Probabilistic Modeling of Power Systems

Transmission and Dispatch

Ricardo Marquez

Working draft

Preface

This book is written for an engineer who knows conservation laws and linear algebra and has never met a probabilistic graphical model. It takes one idea and follows it through: that the structure of a physical system, the way conserved quantities accumulate at places and move between them and cross the system's boundary, is also the structure of every probabilistic question one can ask about that system. A model of the physics is a graph. A model of what we do not know about the physics is the same graph with probability attached to it. Inference on that graph is how a measurement in one place becomes knowledge about another, how a failure is diagnosed, how a decision is justified.

A transmission network is a good system to point that idea at, and in one respect the best. It is the one domain where the idea already has a foothold: operators have estimated the state of their networks from redundant measurements, weighed against the network's own equations, for half a century [82], and the control-room estimator is a small, specialized instance of the construction this book generalizes. Its physics is settled: a power balance at every bus and a flow law on every line. Its instruments are dense by the standards of other physical systems and still leave the questions that matter unanswered. Which reading is wrong? Which breaker status is? How close will tomorrow's peak come to a line's limit after an outage? Its decisions are made under uncertainty at every scale from the next five minutes to the next decade — dispatch, contingency analysis, resource adequacy — on a network whose boundaries are drawn by rule, not by convenience.

Every system worth modeling is a piece cut out of a larger one, trading a conserved quantity with an environment it does not model. For a power network the piece is a balancing area, a region or a substation. The boundary is where one operator's model stops and a neighbor's begins: the tie lines metered for interchange, the generators and loads at the edge of the model, and the lower-voltage network summarized as an injection. That boundary is where the modeler's knowledge is thinnest. As the later chapters show, it is also where the mathematics of inference does its most useful work, because the flows and voltages at a boundary are exactly what the rest of the interconnection needs to know about the inside. Chapter 1 names such a system a *boundary-exchanging system*.

Nothing in the method of Parts I to V is specific to power. The balances, closures and ports are the same whether the conserved quantity is power, charge, mass or heat, and Table 1.3 lines them up. Three companion editions take the same method text through water distribution, district heating and cooling, and data centers. Each is complete in itself, and a reader of this one needs none of them.

Most chapters end twice. First with exercises, which are worked with pencil and paper and whose answers are given for a selection of them. Then with a short list headed *Problems from the studies*. Those are different in kind: each one is a question asked of a real system, stated and solved and committed in the repository beside the study it came from, with a script that

regenerates its answer and a test that checks the answer has not moved. They are there for the reader who wants to run something rather than derive something, and they are the same material the case studies of Part VI are built from, cut down to the size of a problem. The path printed under each is relative to `docs/terra_graph_engine/case_studies/`.

The Introduction says why the book exists, what it inherits from four older traditions, what it adds, and how to read it. Part I is deterministic and establishes the objects. Part II attaches probability to them and states, rather than skips, the places where that attachment needs care. Part III is inference. Part IV asks the questions an operator actually asks. Part V handles time, hierarchy and scale. Part VI is ten complete studies on published test systems, from three lines small enough to enumerate to an annual adequacy study checked against an independent implementation, and each can be reproduced from the material in the earlier parts. Each chapter ends with a short section of notes that says which of its ideas are inherited and from where, and which are the book's own.

Contents

Preface **ii**

0 Introduction **1**
0.1 The questions a grid operator asks . . . 1
0.2 Where the graph comes from . . . 2
0.3 One subject, not four . . . 3
0.4 Probability in the structure, not around the solver . . . 5
0.5 The boundary . . . 6
0.6 From "what is the state" to "what can I know" . . . 6
0.7 What is inherited and what is the book's . . . 7
0.8 How to read this book . . . 8
0.9 What this book does not do . . . 9
Notes and further reading . . . 9

I Physical systems as graphs **10**

1 Conservation, closure, and the boundary **11**
1.1 Three questions about any system . . . 11
1.2 Nodes conserve . . . 12
1.2.1 The balance law . . . 12
1.2.2 Node roles . . . 12
1.2.3 Incidence . . . 13
1.3 Edges close . . . 13
1.3.1 Closure relations . . . 14
1.3.2 A closure is a relation, not an assignment . . . 14
1.3.3 Parameters . . . 15
1.4 The boundary . . . 15
1.4.1 Reservoirs and imposed conditions . . . 15
1.4.2 Exactness at a cut . . . 16
1.4.3 Why the boundary matters most . . . 17
1.5 The residual system . . . 17
1.5.1 Well-posedness . . . 18
1.5.2 Sparsity . . . 18
1.6 Three worked examples . . . 19
1.7 In practice . . . 26

1.8 What is missing . . . 27
Notes and further reading . . . 28
Summary . . . 28
Exercises . . . 29
Solutions to selected exercises . . . 29
Problems from the studies . . . 30

2 From equations to factor graphs **31**
2.1 The rows as a graph . . . 31
2.2 Three languages . . . 32
2.2.1 Directed graphs . . . 32
2.2.2 The directionality problem . . . 33
2.2.3 Undirected graphs . . . 35
2.2.4 Factor graphs . . . 35
2.3 The physical system as a factor graph . . . 36
2.3.1 A bus is not a variable . . . 37
2.3.2 Two domains on one graph . . . 37
2.4 The factor graph is not unique . . . 38
2.5 Vocabulary . . . 40
2.6 In practice . . . 41
2.7 What is missing . . . 41
Notes and further reading . . . 42
Summary . . . 42
Exercises . . . 43
Solutions to selected exercises . . . 43
Problems from the studies . . . 44

II Probability on a physical graph **45**

3 Where probability enters **46**
3.1 A probability on the state . . . 46
3.2 Five kinds of factor . . . 47
3.2.1 Priors . . . 47
3.2.2 Closure factors . . . 48
3.2.3 Conservation factors . . . 48
3.2.4 Measurement factors . . . 49
3.2.5 Failure factors . . . 49
3.2.6 The complete list . . . 49
3.3 The joint as an objective . . . 50
3.3.1 Residuals in units of their widths . . . 51
3.3.2 Conditioning a Gaussian on one reading . . . 51
3.4 A worked example: one sensor, then two . . . 53
3.5 A second example: the triangle with a flow meter . . . 57
3.6 Reading the graph . . . 59
3.7 In practice . . . 60
3.8 What is missing . . . 61

Notes and further reading . . . 61
Summary . . . 61
Exercises . . . 62
Solutions to selected exercises . . . 62
Problems from the studies . . . 63

4 Hard constraints, soft constraints, and the manifold 64
4.1 The hard joint and its manifold . . . 64
4.2 Conditioning on the manifold . . . 65
4.3 The soft joint . . . 66
4.4 Why soften . . . 67
4.5 Choosing the widths . . . 68
4.6 Worked example: the feeder with soft physics . . . 70
4.7 A second example: the triangle with an unmodeled load . . . 72
4.8 In practice . . . 75
4.9 What is missing . . . 75
Notes and further reading . . . 76
Summary . . . 76
Exercises . . . 77
Solutions to selected exercises . . . 77
Problems from the studies . . . 78

5 Factorization, separators, and treewidth 79
5.1 Factorization is not independence . . . 79
5.2 Reading independence off the graph . . . 80
5.3 Explaining away, precisely . . . 81
5.4 Ports are separators . . . 82
5.5 Elimination and fill . . . 83
5.5.1 One elimination . . . 83
5.5.2 Order and width . . . 84
5.5.3 What this means for a physical graph . . . 85
5.6 Worked examples: counting width and fill . . . 86
5.7 Why separators matter for cost . . . 88
5.8 In practice . . . 90
5.9 What is missing . . . 91
Notes and further reading . . . 91
Summary . . . 91
Exercises . . . 92
Solutions to selected exercises . . . 92
Problems from the studies . . . 93

6 The linear-Gaussian case 94
6.1 The Gaussian in information form . . . 94
6.2 The most probable state . . . 95
6.2.1 Gauss–Newton against Newton . . . 96
6.3 Elimination is Cholesky . . . 96
6.3.1 What the factorization costs . . . 97

6.4 The Laplace posterior . . . 99
6.5 Worked example: the feeder in matrix form . . . 99
6.6 Second worked example: the triangle in matrix form . . . 103
6.7 Filters, fields, and state estimators . . . 104
6.8 In practice . . . 105
6.9 What is missing . . . 106
Notes and further reading . . . 106
Summary . . . 107
Exercises . . . 107
Solutions to selected exercises . . . 108
Problems from the studies . . . 109

7 Beyond Gaussian: hybrid states and nonlinearity 110
7.1 Four ways out of the Gaussian world . . . 110
7.2 Hybrid states: the mixture over branches . . . 111
7.2.1 Worked example: a line that may be out . . . 112
7.2.2 Many discrete variables . . . 113
7.3 Piecewise closures and regimes . . . 114
7.3.1 Worked example: a tap changer . . . 114
7.4 Strong nonlinearity . . . 116
7.5 Bounded parameters . . . 117
7.6 What survives . . . 118
7.7 In practice . . . 119
7.8 What is missing . . . 120
Notes and further reading . . . 120
Summary . . . 120
Exercises . . . 121
Solutions to selected exercises . . . 121
Problems from the studies . . . 123

III Inference 125

8 Exact inference: elimination, junction trees, Schur complements 126
8.1 What exact means . . . 126
8.2 Messages on a tree . . . 126
8.3 Loops: the junction tree . . . 128
8.4 Regions and ports: the Schur complement as a message . . . 129
8.5 Reuse . . . 131
8.6 Worked example: two regions, one tie line . . . 132
8.7 Second worked example: two areas with their own references . . . 133
8.8 Discrete variables by region . . . 135
8.9 In practice . . . 136
8.10 What is missing . . . 137
Notes and further reading . . . 137
Summary . . . 138
Exercises . . . 138

Solutions to selected exercises . . . 139
Problems from the studies . . . 140

9 Approximate inference 142
9.1 When exact is too expensive . . . 142
9.2 Loopy belief propagation . . . 143
9.2.1 On the book's models . . . 143
9.3 Variational methods and expectation propagation . . . 145
9.4 Simulation inside a region . . . 147
9.4.1 What a mixture message loses when collapsed . . . 147
9.4.2 Worked example: a generation area as a simulated region . . . 148
9.5 Pruning by evidence . . . 149
9.6 The hierarchy, and how to choose . . . 150
9.7 In practice . . . 150
9.8 What is missing . . . 151
Notes and further reading . . . 151
Summary . . . 152
Exercises . . . 152
Solutions to selected exercises . . . 153
Problems from the studies . . . 154

10 Enumeration, Monte Carlo, and factorized inference 158
10.1 The question, stated honestly . . . 158
10.2 The problem . . . 159
10.3 Enumeration: the product grid . . . 159
10.4 Monte Carlo . . . 161
10.5 Attribution: where the structural difference appears . . . 162
10.6 The factorized route: cuts of the dependency graph . . . 163
10.7 Second worked example: two generation areas . . . 164
10.8 Beyond three inputs . . . 168
10.9 The claim . . . 168
10.10In practice . . . 169
10.11What is missing . . . 170
Notes and further reading . . . 170
Summary . . . 170
Exercises . . . 171
Solutions to selected exercises . . . 171
Problems from the studies . . . 172

11 Sequential evidence 173
11.1 Readings arrive one at a time . . . 173
11.2 One reading, one rank-one update . . . 173
11.3 The innovation . . . 174
11.4 Many readings . . . 175
11.5 Worked example: a stream of flow readings . . . 176
11.6 The Laplace posterior and re-linearization . . . 177
11.7 The value of a reading before it is taken . . . 178

11.8 Cost at scale . . . 180
11.9 Worked example: where to put the next meter . . . 181
11.10In practice . . . 182
11.11What is missing . . . 183
Notes and further reading . . . 183
Summary . . . 184
Exercises . . . 184
Solutions to selected exercises . . . 185

IV The operator's questions 187

12 Conditioning, diagnosis, and virtual sensors 188
12.1 Everything is a readout . . . 188
12.2 Virtual sensors . . . 188
12.3 Is the model adequate? . . . 189
12.4 Where is it wrong? . . . 190
12.5 Worked example: bad data on a power network . . . 191
12.6 Which explanation? . . . 192
12.7 Can this parameter be recovered? . . . 193
12.8 Worked example: can a susceptance drift be seen? . . . 194
12.9 Which sensor next? . . . 195
12.10Should the Gaussian be believed? . . . 197
12.11In practice . . . 198
12.12What is missing . . . 199
Notes and further reading . . . 199
Summary . . . 200
Exercises . . . 200
Solutions to selected exercises . . . 201
Problems from the studies . . . 202

13 Interventions and counterfactuals 204
13.1 Why the graph is not enough . . . 204
13.2 Rows are mechanisms; inputs are exogenous . . . 204
13.3 Seeing versus doing, on the feeder . . . 205
13.4 Counterfactuals: three steps . . . 207
13.5 Decisions as branches . . . 210
13.6 Worked example: an afternoon heat wave . . . 211
13.7 Worked example: holding the tie flow . . . 213
13.8 What the semantics does not decide . . . 214
13.9 In practice . . . 214
13.10What is missing . . . 215
Notes and further reading . . . 215
Summary . . . 216
Exercises . . . 216
Solutions to selected exercises . . . 217
Problems from the studies . . . 218

14 Attribution, sensitivity, and certificates **219**
14.1 Three questions, one derivative . . . 219
14.2 Sensitivities from the solve . . . 219
14.3 Attribution . . . 221
14.4 A census with gradients . . . 222
14.4.1 What the census answers . . . 223
14.4.2 The three-line problem . . . 223
14.5 The kink . . . 224
14.6 Worked example: convexity lost and recovered . . . 226
14.7 Conditions . . . 228
14.8 In practice . . . 229
14.9 What is missing . . . 230
Notes and further reading . . . 230
Summary . . . 231
Exercises . . . 232
Solutions to selected exercises . . . 232
Problems from the studies . . . 233

15 Combinatorial failure **235**
15.1 The question . . . 235
15.2 The prior over failure sets . . . 236
15.2.1 Independent failures . . . 236
15.2.2 Failure as an intervention . . . 237
15.3 The consequence per branch, and how to get it cheaply . . . 237
15.4 Worked example: a six-bus network . . . 238
15.5 The posterior over failure sets . . . 240
15.6 Enumeration by regions . . . 241
15.7 The worst k-set as a query . . . 242
15.8 Common cause and correlated failure . . . 243
15.8.1 A shared shock . . . 243
15.8.2 A mixture over the rate . . . 243
15.9 Worked example: a transformer bank . . . 244
15.10In practice . . . 246
15.11What is missing . . . 246
Notes and further reading . . . 247
Summary . . . 247
Exercises . . . 247
Solutions to selected exercises . . . 248
Problems from the studies . . . 248

V Time, hierarchy, and scale **250**

16 Dynamics and temporal factors **251**
16.1 The accumulation term made live . . . 251
16.2 Integration as a sequence of steady solves . . . 252
16.3 The unrolled graph . . . 254

16.4 Process noise . . . 255
16.5 Filtering, smoothing, and prediction as eliminations . . . 255
16.6 Worked example: tracking drifting losses . . . 256
16.7 Detecting a change in time . . . 257
16.8 Two hypotheses the steady state cannot separate . . . 258
16.9 Events and cascades . . . 260
16.10Worked example: the conductors of the six-bus network . . . 260
16.11In practice . . . 262
16.12What is missing . . . 263
Notes and further reading . . . 263
Summary . . . 264
Exercises . . . 264
Solutions to selected exercises . . . 265
Problems from the studies . . . 265

17 Hierarchy as physical decomposition **266**
17.1 One conservation law used twice . . . 266
17.2 Nested messages . . . 267
17.3 What the parent needs to know . . . 268
17.4 Worked example: a collector substation in three levels . . . 269
17.5 The cut, three times . . . 273
17.6 Worked example: a feeder under a transmission bus . . . 274
17.7 Hierarchy in time . . . 276
17.8 In practice . . . 277
17.9 What is missing . . . 278
Notes and further reading . . . 278
Summary . . . 278
Exercises . . . 279
Solutions to selected exercises . . . 279
Problems from the studies . . . 280

18 Partitioning at scale **281**
18.1 The cost of a cut . . . 281
18.2 Elimination width of physical networks . . . 282
18.3 Choosing the cut . . . 283
18.4 Coupling the regions: messages, propagation, or iteration . . . 285
18.5 Enumeration by region . . . 288
18.6 The second domain . . . 291
18.7 In practice . . . 292
18.8 What is missing . . . 292
Notes and further reading . . . 293
Summary . . . 293
Exercises . . . 294
Solutions to selected exercises . . . 294
Problems from the studies . . . 295

VI Case studies **296**

19 Case studies **297**
19.1 What a case study is here . . . 297
19.2 The studies . . . 298
19.3 Reading a case study . . . 298

20 Three routes on three lines **299**
20.1 The question . . . 299
20.2 The system . . . 299
20.2.1 The model in TERRA . . . 300
20.2.2 One solve . . . 301
20.3 The model as a factor graph . . . 301
20.4 Method . . . 302
20.5 Results . . . 303
20.5.1 The grid, quantity by quantity . . . 304
20.5.2 Sampling . . . 305
20.5.3 Equal accuracy . . . 305
20.5.4 The cut ladder . . . 306
20.6 What surprised . . . 306
20.7 Reproduction . . . 307

21 Nine questions from one census **309**
21.1 The question . . . 309
21.2 The system . . . 310
21.3 The model as a factor graph . . . 310
21.4 Method . . . 311
21.5 Results . . . 313
21.5.1 The certificate and the cost . . . 313
21.5.2 The mean . . . 313
21.5.3 Prices and attribution: Q1 to Q4 . . . 314
21.5.4 New beliefs: Q5 and Q6 . . . 315
21.5.5 The what-if curve: Q7 . . . 315
21.5.6 The tail and the worst case: Q8 and Q9 . . . 316
21.5.7 How large a census . . . 316
21.6 What surprised . . . 317
21.7 Reproduction . . . 318

22 What survives on AC **320**
22.1 The question . . . 320
22.2 The system . . . 321
22.3 The model as a factor graph . . . 321
22.4 Method . . . 322
22.5 Results . . . 324
22.5.1 The certificate . . . 324
22.5.2 What survives . . . 325
22.5.3 Accuracy and cost . . . 325

22.5.4 The answers, question by question . . . 326
22.6 What surprised . . . 328
22.7 Reproduction . . . 329

23 The worst k outages 330
23.1 The question . . . 330
23.2 The system . . . 331
23.3 The model as a factor graph . . . 331
23.4 Method . . . 333
23.4.1 Replication . . . 333
23.4.2 Proof by complete enumeration . . . 333
23.4.3 Search, and what certifies it . . . 333
23.4.4 Conditioning on evidence . . . 333
23.5 Results . . . 334
23.6 What surprised . . . 338
23.7 Reproduction . . . 339

24 Partitioning in practice 341
24.1 The question . . . 341
24.2 The systems . . . 342
24.3 The model as a factor graph . . . 343
24.4 Method . . . 344
24.5 Results . . . 345
24.5.1 Compression on real cuts . . . 345
24.5.2 Two regions at peak load . . . 347
24.5.3 Chains beyond enumeration . . . 348
24.5.4 The second domain . . . 350
24.6 What surprised . . . 350
24.7 Reproduction . . . 352

25 Bad data on fourteen buses 353
25.1 The question . . . 353
25.2 The system . . . 353
25.3 The model as a factor graph . . . 355
25.4 Method . . . 356
25.5 Results . . . 356
25.5.1 Is the model adequate? . . . 356
25.5.2 How checkable is each meter? . . . 357
25.5.3 A gross error where it can be seen . . . 358
25.5.4 A gross error where it cannot . . . 358
25.5.5 One reading at a time . . . 360
25.5.6 Which explanation? . . . 360
25.5.7 Which meter next? . . . 361
25.5.8 Which loads can be recovered? . . . 361
25.5.9 Critical meters and a thinned set . . . 361
25.6 What surprised . . . 363
25.7 Reproduction . . . 364

26 An independent implementation agrees 365
26.1 The question . . . 365
26.2 The workload . . . 365
26.3 What "independent" has to mean . . . 366
26.4 Method . . . 366
26.5 Results . . . 367
26.5.1 The headline . . . 367
26.5.2 Every scenario, not just the total . . . 367
26.5.3 Where they do not agree . . . 368
26.5.4 What the agreement is not . . . 369
26.5.5 What a network-blind model would have said . . . 369
26.5.6 Cost . . . 370
26.6 What surprised . . . 370
26.7 Reproduction . . . 371
Notes . . . 372

27 Where enumeration stops paying 373
27.1 The question . . . 373
27.2 The systems, and what is being counted . . . 373
27.3 Method . . . 374
27.4 Results . . . 374
27.4.1 The cost law . . . 374
27.4.2 Where it dominates, and where it inverts . . . 375
27.4.3 It is not about size . . . 376
27.4.4 Importance sampling collapses with component count . . . 377
27.4.5 The DC watchlist, judged under AC . . . 377
27.4.6 Cost . . . 378
27.5 What surprised . . . 378
27.6 Reproduction . . . 379
Notes . . . 379

28 What a sensor is worth 381
28.1 The question . . . 381
28.2 The two systems . . . 381
28.3 What a publishable rating is . . . 382
28.4 Results . . . 382
28.4.1 Confidence eats the benefit . . . 382
28.4.2 The bound is honest . . . 383
28.4.3 What each sensor is worth . . . 384
28.4.4 A fleet that is already there . . . 385
28.4.5 One circuit loses, and the numbers hold up . . . 385
28.5 What surprised . . . 386
28.6 Reproduction . . . 387
Notes . . . 387

29 Beyond correctness **388**
29.1 The question . . . 388
29.2 Which units failed . . . 388
29.2.1 The setup, and three baselines . . . 388
29.2.2 What it buys . . . 389
29.2.3 Are the stated odds honest? . . . 390
29.2.4 Does it survive a wrong model? . . . 391
29.3 What a megawatt is worth, and where . . . 391
29.3.1 Three prices for the same megawatt . . . 393
29.4 What surprised . . . 393
29.5 Reproduction . . . 394
Notes . . . 394

A Notation **396**
A.1 Conventions . . . 396
A.2 Symbols . . . 397
A.3 Letters with more than one meaning . . . 400

B Gaussian identities **402**
B.1 The two forms of a Gaussian . . . 402
B.2 Block matrices . . . 403
B.3 Marginals and conditionals . . . 404
B.4 Products and linear-Gaussian models . . . 405
B.5 Rank-one changes . . . 406
B.6 Integrals, quadratic forms and divergences . . . 406
B.7 The Laplace approximation . . . 407
B.8 Where the book uses each identity . . . 408

C From the book to the engine **409**
C.1 What TERRA is . . . 409
C.2 A model in a dozen lines . . . 410
C.3 Where each construction lives . . . 410
C.4 The case studies . . . 411

D A catalogue of closures **414**
D.1 How to read an entry . . . 414
D.2 The four patterns in the engine . . . 416
D.3 Electrical . . . 416
D.4 Storage and electrochemistry . . . 418
D.5 Thermal . . . 418
D.6 Fluid . . . 420
D.7 Moist air . . . 422
D.8 Plant equipment . . . 423

Chapter 0

Introduction

A little after six in the morning, while the load ramps and the generators follow it, the state estimator in a transmission control room flags a residual: one line's measured flow disagrees with what every other measurement implies. The list of explanations is short. The transducer has failed. Or a breaker is recorded as closed when it is open, and the network the estimator is solving is not the network in the field. Or the line's parameters are not what the model says. Or the flow is real, and something on the network has changed. The answer matters within the hour, because the estimate feeds the contingency analysis, and the contingency analysis decides how close to its limits the morning's dispatch may run.

The engineer on shift already has the physics of that network: a power balance at every bus, a flow law on every line, a reference angle at the slack bus and a schedule at every generator. Given the injections, a power-flow solver turns those equations into every angle and every flow. What the solver cannot do is run the other way: take a few hundred readings, one of which is wrong, and say which one, how sure it is, what the flow on the suspect line really is, and whether one more phasor measurement unit would have settled it. Probabilistic graphical models supply that machinery. They are usually taught apart from conservation laws, constitutive relations and engineering solvers, and the engineer who wants both has had to learn two subjects and build the bridge alone. This book is the bridge. Its claim is that the bridge is short, because the graph the probabilistic machinery needs is already there in the equations the engineer wrote, and that once the graph is recognized, every question on the shift's list is a readout of one object. This chapter says why that claim is worth a book, what it inherits from four older traditions, what it adds, and how to read it.

0.1 The questions a grid operator asks

A model of a physical system, as engineering teaches it, is a set of equations and a solver. The equations are the balances, the closures and the boundary conditions; the solver turns known inputs into a computed state. The pipeline is

$$\text{inputs} \longrightarrow \text{solver} \longrightarrow \text{outputs}, \tag{0.1}$$

and it answers one question well: given these inputs, what is the state? Most textbooks on physical modeling stop there, and for design that is often enough.

Operation asks different questions. A transmission network is instrumented more densely than most physical systems, and it is still not enough: flows and voltages arrive from SCADA

every few seconds, phasor measurement units cover a fraction of the buses, and every reading passes through a transducer, a channel and a database that can each be wrong. Line parameters come from design data and drift with temperature and age. Loads and renewable output are forecasts. Breaker statuses are telemetered and occasionally misreported. The questions an operator or a transmission planner actually asks are:

- What is the state of the network now, given SCADA and PMU readings and a model I only partly trust?
- Which line's impedance is off, and by how much?
- What is the injection at a bus nobody meters?
- Is this residual a failed meter, a wrong breaker status, a parameter error or a real flow, and on which line?
- How uncertain is each of those answers?
- Where should the next PMU go, and what would it be worth?
- What would happen if I opened that line, redispatched that unit, or moved that phase shifter's tap?
- Which uncertain load or rating drives the risk of an overload after an outage, and is the risk figure I am about to report a floor, a ceiling or a guess?

None of these is a question the pipeline (0.1) can answer. Each asks the model to run in a direction other than inputs to outputs, or to weigh a reading against a law, or to compare explanations, or to say how far its own answer can be trusted. The pipeline has no place to put a reading, no notion of an explanation, and no measure of trust.

The book's starting observation is that the equations do not have to be rewritten to answer these questions. They have to be read differently. A balance says that certain flows sum to zero at a node; a closure says that a flow and two potentials satisfy a relation; a boundary condition says that a quantity has a value. Each is a statement about which combinations of the variables are admissible. Attach to each a statement of how far from admissible the system may be, attach to each uncertain input a statement of what is known about it, attach to each sensor a statement of what it read and how well, and the collection is a probability distribution over the entire state of the system, built from nothing but the physics and the instruments. Every question in the list is then a question about that distribution, and Parts II to IV of this book are about answering them.

0.2 Where the graph comes from

A textbook on probabilistic graphical models begins with random variables and their joint distributions, defines directed and undirected graphs and the factorizations they encode, develops conditional independence, and then teaches inference on the graph. The examples are alarms and burglaries, diseases and symptoms, letters and their noisy transmissions. It is a coherent subject and the machinery is powerful, but an engineer arriving from the physical side asks a question the textbook does not answer: where did the graph come from? For an alarm and a burglary the graph is a modeling decision, drawn from a story about what causes what. For a transmission line, a substation or an interconnected network it is not a decision at all.

For a physical model the graph is the sparsity pattern of the equations. Each balance reads the flows at one node. Each closure reads one flow and the potentials at its ends. Each boundary condition reads one variable. Draw a node for every variable and a node for every equation, join each equation to the variables it reads, and the result is a factor graph, the bipartite graph

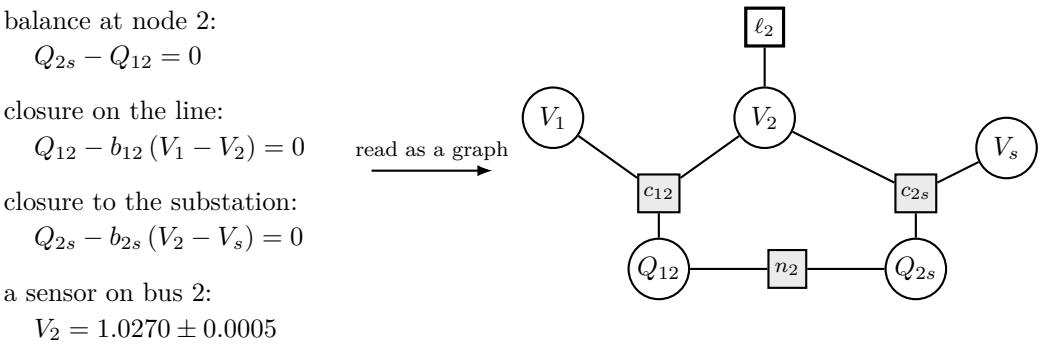


Figure 0.1: The graph comes from the equations. Left: three physics equations of a two-line feeder and one sensor reading, as an engineer writes them. Right: the same four statements as a factor graph. Each equation is a square joined to exactly the variables it reads; the sensor is a square of degree one. Nothing was decided in drawing it.

Table 0.1: What each object of a physical model becomes when probability is attached. The graph is unchanged; only the meaning of the squares is.

In the equations	In the factor graph	With probability attached
balance at a node	a factor on the node's flows	conservation, to within a small width
constitutive relation	a factor on a flow and two potentials	closure, to within its model error
boundary condition	a factor of degree one	a prior on the boundary value
uncertain parameter	a variable in every closure reading it	a prior on the parameter
sensor	(absent)	a likelihood of degree one on its variable
component in or out of service	(absent)	a discrete variable in its closure's scope

on which the probabilistic machinery operates (Figure 0.1). No modeling decision was made in drawing it, because every edge was already an entry of the Jacobian that the solver assembles. The graph is the model's structure, seen.

Attaching probability then changes what the squares mean and not where they are. A balance equation becomes a factor that says the balance holds to within a small tolerance. A closure becomes a factor that says the constitutive law holds to within its model error. An uncertain parameter or boundary condition gets a prior, a square of degree one on its variable. A sensor becomes a likelihood, another square of degree one, on the variable it reads. A component that may be in service or out becomes a discrete variable in the scope of the closure it governs. Table 0.1 lists the correspondence, which Chapter 3 develops. The bridge from the engineer's equations to the graphical model is this table, and it is short.

0.3 One subject, not four

Little in this construction is new in isolation, and it would be wrong to present it as if it were. Four traditions have each built part of it, and the book is written in the conviction that they are one subject that has been taught as four.

Port-based physical modeling. Bond graphs, network thermodynamics and port-Hamiltonian systems supply the domain-independent view of a physical system as conserved quantities at nodes, flows driven by potentials along edges, and conjugate

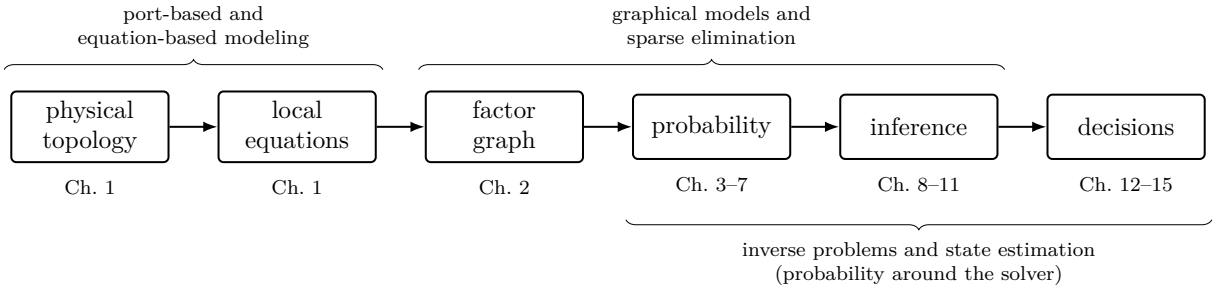


Figure 0.2: The chain of ideas the book follows, with the chapters that cover each link and the traditions that own parts of it. No one tradition carries the object from the topology to the decision.

flow-and-potential pairs at ports, the same structure whether the domain is thermal, fluid, electrical or mechanical [40, 67, 70, 93]. This tradition stops at the equations: it says how to write $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ for any physical system, and then solves.

Equation-based modeling. Acausal modeling languages and the structural analysis of differential-algebraic systems establish that a physical law is a relation among variables, not an assignment of an output from inputs, and they give the bipartite equation-variable structure and its matching theory [5, 20, 64, 69]. This tradition, too, stops at the solve, and it does not attach probability.

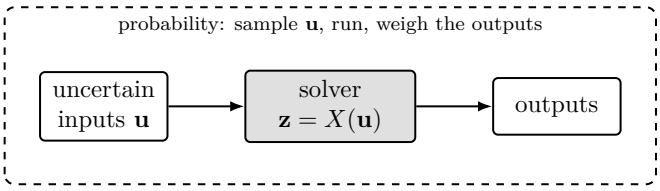
Probabilistic graphical models and sparse elimination. Factor graphs, Markov properties, separators, elimination, fill and treewidth come from the graphical-model and sparse-matrix literatures [16, 46, 48, 50, 78]. This tradition starts from a factorized distribution and asks how to compute with it; it does not say where the factorization comes from, and its graphs are drawn, not derived.

Bayesian inverse problems and state estimation. Priors on uncertain physical inputs, sensors as likelihoods, weighted least-squares estimation under conservation constraints, and the diagnosis of bad data are established in inverse problems, in power-system state estimation, which has run in transmission control rooms since Schweppe and Wildes formulated it [82], and in process data reconciliation [1, 38, 58, 65, 87]. This tradition does attach probability to physics, but usually around the solver: the physics is a black-box map from inputs to observables, and the posterior is over the inputs alone.

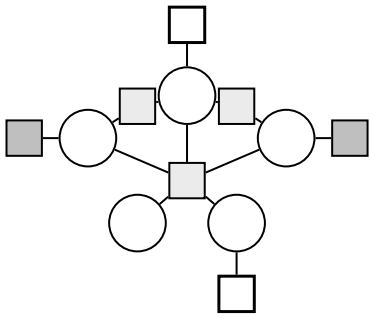
Put the four end to end and they make one chain,

$$\boxed{\text{physical topology} \;\rightarrow\; \text{local equations} \;\rightarrow\; \text{factorization} \;\rightarrow\; \text{probability} \;\rightarrow\; \text{inference} \;\rightarrow\; \text{engineering decisions}} \tag{0.2}$$

of which each tradition owns one or two links and none carries the whole. Figure 0.2 draws it, with the chapters of this book placed along it. The book's purpose is to make the chain feel like one subject: to start where the engineer starts, at the topology and the equations, and to arrive where the operator needs to be, at a decision with its uncertainty stated, without changing the object along the way.



(a) probability around the solver



(b) probability in the structure:
priors (dark) and sensors (open) added to the physics graph

Figure 0.3: Two ways to combine physics and probability. (a) The solver is a black box; probability lives outside it, over its inputs and outputs. (b) The physics equations are the factors; priors and sensors are squares added to the same graph. The book takes the second route and says why.

0.4 Probability in the structure, not around the solver

There is an obvious way to combine a physical model with probability, and most of the fourth tradition uses it: leave the solver as it is, treat it as a function from uncertain inputs to observable outputs, draw inputs from their priors, run the solver, and reason about the outputs. The physics is a black box inside a probabilistic wrapper. Figure 0.3 draws it on the left. It is correct, it is general, and it is how a great deal of uncertainty quantification is done.

The book takes the other route, drawn on the right: the equations themselves are the factors, and probability lives inside the structure, on every variable the physics has. The thesis of the book, in one sentence, is

> Do not put probability around the solver. Put it into the structure of the physical model. (0.3)

Three consequences follow, and they are what the later chapters are about.

There is one object. The deterministic solve is the special case of the probabilistic model in which the priors and sensors are absent and the physics is exact; the two share every row and every Jacobian (Chapter 6 proves this). Adding a sensor adds a row. Adding a question adds nothing: the estimate, the diagnosis, the counterfactual and the risk are readouts of one posterior (Chapter 12).

Structure is preserved, and structure is what computation costs. The wrapper's map from inputs to outputs is dense: every output depends on every input. The physics graph is sparse: every equation reads a few variables. The posterior built in the structure has the sparsity of the physics, and every computation in Part III is affordable because of it. The wrapper's route discards the sparsity the moment it closes the box.

Violation can be seen. A wrapper enforces the physics exactly, because the solver does. When the sensors report something no state of the physics can produce, the wrapper has no answer. The structural model gives each law a width, lets it be violated at a cost, and reports the violation as a diagnosis: the failed transducer, the breaker whose recorded status is wrong, the line whose parameters have drifted (Chapters 4 and 12).

0.5 The boundary

Every system worth modeling is a piece cut out of a larger one. A balancing area trades power with its neighbors, a substation trades power with the distribution feeders below it, a wind farm trades power with the collector system. The piece has an inside that the modeler describes and a boundary across which conserved quantities pass to an outside the modeler does not describe. Chapter 1 calls such a piece a *boundary-exchanging system*, and the book returns to the boundary at every level for a reason that is easiest to state in the language of the third tradition.

In a graphical-model text, a *separator* is a set of variables whose knowledge makes two parts of the graph independent, and finding one is a graph-theoretic exercise whose reward is a cheaper computation. Interconnected power systems drew their separators before any theory asked them to. Balancing areas are joined by tie lines, the tie lines are metered for interchange accounting, and each area's control error is computed from exactly those meters and the frequency. The flows on the tie lines and the voltages and angles at their ends, a flow and its conjugate potential at each port, are exactly the variables that separate an area's interior from the rest of the interconnection (Chapter 5 proves it, and proves that nothing smaller does). So the instruction "find a separator" becomes "cut the interconnection at the tie lines", which is where the interconnection's own rules have already cut it, and the mathematics then explains why that cut is also the cheap place to divide the computation (§8.6 works two regions and one tie line).

This gives one decomposition three readings at once. Physically, a subsystem is summarized to its neighbors by what crosses its ports, exactly, because conservation says the cut flows equal the net source inside. Probabilistically, the ports are a separator, so a subsystem's message to the rest of the system is a function of the ports alone. Computationally, eliminating the interior onto the ports is the Schur complement of sparse elimination, the Kron reduction of network theory, and the region message of a junction tree, which are one operation (Chapter 8). A region can be an equivalent network with error bars, computed once and reused; a large system can be a tree of regions (Chapter 17); and a partition can be judged by its ports (Chapter 18). The three-way connection is the part of the book for which the author has found the least direct precedent, and it is the reason the boundary is in the title.

0.6 From "what is the state" to "what can I know"

The pipeline (0.1) has a direction: inputs in, outputs out. A physical law does not. The relation

$$Q - b\,(V_{\text{tail}} - V_{\text{head}}) \;=\; 0 \tag{0.4}$$

says that four quantities satisfy one constraint. It does not say that Q is computed from the voltages; given Q and V_{tail} it determines V_{head}, and given all three it determines b, which is how a line whose parameters have drifted is found (§12.8). Which variable is unknown is a property of the question, not of the physics (Chapter 2 makes this precise, and shows that for a looped system no fixed direction exists at all). The book keeps the relations as relations, which is what the second tradition insists on and what the first tradition's bond graphs draw, and it is why the factor graph rather than the directed graph is the book's language.

Once uncertainty and observations enter, the absence of direction becomes the point. A reading of one voltage is evidence about everything the physics ties to it. On the export feeder that runs through the book, a single voltage sensor one bus down cuts the uncertainty in the unmeasured reactive injection by a factor of three and a half, and predicts the unmeasured

Table 0.2: The closest prior work, by what it covers. A check marks a substantial treatment; a word marks a partial or specialized one.

	physics graph	equations as factors	probability	general domains	ports as separators	broad inference
bond graphs, port-Hamiltonian [70, 93]	✓	—	—	✓	✓	—
multibody factor graphs [10]	✓	✓	limited	multibody	—	solve, estimate
bond-graph Bayesian networks [55]	✓	indirect	✓	general idea	—	fault diagnosis
factor-graph state estimation [14]	power grid	specialized	✓	power only	regional	state estimation
graphical-model texts [46, 50]	—	—	✓	abstract	graph-theoretic	✓
inverse-problem texts [38, 87]	equations	not generally	✓	✓	—	inverse only
this book	✓	✓	✓	✓	✓	✓

generator-bus voltage to within about a thousandth of a per unit, before any second sensor is read (Chapter 3). That prediction is a *virtual sensor*, and its error bar is part of the answer. The question the model answers has changed from "what is the state, given the inputs" to "what can I know, given what I have", and the second question is the operator's.

0.7 What is inherited and what is the book's

A book that combines old ingredients owes its reader a clear account of which is which. Each chapter closes with a section of notes that does this for its own contents. The account for the whole is short.

Inherited: the port-based view of physics; the acausal reading of its equations; the factor graph, its Markov properties, elimination and treewidth; the Gaussian machinery of least squares, sparse Cholesky and the Laplace approximation; Bayesian inverse problems; state estimation and data reconciliation with their bad-data statistics; the causal calculus of interventions; variance-based attribution; reliability evaluation and contingency analysis. None of these is the book's, and the notes say whose each is.

The book's: the synthesis. Taking an equation-based, conservation-structured physical model and turning its residual structure, unchanged, into the factorization on which inference, diagnosis, uncertainty propagation, attribution and decision queries are performed; the identification of physical ports with probabilistic separators and computational cuts; the intervention semantics for acausal equations; and the insistence, carried through every chapter, that the deterministic model is a corner of the probabilistic one.

The author has looked for this treatment and not found it. Pieces of it exist in several communities that have discovered the connection independently. Multibody dynamics has been written as a factor graph whose sparse structure serves both simulation and estimation, with forward and inverse dynamics as different queries on one graph [10], which is the "relations, not assignments" reading in a single domain. Bond graphs have been converted into Bayesian networks for fault diagnosis [55], which attaches probability to a port-based physical model but does so by assigning the causal directions this book declines to assign. Power systems have been estimated by message passing on a factor graph built from the power-flow equations [14], which is Chapters 6 to 12 in one domain and one query. Table 0.2 sets these beside the four traditions and beside this book. What the table shows is not that any column is empty but that no prior row fills them all, and that several communities have arrived at parts of one construction without a text that says they are parts of one construction. That is the gap the book is written to fill, and "not found" is the strongest claim the author is willing to make for it.

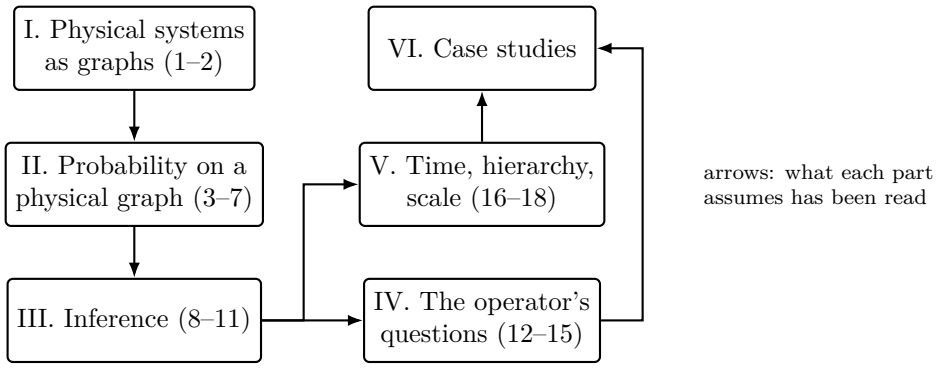


Figure 0.4: The parts of the book and their dependencies. Parts IV and V each need Part III and not each other; Part VI needs both.

0.8 How to read this book

Who it is for. The reader is an engineer who knows conservation laws and linear algebra and has never met a probabilistic graphical model. Nothing about probability beyond the meaning of a density is assumed; Chapter 3 supplies what is needed in a page and Appendix B the Gaussian identities. A reader who knows graphical models and wants the physics will find Chapters 1, 2 and 5 the ones to read first, since they say what the graph is and why the ports separate. The examples are power-system examples; the constructions are not, and a reader who works on heat, water or any other conserved quantity will find nothing in Parts I to V that needs changing but the names.

The parts. Figure 0.4 shows the six parts and what each depends on. Part I is deterministic and builds the objects. Part II attaches probability and states, rather than skips, the places where that attachment needs care. Part III organizes the computation. Part IV asks the operator's questions. Part V handles time, hierarchy and scale. Part VI is a set of complete studies, each reproducible from the material in the earlier parts.

The running examples. Two small systems appear in almost every chapter: an export feeder, a generator that reaches an intermediate bus and then a substation busbar, and a triangle of three buses joined by three lines under the DC approximation of power flow. Both are introduced in Chapter 1, both are small enough to solve by hand, and both acquire priors, sensors, a failed component, a second region, an intervention and a failure set as the chapters go on. The reader who follows them through will have seen every construction of the book on a system whose every number can be checked. Larger systems appear where the small ones cannot show the point: a six-bus network for decomposition and failure, two interconnected areas for common cause, a mesh of buses for cost at scale, and the case studies of Part VI.

The numbers. Every number in every worked example is produced by a short script that accompanies the book, one per chapter, and the text says which. The scripts are a few hundred lines each and use nothing beyond a numerical library; they are meant to be read and rerun, and several exercises ask for exactly that. The case studies of Part VI read their numbers from committed result files of the studies they describe. No number in the book was typed by hand.

Conventions. Each chapter ends with a section on what is in practice, notes on what is inherited and from where, a summary, five exercises, and worked solutions to two of them. Load-bearing conclusions are set apart as numbered principles; there are about three per chapter, and a reader who remembers only those will have the book's argument. Propositions are proved where they are stated. Notation is collected in Appendix A, and the reference implementation that the author used to run every script is mapped to the book's constructions in Appendix C; nothing in the body depends on it.

0.9 What this book does not do

It does not teach physical modeling; it assumes the reader can write a balance and a closure, and it uses the simplest ones. It does not teach probability or graphical models in generality; it teaches the parts a physical model needs, and points to the texts that teach the rest. It does not develop machine learning: no factor in this book is learned from data, though Chapter 12 estimates parameters and Chapter 14 fits a surrogate, and the notes say where the learned versions of its constructions live. It does not treat dynamics faster than a dispatch interval: rotor angles, transient stability, electromagnetic transients and protection timing are absent, and time enters in Chapter 16 only as slow drift, conductor heating and cascading outages.It does not treat continuous space: every system here is a graph of lumped elements, and a field discretized on a mesh appears only as a large graph whose treewidth is a warning. And it does not claim that the constructions it teaches are new; it claims that they are one subject, and it tries to teach them as one.

Notes and further reading

The five gaps that §0.1 to §0.6 describe were articulated in a review of the manuscript, and the positioning of §0.7 follows a literature search made in response to it. The closest works found, Blanco-Claraco, Leanza and Reina's factor-graph multibody dynamics [10], Lo, Wong and Rad's bond-graph Bayesian networks [55], and Chavali and Nehorai's factor-graph state estimation [14], are each discussed in the chapter whose subject they touch. The four traditions of §0.3 are cited there and again in the notes of the chapters that inherit from them; the reader who wants one text per tradition is directed to Karnopp and Rosenberg [40], Benveniste, Caillaud and Malandain [5], Koller and Friedman [46], and Kaipio and Somersalo [38].

Part I

Physical systems as graphs

Chapter 1

Conservation, closure, and the boundary

Every physical system a modeler cares about is a piece cut out of a larger one. A transmission network trades power with the networks around it. A substation trades reactive power with the feeders it supplies. A building trades heat with the weather. The piece has an inside, which the modeler describes in detail, and a boundary, across which conserved quantities pass to an outside the modeler does not describe at all.

This chapter builds the deterministic model of such a piece. It has three ingredients and no probability. The three ingredients are conservation at places, closure along paths, and exchange at the boundary. The reason to build the deterministic model first, and to build it carefully, is that the probabilistic model of Part II inherits its structure entirely: the places, paths and boundary of this chapter become the nodes, factors and separators of the graphical model. Nothing is added to the structure later. Only uncertainty is.

1.1 Three questions about any system

Take any conserved quantity: energy, mass, electric charge, a chemical species. Three questions describe how it lives in a system.

1. Where can it accumulate?
2. How does it move from one place to another?
3. What crosses the boundary?

The first question identifies the *places*. Each place has an amount of the quantity, which may change in time. The amount is an *extensive* state: the energy in a wall, the mass of water in a tank, the charge on a capacitor. Doubling the place doubles the state.

The second question identifies the *paths*. Along a path the quantity flows at some rate: watts, kilograms per second, amperes. What drives the flow is a difference in an *intensive* quantity between the two ends of the path: temperature, pressure, voltage. Doubling the place does not double the intensive quantity; it is a property of the state at a point, not of the amount.

The third question identifies the *boundary*. Some paths lead to places the model does not describe. The quantity that crosses those paths enters or leaves the system. The places on the far side are held at imposed conditions, either an imposed intensive value (the ambient temperature, the grid voltage) or an imposed flow (a heat load, a scheduled injection).

Table 1.1: Four conservation domains and their three quantities.

Domain	Extensive state	Flow	Intensive driver
Energy (thermal)	internal energy E [J]	heat rate Q [W]	temperature T [K]
Mass	mass M [kg]	mass rate $\dot{m}$ [kg/s]	pressure p [Pa]
Charge	charge q [C]	current I [A]	voltage V [V]
Power (DC network)	—	line flow F [W]	angle θ [rad]

Table 1.1 lists the four conservation domains that recur in this book with their extensive state, their flow, and the intensive quantity that drives it. The rows are physically different; the columns are the same three questions. That parallel is the foundation of everything that follows. A method that works on the columns works on every row.

The last row is worth a comment. In the DC approximation of a power network the conserved quantity is real power, the flow on a line is the power it carries, and the driver is the voltage angle difference between its ends. There is no extensive state because the approximation is steady: buses do not store power. The row still fits the three questions. The first answer is simply "nowhere".

1.2 Nodes conserve

The three questions define a graph. The places are nodes; the paths are edges. Write $\mathcal{N}$ for the set of nodes, $\mathcal{E}$ for the set of edges, and $\mathcal{D}$ for the set of conservation domains the model tracks. A model with heat and water flowing through the same pipework has $\mathcal{D} = \{\text{energy}, \text{mass}\}$; a DC power network has $\mathcal{D} = \{\text{power}\}$.

1.2.1 The balance law

At each node n and for each domain d in which n can accumulate, the rate of change of the extensive state equals what flows in, minus what flows out, plus what is generated, minus what is lost:

$$\boxed{\frac{\mathrm{d}X_{n,d}}{\mathrm{d}t} \;=\; \sum_{e \in \mathcal{E}(n)} A_{n,e}\, P_{d,e} \;+\; G_{n,d} \;-\; L_{n,d}} \tag{1.1}$$

Here $X_{n,d}$ is the extensive state of domain d at node n, $\mathcal{E}(n)$ is the set of edges incident on n, $P_{d,e}$ is the flow of d along edge e, $G_{n,d}$ is a source at the node and $L_{n,d}$ a loss. The coefficient $A_{n,e}$ is the sign that says whether edge e delivers to or draws from node n; it is defined in §1.2.3.

Equation (1.1) is the whole of physics that a node contributes. It is linear in the flows, always and exactly. It has no parameters. Whether the domain is energy or mass or charge, whether the node is a wall or a tank or a bus, the balance row looks the same. This is not a modeling convenience. It is the reason the graph is the right object: the graph is the conservation skeleton, and every domain that the model tracks uses that same skeleton.

1.2.2 Node roles

Not every node accumulates. Three roles cover the cases.

Balanced. The node has an extensive state that can change. Equation (1.1) holds with its left side live. A wall, a water tank, a battery.

Zero-storage. The node is a junction or a bus. It has no extensive state, so equation (1.1) holds with $\mathrm{d}X_{n,d}/\mathrm{d}t \equiv 0$: what comes in goes out. A pipe tee, an electrical bus, a mixing point.

Reservoir. The node is on the far side of the boundary. Its intensive state is imposed, not solved. It contributes no balance row, because the model does not claim to know its balance; it appears only as the far end of the edges that cross the boundary. The atmosphere, the utility grid, a feed line.

The first two roles are the inside of the system. The third is the outside. Which nodes are reservoirs is the modeler's first and most consequential decision, and §1.4 returns to it.

A node also declares which domains it is balanced in. A pipe carrying water may need a mass balance at each tee but no energy balance if the pipe is adiabatic; a bus needs a power balance and nothing else. The balance rows exist only where the modeler asks for them. The roles are declared per domain as well: a return manifold whose pressure is held by a pump is a reservoir for mass and yet a balanced node for energy, because the temperature of the water leaving it is the model's to determine. Example 1.3 has such a node.

1.2.3 Incidence

Give every edge a direction, from a tail node to a head node. The direction is a bookkeeping convention, not a physical claim: a flow that runs against the edge's direction is simply negative. The *incidence matrix* $\mathbf{A}$ has one row per non-reservoir node and one column per edge, with entries

$$A_{n,e} = \begin{cases} +1 & \text{edge } e \text{ enters } n \text{ (n is its head)}, \\ -1 & \text{edge } e \text{ leaves } n \text{ (n is its tail)}, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

Every column of the full incidence matrix (with reservoir rows included) has one $+1$ and one -1, so the columns sum to zero. Dropping the reservoir rows breaks that property for the columns of boundary edges, and that is exactly right: the flow on a boundary edge is not balanced by any row of the model. It is what the system exchanges.

With $\mathbf{A}$ in hand, the sum in equation (1.1) is one row of a matrix product. Stack the flows of domain d into a vector $\mathbf{P}_d$ and the balances into a vector; the balance law for all nodes at once is $\mathrm{d}\mathbf{X}_d/\mathrm{d}t = \mathbf{A}\,\mathbf{P}_d + \mathbf{G}_d - \mathbf{L}_d$.

> **Principle 1.1.** *The incidence matrix is shared by every domain. Energy and mass in the same pipework do not have two graphs. They have one graph, and each domain contributes balance rows on the nodes where it is tracked, using the same* $\mathbf{A}$.

An edge that carries one domain but not another, such as a rotating shaft that transmits energy but no mass, or a conduction path through a solid, simply has no flow variable for the absent domain. Its column contributes zero to that domain's balance rows.

1.3 Edges close

The balance law says that the flows at a node sum to the accumulation. It does not say what any flow is. That is the job of the edge.

Table 1.2: Four closure patterns. Each is written as a residual that vanishes when the relation holds.

Pattern	Residual	Typical use
Carrier flow	$P_e - q_e\,\varepsilon_e$	a quantity carried by another: $P = \dot{m}\,h$ (enthalpy carried by mass), $P = I\,V$ (power carried by current)
Gradient flow	$P_e - g(\phi_i, \phi_j; \theta)$	driven by a difference in intensive state: $Q = UA\,(T_h - T_c)$, $F = B\,(\theta_i - \theta_j)$
Property	$y - f(x_1, \ldots, x_k)$	a property table or a device specification: $h = h(p, T)$, $Q = \mathrm{COP} \cdot W$
Piecewise	$y - f_b(x)$	a device that changes regime: a valve that saturates, a limiter, a breaker

1.3.1 Closure relations

Along each edge, for each domain it carries, there is a flow variable and a relation that ties that flow to the states at the edge's ends and to some parameters. The relation is called a *closure* because it closes the system of equations: the balance rows alone have more unknowns than rows, and the closures supply the missing rows.

Definition 1.1 (Closure)**.** A closure is a local relation $r(\mathbf{z}_{\mathrm{loc}}; \theta) = 0$ among a small set of variables $\mathbf{z}_{\mathrm{loc}}$, with parameters θ, written in *residual form*: a function that is zero exactly when the relation holds. The closure declares which variables it reads.

Four patterns cover the great majority of physical closures (Table 1.2).

The gradient pattern is the one the three questions anticipated: a flow driven by an intensive difference. Conduction, convection, DC current, DC power flow and laminar pipe flow all take this form, with different functions g and different parameters θ: a conductance, a heat transfer coefficient times an area, a susceptance. The carrier pattern appears whenever one domain rides on another, as enthalpy rides on mass flow. The property pattern is a catch-all for relations among variables at a single place. The piecewise pattern is the first hint of something the deterministic model handles awkwardly and the probabilistic model handles naturally; it returns in Chapter 7. Appendix D catalogues the closures of the domains this book touches. For each it gives the residual and its pattern, the unknowns and parameters, the Jacobian, where the relation kinks, its shape, and the parameter that usually carries the prior.

1.3.2 A closure is a relation, not an assignment

Write the conduction closure as most textbooks do:

$$Q = UA\,(T_h - T_c). \tag{1.3}$$

The notation suggests a computation: given the two temperatures, compute the heat rate. But nothing in the physics says which quantities are given. If the heat rate and the hot temperature are known, the same relation determines the cold temperature:

$$T_c = T_h - Q/(UA). \tag{1.4}$$

If all three are measured, the relation determines the conductance UA. The physics asserts that four quantities satisfy one constraint. It does not assert a direction.

The residual form makes this explicit:

$$r(Q, T_h, T_c; UA) \;=\; Q - UA\,(T_h - T_c) \;=\; 0. \tag{1.5}$$

Equation (1.5) has no left-hand side to compute. It is a surface in the space of its variables, and every point on the surface is a physically admissible state of the edge.

> **Principle 1.2.** *A closure relates its variables. It does not compute one of them from the others. Which variable is unknown is a property of the question being asked, not of the physics.*

This principle looks like a remark about notation. It is not. It decides, in Chapter 2, which kind of graphical model a physical system should be written in, and it is the reason the arrows of a Bayesian network are not the natural representation of a closure. Hold on to it. The principle itself is old: it is the acausal view of physical laws that bond-graph modeling and equation-based modeling languages are built on [64, 70], and §1.8 says more.

1.3.3 Parameters

The parameters θ of a closure, a conductance, a susceptance, an efficiency, are ordinarily numbers the modeler supplies. Nothing forces that. A parameter the modeler does not know can be left as an unknown, added to the vector of variables, and determined by the equations if enough other things are known. A conductance inferred from a measured heat rate and two measured temperatures is the simplest case of this. The deterministic model needs one extra equation for each parameter it frees; the probabilistic model of Part II needs only a prior.

1.4 The boundary

The boundary is the set of edges with a reservoir at one end. Everything on the reservoir's side is outside the model. This section says what the model knows about the outside, why that knowledge is exact in one specific sense, and why the boundary is the most important part of the system.

1.4.1 Reservoirs and imposed conditions

A reservoir contributes no balance row. It contributes an imposed condition on the boundary edge, of one of two kinds.

Imposed potential. The reservoir's intensive state is fixed: the ambient temperature, the grid voltage angle at the point of connection. The boundary edge's closure then determines the flow across it. This is the Dirichlet condition of boundary-value problems.

Imposed flow. The flow across the boundary edge is fixed: a specified heat load, a scheduled generation injection. The balance row at the inside node then determines the inside potential. This is the Neumann condition.

Each boundary edge is thus a pair, a flow and its conjugate potential, one of which is imposed and the other of which the model determines. The pair is domain-independent (Table 1.3). Call the pair a *port*.

Table 1.3: Ports: the conjugate flow and potential pair at a boundary, by domain.

Domain	Flow across the boundary	Potential at the boundary
Thermal	heat rate Q	temperature T
Fluid	mass rate $\dot{m}$	pressure p
Electrical	current I	voltage V
DC power	line flow F	angle θ

Definition 1.2 (Boundary-exchanging system)**.** A boundary-exchanging system is a conservation graph with at least one reservoir node. It exchanges conserved quantities with its environment through its ports. Its state is determined jointly by its internal balances and closures and by the conditions imposed at its ports.

Every model in this book is boundary-exchanging. A closed system, one with no reservoirs, is a limiting case that occurs in textbooks and almost nowhere else. The term is this book's own. Thermodynamics calls such a system *open*, and port-based modeling calls the places where it trades with its environment *ports* [93]. The definition adds only that the trade goes through reservoir nodes of a conservation graph, which is what lets Chapter 5 treat the ports as separators.

1.4.2 Exactness at a cut

Take any set S of non-reservoir nodes, and consider the edges that cross from S to its complement. Sum the steady balance rows over the nodes in S. Every edge with both ends in S appears twice in the sum, once with $+P_e$ at its head and once with $-P_e$ at its tail; it cancels. Every edge with one end in S appears once. What remains is

$$\sum_{e \text{ crossing } S} A_{n(e),e}\, P_{d,e} \;=\; -\sum_{n \in S} (G_{n,d} - L_{n,d}), \tag{1.6}$$

where $n(e)$ is the end of e inside S. In words: the net flow of domain d across the boundary of S equals the net source inside S, whatever happens inside. Internal detail has no effect on the cut flow.

Proposition 1.1 (Exactness at a cut)**.** *In steady state the sum of the flows crossing the boundary of any node set S equals the net source inside S. The sum depends on S only through its sources and losses, not through its internal structure.*

This is a two-line consequence of conservation, and it is the reason a boundary can be trusted. Whoever sits outside S and sees only the port flows sees a quantity that is exact, not an approximation of the inside. A substation meter that reads the total power delivered by the feeders it supplies reads a number that is identically the station's load, not an estimate of it. Chapter 17 builds an entire theory of hierarchical modeling on Proposition 1.1.

Note what the proposition does not say. It says nothing about the intensive states. The voltage at the boundary of S is not a sum of anything inside S, and no conservation law licenses aggregating it. Extensive flows aggregate by summation across a cut because a conservation law says so. Intensive states must be solved for.

1.4.3 Why the boundary matters most

Two reasons put the boundary at the center of this book.

The first is that it is unavoidable. The modeler who refuses to draw a boundary must model the universe. Every practical model draws one, and what it draws determines what the model can and cannot say. A building model with the weather as a reservoir cannot say anything about the weather; it can only say what the building does given the weather.

The second reason is that the boundary is where knowledge is thinnest. Inside the system the modeler has chosen every closure and knows every parameter, or at least knows which ones are uncertain. At the boundary the modeler has an imposed condition that summarizes an entire unmodeled world into one number per port. That number is a forecast, a schedule, a nominal value, or a measurement taken somewhere else. It is where the model's ignorance is concentrated.

When uncertainty enters in Part II it therefore enters at the boundary first. And when Part III asks how a large system can be solved as a collection of small ones, the answer is that each piece is a boundary-exchanging system in its own right, that its ports carry all that its neighbors need to know about it, and that Proposition 1.1 is what makes the ports sufficient. The machinery of separators and messages in the later chapters is the boundary of this section made probabilistic.

> **Principle 1.3.** *A system is defined by what crosses its boundary. Everything inside is the modeler's to describe; everything outside is summarized, exactly for the flows and by assumption for the potentials, at the ports.*

1.5 The residual system

Collect the pieces. The unknowns of the model are gathered into a single vector $\mathbf{z} \in \mathbb{R}^n$: the intensive state at each inside node, the flow on each edge channel, the extensive state at each balanced node, and any parameters the modeler has freed. Conditions imposed at reservoirs either remove a variable from $\mathbf{z}$ by substituting its value, or add an explicit row $z_k - \bar{z}_k = 0$; either way they are known.

The equations are gathered into a single vector-valued residual $\mathbf{F}(\mathbf{z}) \in \mathbb{R}^m$, whose rows come in a fixed order:

1. balance rows, one per (inside node, tracked domain) pair, equation (1.1) with the accumulation term moved to the right;
2. edge-channel closures, one per edge per domain it carries;
3. storage relations, one per extensive state, tying it to the intensive state at the same node ($E = C\,T$ for a thermal mass, for instance);
4. freestanding closures: properties, specifications, couplings not tied to one edge;
5. explicit boundary-condition rows.

The model is the system

$$\boxed{\mathbf{F}(\mathbf{z}) = \mathbf{0}} \tag{1.7}$$

in steady state, and $\mathbf{F}(\mathbf{z}, \dot{\mathbf{z}}) = \mathbf{0}$ with the accumulation terms live in the dynamic case, which is deferred to Chapter 16.

1.5.1 Well-posedness

Before anything is solved, the system can be checked for shape. It should be square, $m = n$: as many equations as unknowns. Its Jacobian $\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{z}$ should have full structural rank, which means that some assignment of equations to unknowns covers every unknown; this is a property of which variables each row reads, not of their values, and it can be checked on the graph alone. The graph should be connected. Every flow variable should be constrained by some closure, since a balance row alone cannot determine two flows. No variable should be absent from every row.

Each of these checks fails in a way that points at a node, an edge or a variable. A model that fails one is not wrong in a numerical sense. It is underspecified or overspecified, and the failure says where. Making the checks structural, and making them precede any numerical work, is the difference between a modeling error that is reported as "node 7 has no closure on its outlet flow" and one that is reported as a singular matrix somewhere inside a solver.

The first check, squareness, can be done by counting before a single row is written.

Proposition 1.2 (Squareness by counting)**.** *Let a model have N inside nodes each balanced in every domain it touches, E edge channels (one per edge per domain the edge carries), S storage states, Θ freed parameters, C freestanding closures and R explicit boundary-condition rows, and let every inside node carry one potential per domain in which it is balanced. Then the number of unknowns exceeds the number of rows by exactly $\Theta - C - R$. The model is square if and only if every freed parameter is paid for by one freestanding closure or one explicit boundary condition.*

Proof. Count unknowns: one potential per (node, domain) pair, N_d in all; one flow per edge channel, E; one extensive state per storage, S; and Θ parameters. Count rows: one balance per (node, domain) pair, N_d; one closure per edge channel, E; one storage relation per storage, S; then C and R. The first three pairs cancel term by term, leaving $\Theta - (C + R)$. □

The proposition explains a pattern the examples below repeat. A model built from balanced nodes and closed edges is square by construction; it becomes non-square only when the modeler frees a parameter without adding a relation that pins it, or adds a relation without freeing anything. The first is the situation of a model with an unknown conductance and no measurement; the second, of a model with a measurement and nothing for it to determine. Both are exactly the situations that Chapter 3 will handle without counting, because a prior is a row that costs nothing and a measurement is a row that need not be paid for; but in the deterministic model the count must balance, and the proposition says how.

1.5.2 Sparsity

Each balance row has one nonzero entry per incident edge channel and one for its own storage state. Each closure row has one nonzero per variable it declares. The number of nonzeros in $\mathbf{J}$ is therefore of order $|\mathcal{E}|$ plus the total number of closure inputs: linear in the size of the model, not quadratic in the number of unknowns. A network with ten thousand nodes has a Jacobian with tens of thousands of nonzeros, not a hundred million.

The balance rows are exactly linear in the flows, so their entries in the flow columns are the constants $-A_{n,e}$ and never change. All of the nonlinearity, and all of the domain-specific derivative work, lives in the closure rows. The pattern of nonzeros is known before any number is computed, from the closures' declared inputs alone.

Both facts, linear sparsity and a fixed pattern, are what make everything downstream affordable: solving equation (1.7) by Newton's method, eliminating the inside of a subsystem to

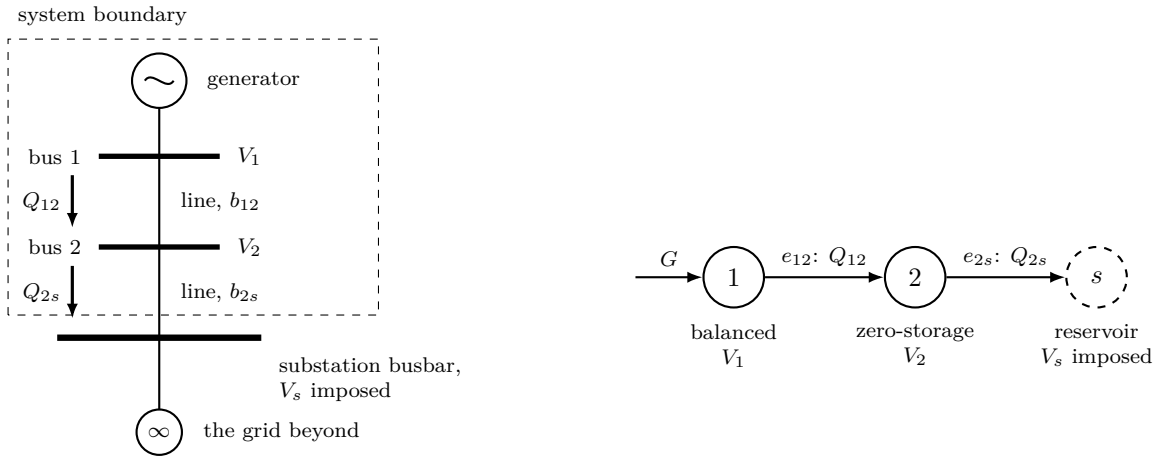


Figure 1.1: The export feeder of Example 1.1. Left: the single-line diagram. A generator injects G into bus 1 at V_1; bus 1 reaches bus 2 at V_2 through a line of susceptance b_{12}; bus 2 reaches the substation busbar at V_s through a line of susceptance b_{2s}. The dashed box is the system boundary; the grid beyond it is not modeled. Right: the same arrangement as a conservation graph. Node 1 is balanced (in steady state its accumulation is zero), node 2 is a junction, and node s, dashed, is the substation reservoir on the far side of the boundary. Two edges, two closures; the source G is a known injection at node 1.

expose its ports, and, in Part II, forming the precision matrix of a Gaussian posterior, which turns out to be $\mathbf{J}^\mathsf{T}\mathbf{J}$ up to weighting and inherits this sparsity exactly.

1.6 Three worked examples

Three examples show every object of this chapter in a form small enough to solve by hand or nearly so. The first is radial and carries one conserved quantity; the second is meshed and is the running example of the book, acquiring a loop here, uncertainty in Chapter 3, and a failed line in Chapter 7; the third carries two domains on one graph. The method of this book is not particular to electric power, and the tables above name the other domains it applies to unchanged; but a book is easier to read when its examples come from one place, and in this one they come from transmission and distribution. Every number is from `ch01_examples.py` in the book's `scripts` directory.

Example 1.1 (A generator exporting through a feeder)**.** An embedded generator injects a known reactive power G into a distribution bus (node 1). That bus reaches an intermediate bus (node 2) through a line of susceptance b_{12}. The intermediate bus reaches the substation busbar (node s) through a second line of susceptance b_{2s}. The substation voltage V_s is held by the transformer's tap changer and is imposed. Figure 1.1 draws the graph.

Unknowns. Two flows, Q_{12} and Q_{2s}, and two potentials, V_1 and V_2. The substation voltage is imposed and substituted out.

Incidence. Rows for nodes 1 and 2, columns for edges e_{12} and e_{2s}, directions as drawn:

$$\mathbf{A} = \begin{pmatrix} -1 & 0 \\ +1 & -1 \end{pmatrix}. \tag{1.8}$$

The reservoir row $(0, +1)$ is dropped; the second column no longer sums to zero, which is the signature of a boundary edge.

Residuals. Two rows come from the balance law. At each inside node, equation (1.1) in steady state is $0 = \sum_e A_{n,e} Q_e + G_n$, written as the residual $r_n = -\sum_e A_{n,e} Q_e - G_n$, which vanishes when the node balances:

$$r_1 = -(-Q_{12}) - G = Q_{12} - G \qquad \text{(balance at node 1)}, \tag{1.9}$$
$$r_2 = -(Q_{12} - Q_{2s}) = Q_{2s} - Q_{12} \qquad \text{(balance at node 2)}. \tag{1.10}$$

Two more come from the closures, one gradient closure on each edge:

$$r_3 = Q_{12} - b_{12}\,(V_1 - V_2) \qquad \text{(closure on } e_{12}\text{)}, \tag{1.11}$$
$$r_4 = Q_{2s} - b_{2s}\,(V_2 - V_s) \qquad \text{(closure on } e_{2s}\text{)}. \tag{1.12}$$

Four rows, four unknowns, and every flow has a closure.

Vector form. Stack the unknowns, flows first, and the residuals in the order of the rows above:

$$\mathbf{z} = \begin{pmatrix} Q_{12} \\ Q_{2s} \\ V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{v} \end{pmatrix}, \qquad \mathbf{F}(\mathbf{z}) = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} Q_{12} - G \\ Q_{2s} - Q_{12} \\ Q_{12} - b_{12}\,(V_1 - V_2) \\ Q_{2s} - b_{2s}\,(V_2 - V_s) \end{pmatrix}.$$

The model is $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, equation (1.7) for this example. Its rows come in two blocks, and the incidence matrix writes each block in one line. Collect the data of the model:

$$\mathbf{s} = \begin{pmatrix} G \\ 0 \end{pmatrix}, \qquad \mathbf{K} = \begin{pmatrix} b_{12} & 0 \\ 0 & b_{2s} \end{pmatrix}, \qquad \mathbf{A}_b = \begin{pmatrix} 0 & +1 \end{pmatrix}.$$

The vector $\mathbf{s}$ holds the sources at the inside nodes, the diagonal matrix $\mathbf{K}$ holds one susceptance per edge, and $\mathbf{A}_b$ is the dropped reservoir row, through which the imposed voltage V_s enters. The balance rows r_1, r_2 are $-\mathbf{A}\mathbf{q} - \mathbf{s}$. For the closure rows, look at the vector $\mathbf{A}^\mathsf{T}\mathbf{v} + \mathbf{A}_b^\mathsf{T} V_s$. Its first entry is $-V_1 + V_2$ and its second is $-V_2 + V_s$: it is the voltage rise along each edge, head minus tail, so its negative is the drop that drives the flow. The closure rows r_3, r_4 are therefore $\mathbf{q} + \mathbf{K}(\mathbf{A}^\mathsf{T}\mathbf{v} + \mathbf{A}_b^\mathsf{T} V_s)$, and

$$\mathbf{F}(\mathbf{z}) = \begin{pmatrix} -\mathbf{A}\mathbf{q} - \mathbf{s} \\ \mathbf{q} + \mathbf{K}(\mathbf{A}^\mathsf{T}\mathbf{v} + \mathbf{A}_b^\mathsf{T} V_s) \end{pmatrix}. \tag{1.13}$$

Jacobian. Differentiate $\mathbf{F}$ with respect to $\mathbf{z}$, one block at a time. The balance block depends on the flows alone, through $-\mathbf{A}$, and not on the voltages. The closure block depends on the flows through the identity and on the voltages through $\mathbf{K}\mathbf{A}^\mathsf{T}$. So

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{z}} = \begin{pmatrix} \partial \mathbf{F}_{\text{bal}}/\partial \mathbf{q} & \partial \mathbf{F}_{\text{bal}}/\partial \mathbf{v} \\ \partial \mathbf{F}_{\text{clo}}/\partial \mathbf{q} & \partial \mathbf{F}_{\text{clo}}/\partial \mathbf{v} \end{pmatrix} = \begin{pmatrix} -\mathbf{A} & \mathbf{0} \\ \mathbf{I} & \mathbf{K}\mathbf{A}^\mathsf{T} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & -b_{12} & b_{12} \\ 0 & 1 & 0 & -b_{2s} \end{pmatrix}. \tag{1.14}$$

The balance rows are constant and the closure rows carry the parameters. The structural rank is full.

Solution. Every model in this book is solved the same way, by Newton's method on $\mathbf{F}(\mathbf{z}) = \mathbf{0}$. Each step replaces $\mathbf{z}$ by $\mathbf{z} - \mathbf{J}^{-1}\mathbf{F}(\mathbf{z})$, one linear solve with the Jacobian, and the steps repeat

until the residual vanishes. Here every closure is linear, so $\mathbf{F}$ is exactly linear in $\mathbf{z}$, $\mathbf{J}$ is constant, and the first step lands on the solution from any starting point. With $G = 0.05$ per unit, $b_{12} = 5$ and $b_{2s} = 2$ per unit, and $V_s = 1.00$ per unit, one step from $\mathbf{z} = \mathbf{0}$ gives $\mathbf{q} = (0.05, 0.05)^{\mathsf{T}}$ and $\mathbf{v} = (1.035, 1.025)^{\mathsf{T}}$: both flows equal the source, bus 1 sits at 1.035 and bus 2 at 1.025, so exporting has raised the far end of the feeder three and a half percent above the busbar. The last fact holds for any numbers. Every column of the full incidence matrix sums to zero, so $\mathbf{1}^{\mathsf{T}}\mathbf{A} + \mathbf{A}_b = \mathbf{0}$; multiplying the balance rows by $\mathbf{1}^{\mathsf{T}}$ gives $\mathbf{A}_b\mathbf{q} = \mathbf{1}^{\mathsf{T}}\mathbf{s}$, which says that the flow into the reservoir, Q_{2s}, is the total source G. That is Proposition 1.1 for $S = \{1, 2\}$.

The solution is also linear in the two inputs, the source and the substation voltage. Collect them in $\mathbf{u} = (G, V_s)^{\mathsf{T}}$; then each voltage is a fixed combination of them,

$$V_1 = \mathbf{h}_1^{\mathsf{T}}\mathbf{u}, \quad \mathbf{h}_1 = \begin{pmatrix} 1/b_{2s} + 1/b_{12} \\ 1 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 1 \end{pmatrix}; \qquad V_2 = \mathbf{h}_2^{\mathsf{T}}\mathbf{u}, \quad \mathbf{h}_2 = \begin{pmatrix} 1/b_{2s} \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}.$$

Each voltage is the busbar voltage plus the reactances between it and the boundary times the flow, exactly as an engineer would write it down. These numbers and these two vectors return in Chapter 3, where the source and the busbar voltage become uncertain and a meter on a voltage reads $\mathbf{h}_1^{\mathsf{T}}\mathbf{u}$ or $\mathbf{h}_2^{\mathsf{T}}\mathbf{u}$. What the residual form adds is that the same four rows answer the other questions without rearrangement: if V_1 is measured and G is unknown, free G, add it to $\mathbf{z}$, and the system is again square, as Proposition 1.2 says it must be when one parameter is freed and one row is added.

Example 1.2 (A three-bus DC power network)**.** Three buses are connected in a triangle by lines of equal susceptance B. Buses 2 and 3 carry known injections P_2 and P_3, negative for load. Bus 1 is the point of connection to a much larger network, whose angle is taken as the reference, $\theta_1 = 0$. Bus 1 is a reservoir: the model does not claim to know the balance of the network beyond it. Figure 1.2 draws the graph.

Unknowns. Three flows, F_{12}, F_{13} and F_{23}, and two angles, θ_2 and θ_3; θ_1 is imposed and substituted out.

Incidence. Rows for buses 2 and 3, columns for the three lines in the directions drawn:

$$\mathbf{A} = \begin{pmatrix} +1 & 0 & -1 \\ 0 & +1 & +1 \end{pmatrix}. \tag{1.15}$$

Residuals. Two rows come from the balance law at buses 2 and 3, which are zero-storage, so there is no accumulation:

$$r_2 = -(F_{12} - F_{23}) - P_2 \qquad \text{(balance at bus 2)}, \tag{1.16}$$
$$r_3 = -(F_{13} + F_{23}) - P_3 \qquad \text{(balance at bus 3)}. \tag{1.17}$$

Three more come from the closures, one gradient closure on each line:

$$r_{12} = F_{12} - B\,(\theta_1 - \theta_2) = F_{12} + B\,\theta_2 \qquad \text{(closure on } e_{12}\text{)}, \tag{1.18}$$
$$r_{13} = F_{13} + B\,\theta_3 \qquad \text{(closure on } e_{13}\text{)}, \tag{1.19}$$
$$r_{23} = F_{23} - B\,(\theta_2 - \theta_3) \qquad \text{(closure on } e_{23}\text{)}. \tag{1.20}$$

Five rows, five unknowns.

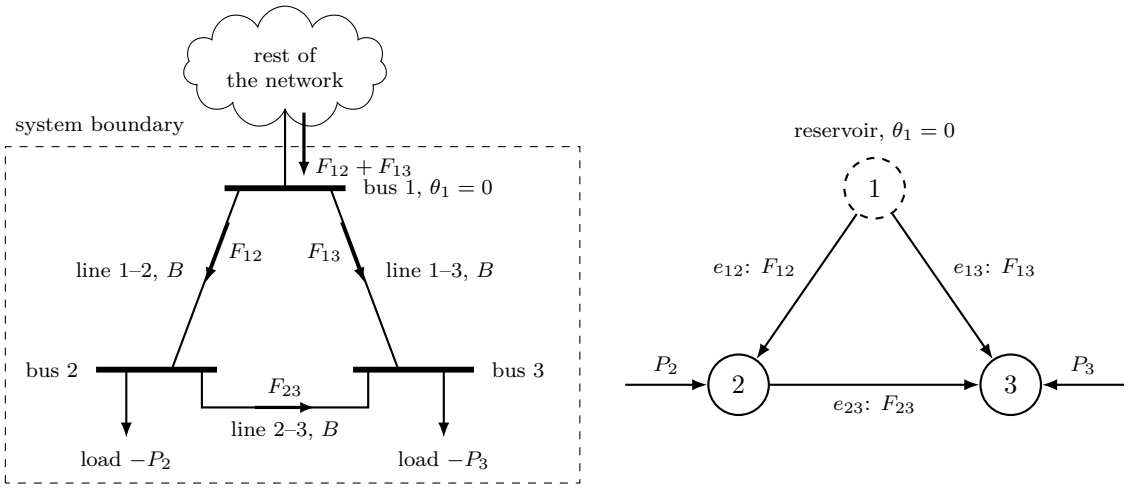


Figure 1.2: The DC triangle of Example 1.2. Left: the one-line diagram. Three bus bars are joined by three lines of equal susceptance B; buses 2 and 3 serve loads; bus 1 is tied to the rest of the network, which lies outside the dashed system boundary and supplies whatever the loads need. Arrows mark the flow directions of the example. Right: the same network as a conservation graph. Buses 2 and 3 are zero-storage; bus 1 is a reservoir at the reference angle. The three lines form a loop, the first loop in this book. Injections P_2, P_3 are imposed.

Vector form. As in Example 1.1, stack the unknowns flows first and the residuals in the order of the rows above:

$$\mathbf{z} = \begin{pmatrix} F_{12} \\ F_{13} \\ F_{23} \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \boldsymbol{\theta} \end{pmatrix}, \qquad \mathbf{F}(\mathbf{z}) = \begin{pmatrix} r_2 \\ r_3 \\ r_{12} \\ r_{13} \\ r_{23} \end{pmatrix}.$$

With the injections $\mathbf{s} = (P_2, P_3)^\mathsf{T}$, one susceptance per line, $\mathbf{K} = B\,\mathbf{I}$, and the reservoir row of bus 1, $\mathbf{A}_b = (-1, -1, 0)$, with its angle θ_1, the residual vector has the two blocks of equation (1.13), with angles in place of temperatures: the balance rows are $-\mathbf{A}\mathbf{F} - \mathbf{s}$ and the closure rows are $\mathbf{F} + \mathbf{K}(\mathbf{A}^\mathsf{T}\boldsymbol{\theta} + \mathbf{A}_b^\mathsf{T}\theta_1)$. Differentiating block by block, as there, gives the Jacobian

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{z}} = \begin{pmatrix} -\mathbf{A} & \mathbf{0} \\ \mathbf{I} & B\,\mathbf{A}^\mathsf{T} \end{pmatrix}.$$

Solution. Newton's method on $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ again. The closures are linear, so from $\mathbf{z} = \mathbf{0}$ it takes one step. With $P_2 = P_3 = -1$ (two equal loads, in per-unit), both angles are $-1/B$ and $\mathbf{F} = (1, 1, 0)^\mathsf{T}$. The line between the two loads carries nothing, by symmetry. The flow into the model across its boundary follows as in Example 1.1: here too $\mathbf{1}^\mathsf{T}\mathbf{A} + \mathbf{A}_b = \mathbf{0}$, so $\mathbf{A}_b\mathbf{F} = \mathbf{1}^\mathsf{T}\mathbf{s}$, which reads $-(F_{12} + F_{13}) = P_2 + P_3$. The boundary flow $F_{12} + F_{13} = 2$ is the total load: Proposition 1.1 again, with $S = \{2, 3\}$. The injection at the reference bus, which a power engineer calls the slack, is precisely the port flow of the boundary-exchanging system.

Now break the symmetry: $P_2 = -1.5$, $P_3 = -0.5$. Then $\mathbf{F} = (1.167, 0.833, -0.333)^\mathsf{T}$. The third entry, F_{23}, is negative: a third of a unit flows from bus 3 to bus 2, against the drawn direction. The loop matters. Bus 2's extra load is served partly by the direct line from bus 1 and

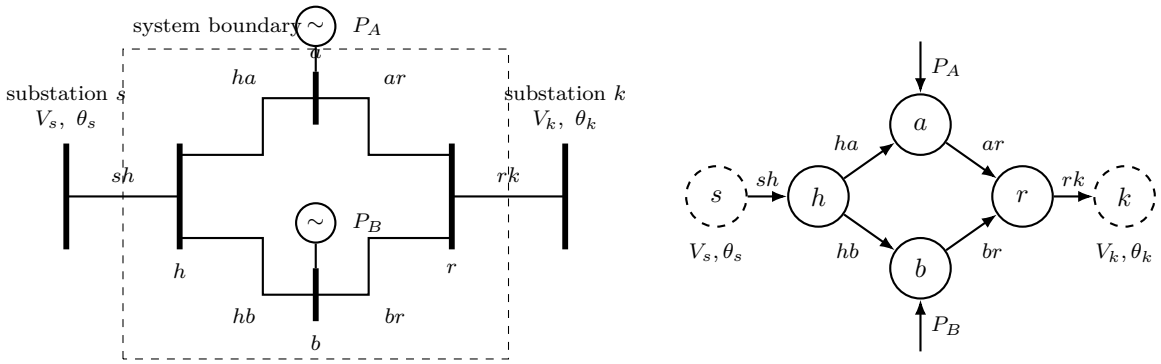


Figure 1.3: The ring feeder of Example 1.3. Left: two substations hold the two ends of a ring; between them two branches carry embedded generation. Right: the same ring as a conservation graph, with the two substation busbars as reservoirs. Every edge carries an active channel and a reactive channel, and the two domains use the same incidence matrix; the injections are active sources at the generator buses.

partly around the other side of the triangle. There is no way to compute F_{23} from a chain of local steps; it depends on the whole loop at once. That dependence, harmless here, is the thing that makes inference on this graph interesting in Part III.

The DC power flow. Because every closure here is linear, the flows can also be eliminated by hand, and that is how a power engineer usually solves such a network. The closure rows give $\mathbf{F} = -\mathbf{K}(\mathbf{A}^\mathsf{T}\boldsymbol{\theta} + \mathbf{A}_b^\mathsf{T}\theta_1)$. Substituting this into the balance rows, with $\theta_1 = 0$, leaves two rows in the two angles, and their solution:

$$B\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} P_2 \\ P_3 \end{pmatrix}, \qquad \begin{pmatrix} \theta_2 \\ \theta_3 \end{pmatrix} = \frac{1}{3B}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} P_2 \\ P_3 \end{pmatrix}, \tag{1.21}$$

after which $\mathbf{F} = -B\,\mathbf{A}^\mathsf{T}\boldsymbol{\theta}$. The matrix $B\,\mathbf{A}\mathbf{A}^\mathsf{T}$ on the left is the reduced susceptance matrix of the network, and this elimination is the DC power flow. The loop is visible in it. Without line e_{23} the incidence matrix would be its first two columns alone, $\mathbf{A}\mathbf{A}^\mathsf{T}$ would be the identity, and each angle would follow from its own injection; the off-diagonal -1 is the line that closes the loop, and it is why F_{23} depends on both injections at once. The elimination is a shortcut that linear closures allow, not a different model. The residual form contains it but does not require it, and once the closures stop being linear the shortcut is gone: only Newton's method on the five-row system remains.

Example 1.3 (A ring feeder carrying active and reactive power)**.** A ring is fed from two substations. From the first (node s), whose busbar is held at V_s and whose angle is θ_s, a circuit runs to a junction (node h). From h two branches run through buses a and b, which host embedded generators injecting P_A and P_B, to a second junction (node r), from which a circuit runs to the second substation (node k), held at V_k and θ_k. Figure 1.3 draws the arrangement and its graph. Two conserved quantities move through the same circuits, active and reactive power, and the point of the example is that they share one graph.

Unknowns. Four inside buses, each with a voltage and an angle, and six edges, each with a reactive flow Q_e and an active flow P_e: twenty unknowns in all.

Incidence. Rows h, a, b, r, columns sh, ha, hb, ar, br, rk:

$$\mathbf{A} = \begin{pmatrix} +1 & -1 & -1 & 0 & 0 & 0 \\ 0 & +1 & 0 & -1 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 & +1 & -1 \end{pmatrix}, \tag{1.22}$$

used twice, once for the reactive balances and once for the active balances, which is Principle 1.1 in a matrix.

Residuals. Eight rows come from the balance law, with the one incidence matrix serving both domains: a reactive balance and an active balance at each inside bus $n \in \{h, a, b, r\}$. The active balance carries the injection P_n as a source, with $P_a = P_A$, $P_b = P_B$ and zero elsewhere:

$$r_n^Q = -\sum_e A_{n,e}\, Q_e \quad \text{(reactive balance)}, \qquad r_n^P = -\sum_e A_{n,e}\, P_e - P_n \quad \text{(active balance)}.$$

Twelve more come from the closures, two on each edge e. The reactive closure is the gradient pattern of Table 1.2, written exactly as the feeder's. The active closure is bilinear: the active power on the edge is its susceptance, times the voltage of the bus it leaves, times the angle it falls through.

$$\begin{aligned} r_e^g &= Q_e - b_e\,(V_{\text{tail}} - V_{\text{head}}) && \text{(reactive closure)}, \\ r_e^c &= P_e - b_e\, V_{\text{tail}}\,(\theta_{\text{tail}} - \theta_{\text{head}}) && \text{(active closure)}, \end{aligned}$$

with the imposed values V_s, V_k, θ_s and θ_k wherever a tail or head is a reservoir. Twenty rows, twenty unknowns; Proposition 1.2 with $\Theta = C = R = 0$.

Vector form. As in Example 1.1, stack the unknowns flows first and the residuals balances first:

$$\mathbf{z} = \begin{pmatrix} \mathbf{q} \\ \mathbf{P} \\ \mathbf{v} \\ \boldsymbol{\theta} \end{pmatrix},$$

with $\mathbf{q}$ and $\mathbf{P}$ the reactive and active flows on the six edges and $\mathbf{v}$ and $\boldsymbol{\theta}$ the voltages and angles at h, a, b, r. The residual vector $\mathbf{F}(\mathbf{z})$ has four blocks, the two balances and then the two closures, and the model sets each to zero:

$$-\mathbf{A}\,\mathbf{q} = \mathbf{0}, \qquad -\mathbf{A}\,\mathbf{P} - \mathbf{P}_s = \mathbf{0}, \tag{1.23}$$

$$\mathbf{q} + \mathbf{B}\,(\mathbf{A}^{\mathsf{T}}\mathbf{v} + \mathbf{A}_b^{\mathsf{T}}\mathbf{v}_b) = \mathbf{0}, \qquad \mathbf{P} + \mathbf{B}\,\mathbf{D}_v\,(\mathbf{A}^{\mathsf{T}}\boldsymbol{\theta} + \mathbf{A}_b^{\mathsf{T}}\boldsymbol{\theta}_b) = \mathbf{0}. \tag{1.24}$$

The pieces are these.

- $\mathbf{P}_s = (0, P_A, P_B, 0)^{\mathsf{T}}$ holds the injections at the buses.
- $\mathbf{B} = \operatorname{diag}(b_{sh}, \ldots, b_{rk})$ holds the susceptances. The left closure is $Q_e = b_e\,(V_{\text{tail}} - V_{\text{head}})$ on every edge, the gradient pattern of Table 1.2 in the reactive domain, written exactly as the feeder's.
- $\mathbf{A}_b$ holds the reservoir rows of s and k, and $\mathbf{v}_b = (V_s, V_k)^{\mathsf{T}}$ their imposed voltages:

$$\mathbf{A}_b = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 \end{pmatrix}.$$

- The right closure is bilinear: the active power on an edge is its susceptance, times the voltage of the bus it leaves, times the angle difference across it. $\mathbf{D}_v = \operatorname{diag}(\mathbf{U}\mathbf{v} + \mathbf{u}_b\,\mathbf{v}_b)$ puts each edge's tail voltage on a diagonal, and $\mathbf{U}$ picks each edge's tail. Row e of $\mathbf{U}$ has a one in the column of e's tail when the tail is an inside bus. The one edge that leaves a reservoir, sh, takes the imposed substation voltage V_s through $\mathbf{u}_b$ instead:

$$\mathbf{U} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{u}_b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Solution. Take $V_s = 1.03$ and $V_k = 1.00$ per unit, $\theta_s = 0.02$ and $\theta_k = 0$ radians, susceptances of 5, 2, 3, 2, 3 and 5 per unit on the six edges in the order above, and injections $P_A = 0.20$ and $P_B = 0.40$ per unit.

The model is solved by Newton's method on $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, as the other two are, from a rough guess: every voltage 1.00 per unit and every angle and every flow zero. It converges in two steps. The largest residual falls from 4.0×10^{-1} to 4.2×10^{-3} and then to 3.3×10^{-13}; the Jacobian below says why so few steps are needed.

The voltages are $\mathbf{v} = (1.0225, 1.0150, 1.0150, 1.0075)^{\mathsf{T}}$ per unit, and the reactive flows are $\mathbf{q} = (0.0375, 0.0150, 0.0225, 0.0150, 0.0225, 0.0375)^{\mathsf{T}}$: 0.0375 through the ring, split 0.0150 through the upper branch and 0.0225 through the lower, in proportion to their susceptances. The two branches are symmetric dividers, so a and b sit at the same 1.0150, halfway between the junctions either side of them. The angles are $\boldsymbol{\theta} = (0.0737, 0.1181, 0.1344, 0.0642)^{\mathsf{T}}$ radians: every inside bus is ahead of both substations, because both generators are exporting. The active flows are $\mathbf{P} = (-0.2767, -0.0906, -0.1861, 0.1094, 0.2139, 0.3233)^{\mathsf{T}}$ per unit, the negative ones running back toward substation s. With two reservoir rows the column identity of Example 1.1 reads $\mathbf{1}^{\mathsf{T}}\mathbf{A} + \mathbf{1}^{\mathsf{T}}\mathbf{A}_b = \mathbf{0}$, and multiplying the active balances by $\mathbf{1}^{\mathsf{T}}$ gives $P_{rk} - P_{sh} = \mathbf{1}^{\mathsf{T}}\mathbf{P}_s$. Active power leaves across the boundary at 0.3233 and enters at -0.2767 — that is, leaves at both ends — and the difference is the 0.60 per unit the two generators inject. That is Proposition 1.1 in the active domain. The reactive balances, which have no sources, give $Q_{rk} = Q_{sh}$, the same proposition in the reactive domain.

The active closures are bilinear, $b_e V_{\text{tail}}\,\Delta\theta_e$, so the model is the first nonlinear one in the book. Its Jacobian, with the unknowns in the order $(\mathbf{q}, \mathbf{P}, \mathbf{v}, \boldsymbol{\theta})$ and the rows in the order of equations (1.23)–(1.24), is

$$\mathbf{J} = \begin{pmatrix} -\mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{B}\mathbf{A}^{\mathsf{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{B}\,\mathbf{D}_\theta\mathbf{U} & \mathbf{B}\,\mathbf{D}_v\mathbf{A}^{\mathsf{T}} \end{pmatrix}, \qquad \mathbf{D}_\theta = \operatorname{diag}(\mathbf{A}^{\mathsf{T}}\boldsymbol{\theta} + \mathbf{A}_b^{\mathsf{T}}\boldsymbol{\theta}_b),$$

where $\mathbf{D}_\theta$ holds the angle differences. The nonlinearity sits in the last row alone, in the blocks $\mathbf{B}\mathbf{D}_v\mathbf{A}^{\mathsf{T}}$ and $\mathbf{B}\mathbf{D}_\theta\mathbf{U}$, which carry the state. It is also one-directional. Put the reactive unknowns and rows first: no reactive row reads $\boldsymbol{\theta}$ or $\mathbf{P}$, so the Jacobian is block lower triangular, and its reactive block is constant. The first Newton step therefore solves the reactive problem exactly, whatever the guess: after it the voltages and reactive flows are right to 10^{-11}, while the angles are still 1.5×10^{-3} radians off. With $\mathbf{v}$ right, the active rows are linear in $\boldsymbol{\theta}$ and $\mathbf{P}$, and the

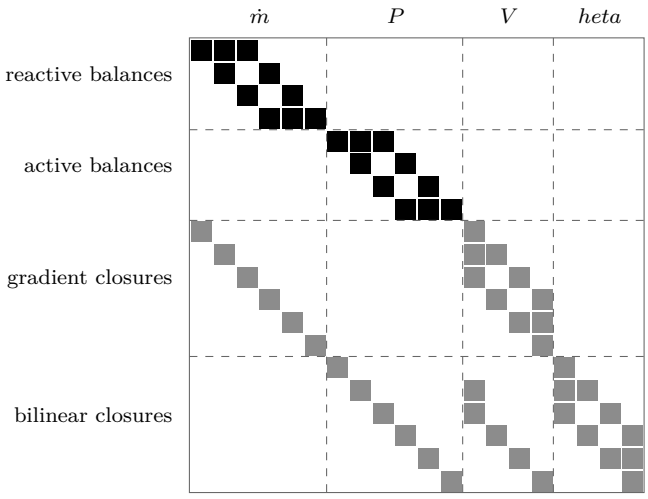


Figure 1.4: Nonzero pattern of the Jacobian of Example 1.3, rows in the fixed order of §1.5 and columns grouped by variable kind, flows first. Black entries are the constant ± 1 of the balance rows; grey entries carry parameters or the state. Fourteen percent of the matrix is nonzero, and the pattern was known before any number was.

second step solves them. This is the decoupling that every fast power-flow method exploits, appearing here as a property of the Jacobian rather than as an approximation. Figure 1.4 draws the Jacobian's pattern: 57 nonzeros in a 20×20 matrix, the balance rows constant, the bilinear rows reading a flow, a voltage, an angle and an active flow each.

Remark 1.1. All three examples have a Jacobian whose balance rows are constants and whose closure rows carry every parameter. All are square with full structural rank. All have a flow across the boundary that equals the net source inside. And in all, the unknowns could be reassigned, freeing a parameter and fixing a state, without changing a single row. The structure is the model. The numbers are an instance.

1.7 In practice

Draw the boundary first. Before any equation, decide which nodes are reservoirs. Everything the model will be unable to say follows from that decision, and so does everything it will be asked. Write the imposed condition at every port, potential or flow, and check that at least one port in every connected domain imposes a potential; a domain with only flows imposed has no level, as Exercise 1.1 shows.

One graph, several domains. Draw the nodes and edges once. Then, domain by domain, declare which nodes are balanced and which edges carry a channel. Do not draw a second graph for the second domain; the shared incidence matrix is what later lets a measurement in one domain inform the other.

Write every closure as a residual. Resist the assignment form. A closure written as $Q = UA(T_h - T_c)$ will be read as a computation, and a model that is a chain of computations

cannot be run backwards, cannot absorb a measurement, and cannot free a parameter. A closure written as $Q - UA(T_h - T_c) = 0$ can.

Count before solving. Proposition 1.2 takes a minute and catches the two commonest errors: a freed parameter with nothing to pin it, and a measurement with nothing to determine. Then check structural rank, which catches the subtler error of a subsystem that is square overall but has three rows on two unknowns in one place and two rows on three in another.

Check the cut. After any solve, sum the port flows of each domain and compare with the net source inside. The identity of Proposition 1.1 costs nothing to verify and fails loudly when a sign in the incidence matrix is wrong, which is the commonest error of all.

Keep the pattern. The Jacobian's nonzero pattern is fixed by the declared inputs of the closures. Record it. Everything that follows, the solver's ordering, the elimination of a subsystem onto its ports, the sparsity of the posterior precision, is a property of that pattern, and the pattern should be inspected once, as in Figure 1.4, before any of it is trusted.

1.8 What is missing

Every number in this chapter was known. The susceptances b_{12} and B were given; the substation voltage and the injections were imposed. Nothing was measured. Nothing failed.

List where a real model's ignorance actually sits.

- *Parameters.* The susceptance of a real line is a nominal value from a data sheet and a length on a drawing, correct to perhaps a few percent; its resistance depends on the conductor's temperature, which depends on what it is carrying.
- *Boundary conditions.* The substation voltage follows the tap changer and the wider grid, and is at best a forecast. The load at bus 2 is a schedule that the customers have not read.
- *Model form.* The line from bus 2 to bus 3 is either in service or it is not, and the closure is different in the two cases. A generator is either regulating its voltage or sitting on a reactive limit.
- *Measurements.* A real system has sensors. Some of its voltages and flows are observed, with noise, and the deterministic model has no place to put an observation. It takes imposed values and returns solved values; it cannot take a noisy reading of a solved value and revise its other solved values in the light of it.

The last item is the decisive one. An observation of V_2 in Example 1.1 is information about b_{12}, about G, about V_s, and about V_1. The deterministic model can use it only by choosing which one unknown it will determine. The probabilistic model uses it for all of them at once, in proportion to how much each is uncertain.

Chapter 2 rewrites $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ as a graph of relations, in which each row is a node in its own right that touches only the variables it reads. Chapter 3 attaches probability to that graph: a prior on each uncertain parameter and boundary condition, a likelihood on each measurement, and a factor for each row of $\mathbf{F}$. The joint distribution over every variable in the system is then a product,

$$p(\mathbf{z}) \;\propto\; \prod_{f} \varphi_f(\mathbf{z}_f), \tag{1.25}$$

in which each factor φ_f touches only a small set $\mathbf{z}_f \subset \mathbf{z}$. The set of factors is precisely the set of rows of this chapter, plus the priors and the sensors. Nothing about the structure changes. That is the claim the rest of the book makes good on.

Notes and further reading

The view of a physical system as conserved quantities at nodes, flows driven by potential differences along edges, and conjugate flow-and-potential pairs at ports, with the same structure across thermal, fluid, electrical and mechanical domains, is the bond-graph tradition begun by Paynter [70] and developed by Karnopp and Rosenberg [40]; what this chapter calls a port is their pair of effort and flow variables, and Table 1.3 is their table. Network thermodynamics [67] extended the same network structure to irreversible processes. The modern statement closest to §1.2 and §1.4, with an explicit incidence matrix, open graphs and compositional interfaces, is the port-Hamiltonian formulation on graphs of van der Schaft and Maschke [93]. The reader who knows any of these will recognize Chapter 1 as a restatement in their terms with the dynamics left for Chapter 16.

That a physical law is a relation and not an assignment (Principle 1.2) is the founding premise of acausal, equation-based modeling languages, of which Modelica [64] is the most widely used: its connectors carry potential variables that are equated and flow variables that are summed to zero at a connection, which is equation (1.1) generated automatically. The mathematical foundations of such languages, including the structural analysis that this chapter's well-posedness checks perform, are set out by Benveniste, Caillaud and Malandain [5]. Structural rank and the matching of equations to unknowns go back to Dulmage and Mendelsohn [20]; the same structural analysis applied to differential-algebraic systems is Pantelides' algorithm [69], and the numerical theory of such systems is in Brenan, Campbell and Petzold [12].

The sparsity argument of §1.5 and the claim that it is what makes everything downstream affordable are standard in the sparse-matrix literature; Davis [16] and Duff, Erisman and Reid [19] are the references, and the observation that power-network equations should be factored in an order chosen for sparsity is Tinney and Walker's [91]. The DC power flow of Example 1.2, its assumptions and its accuracy are reviewed by Stott, Jardim and Alsaç [86].

What is the book's own in this chapter is the emphasis: that the boundary is where uncertainty will enter, and that the exactness of cut flows (Proposition 1.1) will become the sufficiency of ports in Chapter 5.

Summary

- Three questions describe a conserved quantity in any system: where it accumulates, how it moves, what crosses the boundary. Their answers are nodes, edges and reservoirs.
- Nodes conserve. The balance law (1.1) is linear in the flows, has no parameters, and uses one incidence matrix shared by every domain.
- Edges close. A closure is a residual relation among a few variables; it does not assign a direction. Four patterns cover most of physics.
- The boundary is the set of edges to reservoirs. Each carries a port, a conjugate flow and potential pair. Cut flows are exact by conservation; cut potentials are not aggregable.
- The model is $\mathbf{F}(\mathbf{z}) = \mathbf{0}$: square, structurally full-rank, with sparsity linear in model size and a nonzero pattern fixed before any number is computed. It is square by construction,

and stays square exactly when every freed parameter is paid for by a row.
- Two domains on one graph share the incidence matrix; the ring feeder's twenty rows carry active and reactive power together, and the cut identity holds in each domain separately.
- Uncertainty sits in parameters, boundary conditions, model form and measurements. The deterministic model has no place for a measurement. The probabilistic model of Part II will, on the same graph.

Exercises

Exercise 1.1. In Example 1.1, replace the imposed substation voltage by an imposed reactive flow $Q_{2s} = \bar{Q}$ at the boundary, and free the source G. Write $\mathbf{z}$ and $\mathbf{F}$. Is the system square? Compute its structural rank and identify the direction in which the Jacobian is singular. What single additional imposed condition restores a well-posed model, and why must it be a potential?

Exercise 1.2. Add the active-power domain to Example 1.1 by giving each line an active flow alongside its reactive one, $P_e = b_e\,(\theta_{\text{tail}} - \theta_{\text{head}})$, with the substation angle θ_s imposed and a known active injection at node 1. Write the incidence matrix with both domains and confirm it is the same matrix. Which nodes need an active-power balance, and how many unknowns and rows does the model now have?

Exercise 1.3. In Example 1.2, remove line e_{23}. Solve for the angles and flows with $P_2 = -1.5$, $P_3 = -0.5$, and compare F_{12} with the looped case. Then remove line e_{13} instead. Which removal changes the boundary flow at bus 1, and why does Proposition 1.1 say it cannot?

Exercise 1.4. Make bus 1 of Example 1.2 an inside zero-storage node with a known injection P_1, and make there be no reservoir at all. Show that the resulting $\mathbf{F}$ is not structurally full-rank, identify the direction in which $\mathbf{J}$ is singular, and explain in words why a closed DC network cannot fix its own reference angle.

Exercise 1.5. Prove Proposition 1.1 for the dynamic case: show that the net flow across the boundary of S equals the net source inside S minus the rate of change of the total extensive state inside S. Which quantity does a controller outside S then need to know in addition to the port flows?

Solutions to selected exercises

Exercise 1.1. The unknowns are now $\mathbf{z} = (Q_{12}, Q_{2s}, V_1, V_2, G, V_s)$, six of them, and the rows are the four of Example 1.1, with G and V_s moved into the unknowns, plus the boundary row $Q_{2s} - \bar{Q} = 0$: five rows. The system is not square, and Proposition 1.2 says why: two quantities were freed (G and V_s, so $\Theta = 2$) and one row was added ($R = 1$). Its Jacobian,

$$\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -b_{12} & b_{12} & 0 & 0 \\ 0 & 1 & 0 & -b_{2s} & 0 & b_{2s} \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

has rank five, and its null direction is $(0, 0, 1, 1, 0, 1)$: add the same constant to V_1, V_2 and V_s and every row is unchanged, because every voltage enters only through a difference. The flows

are fully determined, $Q_{12} = Q_{2s} = G = \bar{Q}$, and so are the voltage differences $V_1 - V_2 = \bar{Q}/b_{12}$ and $V_2 - V_s = \bar{Q}/b_{2s}$; what is not determined is the level. One imposed potential, any one of the three voltages, restores a square system of full rank; an imposed flow could not, because every flow is already fixed and a sixth flow row would either repeat one or contradict it. This is the voltage form of the reference-angle problem of the next exercise but one — the same failure in the other electrical domain — and the general rule: a connected domain needs at least one port with an imposed potential, or its potentials float together.

Exercise 1.3. With $P_2 = -1.5$ and $P_3 = -0.5$ the looped network of Example 1.2 has $F_{12} = 1.167$, $F_{13} = 0.833$ and $F_{23} = -0.333$. Remove e_{23}: each load is served by its own line alone, $F_{12} = 1.5$, $F_{13} = 0.5$, and the angles are $\theta_2 = -1.5/B$, $\theta_3 = -0.5/B$. Remove e_{13} instead: bus 3 is served through bus 2, $F_{12} = 2.0$ and $F_{23} = 0.5$, with $\theta_2 = -2/B$ and $\theta_3 = -2.5/B$, the far bus now the lowest. In every case the boundary flow $F_{12} + F_{13}$ is 2.0, the total load. Neither removal can change it, because Proposition 1.1 with $S = \{2, 3\}$ says the flow across the cut is the net source inside, and removing an internal line changes the internal structure and nothing else. What the removals change is how the two lines from the reservoir share the boundary flow, from 1.17 and 0.83 to 1.5 and 0.5 or to 2.0 and nothing, and that is the difference between the port flow, which conservation fixes, and the potentials and internal flows, which the closures fix and any change inside can move.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

The radial chain, by hand *conservation at a node; the chain has no loop so flows follow from balance alone; potential as the accumulated drop*
`foundation_electrical/dc_power_flow_audit/problems/radial_chain_by_hand`

The IEEE 14-bus case, from a flat start *two conserved quantities on the same graph; the slack absorbs the loss it cannot know in advance; a nonlinear solve needs a convergence tolerance where a linear one does not*
`subsystem_power_flow/ieee14_ac_pf/problems/ac_from_a_flat_start`

Where the losses are *loss as a residual of two flows on the same edge; loss concentrated on few branches; resistance against current squared*
`subsystem_power_flow/ieee14_ac_pf/problems/where_the_losses_are`

Four ledgers, one solve *several conserved quantities on one graph; an algebraic subsystem coupled to a differential one; efficiency loss as the coupling term*
`power_grid/battery_electrothermal_cosim/problems/four_ledgers_one_solve`

The conductor sets its own limit *a rating as the solution of a heat balance rather than a property of the wire; convection dominating over ambient; what a static rating is assuming*
`transmission/dlr_n11_value/problems/the_conductor_sets_its_own_limit`

Chapter 2

From equations to factor graphs

Chapter 1 ended with a system of equations, $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, and a promise: that its structure is also the structure of every probabilistic question about the system. This chapter makes the structure visible. It draws the equations as a graph, compares the three graphical languages in which such a graph can be drawn, and argues that one of them, the factor graph, is the right one for physics. The argument turns on Principle 1.2 of the previous chapter: a closure relates its variables and does not compute one from the others.

There is still no probability in this chapter. Every factor is a relation that either holds or does not. Chapter 3 replaces "holds or does not" by "holds to within so much", and nothing drawn here will need to be redrawn.

2.1 The rows as a graph

Each row of $\mathbf{F}$ reads a few variables and ignores the rest. That pattern of reading is a graph.

Definition 2.1 (Factor graph of a residual system)**.** The factor graph of $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ is a bipartite graph with one *variable node* for each component of $\mathbf{z}$, one *factor node* for each row of $\mathbf{F}$, and an edge between a factor node and a variable node exactly when that row reads that variable. The set of variables a factor reads is its *scope*.

The factor graph is nothing more than the sparsity pattern of the Jacobian $\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{z}$ drawn as a graph: row k is adjacent to column j when J_{kj} is structurally nonzero. It was already known at compile time in §1.5, before any number was computed. What the drawing adds is that paths and loops become visible.

Figure 2.1 draws the export feeder of Example 1.1. Two things are worth reading off it before going further.

First, the physical graph and the factor graph are different graphs of the same system. The physical graph had three nodes and two edges. The factor graph has four variable nodes and four factor nodes. A physical node became a balance factor. A physical edge became a flow variable together with a closure factor. The correspondence is systematic and is tabulated in §2.3; the point here is only that the two drawings answer different questions. The physical graph says where things are. The factor graph says which equations constrain which unknowns.

Second, the factor graph has a cycle, although the physical graph is a chain. Follow $V_2 \to c_{12} \to Q_{12} \to n_2 \to Q_{2s} \to c_{2s} \to V_2$. Each step is an edge of the figure. The cycle exists because there are two routes by which V_2 and Q_{2s} are tied together: directly, through the closure

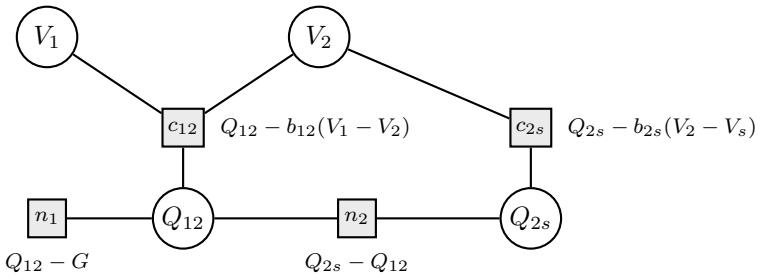


Figure 2.1: The export feeder of Example 1.1 as a factor graph. Circles are the four unknowns; squares are the four rows of $\mathbf{F}$. The two balance rows n_1, n_2 read only flows. The two closures c_{12}, c_{2s} read a flow and the voltages at the ends of their line. The known injection G and the imposed substation voltage V_s are constants inside their factors, not nodes.

of the line into the substation, and indirectly, through the closure of the line above it and the balance at bus 2. Both hold at once. This is not a defect of the drawing. It is a fact about the equations, and it has consequences for inference that Part III will have to face: the factor graph of a physically tree-shaped system is not, in general, a tree.

2.2 Three languages

A factor graph is one of three standard ways to draw a function that splits into local pieces. All three were invented for probability distributions, and Chapter 3 will use them that way. But each is defined by the shape of a factorization, not by what the factors mean, so they can be compared now, on the deterministic system, where the comparison is sharpest.

Throughout this section, a *factorization* of a function Φ of many variables is a way of writing it as a product of functions, each of which reads only a subset of the variables. For the residual system the function is the indicator "every row is satisfied", which is the product over rows of the indicator "this row is satisfied". Each row is a local piece. The three languages differ in what they draw and what they leave out.

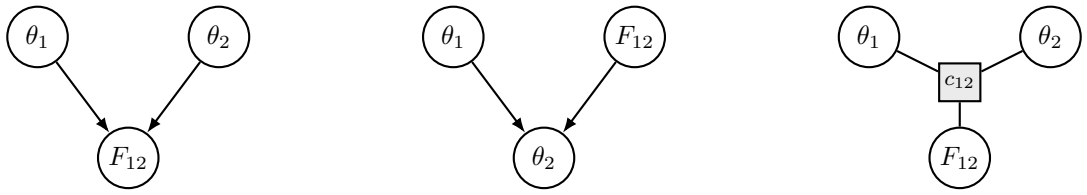
2.2.1 Directed graphs

A *directed graphical model*, or Bayesian network, draws each variable as a node and each factor as a set of arrows into one variable from the variables it depends on. The factorization it encodes is

$$\Phi(X_1, \ldots, X_n) \;=\; \prod_i \phi_i(X_i \mid \mathrm{pa}(X_i)), \tag{2.1}$$

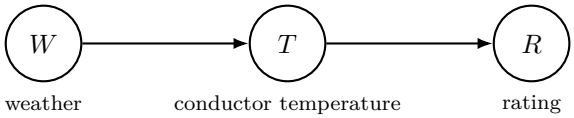
one factor per variable, each reading that variable and its *parents* $\mathrm{pa}(X_i)$, the sources of the arrows into it. The graph must have no directed cycle. When the factors are probabilities the form is $P(A)\,P(B \mid A)\,P(C \mid B)$ for the chain $A \to B \to C$, and the arrows read as "given".

The directed language is the natural one when the system has a generative story: something happens first, and something else depends on it. Weather drives the temperature of a conductor; the conductor's temperature sets its rating. The chain



(a) flow from angles (b) angle from flow (c) the relation itself

Figure 2.2: Three drawings of one closure, $F_{12} - B(\theta_1 - \theta_2) = 0$. Panels (a) and (b) are directed and each commits to a computation; they encode different factorizations of the same relation and are both correct. Panel (c) is a factor graph and commits to nothing beyond the relation.



is a faithful picture. Each arrow points from a cause to an effect, the factorization $\phi(W)\,\phi(T \mid W)\,\phi(R \mid T)$ is the story told in order, and nobody would draw it the other way.

2.2.2 The directionality problem

Now try to draw a closure in the same language. The DC line closure of Example 1.2 is

$$F_{12} = B\,(\theta_1 - \theta_2). \tag{2.2}$$

The directed language needs one variable to be the child and the rest to be its parents. The textbook arrangement is $\theta_1 \to F_{12} \leftarrow \theta_2$, with B as a third parent if it is uncertain. But Principle 1.2 says the physics asserts no such thing. The closure is equally $\theta_2 = \theta_1 - F_{12}/B$, which is $\theta_1 \to \theta_2 \leftarrow F_{12}$, or $B = F_{12}/(\theta_1 - \theta_2)$, which is a third orientation. Figure 2.2 draws the first two.

Each orientation is a correct picture of the same relation, and each encodes a different way of computing it. So the directed drawing is not a property of the physics. It is a property of the physics together with a choice of which quantities are known. That choice changes with the question, and the drawing changes with it.

This might be tolerable if some orientation always existed. It does not. Consider what an orientation of the whole system means: every row is assigned one output variable, no two rows share an output, and the arrows from inputs to outputs contain no directed cycle. An orientation with those properties is exactly an order in which the equations can be solved one at a time by substitution. Each row, taken in order, has all its inputs already computed and delivers its output.

For the export feeder such an order exists. Read Figure 2.1: n_1 delivers $Q_{12} = G$; n_2 then delivers $Q_{2s} = Q_{12}$; c_{2s} then delivers V_2; c_{12} finally delivers V_1. The directed drawing is $G \to Q_{12} \to Q_{2s} \to V_2 \to V_1$, and it is the order in which the example was solved by hand. But now change the boundary condition. Impose the reactive flow $Q_{2s} = \bar{Q}$ into the substation instead of the substation voltage (Exercise 1.1). Then c_{2s} must deliver V_s, n_2 must deliver Q_{12} from Q_{2s}, and n_1 must deliver G. Every arrow reverses. The same physics, the same rows, and a different directed graph, because a different quantity was imposed at the boundary.

For the DC triangle no order exists at all. Try to build one. The balance at bus 2 reads F_{12} and F_{23} and must deliver one of them; the balance at bus 3 reads F_{13} and F_{23} and must deliver one of them. Suppose bus 2 delivers F_{12} and bus 3 delivers F_{13}. Then the closure c_{12} must deliver θ_2 from F_{12}, the closure c_{13} must deliver θ_3 from F_{13}, and c_{23} must deliver F_{23} from θ_2 and θ_3. But bus 2 needed F_{23} to deliver F_{12}, and F_{23} needs θ_2, which needs F_{12}. That is a directed cycle. Every other assignment closes a cycle the same way; Exercise 2.2 asks for the full check. The loop in the network is an irreducible block of two equations in two unknowns, the 2×2 system of equation (1.21), and no sequence of single-equation substitutions solves it. A directed graph with one variable per node cannot represent it without first collapsing the block into one node that holds both angles, which is to say without giving up on drawing the structure at all.

The argument can be made exact, and the exact form is the one that modeling compilers use.

Proposition 2.1 (When an order of computation exists)**.** *A square residual system admits an order of single-row substitutions if and only if there is an assignment of one distinct output variable to each row such that the dependency graph, with an arrow from every input of a row to its output, has no directed cycle. Equivalently, the system's rows decompose into irreducible blocks, sets of rows that must be solved simultaneously, and an order exists exactly when every block has one row.*

Proof. If an order exists, take each row's output to be the variable it delivers; the outputs are distinct because each variable is delivered once, and the dependency graph has no cycle because every arrow points from a variable delivered earlier to one delivered later. Conversely, given such an assignment with an acyclic dependency graph, a topological order of that graph orders the outputs, and hence the rows, so that every row's inputs have been delivered before it is reached. For the second statement, fix any assignment, which is a perfect matching of rows to variables in the bipartite factor graph, and draw an arrow from row r to row s when s reads the variable matched to r. The strongly connected components of this graph are the blocks; they do not depend on which perfect matching was chosen, a fact due to Dulmage and Mendelsohn, and a block with more than one row is a directed cycle among rows that no matching removes. An order exists exactly when every component is a single row. □

The proposition turns the hand argument into a count, and the counts are in the chapter's script, `ch02_factor_graphs.py`, and in Table 2.1. The export feeder has one assignment of outputs to rows, and its dependency graph is acyclic: the order found above is the only one. The DC triangle has three assignments, and every one has a directed cycle; its five rows form a single irreducible block. The ring feeder of Example 1.3, with twenty rows, has exactly two blocks, of ten rows each. One holds the four reactive balances and the six gradient closures; the other holds the four active balances and the six bilinear closures. Neither splits further, because a ring is a loop in both domains at once: the reactive flows divide between the two branches in a proportion no single row fixes, and so do the active flows. The two blocks are solved one after the other rather than together — the reactive block reads no active unknown, which is why the Newton iteration of Example 1.3 settles the voltages on its first step — but neither of them is a sequence of substitutions.

Principle 2.1. *A directed graph encodes an order of computation. Physics supplies relations, not an order. Where an order exists it is fixed by the choice of boundary conditions and changes when they change; where the physics has a loop, no order exists.*

Table 2.1: The three examples of Chapter 1 as factor graphs: rows and variables, edges of the factor graph (nonzeros of $\mathbf{J}$), its cycle rank (edges minus nodes plus one), the sizes of the irreducible blocks of Proposition 2.1, and the edges of the Markov graph (off-diagonal nonzeros of $\mathbf{J}^\mathsf{T}\mathbf{J}$).

system	rows	variables	edges	cycle rank	blocks	Markov edges
export feeder	4	4	8	1	$1, 1, 1, 1$	5
DC triangle	5	5	11	2	5	7
ring feeder	20	20	57	18	$10, 10$	57

2.2.3 Undirected graphs

An *undirected graphical model*, or Markov random field, draws each variable as a node and joins two variables by an edge whenever they appear together in some factor. The factorization it encodes is

$$\Phi(X_1, \dots, X_n) \;=\; \prod_C \psi_C(X_C), \tag{2.3}$$

a product over *cliques* C, sets of mutually adjacent variables, of functions ψ_C that read only the variables in the clique. There are no arrows. A factor reads "these variables are jointly compatible", not "this one is computed from those". That is already the right reading for a closure, and it is why the undirected language is closer to physics than the directed one.

It still forgets something. The undirected graph of the DC line closure is a triangle on θ_1, θ_2, F_{12}: all three are pairwise adjacent because all three appear in one factor. But the same triangle is drawn for three separate pairwise relations, one on each pair, and for one three-way relation, and for a three-way relation together with a pairwise one. The graph records which variables interact and loses how many relations there are and which variables each one reads. For the DC triangle, where two closures both read θ_2 and two balances both read F_{23}, the undirected drawing merges relations that the physics keeps distinct. Since the relations are the physics, this is the wrong thing to forget.

2.2.4 Factor graphs

A *factor graph* is Definition 2.1 with the factors allowed to be any local functions:

$$\Phi(X_1, \dots, X_n) \;=\; \prod_f \varphi_f(X_f), \tag{2.4}$$

where X_f is the scope of factor f. Variables are circles, factors are squares, and a square is joined to exactly the circles in its scope. Nothing is oriented and nothing is merged. Panel (c) of Figure 2.2 is the DC line closure in this language: one square, three circles, and no claim about which is computed from which.

The factor graph keeps what the other two languages lose. It keeps the identity of each relation, which the undirected graph merges into cliques. It keeps the absence of direction, which the directed graph is forced to invent. And it is exactly the sparsity pattern of the Jacobian, so it costs nothing to construct: it is already there in the compiled model.

Principle 2.2. *A factor graph separates variables from the relations among them, and draws both. It is the language in which a residual system is its own picture.*

Table 2.2: What each physical object becomes in the factor graph, and what it will become when probability is attached.

Physical object	Residual system	Factor graph	Part II role
Balanced or zero-storage node	one balance row per tracked domain	a factor whose scope is the incident flows (and the node's storage state)	conservation factor
Edge channel	one flow variable and one closure row	a variable node and a factor whose scope is the flow, the end potentials, and the parameters	closure factor
Reservoir	a substituted constant, or an explicit boundary row	a constant inside the neighboring factors, or a unary factor on one variable	prior on a boundary condition
Freed parameter	a column of **J**	a variable node in the scope of every closure that reads it	prior on a parameter
Sensor	none	none yet	likelihood factor

Directed and undirected graphs are not wrong, and both will return. A directed graph is the right picture for a genuinely generative piece of a model, such as the weather chain above, and Chapter 3 attaches such pieces to the factor graph as priors. An undirected graph is the right picture for asking which variables are conditionally independent of which, and Chapter 5 uses it for exactly that. But the model is built, and inference is organized, on the factor graph.

2.3 The physical system as a factor graph

The correspondence noticed in Figure 2.1 is general. Table 2.2 states it: each object of the physical graph of Chapter 1 becomes a definite set of nodes in the factor graph. The last column anticipates Part II.

Two remarks on the table.

A reservoir can be drawn either way. Substituting its value makes it vanish into the factors that read it, as V_s vanished into c_{2s} in Figure 2.1. Keeping it as a variable with an explicit row $z_k - \bar{z}_k = 0$ makes it a variable node with one unary factor. The two are equivalent now, and the second becomes the natural one the moment $\bar{z}_k$ is uncertain, because the unary factor is then simply replaced by a prior. The factor graph is drawn once, and only the meaning of one square changes.

A sensor has no row in the deterministic system, which is the gap §1.8 identified. In the factor graph it will be a unary factor on the variable it observes. That is the whole of the change Chapter 3 makes to the structure: sensors are added as squares of degree one. The rest of the graph is untouched.

Figure 2.3 draws the DC triangle. It has the cycle traced in thick lines, a second cycle through θ_3 on the right, and a long cycle that goes around the whole figure. The network's single physical loop has become several loops in the factor graph, and the export feeder, which had no physical loop, had one. Counting physical loops does not predict the shape of the factor graph.

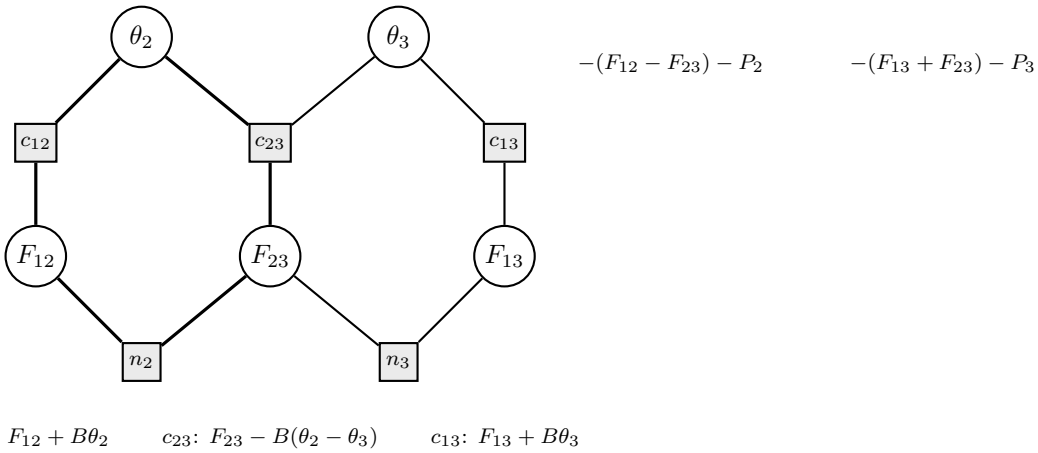


Figure 2.3: The DC triangle of Example 1.2 as a factor graph: five unknowns, five rows. The reference angle $\theta_1 = 0$ and the injections P_2, P_3 are constants inside their factors. The thick edges trace one of the graph's cycles, $\theta_2 \to c_{12} \to F_{12} \to n_2 \to F_{23} \to c_{23} \to \theta_2$; there are others.

2.3.1 A bus is not a variable

The tempting simplification is that a physical node is a variable and a physical edge is a factor. It is a useful first intuition and it is wrong in both halves.

A physical bus in the DC model has one unknown, its angle, and one balance row; that is the only case where the simplification is nearly right, and even there the balance is a factor, not a variable. In an AC model the same bus has an angle and a voltage magnitude, a real and a reactive injection, and two balance rows. A thermal node has a temperature and possibly a stored energy and a storage row. A physical node becomes several variable nodes and one or more factor nodes.

A physical line in the DC model has one flow and one closure. A line whose susceptance is uncertain adds a parameter variable. A line whose rating depends on its temperature, and whose temperature depends on the weather, adds a temperature variable, a thermal closure and a rating closure. A line that may be out of service adds a discrete availability variable, to which Chapter 7 is devoted. A physical edge becomes as many variable nodes and factor nodes as the physics of the edge requires.

Principle 2.3. *The physical topology guides the factorization; it does not dictate a one-to-one map. The closures decide which variables exist and which factors read them.*

2.3.2 Two domains on one graph

The ring feeder of Example 1.3 shows the map at its richest. Its factor graph has forty nodes, twenty variables and twenty factors, joined by fifty-seven edges, one per nonzero of the Jacobian of Figure 1.4. Figure 2.4 draws the part of it that belongs to one edge of the physical graph, the branch from bus h to bus a, together with the balances at its ends. The pattern repeats for every edge.

The figure makes two points. The two domains are two layers of one graph, joined only through the voltages, which the bilinear closures read; a measurement of the active power on the

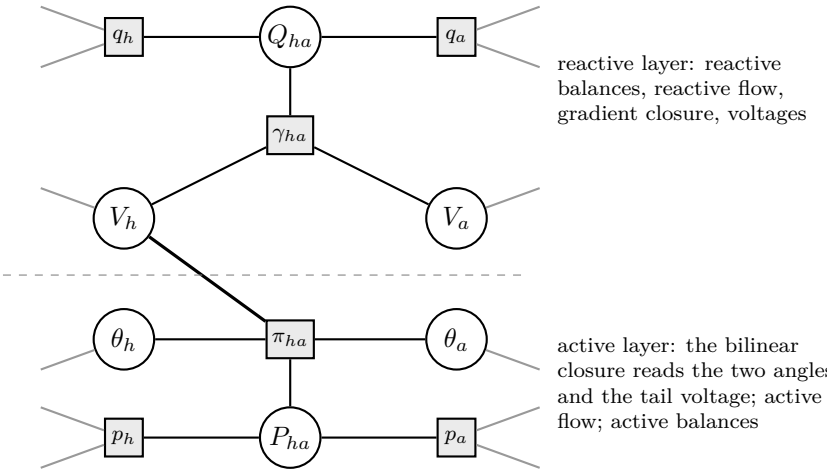


Figure 2.4: One branch of the ring feeder as a factor graph. Above the dashed line, the reactive domain: the gradient closure γ_{ha} reads the two voltages and the reactive flow, and the reactive balances q_h, q_a read the flow. Below it, the active domain: the bilinear closure π_{ha} reads the active flow, the two angles, and the voltage of the bus the branch leaves; the active balances read the active flow. The tail voltage V_h is the only variable in both layers, and the heavy edge is the only one that crosses the line. It is where the domains couple, and it is why an active-power measurement can inform a voltage.

branch constrains V_h through π_{ha}, and V_h constrains the rest of the voltages through γ_{ha} and the closure of the line above it. That path exists because the incidence matrix is shared, and it runs in one direction only: no reactive row reads an angle or an active flow, so the coupling is a one-way street. And the graph is loopy in both layers: each carries the loop of the two parallel branches, and Proposition 2.1 found each as a block of ten rows. The cycle rank of the whole, edges minus nodes plus one, is eighteen.

2.4 The factor graph is not unique

Definition 2.1 draws one factor per row of $\mathbf{F}$. That is the finest factorization the model offers, and it is the canonical one for constructing the model. It is not the only one, and it is not always the one to compute with.

Take the DC triangle and eliminate the flows. Substitute each closure into the balances that read its flow, as was done to reach equation (1.21). The result is two rows in two unknowns,

$$k_2(\theta_2, \theta_3) = 2B\theta_2 - B\theta_3 - P_2, \qquad k_3(\theta_2, \theta_3) = -B\theta_2 + 2B\theta_3 - P_3, \tag{2.5}$$

and its factor graph has two variable nodes and two factor nodes, each factor reading both variables (Figure 2.5). The solution set is unchanged: the angles that satisfy $k_2 = k_3 = 0$ are exactly the angles of the five-row solution, and the flows are recovered from them by the closures. Three variables and three factors have disappeared from the drawing.

The same operation on the export feeder eliminates Q_{12} and Q_{2s} and leaves two rows in V_1, V_2, each reading both. Both reduced graphs still have a cycle, a four-cycle through the two factors, because two factors read the same pair. Merge the two factors into one, which is always allowed since a product of two local functions with the same scope is one local function with that

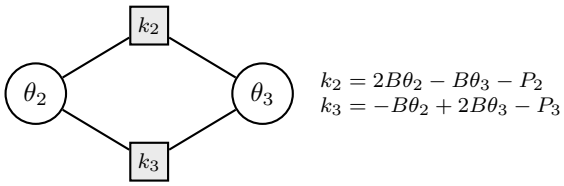


Figure 2.5: The DC triangle after eliminating its flows: the nodal form. Two variables, two factors, each reading both. It is a coarser factorization of the same solution set as Figure 2.3, and it still has a cycle, of length four, because two factors share the same scope.

scope, and the feeder's reduced graph is finally a tree: one factor on $\{V_1, V_2\}$. The triangle's reduced graph, merged, is a single factor on $\{\theta_2, \theta_3\}$, also a tree, and also trivial.

The ring feeder shows the same operation on a larger graph, and shows what it does to the Markov graph. Eliminate the six reactive flows from the reactive layer: substitute each gradient closure into the two reactive balances that read its flow. What remains is four rows on the four voltages, each row reading its own bus's voltage and its neighbors', with the two substation voltages as constants. That is the weighted Laplacian $\mathbf{ABA}^\mathsf{T}$ of Example 1.3, and the pattern of its off-diagonal entries is the ring itself, a four-cycle: h–a, h–b, a–r, b–r.

The Markov graph is denser than that, and the gap is the point. Before elimination the four voltages already had those same four adjacencies, because every gradient closure reads the two voltages at the ends of its line. After elimination they have six: the balance at h now reads V_a and V_b in one row, which makes them adjacent, and the balance at a reads V_h and V_r. The two new edges, a–b and h–r, are the diagonals of the ring, and they are in no physical line. Elimination created them by joining every pair of variables that shared a factor with an eliminated one. That is fill, the subject of Chapter 5: a row that couples a bus to both its neighbors couples the neighbors to each other, and on four buses that is enough to complete the graph. The same substitution in the active layer, with the voltages known, leaves four rows on the four angles with exactly the same pattern and exactly the same two edges of fill. The two domains are solved one after the other, and each pays the same price.

None of these is the wrong graph. Each is a factorization of the same function, and equation (2.4) is satisfied by all of them. They differ in how many variables they expose and how finely they split the relations, and those differences will govern the cost of inference. Chapter 5 shows that the cost of exact inference depends on a property of the factor graph called its treewidth, and treewidth is a property of the drawing, not of the solution set. Choosing the factorization is choosing the treewidth.

Principle 2.4. *The row-per-factor graph is canonical for building a model. Coarser factorizations, obtained by eliminating variables and merging factors, describe the same solution set with a different graph, and the graph is what inference pays for.*

There is a physical reading of elimination that is worth stating now, because Chapter 8 builds on it. Eliminating the flows of the triangle produced the reduced susceptance matrix, which is what a power engineer writes down first. Eliminating everything inside a subsystem except its ports produces a factor on the ports alone: the subsystem as its neighbors see it. That factor is the algebraic form of Proposition 1.1, and computing it is the Schur complement of Chapter 8. The boundary of Chapter 1 is a separator in the factor graph, and eliminating the interior is how a region is summarized to the rest of the system.

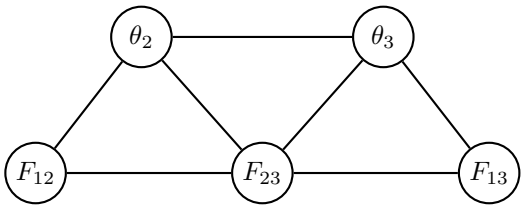


Figure 2.6: The Markov graph of the DC triangle in row-per-factor form. Each edge is a pair of variables that some factor reads together; the closure c_{23}, with scope of size three, contributes a triangle. The graph is the sparsity pattern of $\mathbf{J}^\mathsf{T}\mathbf{J}$ for the Jacobian of Example 1.2, and Chapter 6 will show it is the sparsity pattern of the posterior precision.

2.5 Vocabulary

The rest of the book uses a small set of graph-theoretic terms about the factor graph. They are collected here.

Scope. The set of variables a factor reads. Written $\mathbf{z}_f$ for factor f.

Neighbors. The factors that read a variable, or the variables a factor reads. The two are the two ends of the bipartite edge set.

Degree. The number of neighbors. A prior or a sensor is a factor of degree one. A balance factor has degree equal to the number of incident edge channels plus one for a storage state.

Path, cycle, tree. As in any graph. A factor graph with no cycle is a tree; a factor graph with a cycle is *loopy*. All three examples of this chapter are loopy in row-per-factor form.

Cycle rank. The number of independent cycles, which for a connected graph is the number of edges minus the number of nodes plus one; zero for a tree. Table 2.1 lists it for the examples.

Markov graph. The undirected graph on the variables alone, in which two variables are adjacent when some factor reads both. It is the sparsity pattern of $\mathbf{J}^\mathsf{T}\mathbf{J}$, and it is the graph on which conditional independence is read in Chapter 5.

Separator. A set of variables whose removal disconnects the Markov graph. The ports of a subsystem are a separator between its interior and the rest of the system.

The Markov graph deserves more than a sentence, because the identification with $\mathbf{J}^\mathsf{T}\mathbf{J}$ is not a coincidence.

Proposition 2.2 (The Markov graph is the pattern of $\mathbf{J}^\mathsf{T}\mathbf{J}$)**.** *Two variables are adjacent in the Markov graph of a residual system if and only if the corresponding off-diagonal entry of* $\mathbf{J}^\mathsf{T}\mathbf{J}$ *is structurally nonzero.*

Proof. The entry $(\mathbf{J}^\mathsf{T}\mathbf{J})_{ij} = \sum_k J_{ki}J_{kj}$ is a sum over rows. A term is structurally nonzero exactly when row k reads both variable i and variable j, that is, when both are in the scope of factor k. The sum is structurally nonzero when at least one such row exists, which is the definition of adjacency in the Markov graph. (Numerical cancellation between rows could make the entry zero by accident; the structural statement ignores that, as sparsity analysis always does.) □

Figure 2.6 draws the Markov graph of the DC triangle: five variables, seven edges, one per pair that shares a factor, and the script confirms that the same seven pairs are the off-diagonal nonzeros of $\mathbf{J}^\mathsf{T}\mathbf{J}$. Chapter 6 shows that $\mathbf{J}^\mathsf{T}\mathbf{J}$, suitably weighted, is the precision matrix of the Gaussian posterior. The Markov graph of the physics is therefore the sparsity pattern of the

posterior's precision. That single fact is what makes inference on a physical system affordable at scale, and it was visible in this chapter, before any probability was introduced.

2.6 In practice

Build the row-per-factor graph and keep it. It is the Jacobian's pattern and costs nothing. Coarser graphs are derived from it when a solver needs them; the fine one is the record of what the model asserts.

Name every factor after its row. A factor called "balance at node 7" or "closure on line 12–13" can be pointed at when a check fails, a residual is large, or a measurement disagrees. A factor called φ_{41} cannot. The names of Table 2.2 are the vocabulary the rest of the book uses.

Keep reservoirs as unary factors. Substituting an imposed value into its neighbors is tidy and loses the place where the value will later become a prior. A variable with one square attached is the form that Chapter 3 needs, and it costs one node.

Do not let a bus be a variable. The commonest error of a reader arriving from graphical models is one circle per physical node and one square per physical edge. It is wrong for every model with more than one unknown per node, which is every model beyond the DC approximation, and it hides the closures, which are where the physics and the parameters live.

Expect loops. A physically tree-shaped system has a loopy factor graph whenever a variable is tied to another by two routes, which is always, since a flow appears in a closure and in a balance at each end. Do not build inference on the assumption of a tree; build it on separators, which Chapter 5 supplies.

Run the block decomposition. Proposition 2.1's blocks say which parts of a model can be solved by substitution and which must be solved simultaneously, and modeling compilers compute them in linear time. A block that is unexpectedly large is usually a modeling error, a missing boundary condition or a closure written on the wrong variables, and the block names the rows involved.

2.7 What is missing

The factors of this chapter are indicators. Each is one where its row is satisfied and zero where it is not, and their product is one on the solution set and zero elsewhere. That is a legitimate factorization and the graph is a legitimate factor graph, but it is not yet a model of anything uncertain. The solution set of a square, well-posed system is a point.

Chapter 3 keeps every node and every edge of the factor graph and changes what the squares mean. A balance factor becomes "the balance holds to within a residual of this scale". A closure factor becomes the same for its relation. A reservoir's unary factor becomes a prior. A sensor, drawn for the first time, is a factor of degree one that says how far its reading may be from the variable it observes. The product of the squares is then a function that is large where every relation nearly holds and small elsewhere, and after normalization it is a probability

distribution over the whole state of the system. Every question in Part IV is a question about that distribution, and every one is answered on this graph.

Notes and further reading

Factor graphs in the form used here, a bipartite graph of variables and local functions with the product of the functions as the object of study, were defined by Kschischang, Frey and Loeliger [48]; Loeliger's tutorial [56] is the gentlest introduction, and Dellaert and Kaess [17] show them used for large sparse estimation problems in robotics, which is the closest engineering use to this book's. Directed and undirected graphical models, their factorizations and the relations between them are treated in full by Koller and Friedman [46] and Lauritzen [50]; the point of §2.2 that a directed model can represent a relation only after an orientation has been chosen, and that different orientations encode different factorizations of the same joint, is standard there.

The directionality argument of §2.2.2 is the book's, but its ingredients are not. That a system of equations admits an order of forward substitution exactly when its bipartite equation-variable graph has a matching whose induced dependency graph is acyclic, and that a looped system decomposes into irreducible blocks, is the Dulmage–Mendelsohn decomposition [20]; equation-based modeling languages use it, together with Pantelides' algorithm [69], to decide how to solve a model, and Benveniste, Caillaud and Malandain [5] give the theory. The chapter's contribution is to connect that decomposition to the choice of graphical language: a directed graph is available exactly when the decomposition has no block larger than one.

The bipartite graph of a residual system as the sparsity pattern of its Jacobian, and elimination on it as a graph operation, are the standard graph view of sparse direct methods [16, 78]. The identification of the Markov graph with the pattern of $\mathbf{J}^{\mathsf{T}}\mathbf{J}$ is the same observation as the graph of the normal equations in least-squares theory. The insistence that a physical bus is several variables and several factors is the book's, and it matters because the alternative, one variable per bus and one factor per line, is the model that a reader arriving from the graphical-model side would draw first.

Water distribution networks have been drawn as factor graphs in exactly the sense of §2.3. Han, Eguchi, Mehrotra and Venkatasubramanian [33] write the mass balance at each junction and the Hazen–Williams head loss along each pipe as factors, with the heads and the pipe flows as variables, and estimate the state of a damaged network from a few pressure and flow sensors; Irofti, Romero-Ben, Stoican and Puig [36] chain two such graphs, one estimating the leak-free state over a window of readings and one placing a leak from the residuals of the first, and compare the result with interpolation and Kalman-filter baselines on published networks. Both build on the sparse nonlinear least-squares machinery that Dellaert and Kaess [17] describe for robotics. The point of §2.3 that a physical node is several variables and several factors is visible in both: a junction contributes a head, a demand and a balance factor, a pipe a flow and a closure factor, and a sensor a unary factor on one of them.

Summary

- The factor graph of a residual system is its Jacobian's sparsity pattern drawn as a bipartite graph: variables as circles, rows as squares.
- Physical nodes become balance factors; physical edges become a flow variable and a closure factor; reservoirs become constants or unary factors; sensors will become unary factors. A

bus is not a variable.
- Of the three graphical languages, the directed one encodes an order of computation that physics does not supply and that loops forbid; the undirected one merges distinct relations into cliques; the factor graph keeps both the relations and their lack of direction.
- An order of computation exists exactly when every irreducible block has one row. The feeder has one order; the triangle has three candidate assignments, all cyclic; the ring feeder has two blocks of ten, one per domain, because a ring is a loop in both.
- Two domains are two layers of one factor graph, coupled through the variables the bilinear closures read.
- A physically tree-shaped system can have a loopy factor graph. Counting physical loops does not predict the factor graph's shape.
- The factor graph is not unique. Eliminating variables and merging factors gives coarser factorizations of the same solution set, and the choice governs the cost of inference.
- The Markov graph of the physics is the sparsity pattern of $\mathbf{J}^\mathsf{T}\mathbf{J}$, which Part II will identify as the posterior precision.

Exercises

Exercise 2.1. Draw the factor graph of a tee bus. Three lines leave bus j, one to each of three substation busbars whose voltage and angle are imposed. The bus is balanced in both active and reactive power, and each line carries a gradient closure $Q_i = b_i(V_j - V_{s_i})$ for its reactive flow and a bilinear closure $P_i = b_i V_j(\theta_j - \theta_{s_i})$ for its active flow. Eight unknowns, eight rows. Identify every cycle, and say which of them pass through both domains.

Exercise 2.2. Complete the argument of §2.2.2 for the DC triangle: enumerate every assignment of an output variable to each of the five rows such that no two rows share an output, and show that every such assignment contains a directed cycle. How many assignments are there?

Exercise 2.3. A radial distribution feeder is a tree of n buses fed from a substation reservoir, with one line into each bus and a known load at each bus. Show that its DC residual system admits an order of computation, write it down, and show that it is the order in which a distribution engineer computes the feeder by hand. Then add a single tie line closing one loop and show that the order is destroyed.

Exercise 2.4. For the export feeder, form the reduced two-row system in (V_1, V_2) by eliminating the flows, and confirm by direct substitution that its solutions coincide with those of the four-row system. Draw its factor graph before and after merging the two factors. Which of the three graphs, four-row, two-row, or merged, has the largest factor scope?

Exercise 2.5. Draw the Markov graph of the DC triangle in row-per-factor form, and verify that it is the sparsity pattern of $\mathbf{J}^\mathsf{T}\mathbf{J}$ for the Jacobian of Example 1.2. Then draw the Markov graph of the nodal form (Figure 2.5). Which pairs of variables are adjacent in the first and absent from the second, and why does that not change the solution?

Solutions to selected exercises

Exercise 2.2. The rows and their scopes are n_2 on $\{F_{12}, F_{23}\}$, n_3 on $\{F_{13}, F_{23}\}$, c_{12} on $\{\theta_2, F_{12}\}$, c_{13} on $\{\theta_3, F_{13}\}$ and c_{23} on $\{\theta_2, \theta_3, F_{23}\}$. An assignment gives each row one of its variables with no

repeats. Since θ_2 can be delivered only by c_{12} or c_{23} and θ_3 only by c_{13} or c_{23}, and c_{23} can deliver only one of them, at least one of c_{12}, c_{13} delivers its angle. Enumerating: (1) $c_{12} \to \theta_2$, $c_{13} \to \theta_3$, $c_{23} \to F_{23}$, $n_2 \to F_{12}$, $n_3 \to F_{13}$; (2) $c_{12} \to F_{12}$, $c_{23} \to \theta_2$, $c_{13} \to \theta_3$, $n_2 \to F_{23}$, $n_3 \to F_{13}$; (3) the mirror image of (2) with the roles of buses 2 and 3 exchanged. There are three, and no others. Each has a directed cycle. In (1), c_{12} needs F_{12}, which n_2 delivers from F_{23}, which c_{23} delivers from θ_2, which c_{12} was to deliver: $F_{12} \to \theta_2 \to F_{23} \to F_{12}$. In (2), n_2 delivers F_{23} from F_{12}, c_{12} delivers F_{12} from θ_2, and c_{23} delivers θ_2 from F_{23} and θ_3: again $F_{23} \to \theta_2 \to F_{12} \to F_{23}$. Case (3) is the mirror. The five rows are one irreducible block, as Proposition 2.1 says, and the script confirms the three assignments and the absence of an acyclic one.

Exercise 2.4. Substitute $Q_{12} = b_{12}(V_1 - V_2)$ from c_{12} into n_1 and $Q_{2s} = b_{2s}(V_2 - V_s)$ from c_{2s}, together with Q_{12}, into n_2:

$$k_1 = b_{12}\,(V_1 - V_2) - G, \qquad k_2 = b_{2s}\,(V_2 - V_s) - b_{12}\,(V_1 - V_2).$$

Setting $k_1 = 0$ gives $V_1 - V_2 = G/b_{12}$; substituting into $k_2 = 0$ gives $V_2 = V_s + G/b_{2s}$; and then $V_1 = V_s + G/b_{2s} + G/b_{12}$, which is the four-row solution, 1.025 and 1.035 per unit with the numbers of Example 1.1, and the flows follow from the closures. The two-row factor graph has two variables and two factors, each reading both: a four-cycle, cycle rank one. Merging the factors into one on $\{V_1, V_2\}$ leaves a tree of three nodes, cycle rank zero. The largest scope is three in the four-row form (c_{12} reads V_1, V_2, Q_{12}), two in the two-row form, and two in the merged form; the reduced forms have smaller scopes but a denser Markov graph on the variables that remain, which is the trade that Chapter 5 will price.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Counting the factor graph *variables and factors; a conservation factor per balanced node and a closure factor per edge; the square count is structural*
`foundation_electrical/dc_power_flow_audit/problems/rows_and_unknowns_dc`

Part II

Probability on a physical graph

Chapter 3

Where probability enters

The factor graph of Chapter 2 was drawn with indicator factors: each square is one where its row holds and zero where it does not. This chapter keeps every circle and every square and changes what the squares mean. The change is small to state and large in consequence. A square no longer says "this relation holds". It says "this relation holds to within so much", and the "so much" is a number the modeler supplies. Two new kinds of square appear, priors and sensors, both of degree one. The product of all the squares is then a function over the whole state of the system, large where every relation nearly holds and small elsewhere, and after normalization it is a probability distribution. Every later chapter is about that distribution.

The reader who has not worked with probability on continuous quantities will find the necessary definitions in §3.1. They are few. The reader who has may skip to §3.2.

3.1 A probability on the state

Chapter 1 gathered every unknown of the model into a vector $\mathbf{z} \in \mathbb{R}^n$. The deterministic model picked out one point of $\mathbb{R}^n$, the solution of $\mathbf{F}(\mathbf{z}) = \mathbf{0}$. The probabilistic model instead assigns to every point of $\mathbb{R}^n$ a non-negative number, its *density* $p(\mathbf{z})$, with the density integrating to one over the whole space. The density is large at states the model considers likely and small at states it considers unlikely. A region of state space has probability equal to the integral of p over it.

Three operations on a density are used throughout the book.

Marginalization. The density of some components of $\mathbf{z}$ alone, ignoring the others, is obtained by integrating the others out: $p(z_1) = \int p(z_1, z_2)\,\mathrm{d}z_2$. The marginal of the injection G in a model of the export feeder is what one would report as "the injection is G, give or take".

Conditioning. The density of some components given that others are known is the joint divided by the marginal of the known ones: $p(z_1 \mid z_2) = p(z_1, z_2)/p(z_2)$. Conditioning on a sensor reading is how a measurement changes what is believed about everything else.

Factorization. A density may be a product of local pieces, $p(\mathbf{z}) \propto \prod_f \varphi_f(\mathbf{z}_f)$, each reading a few components. This is the factor graph of Chapter 2 with the factors now non-negative functions instead of indicators. The proportionality sign hides a constant, the *normalizer* $Z = \int \prod_f \varphi_f \,\mathrm{d}\mathbf{z}$, that makes the integral one.

The Gaussian density is used so often that its form should be recognized on sight. In one dimension,

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right), \qquad \text{written } z \sim \mathcal{N}(\mu, \sigma^2), \tag{3.1}$$

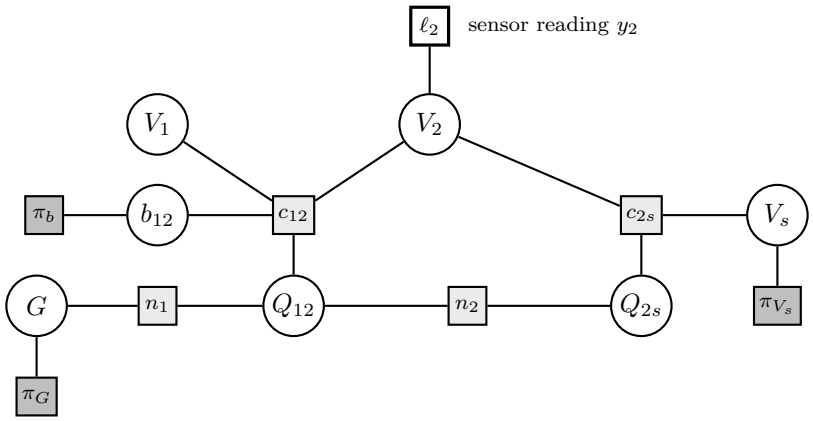


Figure 3.1: The export feeder with probability attached. Compared with Figure 2.1 three variables have been freed, the injection G, the substation voltage V_s and the susceptance b_{12}, and each has acquired a prior (dark squares π). A sensor on V_2 has been added (open square ℓ_2). The four physics factors are the same squares as before, now soft. No edge of the earlier graph has moved.

has mean μ and standard deviation σ. About two thirds of its probability lies within σ of the mean and about 95 percent within 2σ. Its negative logarithm is a quadratic in z, which is the property that makes everything in Chapter 6 work. Appendix B collects the identities the book needs.

One remark on what the numbers mean. A density on the state of a physical system can be read as a frequency, the fraction of days on which the load at bus 2 takes each value, or as a degree of belief, how strongly the modeler expects the susceptance to take each value given a line database. The book uses both readings without apology. The machinery is the same, and the practical question is always the same: what does the modeler know, and what does the modeler not know, about each quantity in the model? The answer is written as a factor.

3.2 Five kinds of factor

Every square in the graph is one of five kinds. Two of them, priors and sensors, are new. Three of them, conservation, closure and failure, are the rows of Chapter 1 read in a new way. Figure 3.1 draws all five on the export feeder and the triangle; the sections that follow define each.

3.2.1 Priors

A *prior* is a unary factor on a quantity the modeler is uncertain about before any measurement is taken: a parameter, a boundary condition, a source. It is what the data sheet, the forecast, or the history says.

The susceptance b_{12} of the line in Example 1.1 is computed from the conductor's geometry and the tower spacing, and the figure that reaches the model carries a spread the database does not state but that experience puts near twenty percent. Written as a factor,

$$\pi_b(b_{12}) = \exp\left(-\frac{(b_{12}-5)^2}{2\cdot 1^2}\right), \tag{3.2}$$

which is $b_{12} \sim \mathcal{N}(5, 1^2)$ up to the constant. The substation voltage is a set point with a deadband, $V_s \sim \mathcal{N}(1.000, 0.003^2)$. The injected reactive power is a schedule with a spread, $G \sim \mathcal{N}(0.05, 0.02^2)$. Each is a square of degree one attached to its variable (Figure 3.1).

Two things distinguish a prior from a boundary condition. A boundary condition is imposed exactly; a prior is imposed softly, and its width is a statement about how far the true value may be from the nominal one. And a boundary condition removes its variable from the unknowns; a prior leaves it in, so that the measurements can move it. The reservoir of Chapter 1, drawn as a unary factor in Table 2.2, is a prior with zero width. Chapter 2 promised that a reservoir becomes a prior by changing the meaning of one square, and this is the change.

Gaussian priors are the default, not the rule. A parameter that must be positive, a load that is bounded, a component whose availability is one or zero, all need other shapes, and Chapter 7 supplies them. For the present chapter everything is Gaussian.

3.2.2 Closure factors

A closure row $r_k(\mathbf{z}) = 0$ becomes the factor

$$\varphi_k(\mathbf{z}_k) = \exp\left(-\frac{r_k(\mathbf{z}_k)^2}{2\sigma_k^2}\right). \tag{3.3}$$

It is one where the relation holds exactly and falls off as the residual grows, on a scale σ_k in the units of the residual. The scope is unchanged: the factor reads exactly the variables the row read.

The scale σ_k is a statement about the closure, not about the system. A branch closure is a model of a physical path. The path has series resistance the lossless form ignores, shunt charging that grows with voltage, and a susceptance that shifts as the conductor heats and sags. The reactive power that actually flows differs from $b_{12}(V_1 - V_2)$ by an amount the modeler does not know but can bound, and σ_k is that bound. Setting σ_k small says the law is trusted; setting it large says the law is a rough guide. In the limit $\sigma_k \to 0$ the factor becomes the indicator of Chapter 2, and the deterministic model is recovered.

3.2.3 Conservation factors

A balance row becomes a factor of the same form,

$$\varphi_n(\mathbf{z}_n) = \exp\left(-\frac{r_n(\mathbf{z}_n)^2}{2\sigma_n^2}\right), \tag{3.4}$$

and here the reader who takes conservation seriously will object. Energy is conserved. There is no model error in a balance row. Why give it a width at all?

Two answers, one practical and one structural. The practical answer is that a balance row is exact for the system as modeled, and the system as modeled may be wrong: an unmodeled leak, a bypass, a load that is drawing from somewhere the graph does not show. A balance that cannot be satisfied by any state is the signature of exactly such a defect, and a model that forbids violation cannot report it. A balance with a small width can. The size of the violation the posterior settles on is the measurement of the defect, and Chapter 12 makes it a diagnostic tool. The structural answer is that a product of factors with zero width is not a density on $\mathbb{R}^n$; it is concentrated on a lower-dimensional surface, and integrating it, conditioning it, marginalizing

it all require more care than the general machinery provides. Chapter 4 is about that surface, and about what the width does to it. For now: every physics row, balance or closure, is given a width, the widths of balance rows are chosen small, and the deterministic solve is the limit in which all of them go to zero.

3.2.4 Measurement factors

A sensor reads a variable, or a function of several, with an error. The reading is y_i; the variable is z_{o_i}; the error is $\nu_i \sim \mathcal{N}(0, \sigma_i^2)$. Since $y_i = z_{o_i} + \nu_i$, the density of the reading given the state is a Gaussian centered on the variable, and read as a function of the state for the fixed reading it is the *likelihood* factor

$$\ell_i(z_{o_i}) = \exp\left(-\frac{(y_i - z_{o_i})^2}{2\sigma_i^2}\right). \tag{3.5}$$

It is a unary factor on the observed variable, drawn as the open square in Figure 3.1. The sensor scale σ_i is the sensor's accuracy, which its data sheet does state.

Three variations cover what real instrumentation does. A sensor that reads a combination, such as a meter on the total power leaving a substation, has a factor whose scope is every variable in the combination. A sensor with a bias has the bias as a variable of its own, with a prior, in the scope of the factor; the bias is then estimated along with everything else. A sensor that reads a quantity not in $\mathbf{z}$, such as a conductor temperature that the model relates to a line's rating by a further closure, needs that closure added as a factor and the conductor temperature added as a variable. In every case the change is local: a square, and perhaps a circle, added to the graph, nothing moved.

This is the promise of §2.7 kept. The deterministic model of Chapter 1 had no place for an observation. The probabilistic model has one, and it is a square of degree one.

3.2.5 Failure factors

Some uncertainty is not about how much but about whether. A line is in service or it is not. A valve is open or stuck. A sensor is working or has failed. Such a quantity is a variable with two values, $a \in \{0, 1\}$, its prior is a probability of each, $\mathbb{P}(a = 1) = 1 - q$ with q the failure rate, and the closure that depends on it reads it in its scope:

$$r_{23}(\theta_2, \theta_3, F_{23}, a_{23}) = F_{23} - a_{23}\, B\,(\theta_2 - \theta_3). \tag{3.6}$$

With $a_{23} = 1$ this is the line closure of Example 1.2. With $a_{23} = 0$ it says the flow is zero whatever the angles: the line is gone. Figure 3.2 draws it. The variable is discrete, the closure's scope has grown by one, and the product of factors is now a density in the continuous variables times a probability mass in the discrete one. Everything in this chapter still applies; what changes in the computation is the subject of Chapter 7.

3.2.6 The complete list

Table 3.1 summarizes. The joint distribution of the whole state is their product,

$$\boxed{p(\mathbf{z} \mid \mathbf{y}) \;\propto\; \prod_j \pi_j(z_j) \prod_k \varphi_k(\mathbf{z}_k) \prod_i \ell_i(z_{o_i})} \tag{3.7}$$

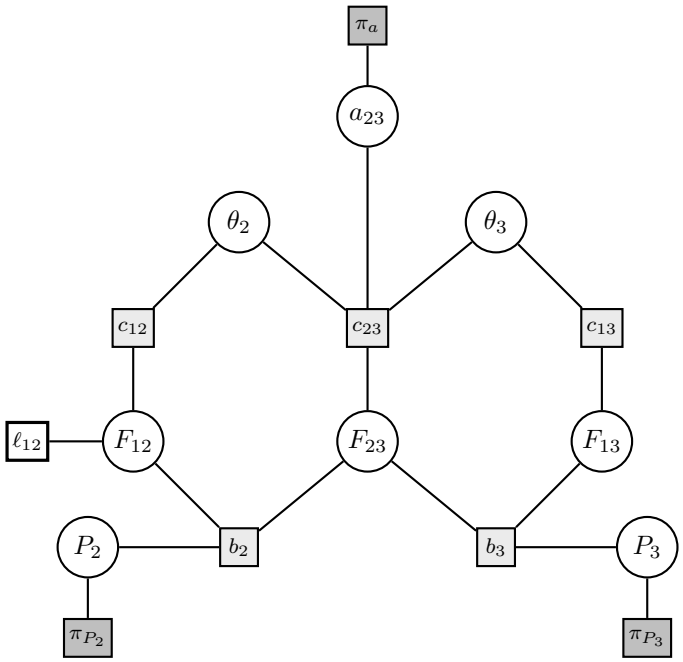


Figure 3.2: The DC triangle with probability attached. The loads P_2, P_3 are freed and given priors; the line from bus 2 to bus 3 has an availability variable a_{23} with a failure prior, in the scope of its closure; a meter ℓ_{12} reads the flow on line 1–2. The structure of Figure 2.3 is intact inside it.

with the failure priors folded into the first product. It is written conditional on the readings $\mathbf{y}$ because the likelihoods depend on them; for fixed readings it is a function of $\mathbf{z}$ alone, and it is called the *posterior*. The word means "after the measurements". The same product with the likelihoods removed is the *prior predictive* distribution, what the model believes about the state before any sensor is read.

Principle 3.1. *Probability describes uncertainty around quantities. Physics factors enforce relations between them. The joint distribution is the product of both, and its factor graph is the factor graph of the physics with unary squares added.*

3.3 The joint as an objective

Take the negative logarithm of equation (3.7). Every factor is an exponential of a negative quadratic, so the logarithm is a sum of quadratics:

$$-\log p(\mathbf{z} \mid \mathbf{y}) \;=\; \sum_j \frac{(z_j - \mu_j)^2}{2\sigma_j^2} + \sum_k \frac{r_k(\mathbf{z}_k)^2}{2\sigma_k^2} + \sum_i \frac{(y_i - z_{o_i})^2}{2\sigma_i^2} + \text{const.} \tag{3.8}$$

The most probable state is the one that minimizes this sum. Read the three terms. The first penalizes departure of each uncertain input from its nominal value, in units of that input's spread. The second penalizes violation of each physics row, in units of that row's width. The third penalizes misfit between each sensor and the variable it reads, in units of the sensor's accuracy.

Table 3.1: The five kinds of factor. The three physics kinds are the rows of Chapter 1; the other two are new.

Kind	Form	Scope	Where it comes from
Prior π	$\exp(-(z-\mu)^2/2\sigma^2)$	one variable	data sheet, forecast, history
Closure φ	$\exp(-r_k(\mathbf{z}_k)^2/2\sigma_k^2)$	the row's variables	a constitutive law and its error
Conservation φ	$\exp(-r_n(\mathbf{z}_n)^2/2\sigma_n^2)$	the row's variables	a balance law; width small
Likelihood ℓ	$\exp(-(y-z)^2/2\sigma_y^2)$	the observed variable	a sensor and its accuracy
Failure π_a	$\mathbb{P}(a)$ on $\{0,1\}$	one discrete variable	a failure rate

The most probable state is the compromise: it moves the inputs away from nominal as little as it can, violates the physics as little as it can, and misfits the sensors as little as it can, with each "as little" weighted by how much the modeler trusts that term.

Equation (3.8) is a weighted least-squares problem. When every row is linear in $\mathbf{z}$ it is a quadratic and its minimizer solves a linear system; when rows are nonlinear it is solved by the same Newton iteration that solved $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, with the extra terms included. Chapter 6 develops this fully. The point to take now is that the deterministic solve of Chapter 1 is the special case of equation (3.8) in which there are no priors, no sensors, and every $\sigma_k \to 0$: the minimum is then zero, reached only where every row holds, which is the solution.

3.3.1 Residuals in units of their widths

Each term of equation (3.8) is half the square of a *standardized residual*: the departure of an input from its nominal value, the violation of a physics row, or the misfit of a sensor, each divided by its own width. Standardizing puts per-unit power, per-unit voltage and radians on one scale, the scale of "how many widths", and it is in that scale that every diagnostic of the book is read. At the most probable state the standardized residuals say where the compromise was struck. For the feeder of §3.4 after its first reading, the injection sits 0.18 prior widths above nominal, the substation voltage 0.06 above, and the sensor is fitted to a hundredth of its accuracy; the sum of the squares, 0.037, is twice the objective at its minimum. After the second reading the four residuals are 0.23, −0.09, 0.08 and −0.07, and the sum is 0.073: two readings, both fitted to a tenth of an accuracy, at the cost of moving the source a quarter of a width. Chapter 4 names this sum the *misfit* and says what value an adequate model should give; Chapter 12 makes it a test. The point to take now is that a posterior is not only a set of estimates but a set of standardized residuals, and the residuals are where the model says whether it was comfortable.

3.3.2 Conditioning a Gaussian on one reading

The worked examples that follow use one operation over and over: a Gaussian belief about a few inputs, and a sensor that reads a linear combination of them. The operation has a closed form, and it is worth deriving once, because every later chapter uses it.

Proposition 3.1 (One linear reading)**.** *Let* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *with* $\boldsymbol{\Sigma}$ *positive definite, and let a sensor read* $y = \mathbf{h}^\mathsf{T}\mathbf{z} + \nu$ *with* $\nu \sim \mathcal{N}(0, \sigma^2)$ *independent of* $\mathbf{z}$*. Then the posterior of* $\mathbf{z}$ *given* y *is Gaussian with*

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{k}\,(y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}), \qquad \boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \mathbf{k}\,\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}, \qquad \mathbf{k} = \frac{\boldsymbol{\Sigma}\mathbf{h}}{\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h} + \sigma^2}. \tag{3.9}$$

Proof. Write $e = y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}$ for the *innovation*, what was read less what the prior predicts, and $s = \mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h} + \sigma^2$ for its variance, so that $\mathbf{k} = \boldsymbol{\Sigma}\mathbf{h}/s$. The proof has four steps.

The negative log posterior. By Bayes' rule $p(\mathbf{z} \mid y) = p(\mathbf{z})\, p(y \mid \mathbf{z})/p(y)$, and $p(y)$ does not depend on $\mathbf{z}$. The prior density is $(2\pi)^{-n/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp[-\tfrac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})]$. Given $\mathbf{z}$, the number $\mathbf{h}^\mathsf{T}\mathbf{z}$ is fixed and ν is still $\mathcal{N}(0,\sigma^2)$, because it is independent of $\mathbf{z}$; so $y \mid \mathbf{z} \sim \mathcal{N}(\mathbf{h}^\mathsf{T}\mathbf{z}, \sigma^2)$, with density $(2\pi\sigma^2)^{-1/2}\exp[-(y-\mathbf{h}^\mathsf{T}\mathbf{z})^2/2\sigma^2]$. Multiplying the two densities and taking the negative logarithm,

$$-\log p(\mathbf{z} \mid y) = J(\mathbf{z}) + K, \qquad J(\mathbf{z}) = \tfrac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}) + \frac{(y-\mathbf{h}^\mathsf{T}\mathbf{z})^2}{2\sigma^2},$$

where K collects $\frac{n}{2}\log 2\pi$, $\frac{1}{2}\log|\boldsymbol{\Sigma}|$, $\frac{1}{2}\log 2\pi\sigma^2$ and $\log p(y)$, none of which depends on $\mathbf{z}$. The posterior is proportional to $\exp[-J(\mathbf{z})]$, so its shape in $\mathbf{z}$ is fixed by J; K only makes it integrate to one.

Completing the square. Measure $\mathbf{z}$ from the prior mean, $\mathbf{x} = \mathbf{z} - \boldsymbol{\mu}$. Then $y - \mathbf{h}^\mathsf{T}\mathbf{z} = e - \mathbf{h}^\mathsf{T}\mathbf{x}$, and expanding the square with $(\mathbf{h}^\mathsf{T}\mathbf{x})^2 = \mathbf{x}^\mathsf{T}\mathbf{h}\mathbf{h}^\mathsf{T}\mathbf{x}$,

$$J = \tfrac{1}{2}\,\mathbf{x}^\mathsf{T}\mathbf{P}\,\mathbf{x} - \frac{e}{\sigma^2}\,\mathbf{h}^\mathsf{T}\mathbf{x} + \frac{e^2}{2\sigma^2}, \qquad \mathbf{P} = \boldsymbol{\Sigma}^{-1} + \frac{\mathbf{h}\mathbf{h}^\mathsf{T}}{\sigma^2}.$$

The matrix $\mathbf{P}$ is positive definite, because $\boldsymbol{\Sigma}^{-1}$ is and $\mathbf{h}\mathbf{h}^\mathsf{T}$ is positive semidefinite. Let $\mathbf{m}$ solve $\mathbf{P}\mathbf{m} = \mathbf{h}\,e/\sigma^2$. Expanding the right-hand side of

$$J = \tfrac{1}{2}(\mathbf{x}-\mathbf{m})^\mathsf{T}\mathbf{P}\,(\mathbf{x}-\mathbf{m}) + \text{const}$$

recovers the line above. A density proportional to $\exp[-\tfrac{1}{2}(\mathbf{x}-\mathbf{m})^\mathsf{T}\mathbf{P}(\mathbf{x}-\mathbf{m})]$ is the Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{P}^{-1}$. So the posterior of $\mathbf{z} = \boldsymbol{\mu} + \mathbf{x}$ is Gaussian with covariance $\boldsymbol{\Sigma}' = \mathbf{P}^{-1}$ and mean $\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{P}^{-1}\mathbf{h}\,e/\sigma^2$.

The covariance. The claim is $\mathbf{P}^{-1} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}/s$. Multiplying, and using that $\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h}$ is a number,

$$\mathbf{P}\Big(\boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}}{s}\Big) = \mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\,\Big(\frac{1}{\sigma^2} - \frac{1}{s} - \frac{\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h}}{\sigma^2 s}\Big) = \mathbf{I},$$

because the bracket is $(s - \sigma^2 - \mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h})/(\sigma^2 s) = 0$. Since $\boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}/s = \mathbf{k}\,\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}$, this is $\boldsymbol{\Sigma}'$ as stated. The identity is the Sherman–Morrison formula for a rank-one change of $\boldsymbol{\Sigma}^{-1}$.

The mean. From the covariance, $\boldsymbol{\Sigma}'\mathbf{h} = \boldsymbol{\Sigma}\mathbf{h} - \boldsymbol{\Sigma}\mathbf{h}\,(\mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h})/s = \boldsymbol{\Sigma}\mathbf{h}\,\sigma^2/s$. Hence $\boldsymbol{\Sigma}'\mathbf{h}/\sigma^2 = \boldsymbol{\Sigma}\mathbf{h}/s = \mathbf{k}$, and $\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{k}\,e$. □

In one variable, with $\mathbf{h} = 1$, the gain is $\boldsymbol{\Sigma}/(\boldsymbol{\Sigma} + \sigma^2)$ and the covariance formula reads $1/\boldsymbol{\Sigma}' = 1/\boldsymbol{\Sigma} + 1/\sigma^2$: the precisions add. The formulas (3.9) never invert $\boldsymbol{\Sigma}$, and they hold for a singular $\boldsymbol{\Sigma}$ as well, for instance when one input is known exactly; the proof above needs the inverse only to write the prior's density, and Proposition B.3 in Appendix B proves the formulas without it.

What h is, and where the physics is. The vector $\mathbf{h}$ has one entry for each unknown in $\mathbf{z}$, and the sensor reads the weighted sum $\mathbf{h}^\mathsf{T}\mathbf{z} = h_1 z_1 + \cdots + h_n z_n$. A sensor attached directly to the jth unknown has $\mathbf{h}$ equal to the jth unit vector, a one in position j and zeros elsewhere. A sensor attached to a quantity that is not itself in $\mathbf{z}$ reads the combination of the unknowns that determines that quantity, and the physics is what says which combination. The proposition

contains no physics. It is the conditioning of any Gaussian vector on any linear reading, and it would hold for prices or pixels. The physics enters through what the proposition is given: which quantities are the unknowns, what the prior says about them, and above all $\mathbf{h}$, the route by which each unknown reaches what the sensor sees. In the example of §3.4 the sensor sits on a voltage, an entry of $\mathbf{z}$, so its $\mathbf{h}$ is a unit vector; the physics is exact and linear, so every entry of $\mathbf{z}$ is a combination of the injection and the substation voltage, and the same sensor becomes a vector over those two inputs. Soft physics fits the same mould. A linear row $r(\mathbf{z}) = \mathbf{a}^\mathsf{T}\mathbf{z} - b$ of width σ_k contributes the factor $\exp[-(b - \mathbf{a}^\mathsf{T}\mathbf{z})^2/2\sigma_k^2]$, which is exactly the likelihood of a sensor with $\mathbf{h} = \mathbf{a}$ that reads the value b with accuracy σ_k. A soft physics row is a reading that always reports that the balance holds, and the proposition folds it in like any other. A nonlinear row, or a sensor on a nonlinear function of $\mathbf{z}$, is linearized at the current state, and $\mathbf{h}$ becomes a row of the Jacobian; Chapter 6 takes that step.

Three things are visible in the formula. The mean moves by the *innovation* $y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}$, the difference between what was read and what was predicted, times a gain $\mathbf{k}$ that is large in the directions the prior is wide and the sensor is accurate. The covariance shrinks by a rank-one amount, in the direction $\boldsymbol{\Sigma}\mathbf{h}$ and in no other: one reading narrows one combination of the inputs and leaves the orthogonal combinations exactly as they were. And the shrinkage does not depend on the reading's value, only on where the sensor sits and how accurate it is, so the posterior spread can be computed before the sensor is read, which is how a sensor's worth is judged before it is bought. Chapter 11 makes this operation the unit of sequential inference.

Several readings at once are handled by the same algebra in a form that is more convenient when there are many.

Proposition 3.2 (Information form)**.** *Write the Gaussian prior by its* precision $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ *and* information vector $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$*. Readings* $\mathbf{y} = \mathbf{H}\mathbf{z} + \boldsymbol{\nu}$ *with* $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ *give a posterior with precision and information*

$$\boldsymbol{\Lambda}' = \boldsymbol{\Lambda} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H}, \qquad \boldsymbol{\eta}' = \boldsymbol{\eta} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{y}, \tag{3.10}$$

and mean $\boldsymbol{\mu}' = \boldsymbol{\Lambda}'^{-1}\boldsymbol{\eta}'$*, covariance* $\boldsymbol{\Sigma}' = \boldsymbol{\Lambda}'^{-1}$*.*

Proof. The negative log posterior is $\frac{1}{2}\mathbf{z}^\mathsf{T}\boldsymbol{\Lambda}\mathbf{z} - \boldsymbol{\eta}^\mathsf{T}\mathbf{z} + \frac{1}{2}(\mathbf{y} - \mathbf{H}\mathbf{z})^\mathsf{T}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{z})$ plus constants; collecting the quadratic and linear terms in $\mathbf{z}$ gives $\frac{1}{2}\mathbf{z}^\mathsf{T}\boldsymbol{\Lambda}'\mathbf{z} - \boldsymbol{\eta}'^\mathsf{T}\mathbf{z}$ with $\boldsymbol{\Lambda}'$ and $\boldsymbol{\eta}'$ as stated, and a Gaussian with that exponent has the mean and covariance claimed. □

In this form a reading *adds* to the precision and the information, and the addition is a sum over readings, one rank-one term each, so readings can be folded in one at a time or all at once with the same result. Proposition 3.1 is the same update expressed in the covariance, the two being related by the matrix identity used in its proof. The information form is the one the rest of the book computes in, because the physics factors of equation (3.7) add to the precision in exactly the same way and keep it sparse; Chapter 6 builds the whole posterior this way.

3.4 A worked example: one sensor, then two

The export feeder, with numbers, shows what the joint distribution does when a sensor is read. The model is the one of Example 1.1, with the same susceptances, $b_{12} = 5$ and $b_{2s} = 2$ per unit, and two changes. The injection and the substation voltage are no longer known: they carry the priors $G \sim \mathcal{N}(0.05, 0.02^2)$ and $V_s \sim \mathcal{N}(1.000, 0.003^2)$ per unit, centred on the values Example 1.1

used. The first says the generator's reactive output is scheduled but drifts; the second says the tap changer holds the busbar to about three parts in a thousand. And a sensor on V_2 with accuracy $\sigma_y = 0.0005$ per unit reads $y_2 = 1.027$. To keep the calculation by hand, the closures and balances are taken as exact ($\sigma_k \to 0$). Every number below is produced by a short script, `ch03_chain_posterior.py` in the book's `scripts` directory, which the reader can run.

The example in vectors. The unknown vector is the one of Example 1.1, flows first, with the two inputs appended because they are now unknown too:

$$\mathbf{z} = \begin{pmatrix} Q_{12} \\ Q_{2s} \\ V_1 \\ V_2 \\ G \\ V_s \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{v} \\ \mathbf{u} \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} G \\ V_s \end{pmatrix}.$$

Exact physics leaves no freedom beyond the inputs. Example 1.1 solved $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ for the flows and voltages in terms of G and V_s: both flows equal G, and $V_1 = \mathbf{h}_1^{\mathsf{T}}\mathbf{u}$, $V_2 = \mathbf{h}_2^{\mathsf{T}}\mathbf{u}$. Every entry of $\mathbf{z}$ is therefore a fixed linear combination of the two inputs,

$$\mathbf{z} = \mathbf{S}\,\mathbf{u}, \qquad \mathbf{S} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0.7 & 1 \\ 0.5 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

whose third and fourth rows are $\mathbf{h}_1^{\mathsf{T}}$ and $\mathbf{h}_2^{\mathsf{T}}$. The prior sits on the inputs,

$$\boldsymbol{\mu}_u = \begin{pmatrix} 0.05 \\ 1.000 \end{pmatrix}, \qquad \boldsymbol{\Sigma}_u = \begin{pmatrix} 0.02^2 & 0 \\ 0 & 0.003^2 \end{pmatrix} = \begin{pmatrix} 4.0 \times 10^{-4} & 0 \\ 0 & 9.0 \times 10^{-6} \end{pmatrix},$$

where the zeros off the diagonal say that the injection and the substation voltage are independent a priori. It passes to the whole vector as $\boldsymbol{\mu}_z = \mathbf{S}\boldsymbol{\mu}_u = (0.05, 0.05, 1.035, 1.025, 0.05, 1.000)^{\mathsf{T}}$ and $\boldsymbol{\Sigma}_z = \mathbf{S}\boldsymbol{\Sigma}_u\mathbf{S}^{\mathsf{T}}$. The sensor on V_2 reads the fourth entry of $\mathbf{z}$, so in Proposition 3.1 it is the simplest kind: $\mathbf{h} = \mathbf{e}_4$, a one in the position of V_2 and zeros elsewhere. Because $\mathbf{z} = \mathbf{S}\mathbf{u}$, the same reading is $\mathbf{e}_4^{\mathsf{T}}\mathbf{S}\mathbf{u} = \mathbf{h}_2^{\mathsf{T}}\mathbf{u}$, a reading of the inputs through $\mathbf{h}_2$. The entries of $\mathbf{h}_2$ are the physics. The first, $1/b_{2s} = 0.5$, is how far bus 2 rises for each per unit of reactive power the generator injects; the second, 1, is how far it rises for each per unit of substation voltage.

The covariance $\boldsymbol{\Sigma}_z$ has rank two, because six unknowns are driven by two inputs, so it is not positive definite. The calculation is therefore done where the freedom is. Proposition 3.1 is applied to $\mathbf{u}$, with $\mathbf{h}_2$, and $\mathbf{S}$ carries the result to all of $\mathbf{z}$: $\boldsymbol{\mu}_z' = \mathbf{S}\boldsymbol{\mu}_u'$ and $\boldsymbol{\Sigma}_z' = \mathbf{S}\boldsymbol{\Sigma}_u'\mathbf{S}^{\mathsf{T}}$. The proposition's formulas applied to $\mathbf{z}$ directly, with $\mathbf{h} = \mathbf{e}_4$ and the rank-two $\boldsymbol{\Sigma}_z$, give the same posterior to twelve digits.

Before the reading. The prior predicts V_2 at $\mathbf{e}_4^{\mathsf{T}}\boldsymbol{\mu}_z = \mathbf{h}_2^{\mathsf{T}}\boldsymbol{\mu}_u = 0.5 \cdot 0.05 + 1 \cdot 1.000 = 1.0250$, with variance $\mathbf{h}_2^{\mathsf{T}}\boldsymbol{\Sigma}_u\mathbf{h}_2 = 0.5^2 \cdot 4.0 \times 10^{-4} + 1^2 \cdot 9.0 \times 10^{-6} = 1.09 \times 10^{-4}$, a spread of 0.01044 per unit. Most of that variance comes from the injection, 1.00×10^{-4} of the 1.09×10^{-4}, against the substation's 9.0×10^{-6}. The same two products with $\mathbf{h}_1$, the third entries of $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$, give the prior for V_1, 1.0350 ± 0.0143.

Table 3.2: The export feeder: prior, posterior after the V_2 sensor, posterior after both sensors. Means and standard deviations, all in per unit; the last two rows are the voltages predicted from each distribution.

	Prior	After $y_2 = 1.027$	After y_2 and $y_1 = 1.038$
G	0.0500 ± 0.0200	0.0537 ± 0.0058	0.0546 ± 0.0029
V_s	1.0000 ± 0.0030	1.0002 ± 0.0029	0.9997 ± 0.0017
$\mathrm{corr}(G, V_s)$	0	-0.985	-0.979
V_2 predicted	1.0250 ± 0.0104	1.0270 ± 0.0005	1.0270 ± 0.0004
V_1 predicted	1.0350 ± 0.0143	1.0377 ± 0.0013	1.0380 ± 0.0005

After one reading. The reading $y_2 = 1.027$ gives the innovation $e = y_2 - \mathbf{h}_2^{\mathsf{T}}\boldsymbol{\mu}_u = 0.0020$. Proposition 3.1 then takes a handful of products:

$$\boldsymbol{\Sigma}_u\mathbf{h}_2 = \begin{pmatrix} 4.0\times10^{-4}\cdot 0.5 \\ 9.0\times10^{-6}\cdot 1 \end{pmatrix} = \begin{pmatrix} 2.00\times10^{-4} \\ 9.00\times10^{-6} \end{pmatrix}, \qquad s = \mathbf{h}_2^{\mathsf{T}}\boldsymbol{\Sigma}_u\mathbf{h}_2 + \sigma^2 = 1.0900\times10^{-4} + 2.5\times10^{-7},$$

$$\mathbf{k} = \frac{\boldsymbol{\Sigma}_u\mathbf{h}_2}{s} = \begin{pmatrix} 1.8307 \\ 0.0824 \end{pmatrix}, \qquad \boldsymbol{\mu}_u' = \boldsymbol{\mu}_u + \mathbf{k}\,e = \begin{pmatrix} 0.05 + 0.003661 \\ 1.000 + 0.000165 \end{pmatrix} = \begin{pmatrix} 0.053661 \\ 1.000165 \end{pmatrix},$$

$$\boldsymbol{\Sigma}_u' = \boldsymbol{\Sigma}_u - \mathbf{k}\,(\boldsymbol{\Sigma}_u\mathbf{h}_2)^{\mathsf{T}} = \begin{pmatrix} 3.3867 & -1.6476 \\ -1.6476 & 0.8259 \end{pmatrix}\times10^{-5},$$

where $\mathbf{k}\,\mathbf{h}_2^{\mathsf{T}}\boldsymbol{\Sigma}_u = \mathbf{k}\,(\boldsymbol{\Sigma}_u\mathbf{h}_2)^{\mathsf{T}}$ because $\boldsymbol{\Sigma}_u$ is symmetric. Read the result off the matrices. The diagonal of $\boldsymbol{\Sigma}_u'$ holds the variances: the injection's falls from 4.0×10^{-4} to 3.39×10^{-5}, a spread of 0.0058 per unit, and the substation voltage's from 9.0×10^{-6} to 8.26×10^{-6}, a spread of 0.0029. The off-diagonal entry, zero before, is now -1.65×10^{-5}, a correlation of -0.985. The posterior of the whole vector follows through $\mathbf{S}$:

$$\boldsymbol{\mu}_z' = \mathbf{S}\boldsymbol{\mu}_u' = (0.0537,\ 0.0537,\ 1.0377,\ 1.0270,\ 0.0537,\ 1.0002)^{\mathsf{T}},$$

and the diagonal of $\mathbf{S}\boldsymbol{\Sigma}_u'\mathbf{S}^{\mathsf{T}}$ gives the spreads $(0.0058, 0.0058, 0.0013, 0.0005, 0.0058, 0.0029)$. Both reactive flows sit at 0.0537 ± 0.0058 per unit, since each equals the injection, and bus 2 at 1.0270 ± 0.0005, the sensor's own accuracy. The posterior, in Table 3.2, is exactly this: the injection up by 0.0037 per unit and the substation voltage by 0.00016; the injection moves more because it had more room to move. The injection's spread falls from 0.0200 to 0.0058 per unit, a factor of three and a half, from a single voltage reading taken at the next bus down. The substation voltage's spread barely moves, from 0.0030 to 0.0029. And the two, independent under the prior, are now correlated at -0.985: the posterior knows that $V_s + 0.5G$ is close to 1.027, so it knows that if the injection is higher the substation voltage is lower, without knowing which. Figure 3.3 draws the prior and the two posteriors as one-standard-deviation ellipses in the (G, V_s) plane; the first posterior is the thin diagonal ridge.

A virtual sensor. V_1 has not been measured, but it is the third entry of $\mathbf{z}$, and the posterior predicts it: $\mathbf{h}_1^{\mathsf{T}}\boldsymbol{\mu}_u' = 0.7\cdot 0.053661 + 1.000165 = 1.0377$, with variance $\mathbf{h}_1^{\mathsf{T}}\boldsymbol{\Sigma}_u'\mathbf{h}_1 = 1.79\times10^{-6}$, a spread of 0.0013. Its prior spread was 0.0143. The reading at bus 2 has tightened the prediction at bus 1 by a factor of ten, through the physics: V_1 and V_2 differ by G/b_{12}, and G is now known

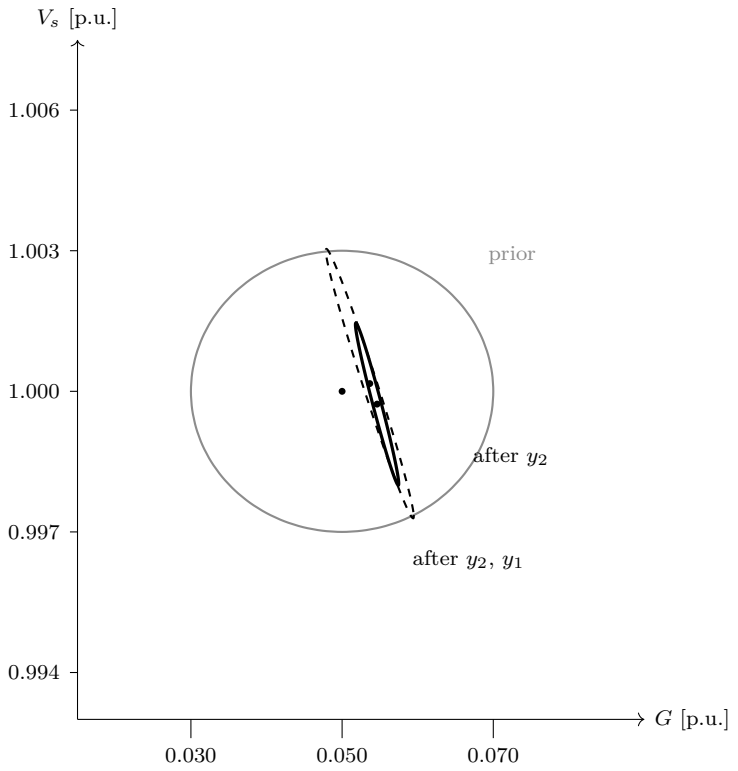


Figure 3.3: One-standard-deviation ellipses of the joint distribution of injection and substation voltage: the prior (grey, axis-aligned, because the two are independent), the posterior after the V_2 reading (dashed, a ridge along $V_s + 0.5G = 1.027$), and the posterior after both readings (solid, shorter along the ridge). Dots mark the means. Generated by the same script as Table 3.2.

to ± 0.0058. This is the first virtual sensor in the book. Chapter 12 is about them. Figure 3.4 draws the three predictive densities of V_1, before any reading, after the first, and after the second, with the second reading marked; the first reading turned a broad prior into a prediction sharp enough to be tested, and the second passed the test.

A second reading. Now a sensor on V_1 reads $y_1 = 1.038$. The reading falls within the virtual sensor's range, which is the first thing to check: had it read 1.030, six standard deviations from the prediction, the model rather than the injection would be the suspect. Proposition 3.1 applies again, with the first posterior in the place of the prior. The sensor reads the third entry of $\mathbf{z}$, so on $\mathbf{z}$ it is $\mathbf{h} = \mathbf{e}_3$, and on the inputs it is $\mathbf{h}_1$. The innovation is $1.038 - \mathbf{h}_1^\mathsf{T}\boldsymbol{\mu}'_u = 0.00027$, and

$$\boldsymbol{\Sigma}'_u\mathbf{h}_1 = \begin{pmatrix} 7.23 \\ -3.28 \end{pmatrix} \times 10^{-6}, \quad s = \mathbf{h}_1^\mathsf{T}\boldsymbol{\Sigma}'_u\mathbf{h}_1 + \sigma^2 = (1.79 + 0.25) \times 10^{-6}, \quad \mathbf{k} = \frac{\boldsymbol{\Sigma}'_u\mathbf{h}_1}{s} = \begin{pmatrix} 3.55 \\ -1.61 \end{pmatrix},$$

$$\boldsymbol{\Sigma}''_u = \boldsymbol{\Sigma}'_u - \mathbf{k}\,(\boldsymbol{\Sigma}'_u\mathbf{h}_1)^\mathsf{T} = \begin{pmatrix} 8.20 & -4.85 \\ -4.85 & 2.99 \end{pmatrix} \times 10^{-6}.$$

The gain on the substation voltage is negative although $\mathbf{h}_1$ has a positive entry for it. The first reading left the injection and the substation voltage correlated, so a generator-bus voltage above

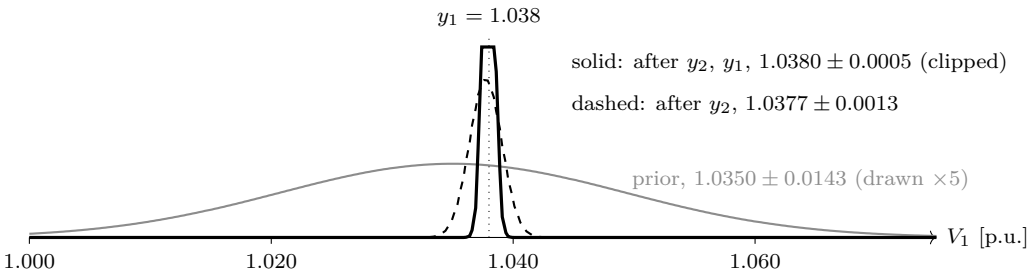


Figure 3.4: The predictive density of the unmeasured generator-bus voltage V_1 under the prior (grey), after the bus-2 reading y_2 (dashed), and after both readings (solid, clipped in height). The dotted line is the reading $y_1 = 1.038$ when it arrives: inside the dashed prediction's range, so the model is not surprised.

its prediction is blamed on the injection, and the correlation pulls the substation voltage down. The mean moves by $\mathbf{k}\, e = (0.00097, -0.00044)$, and the result is the third column of Table 3.2. Proposition 3.2, which folds both readings in at once, gives the same posterior to twelve digits. The injection is now 0.0546 ± 0.0029, the substation voltage 0.9997 ± 0.0017. The second sensor's lever arm on the injection is only the difference $V_1 - V_2 = G/b_{12}$, a slope of $1/b_{12} = 0.2$ against two readings each accurate to 0.0005, which is why the injection improves by a factor of two and not by more. The correlation is still -0.979: two readings of voltage, both of which are "substation voltage plus something times injection", cannot fully separate the two. A sensor on the reactive flow, or on the substation busbar itself, would. Which sensor to add next is a question the posterior can answer before the sensor is bought, and Chapter 12 returns to it.

3.5 A second example: the triangle with a flow meter

The same operation on the electrical example shows what a loop does to it. The model is the one of Example 1.2, with $B = 10$ and the physics exact, and one change: the two loads are no longer known. They carry the priors $P_2 \sim \mathcal{N}(-1.5, 0.3^2)$ and $P_3 \sim \mathcal{N}(-0.5, 0.3^2)$ per unit, independent and centred on the second set of loads Example 1.2 used. Every number is from `ch03_triangle_posterior.py`.

The example in vectors. As for the feeder, the unknown vector is the one of Example 1.2, flows first, with the loads appended:

$$\mathbf{z} = \begin{pmatrix} F_{12} \\ F_{13} \\ F_{23} \\ \theta_2 \\ \theta_3 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \boldsymbol{\theta} \\ \mathbf{u} \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} P_2 \\ P_3 \end{pmatrix}.$$

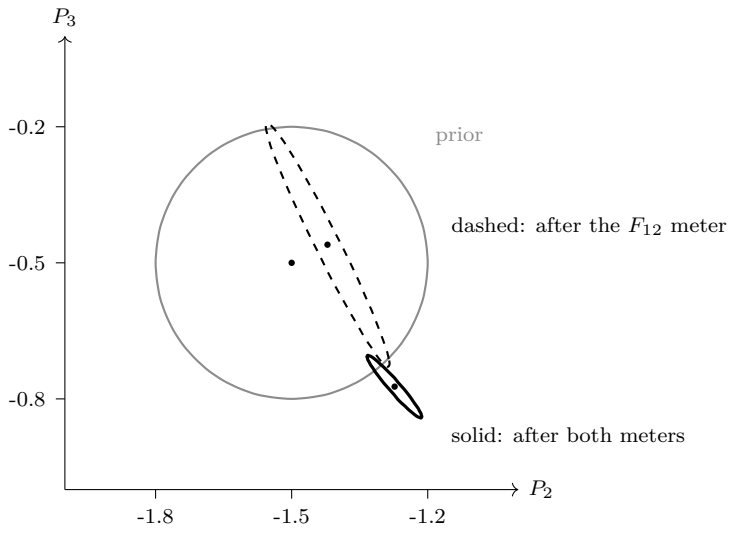


Figure 3.5: One-standard-deviation ellipses of the two loads of the DC triangle: the prior (grey), the posterior after the meter on line 1–2 (dashed), and the posterior after a second meter on the boundary flow (solid). The first meter reads one direction of the load plane and flattens the ellipse along it only; the second reads a different direction and closes it.

The exact physics, equation (1.21) with the flows from the closures, makes every entry a linear combination of the loads,

$$\mathbf{z} = \mathbf{S}\,\mathbf{u}, \qquad \mathbf{S} = \frac{1}{3}\begin{pmatrix} -2 & -1 \\ -1 & -2 \\ 1 & -1 \\ 2/B & 1/B \\ 1/B & 2/B \\ 3 & 0 \\ 0 & 3 \end{pmatrix},$$

and the boundary flow, $F_{12} + F_{13}$, is $-(P_2 + P_3)$. Under the prior the flows are 1.167 ± 0.224, 0.833 ± 0.224 and -0.333 ± 0.141, and the boundary flow 2.000 ± 0.424. A meter on line 1–2 reads the first entry of $\mathbf{z}$: on $\mathbf{z}$ it is $\mathbf{h} = \mathbf{e}_1$, and on the loads it is the first row of $\mathbf{S}$, $(-\frac{2}{3}, -\frac{1}{3})^{\mathsf{T}}$. It reads $F_{12} = 1.10$ with accuracy 0.02, a third of a spread below its prediction. As for the chain, Proposition 3.1 is applied to $\mathbf{u}$ and $\mathbf{S}$ carries the result to $\mathbf{z}$; applied to $\mathbf{z}$ directly, with $\mathbf{h} = \mathbf{e}_1$, it gives the same posterior.

Proposition 3.1 gives the posterior of the loads, $\mathbf{S}$ carries it to the flows, Table 3.3 lists both, and Figure 3.5 draws the loads. The meter reads the direction $(-0.894, -0.447)$ of the load plane, mostly P_2; along it the spread falls from 0.300 to 0.027, and across it, in the direction $(0.447, -0.894)$, it stays at 0.300 exactly. The loads become $P_2 = -1.421 \pm 0.136$ and $P_3 = -0.460 \pm 0.269$, correlated at -0.976: the meter has fixed two thirds of one plus one third of the other, so if one is lower the other must be higher. The unmetered flows move too, through the loop: F_{13} from 0.833 ± 0.224 to 0.780 ± 0.135, F_{23} from -0.333 ± 0.141 to -0.320 ± 0.134, and the boundary flow from 2.000 ± 0.424 to 1.881 ± 0.139. One meter on one line has narrowed the total load by a factor of three, because the total is mostly what the meter reads.

A second meter, on the boundary flow at bus 1, reads 2.05 ± 0.02. The boundary flow is not

Table 3.3: The DC triangle: prior, posterior after the meter on line 1–2, posterior after a second meter on the boundary flow. Loads and flows in per unit, means and standard deviations.

	prior	after $F_{12} = 1.10$	after F_{12} and boundary 2.05
P_2	-1.500 ± 0.300	-1.421 ± 0.136	-1.273 ± 0.060
P_3	-0.500 ± 0.300	-0.460 ± 0.269	-0.773 ± 0.069
$\mathrm{corr}(P_2, P_3)$	0	-0.976	-0.961
F_{12}	1.167 ± 0.224	1.101 ± 0.020	1.107 ± 0.019
F_{13}	0.833 ± 0.224	0.780 ± 0.135	0.940 ± 0.027
F_{23}	-0.333 ± 0.141	-0.320 ± 0.134	-0.167 ± 0.043
boundary flow	2.000 ± 0.424	1.881 ± 0.139	2.047 ± 0.020

an entry of $\mathbf{z}$ but the sum of two, so on $\mathbf{z}$ this meter is $\mathbf{h} = \mathbf{e}_1 + \mathbf{e}_2$, and on the loads it is the sum of the first two rows of $\mathbf{S}$, the direction $(-1, -1)$. That is not the first meter's direction, and the two together pin both loads: $P_2 = -1.273 \pm 0.060$, $P_3 = -0.773 \pm 0.069$, the flow on the line between them -0.167 ± 0.043. The prior had put P_3 at -0.5; the two meters have moved it by nearly a full prior spread, to -0.77, because the boundary meter says the total is 2.05 and the line meter says most of it is not at bus 2. Neither meter reads P_3; both inform it through the loop. This is the meshed twin of the feeder's virtual sensor, and the loop is what makes it work: in a radial network a meter on one feeder says nothing about the load on another.

3.6 Reading the graph

The example computed numbers. The graph shows, before any number is computed, why they came out as they did.

The sensor ℓ_2 in Figure 3.1 touches only V_2. The injection G is three squares away: through c_{2s} to Q_{2s}, through n_2 to Q_{12}, through n_1 to G. Yet the reading moved G more than it moved anything else. Information travels along every path in the factor graph, and how much arrives depends on the factors along the way, not on the number of steps. A stiff closure passes information nearly unattenuated; a wide prior receives it readily. The reading reached G through three exact physics factors and found a prior two hundredths of a per unit wide. It reached V_s through one exact factor and found a prior three thousandths wide. It moved what was loosest.

The two priors π_G and π_{V_s} share no factor, and under the prior their variables are independent. After the reading they are correlated at -0.985. The reading is a factor on V_2, and V_2 is tied to both. Two variables that are independent become dependent when something they both influence is observed.

The size of the correlation has a closed form, and it is worth deriving because it says when the effect is severe. Let two independent inputs $u_1 \sim \mathcal{N}(\mu_1, s_1^2)$ and $u_2 \sim \mathcal{N}(\mu_2, s_2^2)$ be read together through $y = au_1 + bu_2 + \nu$, $\nu \sim \mathcal{N}(0, \sigma^2)$. With $\mathbf{\Sigma} = \mathrm{diag}(s_1^2, s_2^2)$ and $\mathbf{h} = (a, b)$, Proposition 3.1 gives $\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma}\mathbf{h}\mathbf{h}^{\mathsf{T}}\mathbf{\Sigma}/(a^2 s_1^2 + b^2 s_2^2 + \sigma^2)$, whose off-diagonal entry is $-abs_1^2 s_2^2/D$ and whose diagonal entries are $s_1^2(b^2 s_2^2 + \sigma^2)/D$ and $s_2^2(a^2 s_1^2 + \sigma^2)/D$, with D the denominator. The posterior correlation is therefore

$$\rho' \;=\; -\frac{ab\, s_1 s_2}{\sqrt{(a^2 s_1^2 + \sigma^2)(b^2 s_2^2 + \sigma^2)}}. \tag{3.11}$$

It tends to -1 (for $ab > 0$) as the sensor becomes accurate relative to both inputs' contributions, and to zero as the sensor becomes useless; it is large exactly when the sensor reads a combination

it can resolve much better than either prior could. For the feeder, $a = 0.5$, $s_1 = 0.02$, $b = 1$, $s_2 = 0.003$, $\sigma = 0.0005$: $\rho' = -0.5 \cdot 0.02 \cdot 0.003/\sqrt{(1.0025 \times 10^{-4})(9.25 \times 10^{-6})} = -0.985$, the table's value. The sensor resolves $V_s + 0.5G$ to five parts in ten thousand, where the prior knew it to about a hundredth; that is a twenty-fold gain along the ridge and none across it, and the correlation is the ridge's aspect ratio. This is the same fact that Chapter 2 met as "factorization is not independence": local factors, global dependence. It has a name in the graphical-model literature, *explaining away*, and in Chapter 5 it is stated precisely as a rule for reading conditional independence off the graph.

> **Principle 3.2.** *A measurement anywhere is information everywhere. It travels along the factor graph, attenuated by tight factors and absorbed by loose ones, and it correlates whatever it cannot separate.*

3.7 In practice

Write down what you know about every input. A prior is not a nuisance to be set vague; it is the record of the data sheet, the forecast and the history, and its width decides how much a reading moves that input. An input given a width of a thousand will absorb every reading; one given a width of zero is a boundary condition.

Give the physics a width and say why. A closure's width is the modeler's bound on the law's error, and it should be justified in those terms: contact resistance, a neglected radiative term, a linear approximation. Balance rows get a small width so that the model can report a violation rather than refuse one; Chapter 4 says how small.

Check the innovation before believing the update. The innovation, reading minus prediction, in units of the predicted spread, is the first number to look at. A reading three spreads from its prediction is either a discovery or a defect, and the posterior will absorb it either way. Chapter 12 turns this into a test.

Ask what a sensor reads before buying it. By Proposition 3.1 the posterior spread after a reading does not depend on the reading. Compute it for every candidate sensor and buy the one that narrows the quantity you care about; a sensor that reads a direction the prior has already pinned buys nothing.

Expect correlations and report them. Two readings of the same kind leave the inputs correlated, as the feeder's two voltages left G and V_s at -0.98 and the triangle's two meters left the loads at -0.96. A report that gives each input its own error bar and no correlation has thrown away the posterior's most useful content: the combination that is known well and the combination that is not.

Keep the deterministic limit in view. Every posterior in this chapter tends to the deterministic solve as the priors widen and the physics widths vanish. When a posterior surprises, take that limit and see whether the surprise is in the physics or in the widths.

3.8 What is missing

Three things were done in this chapter without justification, and the next three chapters supply it.

The physics rows were softened with widths σ_k, and in the worked example they were then set to zero and the physics eliminated by hand. Chapter 4 says what the product of factors is when the widths are zero, what the widths do to it, and how to choose them.

The graph was read for how information travels and where it correlates. Chapter 5 makes that reading exact: which variables, given which others, are independent, and what a separator is.

The worked example was Gaussian throughout and solved in closed form. Chapter 6 shows that when every row is linear and every factor Gaussian the posterior is Gaussian, that its precision is the weighted normal matrix of the Jacobian with the sparsity of Chapter 2, and that the whole computation is sparse linear algebra. Chapter 7 then asks what survives when a row is nonlinear or a variable is discrete, as a_{23} already is.

Notes and further reading

Attaching priors to the uncertain inputs of a physical model, reading sensors as likelihoods, and asking for the posterior over the physical state is the Bayesian formulation of inverse problems; Stuart [87] gives the mathematical framework and Kaipio and Somersalo [38] the computational one, and both treat the Gaussian case of this chapter's worked example at length. The same objective, weighted least squares over physics residuals and measurement misfits, has two older engineering homes. In power systems it is state estimation, introduced by Schweppe [82] and developed in the books of Monticelli [65] and Abur and Gómez Expósito [1]; in chemical process engineering it is data reconciliation, of which Mah, Stanley and Downing [58] is the classical statement, with conservation constraints, estimation of unmeasured flows and detection of gross errors, all of which reappear in Chapter 12. Both traditions usually take the physics as hard and carry no priors on the inputs; the soft physics and the input priors of this chapter are what let a single formulation cover estimation, diagnosis and the deterministic solve.

The closest precedent for a factor graph over a physical network with message passing on it is Chavali and Nehorai's distributed power-system state estimation [14]. Its scope is one domain and one query; the framework of this book generalizes the graph to multiple conservation domains and the queries to those of Part IV, but the reader should know that the construction of Figure 3.2 has been drawn before.

The remark in §3.1 that a density can be read as a frequency or as a degree of belief, and that the machinery is the same, is the position of MacKay [57], whose book is the best general reference for the reader who wants the probability of this chapter developed with the same practical emphasis. Explaining away, named in §3.6, is Pearl's term [71], and Chapter 5 gives it a precise form for physical graphs.

Summary

- A density on the state assigns a likelihood to every point of $\mathbb{R}^n$. Marginalization integrates out, conditioning divides, factorization multiplies local pieces.

- Five kinds of factor: priors on uncertain inputs, closure and conservation factors with widths, likelihoods from sensors, and failure priors on discrete variables. The physics factors are the rows of Chapter 1; priors and likelihoods are unary squares added to the graph.
- The posterior is the product, equation (3.7). Its negative logarithm is a weighted least-squares objective whose deterministic limit is the solve of Chapter 1.
- One linear reading moves the mean by the innovation times a gain and shrinks the covariance by a rank-one amount in one direction; the shrinkage does not depend on the reading's value.
- One voltage reading a bus away from an uncertain injection cut its spread by a factor of three and a half and predicted an unmeasured voltage to ± 0.0013 per unit. The second reading landed inside that prediction.
- On the triangle, one line meter narrowed the total load by a factor of three and moved an unmetered load through the loop; a second meter in a different direction pinned both.
- Information travels along the graph and is absorbed by the loosest factors; observing a shared effect correlates independent causes.

Exercises

Exercise 3.1. Repeat the worked example with the sensor on V_1 read first and the sensor on V_2 second. Show that the final posterior is the same and that the intermediate one is different. Which single sensor, on V_1 or on V_2, tells more about G, and why?

Exercise 3.2. Add a third sensor to the worked example that reads the substation busbar voltage directly, first with accuracy 0.001 per unit and then with 0.0003. Compute the posterior in each case and show that the correlation between G and V_s falls in magnitude, to about 0.6 in the second case. Explain in terms of Figure 3.1 why this sensor breaks the ridge and the first two did not.

Exercise 3.3. In the worked example, free the susceptance b_{12} with the prior $\mathcal{N}(5, 1^2)$ and keep everything else. The map from inputs to V_1 is now nonlinear in (G, b_{12}). Write the objective of equation (3.8) for this case, find its minimizer numerically, and compare the minimizer with the second column of Table 3.2.

Exercise 3.4. Give the balance row n_2 of the feeder a width σ_n and suppose the two sensors read $y_2 = 1.027$ and $y_1 = 1.050$, inconsistent with any state of the exact model. Show that as $\sigma_n \to 0$ the objective's minimum grows without bound, and that for finite σ_n the minimizer violates the balance by an amount that identifies the size of an unmodeled reactive draw at bus 2.

Exercise 3.5. For the triangle of Figure 3.2, write the joint distribution as a product over the two values of a_{23}: $p(\mathbf{z}) = \mathbb{P}(a_{23} = 1)\, p(\mathbf{z} \mid a_{23} = 1) + \mathbb{P}(a_{23} = 0)\, p(\mathbf{z} \mid a_{23} = 0)$. Draw the factor graph for each of the two terms and state what has happened to the loop in the second.

Solutions to selected exercises

Exercise 3.1. Reading $y_1 = 1.038$ first gives $G = 0.0541 \pm 0.0042$ and $V_s = 1.0001 \pm 0.0029$, correlated at -0.986; reading y_2 second gives $G = 0.0546 \pm 0.0029$, $V_s = 0.9997 \pm 0.0017$, correlated at -0.979, which is the third column of Table 3.2 to fourteen digits. The order cannot matter, because the posterior is the product of the prior and both likelihoods, and a product

does not depend on the order of its factors; Proposition 3.1 applied twice is one way of computing that product, and the intermediate posterior is the only thing that depends on which factor went in first. The sensor on V_1 alone leaves G at ± 0.0042, the sensor on V_2 alone at ± 0.0058. The reason is the slope: $V_1 = V_s + 0.7G$ rises seven tenths of a per unit of voltage for each per unit of injection and $V_2 = V_s + 0.5G$ five tenths, so at equal accuracy the V_1 sensor has the larger lever arm on the injection. Both sensors share the same unit slope on V_s, which is why neither separates the two.

Exercise 3.2. With the two voltage sensors read, a busbar sensor of accuracy 0.001 per unit reading 0.9995 gives $G = 0.0549 \pm 0.0015$, $V_s = 0.9996 \pm 0.0009$, and a correlation of -0.924; with accuracy 0.0003 it gives $G = 0.0550 \pm 0.0008$, $V_s = 0.9995 \pm 0.0003$, and -0.636. The first two sensors both read "substation voltage plus a multiple of the injection" and so both lie along the ridge $V_s + cG$; in Figure 3.1 they attach to V_2 and V_1, which are tied to V_s and G through the same chain of factors, and the ridge is what that chain cannot distinguish. The busbar sensor attaches to V_s alone, reads the direction $(0, 1)$ of the (G, V_s) plane, which crosses the ridge, and Proposition 3.1 shrinks the covariance in a direction that no earlier reading had touched. The correlation does not vanish, because the ridge is only partly crossed at accuracy 0.001 and mostly crossed at 0.0003; a sensor on G itself, or on a reactive flow, would cross it at right angles.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

A row you do not trust *a physics row is a prior with a width; where probability enters a deterministic model; a mis-specified row held tight drags the estimate*
`foundation_electrical/physics_trust_calibration/problems/a_row_you_do_not_trust`

A failure rate from a short record *a count is not a Gaussian; a parameter that must stay positive; what changes when the unknown is the log of a rate*
`power_grid/reliability_rate_inference/problems/a_rate_from_a_short_record`

How long until the rate is known? *what accumulates and what does not as a record lengthens; precision set by events counted rather than years elapsed; the inverse square root law stated exactly*
`power_grid/reliability_rate_inference/problems/how_long_until_the_rate_is_known`

The rating you can publish *a rating as a quantile rather than a point estimate; the minimum of several uncertain quantities; why uncertainty costs capacity even at the median*
`transmission/dlr_rating_uncertainty/problems/the_rating_you_can_publish`

Chapter 4

Hard constraints, soft constraints, and the manifold

Chapter 3 gave every physics row a width and then, in its worked example, set the widths to zero and eliminated the physics by hand. Both moves were made without justification. This chapter supplies it. It says what the product of factors is when the widths are zero, what the widths do when they are not, why a working model uses finite widths, and how to choose them. The chapter ends with the same export feeder as before, now with all six unknowns kept and the physics rows soft, and shows the soft posterior converge to the hard one as the widths shrink, and the price paid for shrinking them.

4.1 The hard joint and its manifold

Split the unknowns into two groups. The *inputs* $\mathbf{u}$ are the quantities that carry priors: parameters, boundary conditions, sources. The *solved variables* $\mathbf{x}$ are everything else: potentials, flows, storage states. The physics rows read both, $\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, and there are as many rows as solved variables, $m = n_x$, with the Jacobian in the solved variables, $\mathbf{J}_x = \partial\mathbf{F}/\partial\mathbf{x}$, invertible. That is the well-posedness of §1.5 restated: for each value of the inputs the physics determines the solved variables. Write the determined value as $\mathbf{x} = X(\mathbf{u})$. The function X is what a solver computes; it is not available in closed form except for the smallest models, but it exists, and it is smooth wherever $\mathbf{J}_x$ is invertible.

Now take the widths of Chapter 3 to zero. A physics factor $\exp(-r_k^2/2\sigma_k^2)$, divided by its normalizer $\sqrt{2\pi}\sigma_k$, becomes the delta function $\delta(r_k)$: zero away from $r_k = 0$, and with unit integral across it. The product of the m physics factors is $\delta(\mathbf{F}(\mathbf{x}, \mathbf{u}))$, which is zero except on the set

$$\mathcal{M} \;=\; \{(\mathbf{x}, \mathbf{u}) : \mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{0}\} \;=\; \{(X(\mathbf{u}), \mathbf{u})\}, \tag{4.1}$$

the graph of X. This set is the *constraint manifold*. It has dimension n_u, the number of inputs, inside a space of dimension $n_x + n_u$. Every physically admissible state of the system lies on it, and no state off it is admissible.

The hard joint, prior times physics, is therefore not a density on $\mathbb{R}^n$ at all. It is a density on $\mathcal{M}$: a distribution over the inputs, carried along by X to the solved variables. The natural way to write it is

$$p(\mathbf{x}, \mathbf{u}) \;=\; p(\mathbf{u})\; \delta\big(\mathbf{x} - X(\mathbf{u})\big), \tag{4.2}$$

which says: draw the inputs from their prior, solve, and the solved variables are what the solve returns. Its marginal over the inputs is the prior, as it should be, since physics alone says nothing about which inputs are likely. Its marginal over any solved variable is the distribution of that variable under the prior, the *prior predictive*, which the worked example of §3.4 computed for V_2 as 70.0 ± 10.4.

Remark 4.1 (Two delta functions). Equation (4.2) and the product of row factors $p(\mathbf{u})\,\delta(\mathbf{F}(\mathbf{x},\mathbf{u}))$ are not the same function. Integrating the second over $\mathbf{x}$ gives $p(\mathbf{u})/|\det \mathbf{J}_x(X(\mathbf{u}),\mathbf{u})|$, by the change of variables $\mathbf{r} = \mathbf{F}(\mathbf{x},\mathbf{u})$ at fixed $\mathbf{u}$. The two agree when $\mathbf{J}_x$ is constant, which is when the physics is linear in the solved variables, and differ otherwise by a factor that weights inputs according to how compliant the physics is there. The general statement is the coarea formula of geometric measure theory [22, 23]: a product of delta functions of residuals is not invariant under a reparameterization of the residuals unless the corresponding Jacobian factor is carried along, and equation (4.2) is the parameterization-free object. Within one model the factor matters only where the physics is nonlinear, and for the models of this book it is small there. Between two models whose physics rows differ it is neither small nor common: comparing a branch in which a line is in service with one in which it is out, by the evidence of their soft row factors, weights each by $1/|\det \mathbf{J}_x|$ of its own physics, and on the DC triangle the two determinants differ by a factor of three. Chapter 7 shows the error this makes and states the correction (Proposition 7.1).

4.2 Conditioning on the manifold

A sensor reads a solved variable x_j with likelihood $\ell(x_j)$. Conditioning the hard joint on the reading multiplies equation (4.2) by $\ell(x_j)$ and, since $x_j = X_j(\mathbf{u})$ on the manifold, gives

$$p(\mathbf{u} \mid y) \;\propto\; p(\mathbf{u})\,\ell\big(X_j(\mathbf{u})\big). \tag{4.3}$$

This is the *input-space* form of the posterior. It is a distribution over the inputs alone. The solved variables are recovered from it by X: their posterior is the distribution of $X(\mathbf{u})$ when $\mathbf{u}$ is drawn from equation (4.3).

The worked example of §3.4 was exactly this computation. Its inputs were $\mathbf{u} = (G, V_s)$; its X was the linear map $\mathbf{z} = \mathbf{S}\mathbf{u}$ from the inputs to the whole unknown vector; its posterior was equation (4.3) with Gaussian prior and Gaussian likelihood, which is Gaussian, and Table 3.2 tabulated it. The manifold was a plane in $\mathbb{R}^6$ parameterized by two numbers, and the whole calculation lived on that plane.

For the feeder the map is explicit. With inputs $\mathbf{u} = (G, V_s)$ and solved variables $\mathbf{x} = (Q_{12}, Q_{2s}, V_1, V_2)$, flows first as in Example 1.1,

$$X(\mathbf{u}) = \begin{pmatrix} G \\ G \\ V_s + 0.7\,G \\ V_s + 0.5\,G \end{pmatrix}, \qquad \frac{\partial X}{\partial \mathbf{u}} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0.7 & 1 \\ 0.5 & 1 \end{pmatrix}, \tag{4.4}$$

which is the matrix $\mathbf{S}$ of §3.4 without its last two rows, the ones that return the inputs themselves. The derivative is dense: every solved variable depends on the injection, and the two voltages on both inputs, although no single physics row read more than three variables. Exercise 4.1 asks for the general statement.

The input-space form is exact, and it is always available in principle. It has two costs. The first is that every evaluation of the right-hand side of equation (4.3) at a new $\mathbf{u}$ needs $X(\mathbf{u})$,

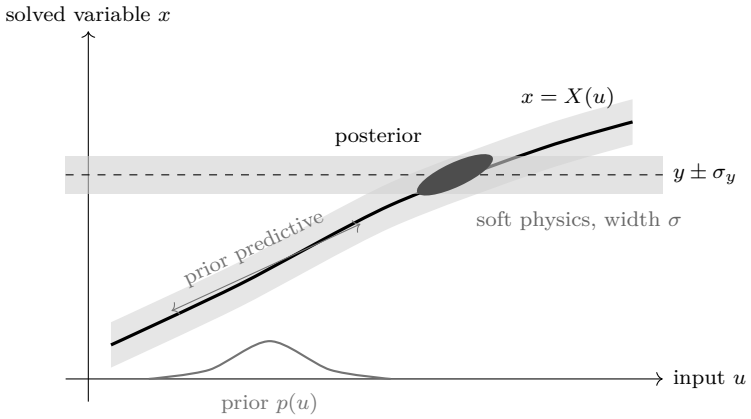


Figure 4.1: One input, one solved variable. The thick curve is the constraint manifold $x = X(u)$; the hard joint lives on it. The grey band around the curve is where the soft joint puts its mass, a distance of order σ from the curve. The prior on u sits on the horizontal axis and is carried up the curve to the prior predictive on x. A sensor reading y is a horizontal band. The posterior is the product: the dark region where the sensor band meets the curve, weighted by how much prior mass lies below it. Softening thickens the curve; it does not move the intersection.

which is a solve of the physics; and every gradient needs $\partial X/\partial \mathbf{u} = -\mathbf{J}_x^{-1}\mathbf{J}_u$, which is a solve per input or an adjoint solve per output. Chapter 14 makes much of this derivative, but it is not free. The second cost is structural. The physics of Chapter 1 was sparse: each row read a few variables. The map X is not. Every solved variable depends, in general, on every input, because information travels along the whole graph, so $\partial X/\partial \mathbf{u}$ is a dense $n_x \times n_u$ matrix and the factor graph of equation (4.3) is a single factor of scope $\mathbf{u}$. Eliminating the physics eliminates the sparsity with it.

4.3 The soft joint

Keep the widths finite. The joint is then

$$p_\sigma(\mathbf{x}, \mathbf{u} \mid y) \;\propto\; p(\mathbf{u})\, \ell(x_j) \prod_{k=1}^{m} \exp\!\left(-\frac{r_k(\mathbf{x}, \mathbf{u})^2}{2\sigma_k^2}\right), \tag{4.5}$$

a density on all of $\mathbb{R}^{n_x+n_u}$, positive everywhere, largest near the manifold and falling off away from it on a scale set by the widths. Figure 4.1 draws the situation for one input and one solved variable.

Three things happen as the widths go to zero, and they are what the chapter promised to establish.

Proposition 4.1 (Zero-width limit)**.** *Let the physics be linear in* $(\mathbf{x}, \mathbf{u})$*, the prior and the likelihood Gaussian, and all widths* $\sigma_k = \sigma$*. Then the soft posterior* (4.5) *is Gaussian for every* $\sigma > 0$*, and as* $\sigma \to 0$*:*

1. *its marginal over the inputs converges to the hard posterior* (4.3)*;*
2. *its mean converges to the point of* $\mathcal{M}$ *that minimizes the prior and likelihood terms of equation* (3.8) *subject to* $\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$*;*

3. *the posterior standard deviation of each physics residual r_k is at most σ, so the mass off the manifold vanishes with the width.*

Proof. Write the linear physics as $\mathbf{F} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with $\mathbf{A}$ square and invertible, so that $X(\mathbf{u}) = -\mathbf{A}^{-1}\mathbf{B}\mathbf{u}$. The soft joint is the exponential of a negative quadratic in $(\mathbf{x}, \mathbf{u})$, hence Gaussian for every $\sigma > 0$.

For item 1, change variables from $(\mathbf{x}, \mathbf{u})$ to $(\mathbf{r}, \mathbf{u})$ with $\mathbf{r} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$; the map is linear with constant Jacobian $\det \mathbf{A}$, so densities transform by a constant. In the new variables $\mathbf{x} = X(\mathbf{u}) + \mathbf{A}^{-1}\mathbf{r}$ and the soft joint is $p(\mathbf{u})\,\ell(X_j(\mathbf{u}) + (\mathbf{A}^{-1}\mathbf{r})_j)\,\exp(-\|\mathbf{r}\|^2/2\sigma^2)$. The last factor, normalized, is a Gaussian on $\mathbf{r}$ of spread σ about zero; integrating $\mathbf{r}$ out gives $p(\mathbf{u})$ times the average of $\ell(X_j(\mathbf{u}) + \epsilon)$ over $\epsilon \sim \mathcal{N}(0, \sigma^2\|(\mathbf{A}^{-1})_{j\cdot}\|^2)$, which for the Gaussian ℓ is a Gaussian in $X_j(\mathbf{u})$ whose spread is the sensor's with $\sigma^2\|(\mathbf{A}^{-1})_{j\cdot}\|^2$ added in quadrature. As $\sigma \to 0$ this tends to $\ell(X_j(\mathbf{u}))$, and the marginal to the hard posterior (4.3).

For item 2, let $\mathbf{z}_\sigma$ minimize $\Phi_\sigma(\mathbf{z}) = \Phi_0(\mathbf{z}) + \|\mathbf{F}(\mathbf{z})\|^2/2\sigma^2$, with Φ_0 the prior and likelihood terms, and let $\mathbf{z}^\star$ minimize Φ_0 subject to $\mathbf{F} = \mathbf{0}$. Since $\mathbf{z}^\star$ is feasible, $\Phi_\sigma(\mathbf{z}_\sigma) \le \Phi_\sigma(\mathbf{z}^\star) = \Phi_0(\mathbf{z}^\star)$, so $\|\mathbf{F}(\mathbf{z}_\sigma)\|^2 \le 2\sigma^2(\Phi_0(\mathbf{z}^\star) - \Phi_0(\mathbf{z}_\sigma))$, which tends to zero because Φ_0 is bounded below. Every limit point of $\mathbf{z}_\sigma$ is therefore feasible, and has $\Phi_0 \le \Phi_0(\mathbf{z}^\star)$ by the same inequality, so it is a constrained minimizer; for a strictly convex Φ_0 the minimizer is unique and $\mathbf{z}_\sigma \to \mathbf{z}^\star$.

For item 3, the posterior precision contains the term $\mathbf{h}_k\mathbf{h}_k^\mathsf{T}/\sigma^2$ for the residual $r_k = \mathbf{h}_k^\mathsf{T}\mathbf{z}$, and the remaining terms are positive semidefinite. Adding a positive semidefinite matrix to a precision cannot increase any variance, so the posterior variance of r_k is at most its variance under the precision $\mathbf{h}_k\mathbf{h}_k^\mathsf{T}/\sigma^2$ alone, restricted to the direction $\mathbf{h}_k$, which is σ^2; Exercise 4.2 asks for the one-line matrix argument. □

For nonlinear physics the same statements hold locally, with the volume factor of Remark 4.1 appearing in the first.

The proposition is what licenses the two shortcuts of Chapter 3. The worked example eliminated the physics and computed on the manifold; that is the $\sigma \to 0$ limit of the soft model, and §4.6 confirms it numerically. And the objective of equation (3.8) was described as having the deterministic solve as its limit; that is item 2 with no prior and no sensor.

4.4 Why soften

If the hard model is exact and the soft model converges to it, why not use the hard model? Three reasons, in increasing order of importance.

The soft model is an unconstrained problem. Minimizing the prior and sensor terms subject to $\mathbf{F} = \mathbf{0}$ is a constrained problem, and its optimality conditions are a saddle-point system in the unknowns and m Lagrange multipliers. Minimizing equation (4.5)'s negative logarithm is an unconstrained least-squares problem whose optimality conditions are a positive definite linear system, or, for nonlinear physics, a sequence of them. The second is what the Newton iteration of Chapter 1 already solves, with more rows. The width is the reciprocal square root of a penalty weight, and the soft model is the quadratic penalty method for the constrained one. That method is old, and its limitation is known: the penalty weight cannot be made arbitrarily large, because the conditioning of the linear system degrades with it. §4.5 returns to this.

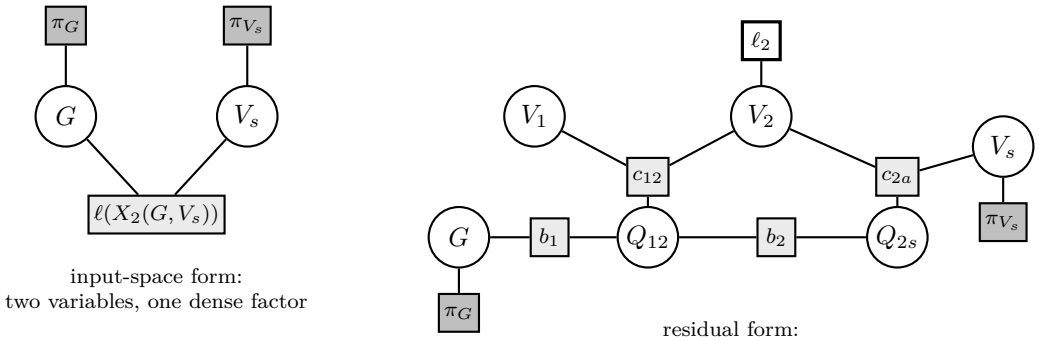


Figure 4.2: The same posterior in two forms. Left: the input-space form of §4.2, in which the physics has been solved and the likelihood is one factor reading every input. Right: the residual form of the soft joint, in which the physics rows remain as factors of small scope and the sensor reads one variable. On six variables the difference is cosmetic; on a network it is the difference between a dense matrix on the inputs and a sparse one on everything.

The soft model keeps the sparsity. This is the structural reason, and Figure 4.2 draws it. The factor graph of equation (4.5) is the factor graph of Chapter 3: physics squares of small scope, unary priors and sensors. Its precision matrix, which Chapter 6 constructs, has the sparsity of the Markov graph of §2.5, linear in the size of the model. The hard model in input-space form has thrown that away: a single dense factor on the inputs. For six unknowns the difference is nothing. For a network of ten thousand nodes it is the difference between a sparse factorization that takes a fraction of a second and a dense one that does not fit in memory. Every decomposition method of Part III, and every scaling argument of Part V, works on the soft graph.

The soft model can be wrong in a way that shows. This is the reason that matters most in practice. The hard model asserts that the state lies on $\mathcal{M}$. If the sensors report something that no point of $\mathcal{M}$ can produce, the hard model has no answer: the likelihood is zero everywhere on the manifold, and the posterior is undefined. A real system produces such readings routinely, because the model is missing something: an unmodeled draw, a bypass, a load the graph does not show, a sensor that has drifted. The soft model always has an answer. It places its posterior off the manifold, by an amount that trades the physics widths against the prior and sensor widths, and the residuals r_k at that posterior say which rows were violated and by how much. Those residuals are the diagnostic. A model that forbids violation cannot report it; a model that permits it, at a cost, reports both the violation and the cost. Chapter 12 builds fault detection on exactly this, and §4.6 shows a first instance.

> **Principle 4.1.** *Physics is softened not because it is uncertain but because the model of it is. A finite width keeps the problem unconstrained, keeps the graph sparse, and lets a violated law be seen and measured rather than forbidden.*

4.5 Choosing the widths

The widths are numbers the modeler supplies, and there are four sources of them.

Table 4.1: Where the widths come from, with the values used in the book's examples. Relative widths are fractions of the row's scale after the equilibration described in the text.

factor	width is	source	examples in the book
sensor	instrument accuracy	data sheet, calibration	0.0005 p.u. voltage transducer; 0.02 p.u. flow meter
prior	spread of the input	data sheet, forecast, history	injection ±0.02 p.u.; load ±0.3 p.u.; set point ±0.003 p.u.
closure	bound on the law's error	neglected terms, handbook scatter	10^{-3} relative when the law is trusted
balance	admissible violation	model completeness, numerics	10^{-3} relative; loosened to find an unmodeled draw

Sensor widths come from the instrument. A thermocouple's data sheet states its accuracy; a power meter's states its class. These are the least negotiable numbers in the model, and the worked examples use them as given.

Prior widths come from data sheets, forecasts, and history, as §3.2 said. They are statements about the modeler's ignorance and should be honest: a susceptance known to twenty percent gets a twenty percent width, not five, and a set point whose past errors had a spread of three parts in a thousand gets three parts in a thousand.

Closure widths come from what is known about the law's error. A linearized law that neglects a term of known size gets a width of that size. A correlation from a handbook with a stated scatter gets that scatter. A law believed exact, such as a DC line flow at fixed susceptance, gets a small width, but not zero, for the reasons of the previous section.

Balance widths are the small ones. A balance is exact for the system as modeled; its width is there to admit violation when the model is incomplete, and to keep the problem unconstrained. It should be small compared with the flows it balances, and no smaller than the numerics allow. Table 4.1 collects the four sources with the values the book's examples use.

That last clause has a definite content. The physics rows have units, a balance in per-unit power, a closure in per-unit power or voltage, and a width must be stated in the row's units. It is convenient, and in a working implementation necessary, to scale each row so that its typical magnitude is of order one, and then to state widths as fractions of that scale. Call the fraction the *relative width*. With rows so scaled, the precision matrix of the soft model has entries of order one from the priors and sensors and of order $1/\rho^2$ from the physics at relative width ρ, and its condition number grows as $1/\rho^2$. The connection between condition number and lost accuracy is a standard result of numerical analysis [34]: a backward-stable solve loses about as many decimal digits as the condition number has, so in double precision a condition number near 10^{10} leaves six digits and one near 10^{14} leaves two. The widths that follow from this are implementation heuristics, not universal constants; they depend on the row scaling, the solver and the precision. With rows equilibrated as described, a relative width of 10^{-3} for the balance rows is a comfortable choice, 10^{-4} is near the edge, and 10^{-5} fails on models of moderate size. The next section shows the growth on the feeder, and a reader with a different normalization should repeat the sweep on their own model.

Failure modes. A width too large lets the posterior satisfy the sensors by breaking the physics rather than by moving an input, and the estimates of the inputs stay near their priors when the data should have moved them. A width too small does the opposite: the posterior moves inputs to implausible values, and blames sensors, rather than admit a violation that the model should have allowed. Neither failure is silent. Both show in the *misfit*: the sum of squared standardized residuals at the posterior over every factor, physics, prior and sensor, which is twice the objective (3.8) there. Chapter 12 turns the misfit into a test, and the statement behind the test is worth deriving here, because it is what gives the number a scale.

Proposition 4.2 (The misfit of an adequate model)**.** *Let a model have m linear Gaussian factors, each of the form $\mathbf{h}_k^\mathsf{T}\mathbf{z} - c_k \sim \mathcal{N}(0, \sigma_k^2)$ with independent errors, on $n < m$ unknowns, with the stacked standardized rows $\mathbf{W}^{1/2}\mathbf{J}$ of full column rank. If the data were generated by the model, the misfit at the posterior mean, the sum of the squared standardized residuals, is a chi-squared variable with $m - n$ degrees of freedom; its expectation is $m - n$ and its standard deviation $\sqrt{2(m-n)}$.*

Proof. Standardize every factor: let $\mathbf{a}_k = \mathbf{h}_k/\sigma_k$ and $b_k = c_k/\sigma_k$, so that the model says $\mathbf{A}\mathbf{z} = \mathbf{b} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ and $\mathbf{A}$ the $m \times n$ matrix of rows $\mathbf{a}_k^\mathsf{T}$. The posterior mean minimizes $\|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2$, so it is the least-squares solution $\hat{\mathbf{z}} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{b}$, and the standardized residual vector is $\mathbf{b} - \mathbf{A}\hat{\mathbf{z}} = (\mathbf{I} - \mathbf{P})\mathbf{b}$ with $\mathbf{P} = \mathbf{A}(\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}$ the orthogonal projector onto the column space of $\mathbf{A}$. If the data came from the model, $\mathbf{b} = \mathbf{A}\mathbf{z}_{\text{true}} - \boldsymbol{\epsilon}$, and since $(\mathbf{I} - \mathbf{P})\mathbf{A} = \mathbf{0}$ the residual is $-(\mathbf{I} - \mathbf{P})\boldsymbol{\epsilon}$: the true state has dropped out. The misfit is $\boldsymbol{\epsilon}^\mathsf{T}(\mathbf{I} - \mathbf{P})\boldsymbol{\epsilon}$, a standard Gaussian vector projected onto the orthogonal complement of an n-dimensional subspace, which is the sum of $m - n$ independent squared standard normals: chi-squared with $m - n$ degrees of freedom. □

The proposition carries its assumptions: linear rows, Gaussian errors with the stated widths, independence. For nonlinear rows it holds locally; for widths that are wrong it fails in a way that is itself informative, since a misfit systematically below $m - n$ says the widths are too wide and one systematically above says they are too narrow or the model is missing something. A value far above $m - n$, by more than a few multiples of $\sqrt{2(m-n)}$, says the model, the widths, or the data are inconsistent. The statement is not a property of an arbitrary factor graph, and the example below shows one failure of each kind.

4.6 Worked example: the feeder with soft physics

Take the export feeder of Figure 3.1 with all six unknowns, $\mathbf{z} = (Q_{12}, Q_{2s}, V_1, V_2, G, V_s)$, the priors $G \sim \mathcal{N}(0.05, 0.02^2)$ and $V_s \sim \mathcal{N}(1.000, 0.003^2)$, the susceptances $b_{12} = 5$ and $b_{2s} = 2$ known, and the four physics rows each given the same width σ in per unit. Everything is linear, so the posterior is Gaussian and is computed by the script `ch04_soft_physics.py` in the book's `scripts` directory; every number below is from its output.

Convergence. With the single sensor $y_2 = 1.027$ of §3.4, Table 4.2 sweeps the width from 10^{-2} per unit down to 10^{-6}. At $\sigma = 10^{-2}$, a fifth of the flow, the physics is loose: the injection's posterior spread is 0.0135, the correlation with the substation voltage is only -0.25, and the largest physics residual at the posterior mean is five ten-thousandths of a per unit. The reading has been partly absorbed by bending the physics rather than by moving the inputs. At $\sigma = 10^{-3}$

Table 4.2: The feeder with soft physics, one sensor $y_2 = 1.027$: posterior of the injection and the substation voltage as the physics width shrinks. The last column is the condition number of the posterior precision matrix. Hard-physics reference from Table 3.2: $G = 0.05366 \pm 0.00582$, V_s to ± 0.00287, correlation -0.985. All quantities in per unit.

σ	mean G	sd G	sd V_s	corr	max $\lvert r_k \rvert$	cond
10^{-2}	0.05217	0.01352	0.00293	-0.247	5.4×10^{-4}	1.6×10^{3}
10^{-3}	0.05364	0.00603	0.00287	-0.944	9.1×10^{-6}	6.4×10^{3}
10^{-4}	0.05366	0.00582	0.00287	-0.985	9.2×10^{-8}	5.9×10^{5}
10^{-5}	0.05366	0.00582	0.00287	-0.985	9.2×10^{-10}	5.9×10^{7}
10^{-6}	0.05366	0.00582	0.00287	-0.985	9.2×10^{-12}	5.9×10^{9}

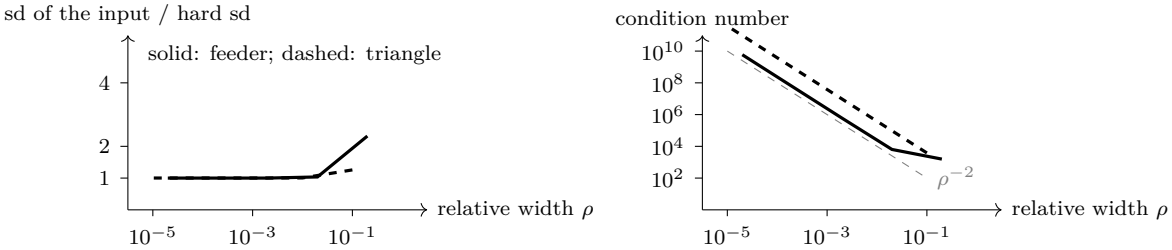


Figure 4.3: The width sweep on the feeder (solid) and on the DC triangle of §4.7 (dashed), against the relative width, the physics width divided by the scale of the flows (0.05 per unit for the feeder, one per unit for the triangle). Left: the posterior spread of the uncertain input relative to its hard-physics value; both reach the hard value at a relative width of 10^{-3}. Right: the condition number of the posterior precision, which grows as the inverse square of the relative width on both models.

the estimates are within a few percent of the hard values. At $\sigma = 10^{-4}$ and below they agree with Chapter 3's hard posterior to every digit shown, the residuals are below 10^{-7}, and the posterior spread of each residual equals the width, as Proposition 4.1 says. The soft model has become the hard one.

The last column is the price, and Figure 4.3 draws it together with the same sweep on the triangle of §4.7. Each factor of ten in the width costs a factor of a hundred in the condition number, as §4.5 predicted. At 10^{-6} per unit, which is a relative width of 2×10^{-5} against flows of five hundredths of a per unit, the condition number is 6×10^{9} on a six-variable model; a larger model with the same relative width would be past the edge. A width of 10^{-5} to 10^{-4} per unit gives the hard answer at a condition number a solver does not notice.

An inconsistent reading. Now keep every width at 10^{-5} per unit and read both sensors, $y_2 = 1.027$ as before and $y_1 = 1.050$. In the exact model $V_1 - V_2 = G/b_{12}$, so the two readings imply $G = 0.115$ per unit; but then $Q_{2s} = 0.115$ and $V_2 = V_s + 0.0575$, which with $V_2 = 1.027$ puts the substation busbar at 0.9695, ten prior spreads below its set point. No state of the exact model is consistent with both readings and both priors. The soft model with tight physics has to answer anyway, and its answer is in the first row of Table 4.3: injection 0.0972, substation voltage 0.9804, the V_2 sensor missed by two thousandths and the V_1 sensor by one and a half. Every physics residual is zero. The model has kept its laws and thrown away its priors and

Table 4.3: Inconsistent readings $y_2 = 1.027$, $y_1 = 1.050$: posterior as the width of the bus-2 balance is loosened while every other row stays at 10^{-5} per unit. The draw estimate is the violation of that balance at the posterior mean, in thousandths of a per unit. Last column: misfit at the posterior; about 2 is expected of an adequate model.

σ_{n_2}	draw [10^{-3} p.u.]	G	V_s	V_2, V_1 fitted	misfit
10^{-5}	0.0 ± 0.0	0.0972 ± 0.0029	0.9804 ± 0.0017	1.0291, 1.0485	74.0
10^{-3}	1.1 ± 1.0	0.0975 ± 0.0029	0.9808 ± 0.0018	1.0290, 1.0485	72.8
10^{-2}	38.1 ± 5.9	0.1075 ± 0.0033	0.9931 ± 0.0026	1.0279, 1.0493	32.6
10^{-1}	58.3 ± 7.3	0.1129 ± 0.0035	0.9999 ± 0.0030	1.0272, 1.0498	10.6

its sensors: the substation voltage sits six and a half of its prior spreads below the set point, and each sensor is off by three or four of its accuracies. The misfit at this posterior, the sum of squared standardized residuals over all eight factors, is 74; for an adequate model with eight factors and six unknowns it would be about 2. The model is telling the reader, loudly, that something is wrong, and it is telling the reader the wrong thing about what.

Suppose the modeler suspects the balance at bus 2: the bus may be drawing reactive power somewhere the graph does not show, into a station-service transformer say. Loosen that one row and leave the others tight. The remaining rows of Table 4.3 show what happens. At a width of 10^{-3} nothing changes; the row is still too stiff to absorb six hundredths of a per unit. At 10^{-2} the posterior has moved: the balance is violated by 0.038, the substation voltage has come back to 0.9931, the sensors are fitted to within a thousandth. At 10^{-1} the posterior is coherent: the balance at bus 2 is violated by 0.0583 ± 0.0073 per unit, the injection is 0.113, the substation voltage is 0.9999, a ten-thousandth from its set point, and both sensors are fitted to two ten-thousandths. The misfit has fallen from 74 to 10.6. It is still above 2, and the reason is visible in the table: the injection is three of its prior spreads above nominal. That is either a second thing wrong or a prior that was too narrow, and the modeler now knows to ask.

Read the residual of the loosened row as a number. The balance at bus 2 said reactive power in equals reactive power out. At the posterior, in exceeds out by 0.058 per unit. That is reactive power leaving bus 2 by a path the model does not contain, and the model has estimated it, with an error bar, from two voltage readings, without being told a draw exists. The row was violated, and the violation is the measurement.

For comparison, the consistent readings of §3.4, $y_2 = 1.027$ and $y_1 = 1.038$, through the same soft model with every width at 10^{-5} per unit, give $G = 0.0546 \pm 0.0029$ and $V_s = 0.9997 \pm 0.0017$, identical to the hard result, with a misfit of 0.07. An adequate model with consistent data sits near the bottom of the misfit's range; an inadequate one does not. Figure 4.4 draws the loosening sweep, beside the same sweep on the triangle that follows.

4.7 A second example: the triangle with an unmodeled load

The DC triangle of §3.5 carries the same lessons into the active-power domain, where the violation is a load the graph does not show. Keep all seven unknowns of §3.5, flows first: the three flows, the two angles and the two loads. Give the five physics rows a common width σ in per unit, the loads their priors -1.5 ± 0.3 and -0.5 ± 0.3, and read meters of accuracy 0.02. Every number is from `ch04_triangle_soft.py`.

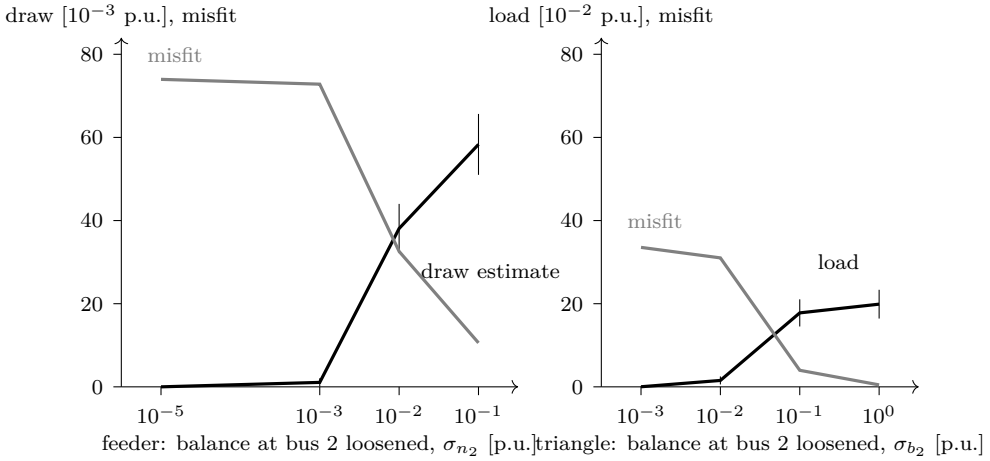


Figure 4.4: Loosening one balance row against inconsistent readings. Left: the feeder, the balance at bus 2 loosened; the draw estimate with its error bar (black) rises from nothing to 58 ± 7 thousandths of a per unit as the width passes 10^{-2}, and the misfit (grey) falls from 74 toward the value an adequate model would give. Right: the DC triangle of §4.7, the balance at bus 2 loosened; the unmodeled load rises to 0.20 per unit and the misfit falls from 34 to below one. In both, the row that is allowed to be violated is the row that reports the defect.

Table 4.4: The DC triangle with soft physics, one meter $F_{12} = 1.10$: posterior of the loads as the physics width shrinks. Hard reference from Table 3.3: $P_2 = -1.421 \pm 0.136$, $P_3 = -0.460 \pm 0.269$, correlation -0.976.

σ [p.u.]	P_2	P_3	corr	max $\lvert r_k \rvert$	cond
0.1	-1.433 ± 0.171	-0.466 ± 0.274	-0.649	7.5×10^{-3}	3.8×10^{3}
0.01	-1.421 ± 0.137	-0.460 ± 0.269	-0.971	8.8×10^{-5}	3.8×10^{5}
0.001	-1.421 ± 0.136	-0.460 ± 0.269	-0.976	8.8×10^{-7}	3.8×10^{7}
10^{-4}	-1.421 ± 0.136	-0.460 ± 0.269	-0.976	8.8×10^{-9}	3.8×10^{9}
10^{-5}	-1.421 ± 0.136	-0.460 ± 0.269	-0.976	8.8×10^{-11}	3.8×10^{11}

Convergence. With the single meter $F_{12} = 1.10$ of Chapter 3, Table 4.4 sweeps the width from 0.1 per unit, a tenth of the flows, down to 10^{-5}. At 0.1 the posterior is loose, the correlation between the loads only -0.65, and the physics residuals are of order 10^{-2}: the meter has been partly absorbed by bending the loop rather than by moving the loads. At 0.01 the loads are within a thousandth of the hard values and at 0.001 they agree to every digit shown, with the correlation -0.976 of Table 3.3. The condition number grows by a hundred per decade, exactly as on the feeder, and Figure 4.3 shows the two models' curves lying on top of each other once the width is expressed relative to the flows. A relative width of 10^{-3} is the comfortable choice on both.

Inconsistent readings. Now read three meters: the flow on line 1–2 at 1.10, the flow on line 2–3 at -0.20, and the load at bus 2 itself, from a meter on the feeder it supplies, at -1.50. The balance at bus 2 says that what arrives on the two lines is what the load draws, $F_{12} - F_{23} = -P_2$; the meters say 1.30 on the left and 1.50 on the right. No state of the exact model satisfies all

Table 4.5: Inconsistent meters on the triangle, $F_{12} = 1.10$, $F_{23} = -0.20$, $P_2 = -1.50$, each to 0.02: posterior as the width of the bus-2 balance is loosened while every other row stays at 0.001. The unmodeled load is the violation of that balance at the posterior mean. About 3 is expected of an adequate model.

σ_{b_2}	unmodeled load	P_2	P_3	F_{12}, F_{23} fitted	misfit
0.001	0.000 ± 0.001	-1.434 ± 0.016	-0.631 ± 0.043	1.166, -0.268	33.5
0.01	0.015 ± 0.010	-1.439 ± 0.017	-0.636 ± 0.043	1.161, -0.263	31.0
0.1	0.178 ± 0.033	-1.493 ± 0.020	-0.689 ± 0.044	1.106, -0.209	4.0
1	0.199 ± 0.035	-1.500 ± 0.020	-0.696 ± 0.044	1.099, -0.202	0.5

Table 4.6: The misfit at the posterior against its expectation under Proposition 4.2, for the cases of this chapter. The expectation is the number of factors minus the number of unknowns; the spread of an adequate model's misfit is the square root of twice that.

case	factors, unknowns	expected	misfit
feeder, consistent readings, tight physics	8, 6	2 ± 2	0.07
feeder, inconsistent readings, tight physics	8, 6	2 ± 2	74
feeder, inconsistent, bus-2 balance at 10^{-1}	8, 6	2 ± 2	10.6
triangle, inconsistent meters, tight physics	10, 7	3 ± 2.4	33.5
triangle, inconsistent, bus-2 balance at 1 p.u.	10, 7	3 ± 2.4	0.5

three to their accuracies, and the tight model of the first row of Table 4.5 answers by keeping its laws and spoiling its meters: the line flows are fitted at 1.166 and -0.268, three accuracies from their readings, the load at -1.434, three from its meter, and the misfit is 34 where an adequate model with ten factors and seven unknowns would give about three. Loosen the balance at bus 2 and the posterior finds the explanation: at a width of 0.1 the balance is violated by 0.18 ± 0.03 per unit and the misfit is 4; at a width of one it is violated by 0.20 ± 0.04, every meter is fitted to within an accuracy, and the misfit is 0.5. The violation is a draw of 0.2 per unit at bus 2 that the graph does not show, a load on a feeder the meter does not cover, or a tap the operator does not know about, and the model has measured it from three meters without being told it exists. The right panel of Figure 4.4 draws the sweep beside the feeder's.

Table 4.6 sets every misfit of the chapter beside its expectation. The consistent feeder and the loosened triangle sit below expectation, which says their widths are, if anything, generous; the two tight inconsistent models sit thirty and forty standard deviations above it, which is not a subtle signal; and the loosened feeder, at 10.6 against 2 ± 2, sits four deviations above, which is the residual doubt the text described.

Two differences from the feeder are instructive. The triangle's load is found at a width of 0.1 per unit, a tenth of the flows, whereas the feeder's needed 10^{-2} to 10^{-1}, a fifth to twice a whole flow; the reason is that the triangle's three meters pin the violation from both sides, while the feeder's two voltage sensors see the draw only through the injection prior. And the triangle's misfit falls all the way to 0.5, below its expected value, while the feeder's stops at 10.6; the triangle's data are explained entirely by one unmodeled load, and the feeder's are not, which is the signal that the feeder has a second thing wrong or a prior too narrow.

4.8 In practice

Scale the rows, then set relative widths. Divide every physics row by the typical size of what it balances so that its residual is of order one, and state widths as fractions of that. Relative widths of 10^{-3} reproduce the hard model on every example in this chapter at a condition number a solver does not notice; 10^{-5} is past the edge on a model of any size. Without the scaling, a width that is small for a transmission-level balance is enormous for a distribution-level one.

Sweep the width once. Run the model at three widths a decade apart and check that the posterior has stopped moving. If it has not, the widths are too loose and the physics is being bent to fit the data; if the condition number has grown past 10^{10}, they are too tight. The sweep is cheap and it is the only way to know where a given model sits.

Read the misfit before the estimates. The sum of squared standardized residuals over every factor, compared with the number of factors minus the number of unknowns, is the first number to report. A misfit near its expectation says the model, the widths and the data are consistent; a misfit far above it says they are not, and no estimate from that posterior should be believed until the reason is found.

Loosen the row you suspect, not every row. An inconsistency is explained by violating some row. Loosening all of them lets the posterior spread the violation wherever it is cheapest, which is usually wrong and always uninformative. Loosening one row asks a question, "is it this?", and the misfit answers it: the feeder's data were explained by a draw at bus 2 and not by anything at bus 1.

Report the violation as a measurement. The residual of a loosened row at the posterior, with its posterior spread, is an estimate of the unmodeled flow, and it should be reported in the row's units with its error bar, exactly as a measured quantity would be.

Keep the hard limit as a check. When the soft model surprises, tighten the widths and compare with the input-space form. If the two disagree, the widths were doing work the modeler did not intend.

4.9 What is missing

The chapter has said what the widths mean and shown what they do on a six-variable model where every posterior was computed by inverting a six-by-six matrix. It has not said how the same computation is organized when the matrix is a million by a million and sparse, which is Chapter 6, nor which variables can be computed without the others, which is Chapter 5. And it has assumed that every row is linear, so that the soft joint is Gaussian and the manifold is a plane. A nonlinear closure makes the manifold curved, the soft joint non-Gaussian, and Proposition 4.1 local rather than global; a discrete availability variable makes the manifold a union of pieces, one per value. Chapter 7 takes those up.

Notes and further reading

The soft model is the quadratic penalty method for equality-constrained least squares, and §4.4 and §4.5 restate what Nocedal and Wright [66] say about it: the penalized problem converges to the constrained one as the penalty weight grows, and its conditioning degrades as the weight grows, which is why the weight is chosen finite and why augmented-Lagrangian methods exist. The book keeps the plain penalty form because its normal equations are the same sparse system as the deterministic solve (Proposition 6.2 in Chapter 6) and because a finite width is wanted for its own sake, to admit violation. The relation between condition number and lost digits is in Higham [34].

Treating physics as a soft constraint with a modeler-chosen width, and letting the size of the violation at the posterior carry diagnostic meaning, is the practice of power-system state estimation, where zero-injection buses are handled either as hard constraints or as pseudo-measurements of very high weight [1, 65], and of process data reconciliation, where the residual of a balance is the signature of a leak or a bad meter [58]. The draw example of §4.6 is the data-reconciliation argument on the smallest possible flowsheet. The chi-squared test named in §4.5 is the standard adequacy test of both fields and of least-squares theory generally; its assumptions are stated there because they are often omitted.

The input-space form of §4.2, with the physics eliminated and the posterior written over the inputs alone, is how Bayesian inverse problems are usually posed [38, 87]: the forward map is $X(\mathbf{u})$, and its derivative is the sensitivity that adjoint methods compute. The observation that this form is dense while the residual form is sparse, and that the choice between them is the choice between Chapter 14's gradients and Chapter 8's elimination, is the book's framing.

Summary

- With zero widths the joint lives on the constraint manifold $\mathbf{x} = X(\mathbf{u})$, of dimension equal to the number of inputs. Its natural form is the prior on the inputs carried along by the solve.
- Conditioning on the manifold gives the input-space posterior $p(\mathbf{u})\,\ell(X(\mathbf{u}))$: exact, dense, and one solve per evaluation. The worked example of Chapter 3 was this.
- With finite widths the joint is a density on the full space, concentrated within about a width of the manifold. As the widths go to zero it converges to the hard posterior; the soft model is the quadratic penalty form of the constrained one.
- Soften because it keeps the problem unconstrained, keeps the graph sparse, and makes a violated law visible and measurable.
- Widths come from instruments, data sheets, forecasts, and known model error; balance widths are small relative to their flows, and the condition number grows as the inverse square of the relative width.
- On the feeder, widths of 10^{-4} to 10^{-5} per unit reproduce the hard posterior exactly; inconsistent readings through tight physics produce an absurd state with a misfit of 74; loosening one balance turns the inconsistency into an unmodeled draw of 0.058 ± 0.007 per unit and a misfit of 10.6.
- On the triangle the same sweep converges at a relative width of 10^{-3}, and three inconsistent meters become an unmodeled load of 0.20 ± 0.04 per unit at bus 2 with the misfit falling from 34 to 0.5. Loosening the wrong row explains nothing, and the misfit says so.

Exercises

Exercise 4.1. For the linear physics $\mathbf{F} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with $\mathbf{A}$ square and invertible, write $X(\mathbf{u})$ explicitly, verify that $\partial X/\partial \mathbf{u} = -\mathbf{A}^{-1}\mathbf{B}$, and show that this matrix is dense for the export feeder even though $\mathbf{A}$ and $\mathbf{B}$ are sparse.

Exercise 4.2. Prove item 3 of Proposition 4.1: in a Gaussian posterior whose precision includes the term $\mathbf{h}\mathbf{h}^{\mathsf{T}}/\sigma^2$ for a linear residual $r = \mathbf{h}^{\mathsf{T}}\mathbf{z}$, the posterior variance of r is at most σ^2. Use the fact that adding a positive semidefinite matrix to a precision cannot increase any variance.

Exercise 4.3. Re-run the sweep of Table 4.2 with the two closure rows at width 10^{-3} per unit and only the two balance rows swept. Explain, in terms of Figure 4.1, why the estimate of G converges to a different limit than in the table, and compute that limit by treating the closure widths as model error in the input-space form.

Exercise 4.4. In the unmodeled-draw example, loosen the balance at bus 1 instead of bus 2. Show that the posterior finds a different explanation of the same readings, state what physical defect it corresponds to, and compare the two misfits. Which explanation should the modeler prefer, and can the data alone decide?

Exercise 4.5. Derive the volume factor of Remark 4.1 for one solved variable and one input, $r(x, u) = 0$, by substituting r for x in the integral $\int \delta(r(x, u))\,\mathrm{d}x$. Then evaluate it for the closure $Q - b_{12}(V_1 - V_2)$ solved for V_1 with b_{12} as the input, and state when it is constant.

Solutions to selected exercises

Exercise 4.2. Write the posterior precision as $\mathbf{\Lambda} = \mathbf{h}\mathbf{h}^{\mathsf{T}}/\sigma^2 + \mathbf{M}$ with $\mathbf{M}$ positive semidefinite, the sum of every other factor's contribution. The posterior variance of $r = \mathbf{h}^{\mathsf{T}}\mathbf{z}$ is $\mathbf{h}^{\mathsf{T}}\mathbf{\Lambda}^{-1}\mathbf{h}$. For any positive definite $\mathbf{\Lambda}_1$ and positive semidefinite $\mathbf{M}$, $\mathbf{\Lambda}_1 + \mathbf{M} \succeq \mathbf{\Lambda}_1$ implies $(\mathbf{\Lambda}_1 + \mathbf{M})^{-1} \preceq \mathbf{\Lambda}_1^{-1}$, so adding $\mathbf{M}$ to a precision cannot increase any variance. But $\mathbf{h}\mathbf{h}^{\mathsf{T}}/\sigma^2$ alone is singular, so take $\mathbf{\Lambda}_1 = \mathbf{h}\mathbf{h}^{\mathsf{T}}/\sigma^2 + \epsilon\mathbf{I}$ with $\epsilon > 0$ small and $\mathbf{M} - \epsilon\mathbf{I} \succeq 0$ for ϵ below the smallest eigenvalue of $\mathbf{M}$ restricted to the directions where it is positive; then $\mathbf{h}^{\mathsf{T}}\mathbf{\Lambda}^{-1}\mathbf{h} \le \mathbf{h}^{\mathsf{T}}\mathbf{\Lambda}_1^{-1}\mathbf{h}$, and by the Sherman–Morrison identity $\mathbf{h}^{\mathsf{T}}\mathbf{\Lambda}_1^{-1}\mathbf{h} = \|\mathbf{h}\|^2/\epsilon \cdot (1 - \|\mathbf{h}\|^2/(\epsilon\sigma^2 + \|\mathbf{h}\|^2)) = \sigma^2\|\mathbf{h}\|^2/(\epsilon\sigma^2 + \|\mathbf{h}\|^2) < \sigma^2$. When $\mathbf{M}$ is singular in the direction $\mathbf{h}$ the bound is attained in the limit, which is why the posterior spread of every residual in Table 4.2 equals the width: the physics row is the only factor that constrains its own residual.

Exercise 4.4. Loosening the balance at bus 1 to 10^{-1}, with every other row at 10^{-5}, gives a violation of -0.046 ± 0.020 per unit, an injection of 0.052 ± 0.020, a substation voltage of 0.9799 ± 0.0017, and a misfit of 68.5; at 10^{-2} the violation is -0.009, and at 10^{-3} nothing has moved from the tight solution. The violation's sign says that 0.046 per unit more than the generator injects is leaving bus 1 down the line: a second, hidden source at bus 1. But the posterior has not been helped: the substation voltage is still two hundredths below its set point, nearly seven of its prior spreads, the sensors are still off by three or four accuracies, and the misfit has fallen from 74 only to 68.5. The reason is in the physics. The two readings fix $V_1 - V_2$ near 0.0196 and hence $Q_{12} = 0.098$ whatever balance is loosened; a draw at bus 2 lets Q_{2s} be less than Q_{12}, so that $V_2 - V_s$ can be 0.027 instead of 0.049 and the substation voltage can return to its set point, while a defect at bus 1 changes only where Q_{12} came from and leaves $Q_{2s} = Q_{12}$, so

the substation voltage must still sit near 0.980. The data decide: the bus-2 explanation reaches a misfit of 10.6 and the bus-1 explanation does not leave 68, and the modeler should prefer the first by a margin of nearly sixty units of misfit, which Chapter 12 will turn into a likelihood ratio. What the data cannot decide is whether the bus-2 draw is the whole story, since 10.6 is still above the expected 2; that residual doubt is about the injection prior, and only a meter on the generator or a better schedule can remove it.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Ratios, not reactances *the loop equation decides the split; flows are invariant to a common scaling of reactance; angles are not*
`foundation_electrical/dc_power_flow_audit/problems/ratios_not_reactances`

The slack is not optional *the conservation rows are linearly dependent by one; a connected lossless network determines the potential only up to a constant; why a slack bus is a gauge choice and not a modelling convenience*
`foundation_electrical/dc_power_flow_audit/problems/the_slack_is_not_optional`

Chapter 5

Factorization, separators, and treewidth

Chapter 3 read the factor graph informally: a measurement travels along every path, and two inputs that share an observed effect become correlated. This chapter makes the reading exact. It states what a factorization does and does not say about independence, gives the rule for reading conditional independence off the graph, shows that the ports of Chapter 1 are exactly the sets that rule singles out, and then turns to cost: what it takes to compute a marginal on a factor graph, why the answer is governed by a number called treewidth, and why a physical cut with few ports is a cheap place to divide a computation. Nothing in the chapter requires Gaussians; everything in it holds for any positive factorized density.

5.1 Factorization is not independence

Two variables are *independent* if their joint density is the product of their marginals, $p(x_1, x_3) = p(x_1)\, p(x_3)$; knowing one says nothing about the other. They are *conditionally independent given* a third if the same holds for every fixed value of the third:

$$p(x_1, x_3 \mid x_2) \;=\; p(x_1 \mid x_2)\, p(x_3 \mid x_2), \qquad \text{written } x_1 \perp x_3 \mid x_2. \tag{5.1}$$

Neither implies the other.

Take the simplest factor graph with three variables, the chain of Figure 5.1:

$$p(x_1, x_2, x_3) \;\propto\; \varphi_A(x_1, x_2)\, \varphi_B(x_2, x_3). \tag{5.2}$$

Are x_1 and x_3 independent? In general, no. Integrate out x_2:

$$p(x_1, x_3) \;\propto\; \int \varphi_A(x_1, x_2)\, \varphi_B(x_2, x_3)\, \mathrm{d}x_2, \tag{5.3}$$

which is a function of x_1 and x_3 together that does not split into a product unless the factors are very special. Information passes from x_1 through x_2 to x_3, and the feeder of Chapter 3 showed it passing: a reading two nodes away moved the source.

Are they independent *given* x_2? Yes, always. Fix x_2. The right side of equation (5.2) is then a function of x_1 alone times a function of x_3 alone, and a joint density that splits that way is a product of its marginals. That is equation (5.1), with one line of proof.

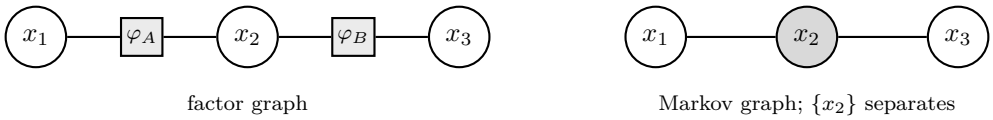


Figure 5.1: The three-variable chain. Left, as a factor graph. Right, its Markov graph: variables adjacent when they share a factor. Removing x_2 disconnects x_1 from x_3, so $x_1 \perp x_3 \mid x_2$. Nothing disconnects them otherwise, so in general $x_1 \not\perp x_3$.

So a factorization into local pieces does two opposite things at once. It creates global dependence, because every variable is reachable from every other along the factors. And it creates conditional independence, because fixing the variables on a path blocks it. The first is what makes a measurement anywhere informative everywhere. The second is what makes inference tractable, because it says which variables can be dealt with separately once a few others are known.

Principle 5.1. *Local factors create global dependence and, at the same time, conditional independence. Which variables are independent given which others is a property of the graph alone, and can be read before any number is computed.*

5.2 Reading independence off the graph

The reading uses the Markov graph of §2.5: one node per variable, an edge between two variables when some factor reads both.

Definition 5.1 (Separator)**.** A set of variables S *separates* two disjoint sets A and B in the Markov graph if every path from a variable in A to a variable in B passes through some variable in S. Equivalently, removing the nodes of S leaves A and B in different connected components.

Proposition 5.1 (Separation implies conditional independence)**.** *Let $p(\mathbf{z}) \propto \prod_f \varphi_f(\mathbf{z}_f)$ be a factorized density and let S separate A from B in its Markov graph. Then $A \perp B \mid S$.*

Proof. Every factor's scope is a set of mutually adjacent variables in the Markov graph, by construction. A set of mutually adjacent variables cannot contain one variable from A and one from B, since those two would then be adjacent and the path of length one between them would avoid S. So every factor reads variables from $A \cup S$ only, or from $B \cup S$ only (or from S only, which is both). Group the factors: $p(\mathbf{z}) \propto f(\mathbf{z}_A, \mathbf{z}_S)\, g(\mathbf{z}_B, \mathbf{z}_S)$. For fixed $\mathbf{z}_S$ this is a product of a function of $\mathbf{z}_A$ and a function of $\mathbf{z}_B$, which is conditional independence. □

The proof is the chain argument of the previous section, with "the middle variable" replaced by "a set that every path must cross". The proposition is the useful direction of a two-way result; the other direction, that every conditional independence a positive density has is visible as a separation in some factorization of it, is the Hammersley–Clifford theorem and is not needed here. Two things about it matter for physical models.

First, the proposition asks nothing of the factors except that they exist. They may be Gaussian, discrete, nonlinear, or indicator functions. So the independence structure of a model is settled by its factor graph, which is settled by its physics, before any width, prior or sensor is chosen.

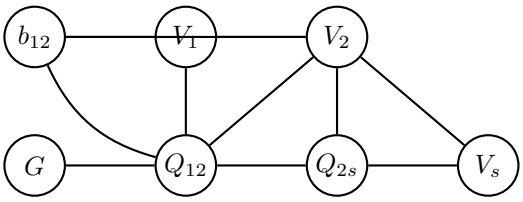


Figure 5.2: The Markov graph of the soft export feeder with its three freed inputs (Figure 3.1), before any sensor is attached. Every edge is a pair read by one factor: the balances join the flows to each other and to the injection, the closures join each flow to the voltages at its ends and, for c_{12}, to the susceptance. The path from G to V_s carries no marginal dependence, and every path from either to the other crosses the two flows.

Second, the converse direction needs the density to be positive everywhere, and a model with hard physics is not: the delta factors of Chapter 4 put zero density off the manifold. Hard physics can therefore carry independencies that no separation shows. The soft model is positive, and the correspondence between graph and independence is then as clean as the proposition makes it. This is one more reason, after the three of §4.4, to compute on the soft graph.

5.3 Explaining away, precisely

Chapter 3 found that the injection G and the substation voltage V_s, independent under the prior, were correlated at -0.985 after the reading of V_2. Proposition 5.1 says when two variables *are* independent given a set; it does not say when they are not, and it says nothing about marginal independence, which is independence given the empty set. Both are needed to make the observation exact.

Look at the Markov graph of Figure 3.1, drawn in Figure 5.2. The injection is adjacent to Q_{12} through the balance n_1; the substation voltage is adjacent to Q_{2s} and V_2 through the closure c_{2s}; and Q_{12}, Q_{2s}, V_2 are all adjacent to one another through n_2 and c_{12}. There is a path from G to V_s, so the empty set does not separate them, and the proposition is silent about their marginal independence.

Yet separate unary factors, and Chapter 4 showed that for linear physics the physics factors integrate out to a constant, leaving the product of the priors. The Markov graph draws an edge path that carries no marginal dependence.

The reason is that the physics between them is, in the hard limit, a deterministic map from inputs to solved variables. Marginalizing a deterministic map's outputs returns its inputs untouched. The undirected graph cannot express "this path carries dependence only when something on it is observed", because it has no notion of input and output. The directed language of §2.2 can: the pattern $G \to V_2 \leftarrow V_s$, two arrows meeting head to head at a common effect, is called a *collider*, and the rule for directed graphs is that a path through a collider is blocked when the collider is *not* observed and opened when it is, or when anything downstream of it is. That is the opposite of the rule for a chain, and it is the whole of explaining away: two independent causes of a common effect become dependent once the effect is seen, because having seen the effect, the more one cause explains it the less the other needs to.

For the models of this book the statement can be made without the directed formalism, because the physics has the input-output structure of Chapter 4 built in.

Proposition 5.2 (Explaining away on a physical graph)**.** *Let the physics be linear with widths $\sigma > 0$, and let two inputs u_1, u_2 carry independent priors. Then $u_1 \perp u_2$ under the prior predictive; and if a solved variable x_j that depends on both, $\partial X_j/\partial u_1 \neq 0 \neq \partial X_j/\partial u_2$, is observed, then in general $u_1 \not\perp u_2 \mid y_j$.*

Proof. For the first half, write the physics as $\mathbf{F} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with $\mathbf{A}$ square and invertible, as in Chapter 4. The prior predictive of the inputs is the joint with the solved variables integrated out, $p(\mathbf{u}) \int \exp(-\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\|^2/2\sigma^2)\,\mathrm{d}\mathbf{x}$. Substitute $\mathbf{x} = -\mathbf{A}^{-1}\mathbf{B}\mathbf{u} + \mathbf{A}^{-1}\mathbf{r}$; the Jacobian of the substitution is $1/|\det \mathbf{A}|$, a constant, and the integral becomes $|\det \mathbf{A}|^{-1} \int \exp(-\|\mathbf{r}\|^2/2\sigma^2)\,\mathrm{d}\mathbf{r}$, which does not depend on $\mathbf{u}$. So the prior predictive of the inputs is the prior, $p(u_1)\,p(u_2)$, a product, and $u_1 \perp u_2$. For the second half, condition the same joint on a reading of x_j; by equation (4.3) the posterior of the inputs is $p(u_1)\,p(u_2)\,\ell(X_j(u_1, u_2))$ in the hard limit, and for finite σ the same with ℓ smoothed as in the proof of Proposition 4.1. A product of a function of u_1, a function of u_2, and a function of both factorizes into a function of u_1 times a function of u_2 only if the third factor does, which for a Gaussian ℓ of a linear X_j with both derivatives nonzero it does not: $\ell(au_1 + bu_2)$ with $ab \neq 0$ contains the cross term $-abu_1u_2/\sigma_y^2$ in its exponent, and a joint Gaussian with a nonzero cross term is not a product. Hence $u_1 \not\perp u_2 \mid y_j$. □

The second half is read from the input-space posterior, and the cross term is the correlation of equation (3.11). In the worked example X_j was $V_s + 0.5\,G$, the likelihood was a Gaussian in that combination, and the posterior was a ridge along it. The ridge is the picture of explaining away: along the ridge, a higher injection is paid for by a lower substation voltage.

> **Principle 5.2.** *Independent inputs stay independent while nothing downstream of them is observed, and become dependent the moment a shared consequence is. The undirected graph over-connects them; the input-output structure of the physics is what says which is which.*

5.4 Ports are separators

Now return to the boundary. Chapter 1 cut a physical graph into a set S of nodes and its complement, and found that the flows across the cut sum to the net source inside, whatever the inside does (Proposition 1.1). The probabilistic version of that statement is that the cut carries everything one side needs to know about the other.

Take a physical cut through a set of edges $\mathcal{C}$. For each cut edge e with inside end i and outside end j, the closure factor c_e reads the flow F_e, both potentials ϕ_i and ϕ_j, and the edge's parameters. Every other factor reads variables from one side only, together with the cut flows: the balance at i reads F_e and the inside flows at i; the balance at j reads F_e and the outside flows at j. Figure 5.3 draws the Markov graph near one cut edge.

Remove the pair $\{F_e, \phi_j\}$. The closure's remaining scope is $\{\phi_i\}$ and the edge's parameters, all inside; the outside balance at j has lost its only inside-facing variable. No path remains from inside to outside through edge e. Do the same for every cut edge and the two sides are disconnected.

Proposition 5.3 (Ports separate)**.** *For a physical cut $\mathcal{C}$, the set of port pairs $\{(F_e, \phi_{j(e)}) : e \in \mathcal{C}\}$, one flow and one potential per cut edge, separates the inside variables from the outside variables in the Markov graph of the physics. Hence, given the ports, the inside and the outside are conditionally independent.*

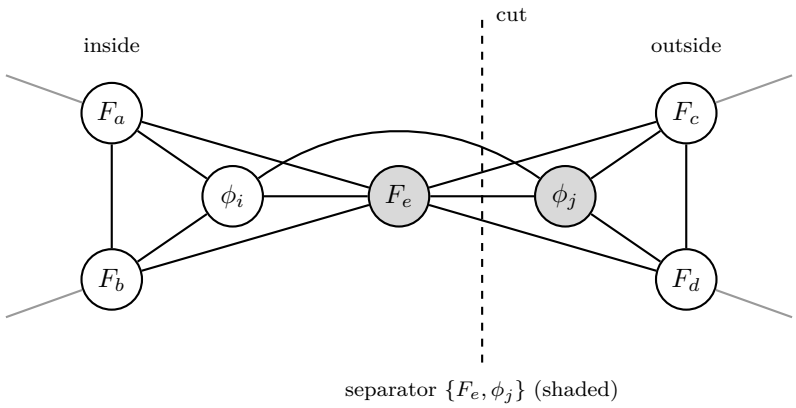


Figure 5.3: The Markov graph around one cut edge e from inside node i to outside node j. The closure c_e makes $\{F_e, \phi_i, \phi_j\}$ a clique; the balance at i joins F_e to the other inside flows F_a, F_b; the balance at j joins it to the outside flows F_c, F_d. Removing the shaded pair $\{F_e, \phi_j\}$ leaves no path from the inside cluster to the outside one. The pair is the port of the edge.

Two remarks. The separator is not unique; $\{F_e, \phi_i\}$ works as well, and so does $\{F_e, \phi_i, \phi_j\}$, and which potential is included is the choice of which side owns the cut's potential. But one flow and one potential per edge is the smallest set that works, and neither alone suffices: the potentials without the flow leave the balances at i and j joined through F_e, and the flow without a potential leaves the closure joining ϕ_i to ϕ_j. The pair that Chapter 1 called a port, on physical grounds, is the pair that separation requires on graphical grounds.

And the proposition is the probabilistic reading of Proposition 1.1. That proposition said the cut flows are exact: the inside's detail cancels, and the outside sees only what crosses. This one says the cut variables are *sufficient*: given them, the inside's detail is irrelevant to the outside, and vice versa. A region can be summarized to its neighbors by a function of its ports alone, with nothing lost. What that function is, and how it is computed, is the subject of Chapter 8.

Principle 5.3. *The ports of a physical cut are a separator of the factor graph. Given the ports, each side of the cut is conditionally independent of the other, so each side can be summarized to the other by a function of the ports alone.*

5.5 Elimination and fill

Independence says what can be computed separately. Cost says what it takes. The basic operation of exact inference is to remove one variable from the factor graph by integrating it out, and the cost of inference is the cost of doing that repeatedly in some order.

5.5.1 One elimination

Suppose x_1 appears in two factors, $\varphi_A(x_1, x_2)$ and $\varphi_C(x_1, x_3)$, and nowhere else. Integrating it out of the joint replaces the two factors by one,

$$g(x_2, x_3) \;=\; \int \varphi_A(x_1, x_2)\, \varphi_C(x_1, x_3)\, \mathrm{d}x_1, \tag{5.4}$$

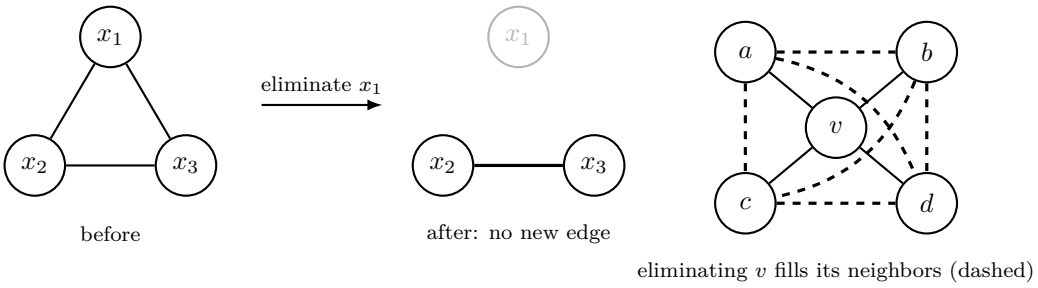


Figure 5.4: Elimination and fill. Left: eliminating x_1 from the triangle creates a factor on $\{x_2, x_3\}$, but they were already adjacent, so the graph gains no edge. Right: eliminating a variable with four neighbors that were not adjacent to each other creates six fill edges; the neighbors become a clique of four, and the factor on that clique is the largest object the computation must hold.

which reads every variable the eliminated factors read except x_1. Nothing else in the joint changes, since no other factor read x_1. The result is a factor graph on one fewer variable that has exactly the marginal of the original over the remaining ones.

The new factor is the point. Before elimination, x_2 and x_3 shared no factor; after it, they share g. In the Markov graph an edge has appeared between x_2 and x_3. Such an edge is called *fill*, and eliminating a variable always fills in every pair of its neighbors that was not already adjacent: the neighbors of the eliminated variable become a clique. Figure 5.4 shows it on the triangle and on a variable with four neighbors.

5.5.2 Order and width

Eliminate all the variables but one, in some order, and what remains is the marginal of that one. The cost is dominated by the largest factor created along the way. For discrete variables with K states, a factor on $w+1$ variables is a table of K^{w+1} numbers, and creating it costs that many operations. For Gaussian variables a factor on $w+1$ variables is a $(w+1) \times (w+1)$ block, and eliminating a variable from it costs of order w^2; the total for the whole graph is what a sparse Cholesky factorization costs, and the fill edges are the nonzeros of the factor that were not in the original matrix.

The largest factor depends on the order. Eliminate the middle of a chain first and the two ends become adjacent; eliminate from one end and no fill ever appears. For the chain the best order creates factors on two variables at most; for the four-neighbor star of Figure 5.4, eliminating the center first creates a factor on four, while eliminating the leaves first creates factors on two. The number that measures how well the best order can do is:

Definition 5.2 (Treewidth)**.** The *width* of an elimination order is one less than the size of the largest factor scope it creates. The *treewidth* of a graph is the minimum width over all elimination orders.

A tree has treewidth one: eliminate leaves inward and every factor created has two variables. A single cycle has treewidth two. A $\sqrt{n} \times \sqrt{n}$ grid has treewidth $\sqrt{n}$. A complete graph on n variables has treewidth $n-1$. Finding the optimal order is hard in general, but good orders are found in practice by heuristics that eliminate low-degree variables first, or that recursively cut the graph at small separators and eliminate the pieces before the separator.

Two bounds fix the number without a search.

Lemma 5.1 (Bounds on the width)**.** *The treewidth of a graph is at least the size of its largest clique minus one, and at most the width of any elimination order. A graph is a tree if and only if its treewidth is one.*

Proof. Consider a largest clique Q and the first of its vertices to be eliminated in any order. Every other vertex of Q is still present and adjacent to it, since elimination never removes an edge, so its front has at least $|Q| - 1$ vertices; the width of the order, and hence the minimum over orders, is at least $|Q| - 1$. The upper bound is the definition. For the last statement, a tree has an order of width one, leaves inward, and a graph with a cycle contains, after eliminating everything off the cycle, a cycle whose first eliminated vertex has two neighbors, so its width is at least two. □

The lower bound is often tight on physical graphs. The DC triangle's row-per-factor Markov graph has a clique of three, the scope of the line closure c_{23}, so its treewidth is at least two, and an order of width two exists (Exercise 5.3); every closure that reads a flow and two potentials plants such a triangle, which is why no row-per-factor physical graph is a tree.

> **Principle 5.4.** *Exact inference costs what the largest factor created by elimination costs, and that is governed by the treewidth of the factorization, not by the number of variables. Sparse is not enough; a sparse graph with large treewidth is expensive.*

5.5.3 What this means for a physical graph

Return to the two examples. The export feeder in its row-per-factor form has a six-cycle (§2.1), so its treewidth is two; in nodal form, with the flows eliminated first, it is a path and its treewidth is one. Eliminating the flows is what the nodal form of §2.4 did by hand. A flow appears in one closure and two balances, and eliminating it joins everything those three factors read: the potentials at its ends and the other flows at both its nodes. That fill among flows is temporary. Once every flow is gone, what remains joins two potentials exactly when their nodes share an edge, which is the network's own graph, and no fill survives. That is why power engineers write the nodal equations first: the reduced susceptance matrix has the sparsity of the network itself. Whether to eliminate flows first or last in a numerical factorization is a different question, answered by counting the fill, and Chapter 6 counts it.

A radial distribution feeder is a tree, so in nodal form its treewidth is one and exact inference on it is linear in its size. A meshed transmission network has treewidth larger than one, but it is a sparse, nearly planar graph, and planar graphs have separators of size $O(\sqrt{n})$ [54], so their treewidth grows no faster than the square root of the number of buses and in practice much more slowly; the measured values for the standard test networks are reported in Chapter 18. Exact inference on them is affordable. A finite-element discretization of a three-dimensional solid is harder: its separators, hence its treewidth, grow as $n^{2/3}$ [62], so sparse direct factorization becomes expensive at scale, which is one motivation for iterative and domain-decomposition solvers.

The physical cut of §5.4 is the elimination strategy the last paragraph hinted at. Cut the network at a set of ports; eliminate everything inside each region, which creates fill only within the region and among its ports; then what remains is a factor graph on the ports alone. The largest factor created inside a region is bounded by the region's own treewidth, and the factor left on the ports has scope equal to the number of ports. The cost of the whole is set by the

Table 5.1: Elimination orders on the DC triangle in row-per-factor form, five variables: the width distribution over all 120 orders, and the widths and fill of the orders that eliminate the flows first or the angles first.

orders	count	width	fill edges
all orders	120	2: 24 orders; 3: 72; 4: 24	–
flows first	12	2 to 4	0 to 3
angles first	12	3	2
F_{23} first	24	4	3 or more

region sizes and the port counts, and a cut with few ports is a cheap cut. Chapter 8 carries this out for Gaussian models, where it is the Schur complement, and Chapter 18 asks how to choose the cut.

5.6 Worked examples: counting width and fill

The claims of the last section can be counted on the book's examples, and the counts are in `ch05_separators.py`.

The triangle, every order. The DC triangle in row-per-factor form has five variables and seven Markov edges (Figure 2.6); Table 5.1 counts its orders. Of its 120 elimination orders, 24 have width two, 72 width three, and 24 width four; none has width one, because the graph contains the triangle $\theta_2\theta_3F_{23}$ and a graph with a triangle is not a tree. The twelve orders that eliminate the three flows first have widths from two to four depending on the order among the flows, and create between none and three fill edges, all among the flows themselves; what remains is the nodal form, two angles joined by the edge that c_{23} had already drawn. The twelve orders that eliminate the two angles first all have width three and create two fill edges, F_{12} to θ_3 and then F_{12} to F_{13}, joining flows that no factor read together. Flows first is the better family, but not uniformly: within it, eliminating F_{23} first, the flow on the loop, costs width four.

The six-bus network. Table 5.2 counts the same things on the network of two triangles and a tie that Chapters 8 and 15 use, drawn in Figure 5.5. Its nodal Markov graph, the five angles with the network's own edges, has treewidth two. Its row-per-factor graph, twelve variables and twenty-six edges, has treewidth at most four by the minimum-fill heuristic, which found an order of width four with three fill edges. Eliminating all seven flows first has width six and creates fourteen fill edges before the flows are gone, because a flow's elimination joins every other flow at both its buses; eliminating the angles first has width eight and twenty-six fill edges. The lesson of §5.5 stands corrected in one respect: the nodal form is the right *result*, with the network's own sparsity, but reaching it by eliminating every flow before any angle is not the cheapest *route*, and a heuristic that interleaves them is. The same table checks Proposition 5.3: removing the tie flow F_{34} and the angle θ_4 disconnects the left triangle from the right, and removing either alone, or the two angles at the tie's ends without the flow, does not.

Lemma 5.1 brackets these numbers. The six-bus graph's largest clique is the scope of a line closure, three variables, so its treewidth is at least two, and the minimum-fill order shows it is at most four; the true value lies between, and finding it exactly is the hard problem the Notes cite. On graphs of this size the gap is harmless, since a width of four is as cheap as a width of two;

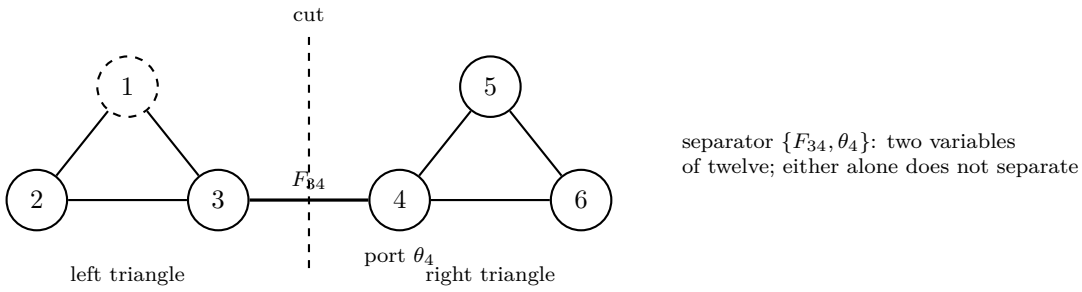


Figure 5.5: The six-bus network cut at its tie. The port pair of the cut edge, the tie flow and the angle on the far side, separates the two triangles in the row-per-factor Markov graph; the script confirms that neither variable alone does, nor the two end angles without the flow.

Table 5.2: Elimination on the six-bus network in row-per-factor form (twelve variables): width and fill under three orders, and which variable sets separate the two triangles.

order	width	fill edges
all flows first, then angles	6	14
all angles first, then flows	8	26
minimum fill (interleaved)	4	3
removed	left and right triangles	
$\{F_{34}, \theta_4\}$	separated	
$\{F_{34}\}$ or $\{\theta_4\}$ alone	connected	
$\{\theta_3, \theta_4\}$	connected	

on a network of a thousand buses the heuristics' orders are what a solver uses, and Chapter 18 reports what they find.

The ring feeder. The two-domain model of Example 1.3 has a Markov graph of twenty variables and fifty-seven edges with treewidth at most six by the heuristics and at least three by its largest clique, which has four variables. Its interesting separator is not a physical cut but a domain cut: remove the four bus voltages and the six reactive flows are disconnected from the four angles and six active flows. Given the voltages, the reactive layer and the active layer are conditionally independent, which is the probabilistic statement of what Chapter 2's Figure 2.4 showed structurally: the domains couple only through the tail voltages the bilinear closures read. An active-power measurement informs a reactive flow only by way of what it says about a voltage. No proper subset of the four will do, because every inside bus is the tail of some branch; but the minimum separator between one particular reactive flow and one particular angle is smaller, three variables, $\{Q_{ha}, Q_{hb}, V_h\}$ between Q_{sh} and θ_r, which is the physical statement that what the substation pushes into the ring reaches the far end's angle only through bus h.

The ring, around and across. A ring of twelve buses, each tied to its two neighbors, has treewidth two: eliminate the buses in order around the ring and every elimination joins the two neighbors of the eliminated bus, a front of two, creating nine fill edges in all (the last two eliminations find their neighbors already adjacent). Figure 5.6 shows the alternative. Cut the ring at two opposite buses, which are then the ports of two chains of five buses each; eliminate

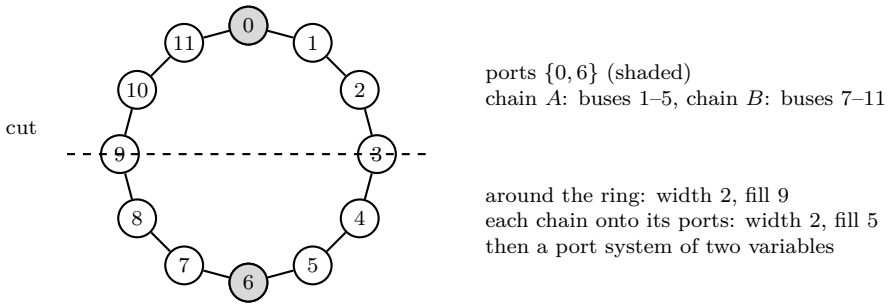


Figure 5.6: A ring of twelve buses cut at two opposite buses into two chains. Eliminating each chain's interior onto the shaded ports costs the same width as going around the ring, and leaves a two-variable port system that the chains share; the gain is independence and reuse, not width.

each chain's interior onto its ports, five fill edges and a front of two per chain; and what remains is a port system on two variables. Nothing has been gained in width, which was already two, but the two chains can be eliminated independently, each holds only its own factors, and the port system is what the chains need to exchange. This is the structure of Chapter 8's region messages in the smallest possible case, and the reason to prefer it is not the width but the reuse: a change in one chain recomputes that chain's five eliminations and the two-variable port solve, not the ring.

Grids against networks. Figure 5.7 plots the elimination width of square grids, the graph of a mesh of buses laid out $s \times s$ nodes, from 4×4 to 16×16, against the number of nodes, with $\sqrt{n}$ for comparison. The width grows as $\sqrt{n}$ and a little faster, 22 at 256 nodes, as the separator theorem for planar graphs says it must. The six-bus network and the ring feeder sit far below the curve at their sizes. A uniform mesh is the hardest two-dimensional topology there is for elimination, because every bus has four neighbors and no cut is thin; a network built by engineers has long thin regions joined at few ports, and Chapter 18 measures how far below the curve real networks of a thousand buses sit.

5.7 Why separators matter for cost

The chapter closes with the count that motivates Part III.

Take a system that divides into three regions, each containing twenty binary uncertain quantities (component availabilities, say), and suppose the physics makes the regions conditionally independent given a separator S of k variables between them. The joint has 2^{60} states, about 10^{18}. A method that must visit each state cannot be run.

Proposition 5.1 says the joint factorizes across the separator,

$$p(R_A, R_B, R_C, S) \;\propto\; f_A(R_A, S)\, f_B(R_B, S)\, f_C(R_C, S), \tag{5.5}$$

for suitable factors, where R_A denotes region A's variables. Then any marginal of interest can be found by eliminating each region into a message on the separator,

$$m_A(S) = \sum_{R_A} f_A(R_A, S), \quad m_B(S) = \sum_{R_B} f_B(R_B, S), \quad m_C(S) = \sum_{R_C} f_C(R_C, S), \tag{5.6}$$

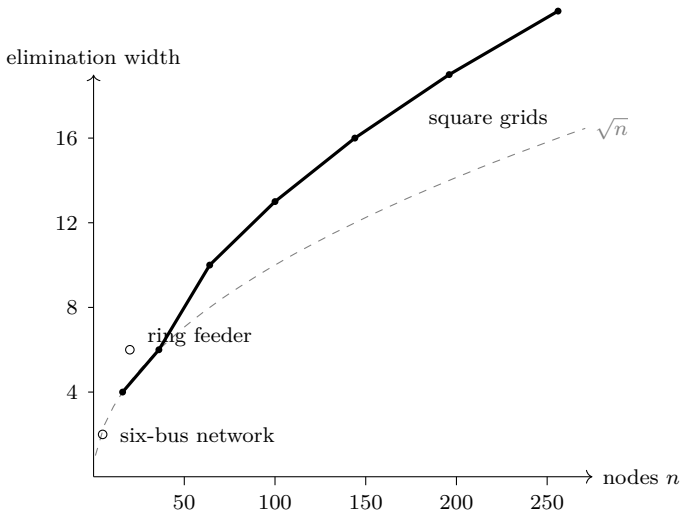


Figure 5.7: Elimination width of square grids of n nodes (filled points, minimum-fill ordering) against $\sqrt{n}$ (dashed). The six-bus network and the ring feeder are marked at their sizes. A grid's width grows with the square root of its size; the networks' widths are set by their ports.

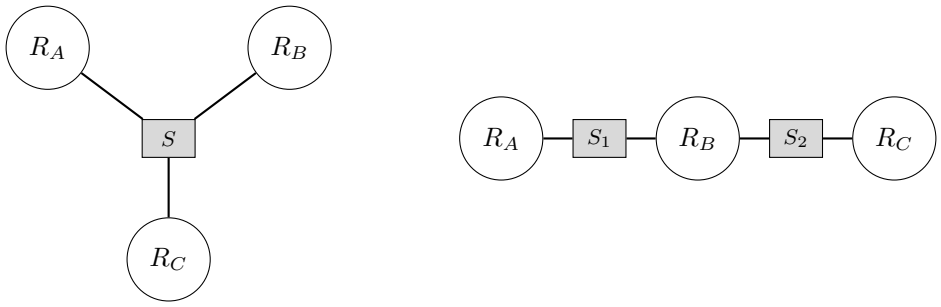
Table 5.3: Operations to marginalize three regions of n binary variables exactly: joined at one separator of k variables (star), or in a chain with two separators of k each, against enumeration of the joint.

n	k	star	chain	ratio	enumeration 2^{3n}
20	5	1.0×10^{8}	1.1×10^{9}	11	1.2×10^{18}
20	3	2.5×10^{7}	8.4×10^{7}	3.3	1.2×10^{18}
30	5	1.0×10^{11}	1.2×10^{12}	11	1.2×10^{27}

and combining them, $p(S) \propto m_A(S)\, m_B(S)\, m_C(S)$. Each message costs $2^{20} \cdot 2^k$ operations at most, about $10^6 \cdot 2^k$, and the combination costs 2^k. For a separator of five variables the whole computation is a few times 10^7 operations, against the 10^{18} of enumeration. The exponent that matters is k, the size of the separator, not 60, the number of uncertain quantities.

Table 5.3 tabulates the count for three sizes, beside the chain arrangement of Exercise 5.5 in which the middle region touches two separators and pays for both; Figure 5.8 draws the two arrangements.

This is the central fact about factorized inference, and it is worth stating what it does and does not claim. It claims that the cost of *exact* inference scales with the separator, and that a physical system with a natural cut of few ports has a small separator for free. It does not claim that a method which samples states instead of enumerating them is defeated by 2^{60}; a sampler draws a hundred thousand states from a space of 10^{18} without difficulty, and the count above is not an argument against it. The argument that does bear on samplers is about what each state costs to evaluate, and about how much of that evaluation the messages (5.6) let one reuse across related questions. Chapter 10 makes that argument with numbers.



star: each region pays 2^{n+k} chain: the middle region pays 2^{n+2k}

Figure 5.8: Three regions joined at one separator (left) and in a chain through two (right). In the chain the middle region's message must be computed jointly over both separators it touches, and its cost is exponential in their total size.

5.8 In practice

Read independence before computing. The Markov graph is free, and Proposition 5.1 answers, without a number, which quantities a proposed sensor can and cannot inform. If every path from the sensor to the quantity of interest crosses a set of variables that are already known, the sensor is wasted.

Remember which paths are open. Two independent inputs of a physical model are correlated only after something downstream of both is observed. A report that lists the sources as independent after the readings are in has misread the graph.

Cut at the ports. A physical cut's port pairs are the minimal separator; the potentials alone or the flows alone are not. A region summarized on anything less than its ports leaks information, and on anything more wastes it.

Do not trust "flows first". The nodal form is the right final graph; the cheapest route to it interleaves flows and potentials. Let a fill-reducing heuristic choose the order, and check the width it reports against the network's own sparsity.

Measure the width of your graph. A minimum-degree ordering gives it in milliseconds. A physical network of any size has a small width; a discretized continuum does not; and a model whose width is unexpectedly large usually contains a closure that reads more variables than the physics requires.

Count the separator, not the state space. For an exact computation the cost is exponential in the separator's size and linear in the number of regions. Before declaring a problem intractable at 2^{60} states, find the cut, count its ports, and count again.

5.9 What is missing

The chapter has read independence off the graph and counted the cost of elimination without saying what a factor is or how one is integrated out. For the Gaussian model both are matrices, elimination is Gaussian elimination, fill is fill, and the treewidth argument becomes a statement about sparse Cholesky. That is Chapter 6. Chapter 7 asks what elimination means when a factor is not Gaussian, and Chapter 8 organizes elimination by regions and ports, as §5.4 promised.

Notes and further reading

Markov properties of undirected graphs, the global Markov property that Proposition 5.1 states, the positivity condition and the Hammersley–Clifford theorem that supplies the converse are treated in Lauritzen [50] and Koller and Friedman [46]; the theorem's first published proof is Besag's [7]. Explaining away, colliders and the d-separation rule that §5.3 paraphrases are Pearl's [71]. Proposition 5.2 is the book's specialization of those rules to a physical graph whose input-output structure is supplied by the physics rather than by declared arrows.

Vertex elimination, fill, and the correspondence between elimination orders and chordal completions are due to Rose, Tarjan and Lueker [78]; that minimizing fill is NP-complete is Yannakakis's result [99], and that deciding treewidth is NP-complete is Arnborg, Corneil and Proskurowski's [3]. Elimination as the basis of exact inference on graphical models, and the junction tree that Chapter 8 uses, are from Lauritzen and Spiegelhalter [51]. Nested dissection, the strategy of §5.5's last paragraph, is George's [25]; the separator theorems that bound treewidth for planar and finite-element graphs are Lipton and Tarjan's [54] and Miller, Teng, Thurston and Vavasis's [62]. Davis [16] is the reference for all of this from the sparse-matrix side.

Proposition 5.3 is the book's. Its two halves are inherited: the port as a conjugate flow-potential pair is the bond-graph notion [40, 70, 93], and separation as sufficiency is the graphical-model notion. Their combination, that the physical port is the minimal probabilistic separator of the physics, is the construction on which Part III and Part V rest. The three-region count of §5.7 is the standard argument for the sum-product algorithm [48]; the caveat about samplers is the book's, and Chapter 10 is its development.

Proposition 5.3 has a concrete consequence in water networks. Han, Eguchi, Mehrotra and Venkatasubramanian [33] partition a network at its articulation junctions and estimate the biconnected components separately. An articulation junction's head is a separator of the network graph, but by itself it is not a separator of the factor graph: the balance factor at that junction contains the flows of every incident pipe, so the two sides stay coupled through the flow of the bridge until that flow joins the head in the separator, which is the head-flow port of §5.4. On EPANET's Net3 example, eliminating the 31 articulation heads alone leaves one connected interior block; eliminating the 31 heads and the 32 bridge flows splits it into 18. The elimination orderings and fill heuristics of §5.5 are used in the same way, and with the same sparse-matrix vocabulary, by Dellaert and Kaess [17], whose Bayes tree is the junction tree of an elimination order read as a directed structure.

Summary

- A factorization creates global dependence and conditional independence together. Fixing a separator blocks every path.

- Separation in the Markov graph implies conditional independence, for any factors. Hard physics breaks the converse; soft physics restores it.
- Independent inputs stay independent while nothing downstream is observed and become dependent once a shared consequence is. The undirected graph over-connects; the physics' input-output structure decides.
- The port pairs of a physical cut, one flow and one potential per cut edge, are a separator: the probabilistic form of exactness at a cut.
- Elimination fills the neighbors of the eliminated variable. Cost is set by the largest factor created, hence by treewidth, hence by the factorization and the order. Nodal form is the right result; an interleaved order is the cheap route to it: on the six-bus network, width four and three fill edges against width six and fourteen for flows-first.
- A grid's width grows as $\sqrt{n}$; a physical network's is set by its ports. The ring feeder's two domains are separated by its four bus voltages.
- Three regions of twenty binary variables joined at a five-variable separator cost a few times 10^7 operations to marginalize exactly, not 10^{18}. The exponent is the separator, not the state count.

Exercises

Exercise 5.1. Draw the Markov graph of the soft feeder of Figure 3.1 with all six variables and the sensor on V_2. Find the smallest separator between G and V_s, and the smallest between b_{12} and V_s. Verify by Proposition 5.1 that $G \perp V_s \mid \{Q_{12}, Q_{2s}\}$, and explain physically why knowing both flows makes the injection and the substation voltage irrelevant to each other.

Exercise 5.2. Prove the first half of Proposition 5.2: for $\mathbf{F} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with $\mathbf{A}$ square and invertible, show $\int \exp(-\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\|^2/2\sigma^2)\,\mathrm{d}\mathbf{x}$ does not depend on $\mathbf{u}$. Then give an example of a nonlinear physics for which it does, and say what that means for the prior predictive of the inputs.

Exercise 5.3. For the DC triangle in row-per-factor form (Figure 2.3), find an elimination order of width two and prove no order of width one exists. Then show that eliminating the three flows first produces the nodal form of Figure 2.5, and find the order among the flows that creates no fill and the one that creates the most.

Exercise 5.4. A ring of n buses, each connected to its two neighbors, has treewidth two. Give an elimination order that achieves it and the fill it creates. Now cut the ring at two opposite buses into two chains and eliminate each chain onto its ports. What is the largest factor created, and how does it compare with eliminating the ring in the naive order around it?

Exercise 5.5. Repeat the count of §5.7 for regions of n binary variables each joined at a separator of k, and for regions joined in a chain $A - S_1 - B - S_2 - C$ rather than at a single separator. Which arrangement is cheaper, and by what factor, when $k = 5$ and $n = 20$?

Solutions to selected exercises

Exercise 5.3. An order of width two: F_{12}, F_{13}, F_{23}, θ_2, θ_3. Eliminating F_{12} joins its neighbors θ_2 (from c_{12}) and F_{23} (from b_2); they were not adjacent, so one fill edge appears and the front had two variables. Eliminating F_{13} likewise joins θ_3 to F_{23}, a second fill edge, front of two.

Eliminating F_{23} now has neighbors θ_2 and θ_3, already adjacent through c_{23}: front of two, no fill. The two angles remain, joined, and the order's width is two. No order of width one exists because the Markov graph contains the triangle θ_2, θ_3, F_{23}: whichever of its three vertices is eliminated first has the other two as neighbors, a front of at least two. Among flows-first orders the script finds widths two, three and four: eliminating F_{23} first is worst, since its four neighbors θ_2, θ_3, F_{12}, F_{13} become a clique with three fill edges and a front of four; eliminating it last, after F_{12} and F_{13} have each been joined to it, is the order above, and its two fill edges are the ones that the nodal form needs and no more. The claim in §5.5 that flows-first creates no lasting fill is right about what survives, the angle-to-angle edges of the network, and silent about the temporary fill among flows, which the order among the flows controls.

Exercise 5.5. With three regions of n binary variables meeting at one separator of k, each message costs $2^n \cdot 2^k$ and the combination 2^k: about $3 \cdot 2^{n+k}$. For $n = 20$, $k = 5$ that is 1.0×10^8 operations. In the chain $A - S_1 - B - S_2 - C$, the end regions send messages on one separator each, 2^{n+k}, but the middle region's message must carry both separators, since B's factor reads S_1 and S_2 together: $2^n \cdot 2^{2k}$. The total is $2^{n+k} \cdot 2 + 2^{n+2k} + 2^{2k}$, about 1.1×10^9 for the same n and k: the chain costs eleven times more, and the factor is $2^k/3$ in general, because the middle region pays for two separators at once. Both are negligible beside enumeration's $2^{60} = 1.2 \times 10^{18}$. The lesson for a partition is that what a region pays for is the total size of the separators it touches, so a region in the middle of a chain is charged twice, and a star of regions around one separator is cheaper than a chain of the same regions.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

The direction the meters cannot see *observability as the rank of an information matrix; the least-determined direction as an eigenvector rather than a variable; a difference measured without its level*
`subsystem_power_flow/state_estimation_diagnostics/problems/the_direction_the_meters_cannot_see`

Two circuits, one signature *structural non-identifiability; a posterior symmetric under swapping two parameters; the combination that is measured against the one that is not*
`power_grid/branch_fault_localization/problems/two_circuits_one_signature`

Spans that share their weather *correlation between the elements of a minimum; dropping off-diagonal terms as a modelling choice with a direction; conservatism that is not free*
`transmission/dlr_rating_uncertainty/problems/spans_that_share_their_weather`

The junction cut *a variable that separates two parts of a model; marginalizing onto a sufficient combination; a cut of the dependency graph rather than of the network*
`foundation_power_flow/terra_vs_monte_carlo_three_line/problems/the_junction_cut`

Chapter 6

The linear-Gaussian case

Everything so far has been about structure: which variables share a factor, which are independent given which, what elimination costs. This chapter is about numbers. When every row is linear and every factor is Gaussian, the posterior is Gaussian, and a Gaussian is two objects: a matrix and a vector. The factor graph becomes the sparsity pattern of the matrix. Elimination becomes Gaussian elimination. Fill becomes fill. The most probable state is the solution of one sparse linear system, and the uncertainty of every variable is read from the same factorization that produced it. When a row is mildly nonlinear the same machinery runs a few times instead of once and returns a Gaussian approximation whose quality can be measured. The chapter ends with the export feeder in matrix form, the fill of three elimination orders counted, and a nonlinear closure checked against exact quadrature.

6.1 The Gaussian in information form

A Gaussian density on $\mathbf{z} \in \mathbb{R}^n$ is usually written by its mean and covariance, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For a factor graph the other parameterization is the natural one:

$$p(\mathbf{z}) \;\propto\; \exp\Big(-\tfrac{1}{2}\,\mathbf{z}^{\mathsf{T}}\boldsymbol{\Lambda}\,\mathbf{z} + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{z}\Big), \qquad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}, \quad \boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}. \tag{6.1}$$

$\boldsymbol{\Lambda}$ is the *precision* or *information matrix* and $\boldsymbol{\eta}$ the *information vector*. The reason to prefer them is one line: the product of two Gaussians in information form is the Gaussian whose precision is the sum of the precisions and whose information vector is the sum of the information vectors. Multiplying factors is adding matrices.

Every factor of Chapter 3 is a Gaussian in a linear function of $\mathbf{z}$. Write a generic linear row as $r(\mathbf{z}) = \mathbf{h}^{\mathsf{T}}\mathbf{z} - c$ with width σ and weight $w = 1/\sigma^2$. Its factor $\exp(-w\,r^2/2)$ expands to

$$\exp\Big(-\tfrac{1}{2}\,w\,\mathbf{z}^{\mathsf{T}}\mathbf{h}\mathbf{h}^{\mathsf{T}}\mathbf{z} + w\,c\,\mathbf{h}^{\mathsf{T}}\mathbf{z}\Big) \times \text{const}, \tag{6.2}$$

which is equation (6.1) with $\boldsymbol{\Lambda} = w\,\mathbf{h}\mathbf{h}^{\mathsf{T}}$ and $\boldsymbol{\eta} = w\,c\,\mathbf{h}$: a rank-one matrix on the row's scope and a vector supported there. A prior on z_j is the case $\mathbf{h} = \mathbf{e}_j$, $c = \mu_j$: it adds w_j to one diagonal entry and $w_j\mu_j$ to one component. A sensor on z_o is the same with $c = y$. A physics row is $\mathbf{h}$ equal to the row's gradient.

Sum over all factors. Stack every residual, physics, prior and sensor, into one vector $\mathbf{g}(\mathbf{z})$ of length m_g, let $\mathbf{J}_g = \partial\mathbf{g}/\partial\mathbf{z}$ be its Jacobian and $\boldsymbol{\Omega} = \mathrm{diag}(w)$ its weights. Then

$$\boxed{\boldsymbol{\Lambda} \;=\; \mathbf{J}_g^{\mathsf{T}}\,\boldsymbol{\Omega}\,\mathbf{J}_g} \tag{6.3}$$

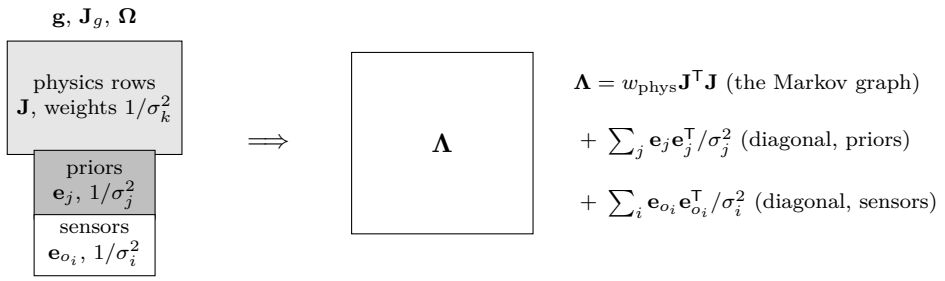


Figure 6.1: Assembling the precision. The three blocks of the stacked residual contribute three terms to $\mathbf{\Lambda}$: the physics rows give the sparse matrix whose pattern is the Markov graph, and priors and sensors add to the diagonal only. Removing the last two blocks and letting the physics weights grow recovers the deterministic solve (Proposition 6.2).

is the precision of the posterior, and the physics part of it is $w_{\text{phys}}\mathbf{J}^{\mathsf{T}}\mathbf{J}$ with $\mathbf{J}$ the Jacobian of Chapter 1, exactly the matrix that §2.5 said would appear. Figure 6.1 draws the assembly: the stacked residual has a block of physics rows, a block of priors and a block of sensors, and each block adds its own rank-m term to the same matrix.

Proposition 6.1 (Sparsity of the precision)**.** $\mathbf{\Lambda}_{ij} \neq 0$ *only if some row of* $\mathbf{g}$ *reads both* z_i *and* z_j*. The off-diagonal sparsity pattern of* $\mathbf{\Lambda}$ *is the Markov graph of the factor graph.*

The proof is that $(\mathbf{J}_g^{\mathsf{T}}\mathbf{\Omega}\mathbf{J}_g)_{ij} = \sum_k w_k (\mathbf{J}_g)_{ki}(\mathbf{J}_g)_{kj}$, and a term is nonzero only when row k has nonzeros in both columns. Two consequences follow at once. The number of nonzeros of $\mathbf{\Lambda}$ is linear in the size of the model, by the sparsity argument of §1.5. And every structural statement of Chapter 5, separation, fill, treewidth, applies to $\mathbf{\Lambda}$ verbatim, because the Markov graph is $\mathbf{\Lambda}$'s graph.

6.2 The most probable state

The negative log posterior, equation (3.8), in the stacked notation is

$$C(\mathbf{z}) \;=\; \tfrac{1}{2}\,\mathbf{g}(\mathbf{z})^{\mathsf{T}}\,\mathbf{\Omega}\,\mathbf{g}(\mathbf{z}), \tag{6.4}$$

and the most probable state, the *maximum a posteriori* or MAP estimate $\mathbf{z}^\star$, minimizes it. When $\mathbf{g}$ is linear, $\mathbf{g}(\mathbf{z}) = \mathbf{J}_g\mathbf{z} - \mathbf{c}$, the minimizer solves the normal equations

$$\mathbf{\Lambda}\,\mathbf{z}^\star \;=\; \mathbf{J}_g^{\mathsf{T}}\mathbf{\Omega}\,\mathbf{c} \;=\; \boldsymbol{\eta}, \tag{6.5}$$

one sparse symmetric positive-definite system. This is weighted least squares, and $\mathbf{z}^\star$ is also the posterior mean, since for a Gaussian the mode and the mean coincide.

When some rows are nonlinear, linearize at the current iterate, $\mathbf{g}(\mathbf{z}+\boldsymbol{\delta}) \approx \mathbf{g}(\mathbf{z}) + \mathbf{J}_g(\mathbf{z})\boldsymbol{\delta}$, and minimize the resulting quadratic:

$$\mathbf{\Lambda}(\mathbf{z})\,\boldsymbol{\delta} \;=\; -\,\mathbf{J}_g(\mathbf{z})^{\mathsf{T}}\mathbf{\Omega}\,\mathbf{g}(\mathbf{z}), \qquad \mathbf{z} \leftarrow \mathbf{z} + t\,\boldsymbol{\delta}. \tag{6.6}$$

This is the Gauss–Newton iteration. Because $\mathbf{\Lambda}$ is positive definite at a well-posed problem, $\boldsymbol{\delta}$ is a descent direction for C, and a step length $t \le 1$ chosen by backtracking until C decreases makes the iteration converge from a poor start. Near the solution $t = 1$ and convergence is quadratic when the residuals are small, which for a physical model with tight physics they are.

Proposition 6.2 (Determinism is a corner of inference)**.** *Let the model carry no prior and no sensor, let the physics be square ($m = n$) with $\mathbf{J}$ nonsingular, and let all physics weights be equal. Then the Gauss–Newton step* (6.6) *is the Newton step for $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, and any stationary point of C with $C = 0$ is a solution of the physics.*

Proof. With no other factors, $\mathbf{g} = \sqrt{w}\,\mathbf{F}$ and $\mathbf{J}_g = \sqrt{w}\,\mathbf{J}$. Equation (6.6) reads $w\mathbf{J}^\mathsf{T}\mathbf{J}\boldsymbol{\delta} = -w\mathbf{J}^\mathsf{T}\mathbf{F}$; cancel w and multiply by $\mathbf{J}^{-\mathsf{T}}$ to get $\mathbf{J}\boldsymbol{\delta} = -\mathbf{F}$, which is Newton's step. And $C = \frac{1}{2}w\|\mathbf{F}\|^2$ vanishes only where $\mathbf{F}$ does. □

The proposition is the precise form of a claim made twice already: the deterministic solve of Chapter 1 and the probabilistic estimate of Chapter 3 are one computation. There is one set of rows, one Jacobian, one iteration. Adding a sensor adds a row; adding a prior adds a row; softening the physics changes a weight. The deterministic model is the corner of the probabilistic one in which the extra rows are absent, and the two can never disagree about the physics because they share it.

6.2.1 Gauss–Newton against Newton

The Gauss–Newton matrix $\mathbf{J}_g^\mathsf{T}\boldsymbol{\Omega}\mathbf{J}_g$ is not the Hessian of C. Differentiating equation (6.4) twice gives

$$\nabla^2 C \;=\; \mathbf{J}_g^\mathsf{T}\boldsymbol{\Omega}\mathbf{J}_g \;+\; \sum_k w_k\, g_k(\mathbf{z})\, \nabla^2 g_k(\mathbf{z}), \tag{6.7}$$

the Gauss–Newton term plus a sum, over every row, of that row's residual times its own curvature. The second term vanishes when the rows are linear, since then $\nabla^2 g_k = \mathbf{0}$, and it is small when the residuals at the solution are small, whatever the curvature, which is the case for a physical model whose physics rows are tight and whose sensors are fitted to their accuracies. Dropping it costs nothing in the linear case, keeps the matrix positive definite in every case, since a sum of weighted outer products cannot be indefinite while $\nabla^2 g_k$ can, and makes the iteration a sequence of least-squares solves that never needs a second derivative. What it costs is the rate: with the second term present Newton's method converges quadratically regardless of the residuals, and without it Gauss–Newton converges quadratically only when they vanish and linearly otherwise, at a rate set by the size of the dropped term relative to the kept one. On the models of this book the residuals at the mode are the physics widths and the sensor accuracies, both small, and the examples of §6.5 and §6.6 show the quadratic tail; a model with a badly fitted sensor, one whose residual at the mode is many accuracies, is the case where the dropped term matters, and it is also the case where Chapter 12 says something is wrong with the model.

A singular $\boldsymbol{\Lambda}$ is not a numerical accident but a statement: some direction in $\mathbf{z}$ is constrained by no physics row, no prior and no sensor, and the posterior is flat along it. The remedy is a prior on a variable in that direction, and the factorization that fails names the variable; §6.6 shows it on the triangle. Chapter 12 treats this as identifiability.

6.3 Elimination is Cholesky

Chapter 5 eliminated a variable by integrating it out and found that the neighbors fill in. For a Gaussian the integral is a formula. Partition $\mathbf{z}$ into a block $\mathbf{x}$ to eliminate and a block $\mathbf{u}$ to keep, and partition $\boldsymbol{\Lambda}$ and $\boldsymbol{\eta}$ accordingly. Then

$$p(\mathbf{u}) \;\propto\; \int p(\mathbf{x},\mathbf{u})\,\mathrm{d}\mathbf{x} \;=\; \mathcal{N}\big(\,\cdot\,;\; \boldsymbol{\Lambda}_{uu} - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\Lambda}_{xu},\; \boldsymbol{\eta}_u - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\eta}_x\big) \tag{6.8}$$

in information form. The matrix $\mathbf{\Lambda}_{uu} - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu}$ is the *Schur complement* of $\mathbf{\Lambda}_{xx}$. Its sparsity is that of $\mathbf{\Lambda}_{uu}$ plus every pair of kept variables that are both adjacent to the eliminated block, which is the fill of §5.5 written as a matrix.

Proposition 6.3 (Marginalization is the Schur complement)**.** *Equation* (6.8) *holds: the marginal of a Gaussian in information form over a block of variables is Gaussian with the Schur complement as precision and* $\boldsymbol{\eta}_u - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\boldsymbol{\eta}_x$ *as information vector. The Schur complement is positive definite when* $\mathbf{\Lambda}$ *is.*

Proof. Write the exponent of the joint as $-\frac{1}{2}(\mathbf{x}^\mathsf{T}\mathbf{\Lambda}_{xx}\mathbf{x} + 2\mathbf{x}^\mathsf{T}\mathbf{\Lambda}_{xu}\mathbf{u} + \mathbf{u}^\mathsf{T}\mathbf{\Lambda}_{uu}\mathbf{u}) + \boldsymbol{\eta}_x^\mathsf{T}\mathbf{x} + \boldsymbol{\eta}_u^\mathsf{T}\mathbf{u}$. For fixed $\mathbf{u}$ it is a quadratic in $\mathbf{x}$; complete the square with $\mathbf{m} = \mathbf{\Lambda}_{xx}^{-1}(\boldsymbol{\eta}_x - \mathbf{\Lambda}_{xu}\mathbf{u})$:

$$-\tfrac{1}{2}(\mathbf{x}-\mathbf{m})^\mathsf{T}\mathbf{\Lambda}_{xx}(\mathbf{x}-\mathbf{m}) + \tfrac{1}{2}\mathbf{m}^\mathsf{T}\mathbf{\Lambda}_{xx}\mathbf{m} - \tfrac{1}{2}\mathbf{u}^\mathsf{T}\mathbf{\Lambda}_{uu}\mathbf{u} + \boldsymbol{\eta}_u^\mathsf{T}\mathbf{u}.$$

Integrating over $\mathbf{x}$ removes the first term and contributes a constant, $\sqrt{(2\pi)^{n_x}/\det\mathbf{\Lambda}_{xx}}$, that does not depend on $\mathbf{u}$. Expanding $\frac{1}{2}\mathbf{m}^\mathsf{T}\mathbf{\Lambda}_{xx}\mathbf{m} = \frac{1}{2}(\boldsymbol{\eta}_x - \mathbf{\Lambda}_{xu}\mathbf{u})^\mathsf{T}\mathbf{\Lambda}_{xx}^{-1}(\boldsymbol{\eta}_x - \mathbf{\Lambda}_{xu}\mathbf{u})$ and collecting the terms in $\mathbf{u}$ gives the quadratic $-\frac{1}{2}\mathbf{u}^\mathsf{T}(\mathbf{\Lambda}_{uu} - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu})\mathbf{u} + (\boldsymbol{\eta}_u - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\boldsymbol{\eta}_x)^\mathsf{T}\mathbf{u}$, which is the claim. For positive definiteness, take any $\mathbf{u} \neq \mathbf{0}$ and set $\mathbf{x} = -\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu}\mathbf{u}$; then $(\mathbf{x},\mathbf{u})^\mathsf{T}\mathbf{\Lambda}(\mathbf{x},\mathbf{u}) = \mathbf{u}^\mathsf{T}(\mathbf{\Lambda}_{uu} - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu})\mathbf{u}$, and the left side is positive because $\mathbf{\Lambda}$ is. □

Eliminating one variable at a time, in some order, is exactly what the Cholesky factorization $\mathbf{\Lambda} = \mathbf{L}\mathbf{L}^\mathsf{T}$ does: each step pivots on one diagonal entry and updates the trailing submatrix by the rank-one Schur complement. The nonzeros of $\mathbf{L}$ that are not nonzeros of $\mathbf{\Lambda}$ are the fill edges, the order of elimination is the order of the pivots, and the treewidth of $\mathbf{\Lambda}$'s graph is what bounds the largest dense block that $\mathbf{L}$ contains. Chapter 5's statements about cost are statements about the size of $\mathbf{L}$.

6.3.1 What the factorization costs

A column-oriented Cholesky factorization that finds c_j nonzeros below the diagonal in column j of $\mathbf{L}$ spends about $c_j(c_j+3)/2$ multiply-adds on that column: one square root, c_j divisions, and a rank-one update of the $c_j \times c_j$ trailing block that the column touches. The total is a sum over columns of a quadratic in the column counts, and the column counts are the fronts of Chapter 5's elimination. So the cost is set by the order, through the fill it creates, and Table 6.1 counts it on the book's two small models and on square grids, the graph of a mesh of buses, where the difference between orders is large enough to see.

On the 1 600-node grid the minimum-degree order factors in 2.9×10^5 multiply-adds, four and a half times fewer than the natural order and more than two thousand times fewer than a dense factorization, and its factor holds 1.7 percent of the entries a dense one would. The ratio to dense grows with the size, since the dense cost grows as n^3 and the sparse one, for a plate, as $n^{3/2}$. The physical networks of Chapter 18, with their smaller elimination widths, do better still. This is the arithmetic behind Principle 5.4.

With $\mathbf{L}$ in hand:

- the MAP is two triangular solves, $\mathbf{L}\mathbf{v} = \boldsymbol{\eta}$ then $\mathbf{L}^\mathsf{T}\mathbf{z}^\star = \mathbf{v}$;
- the marginal variance of z_k is $(\mathbf{\Lambda}^{-1})_{kk}$, and it is obtained by solving $\mathbf{\Lambda}\mathbf{x} = \mathbf{e}_k$ with the same two triangular solves and reading x_k, never by forming $\mathbf{\Lambda}^{-1}$, which is dense. This is *selected inversion*, and it costs one solve per variable asked about;

Table 6.1: Nonzeros of the Cholesky factor and multiply-adds of the factorization, for the feeder and the triangle under the orders of this chapter and for square grids of n nodes under the natural (row by row) order and a minimum-degree order, against a dense factorization of the same size.

model	order	nonzeros of $\mathbf{L}$	multiply-adds
feeder, 6 variables	inputs first	14	25
	flows first	20	50
triangle, 7 variables	minimum degree	20	42
	flows first	23	55
grid 10×10, $n = 100$	natural	1 009	5.9×10^3
	minimum degree	656	2.9×10^3
	dense	5 050	1.7×10^5
grid 40×40, $n = 1\,600$	natural	63 895	1.3×10^6
	minimum degree	21 508	2.9×10^5
	dense	1 280 800	6.8×10^8

- the marginal of any subset $\mathbf{u}$ is the Schur complement (6.8), and when $\mathbf{u}$ is the set of ports of a region, that Schur complement is the region summarized to its neighbors. Chapter 8 is about that.

Two of the three items deserve their own statements, because later chapters lean on them.

Proposition 6.4 (Selected inversion and sampling)**.** *Let $\mathbf{\Lambda} = \mathbf{L}\mathbf{L}^\mathsf{T}$. Then the marginal variance of z_k is $\|\mathbf{L}^{-1}\mathbf{e}_k\|^2$, obtained by one forward triangular solve; and if $\boldsymbol{\xi}$ is a vector of independent standard normals, then $\mathbf{z} = \boldsymbol{\mu} + \mathbf{L}^{-\mathsf{T}}\boldsymbol{\xi}$ is a sample from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Lambda}^{-1})$, obtained by one backward triangular solve.*

Proof. $(\mathbf{\Lambda}^{-1})_{kk} = \mathbf{e}_k^\mathsf{T}(\mathbf{L}\mathbf{L}^\mathsf{T})^{-1}\mathbf{e}_k = (\mathbf{L}^{-1}\mathbf{e}_k)^\mathsf{T}(\mathbf{L}^{-1}\mathbf{e}_k)$, the squared norm of the solution of $\mathbf{L}\mathbf{v} = \mathbf{e}_k$. For the sample, $\mathbf{z} - \boldsymbol{\mu} = \mathbf{L}^{-\mathsf{T}}\boldsymbol{\xi}$ is Gaussian with mean zero and covariance $\mathbf{L}^{-\mathsf{T}}\,\mathbf{I}\,\mathbf{L}^{-1} = (\mathbf{L}\mathbf{L}^\mathsf{T})^{-1} = \mathbf{\Lambda}^{-1}$. □

Both cost a triangular solve, which for a sparse factor is linear in the factor's nonzeros. So every marginal variance of a model with a million variables costs a million sparse solves, which is affordable, and a posterior sample costs one, which is how Chapter 10 draws samples from a Gaussian region without ever forming a covariance.

Proposition 6.5 (Conditioning is the other Schur complement)**.** *In information form, conditioning on the block $\mathbf{x} = \mathbf{x}_0$ leaves $\mathbf{u}$ Gaussian with precision $\mathbf{\Lambda}_{uu}$, unchanged, and information vector $\boldsymbol{\eta}_u - \mathbf{\Lambda}_{ux}\mathbf{x}_0$.*

Proof. Fix $\mathbf{x} = \mathbf{x}_0$ in the exponent of equation (6.1) and keep the terms in $\mathbf{u}$: $-\frac{1}{2}\mathbf{u}^\mathsf{T}\mathbf{\Lambda}_{uu}\mathbf{u} + (\boldsymbol{\eta}_u - \mathbf{\Lambda}_{ux}\mathbf{x}_0)^\mathsf{T}\mathbf{u}$ plus a constant. □

Marginalizing and conditioning are thus the two halves of one block elimination. Marginalizing $\mathbf{x}$ changes the precision on $\mathbf{u}$ by the Schur complement and needs a solve; conditioning on $\mathbf{x}$ leaves the precision alone and needs only a product. The asymmetry is the information form's whole convenience: the questions that Part IV asks most, "given these readings" and "given this intervention", are conditionings, and they are cheap.

Principle 6.1. *For a linear-Gaussian model the whole of inference is one sparse factorization. The mode, every marginal variance, every conditional and every region summary are solves against it.*

6.4 The Laplace posterior

When some rows are nonlinear the posterior is not Gaussian, and its exact marginals are integrals with no closed form. But the Gauss–Newton iteration (6.6) still converges to the mode $\mathbf{z}^\star$, and the matrix assembled to take the last step, $\mathbf{\Lambda}(\mathbf{z}^\star) = \mathbf{J}_g(\mathbf{z}^\star)^\mathsf{T}\mathbf{\Omega}\mathbf{J}_g(\mathbf{z}^\star)$, is the curvature of C at the mode. The *Laplace approximation* replaces the posterior by the Gaussian with that mode and that curvature,

$$p(\mathbf{z} \mid \mathbf{y}) \;\approx\; \mathcal{N}\big(\mathbf{z}^\star,\; \mathbf{\Lambda}(\mathbf{z}^\star)^{-1}\big). \tag{6.9}$$

It is exact when every row is linear, since then $\mathbf{\Lambda}$ does not depend on $\mathbf{z}$ and the posterior is the Gaussian it describes. Otherwise it is the second-order expansion of $\log p$ about its maximum, and it is accurate when the posterior is concentrated enough that the rows are nearly linear across its width. A closure that curves by a small fraction of its slope over one posterior standard deviation is nearly linear in that sense; a closure that saturates or switches regime inside the posterior's width is not, and Chapter 7 is about those.

The Laplace posterior costs nothing beyond the MAP: the matrix is already assembled. What it does not provide is a guarantee, and Chapter 12 supplies tests that probe its validity from the model itself. The worked example below supplies the simplest test, comparison with exact quadrature, in the one case where quadrature is affordable.

6.5 Worked example: the feeder in matrix form

Take the soft export feeder of Chapter 4 with both sensors: six unknowns, four physics rows at width 10^{-5} per unit, two priors, two sensors, eight rows of $\mathbf{g}$ in all. Every number here is from the script `ch06_gaussian_chain.py` in the book's `scripts` directory.

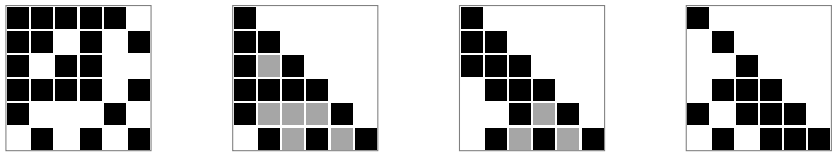
The Jacobian and the precision. Table 6.2 shows which variables each row of $\mathbf{g}$ reads. Form $\mathbf{\Lambda} = \mathbf{J}_g^\mathsf{T}\mathbf{\Omega}\mathbf{J}_g$ and mark its nonzeros:

$$\mathbf{\Lambda} \;\sim\; \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \cdot \\ \bullet & \bullet & \cdot & \bullet & \cdot & \bullet \\ \bullet & \cdot & \bullet & \bullet & \cdot & \cdot \\ \bullet & \bullet & \bullet & \bullet & \cdot & \bullet \\ \bullet & \cdot & \cdot & \cdot & \bullet & \cdot \\ \cdot & \bullet & \cdot & \bullet & \cdot & \bullet \end{pmatrix} \qquad (Q_{12},\, Q_{2s},\, V_1,\, V_2,\, G,\, V_s). \tag{6.10}$$

The off-diagonal nonzeros are the eight edges V_1V_2, V_1Q_{12}, V_2Q_{12}, V_2Q_{2s}, V_2V_s, $Q_{12}Q_{2s}$, $Q_{12}G$, $Q_{2s}V_s$, which is the Markov graph of Figure 3.1 read off Table 6.2: two variables are adjacent when some row has a mark in both columns. Proposition 6.1, by inspection.

Table 6.2: Rows of the stacked residual $\mathbf{g}$ for the feeder with two sensors, and the variables each reads. The pattern of $\mathbf{J}_g$.

Row	Residual	Q_{12}	Q_{2s}	V_1	V_2	G	V_s
n_1	$Q_{12} - G$	•				•	
n_2	$Q_{2s} - Q_{12}$	•	•				
c_{12}	$Q_{12} - b_{12}(V_1 - V_2)$	•		•	•		
c_{2s}	$Q_{2s} - b_{2s}(V_2 - V_s)$		•		•		•
π_G	$G - 0.05$					•	
π_{V_s}	$V_s - 1.000$						•
ℓ_2	$V_2 - 1.027$				•		
ℓ_1	$V_1 - 1.038$			•			



$\mathbf{\Lambda}$
14 in the lower triangle

$\mathbf{L}$, flows first
fill 6

$\mathbf{L}$, potentials first
fill 3

$\mathbf{L}$, inputs first
fill 0

Figure 6.2: The feeder's precision $\mathbf{\Lambda}$ and its Cholesky factor $\mathbf{L}$ under three orderings, as nonzero patterns. Black entries of $\mathbf{L}$ are nonzeros of the lower triangle of the permuted $\mathbf{\Lambda}$; grey entries are fill. Inputs first has none; potentials first has three; flows first, the order in which the unknowns are listed, has six.

The MAP and the marginals. One solve of equation (6.5) gives the MAP: $Q_{12} = Q_{2s} = 0.05463$, $V_1 = 1.03797$, $V_2 = 1.02704$, $G = 0.05463$, $V_s = 0.99973$. The two flows and the source agree to every digit because the physics rows are tight. The marginal standard deviations, by selected inversion against the Cholesky factor, are 0.00286, 0.00286, 0.00047, 0.00044, 0.00286, 0.00173 in the same order, and they agree with the diagonal of the dense inverse to four decimals, as they must. The Schur complement of $\mathbf{\Lambda}$ onto the two inputs (G, V_s) gives standard deviations 0.00286 and 0.00173 with correlation -0.979, which is the third column of Table 3.2. Chapter 3's hand calculation on the manifold and this chapter's sparse factorization of the soft model are, as Proposition 4.1 said, the same answer.

Fill under three orderings. The lower triangle of $\mathbf{\Lambda}$ has 14 nonzeros. Factor it in three orders and count the nonzeros of $\mathbf{L}$:

Elimination order	nonzeros of $\mathbf{L}$	fill
$Q_{12}, Q_{2s}, V_1, V_2, G, V_s$ (flows first, as listed)	20	6
$V_1, V_2, Q_{12}, Q_{2s}, G, V_s$ (potentials first)	17	3
$G, V_s, V_1, V_2, Q_{12}, Q_{2s}$ (inputs first)	14	0

Figure 6.2 draws the three factors. Inputs first creates no fill at all: G has one neighbor, V_s has two that are already adjacent, and so on down the list, each variable's neighbors already a clique when its turn comes. That is a perfect elimination order, and it exists because the Markov

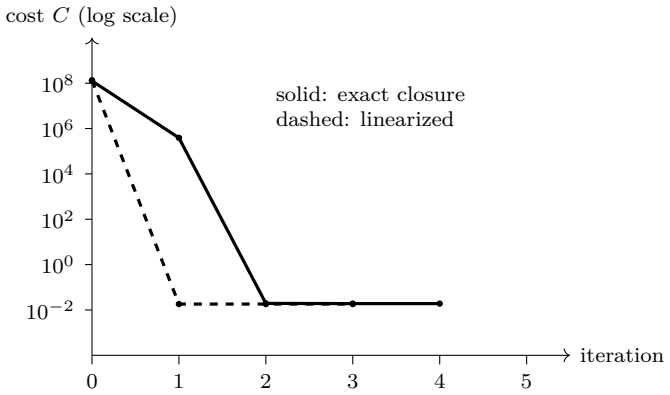


Figure 6.3: Gauss–Newton on the feeder with the exact lossless closure, from the cold start, one sensor: the cost at each iteration on a logarithmic scale (solid), against the linearized closure (dashed), which is exact in one step. The first step removes three orders of magnitude and the second seven; the convergence is quadratic once the iterate is close.

graph of the feeder is chordal. Flows first is the worst of the three: Q_{12} has four neighbors, V_1, V_2, Q_{2s}, G, of which only some pairs are adjacent, and eliminating it fills the rest. The fill is temporary in the sense of §5.5, since after every flow is gone the graph on the potentials is the network's, but $\mathbf{L}$ remembers it.

Flows first is also the order in which this book lists the unknowns, from Chapter 1 on, and there is no contradiction in that. The order in which variables are listed is for reading a model: the flows, then the potentials they connect, then the inputs. The order in which they are eliminated is for computing with it, and a sparse solver chooses that order separately, from the graph, before it does any arithmetic. On six variables none of this matters. On a million it decides whether the factorization fits in memory, and it is why a sparse solver spends effort choosing the order before it spends any on arithmetic.

A nonlinear closure. Replace the linearized branch closure into the substation by the exact lossless one, $Q_{2s} = B_{\text{ac}}\, V_2\,(V_2 - V_s)$, with B_{ac} chosen so that the two closures agree at the nominal operating point (b_{2s} divided by $V_2 = 1.025$, or 1.9512). The linear closure of Chapter 1 is this one with the tail voltage held at its nominal value; the row is now quadratic in V_2 and bilinear in V_2 and V_s. Start Gauss–Newton from a deliberately poor guess, flows of 0.08, $V_1 = 0.98$, $V_2 = 0.96$, inputs at their prior means. With one sensor the cost falls from 1.3×10^8 to 3.9×10^5 to 0.019 in three iterations, with full steps throughout, and the step length is below 10^{-7} by the third. With two sensors it is the same story in four. Convergence is quadratic once the iterate is close, as the theory said. Figure 6.3 plots the cost per iteration for the exact closure and for the linearized one, which is solved in a single step because it is linear; the second script, `ch06_triangle_gaussian.py`, produces the figures of this section and the second example that follows.

Table 6.3 compares the Laplace posterior at the mode with the exact posterior, computed by quadrature on an 801×601 grid over the two inputs in the input-space form of Chapter 4. They agree to five decimals in both means and both spreads, with one sensor and with two; the exact posterior's mean sits 0.008 of a standard deviation above its mode with one sensor and 0.030 with two. This is what a nonlinear closure looks like when it is barely nonlinear, and the reason

Table 6.3: Exact lossless closure: Laplace posterior at the Gauss–Newton mode against exact quadrature over (G, V_s). All quantities in per unit.

Sensors		G	V_s
y_2 only	Laplace	0.05376 ± 0.00583	1.00017 ± 0.00287
	exact	0.05377 ± 0.00583	1.00016 ± 0.00287
y_2, y_1	Laplace	0.05466 ± 0.00286	0.99976 ± 0.00173
	exact	0.05466 ± 0.00286	0.99976 ± 0.00173

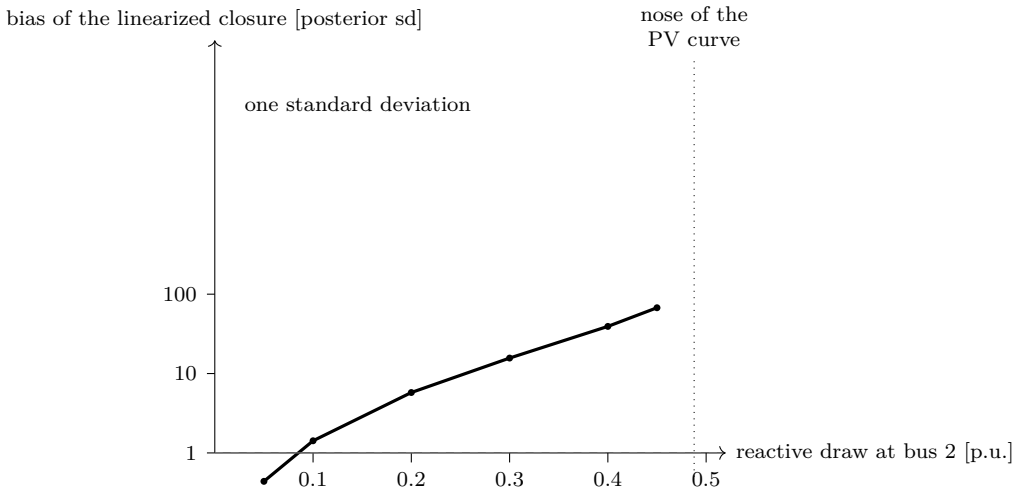


Figure 6.4: The cost of linearizing the branch closure, as the feeder is loaded down its PV curve. The vertical axis is the gap between the linearized model's estimate of the reactive draw and the exact model's, in units of the exact posterior's standard deviation, on a logarithmic scale. The dotted line is the nose of the PV curve at 0.4878 per unit, past which the exact row has no solution. Near nominal voltage the linearization costs a fraction of a standard deviation; at half the nose it costs six; approaching the nose it costs tens.

is worth stating plainly, because it is a fact about per unit rather than about Laplace. Over the posterior's width the tail voltage moves by about a thousandth, so $B_{\mathrm{ac}}V_2$ moves by a tenth of a percent: the exact closure is almost exactly its own linearization. That is why the linear closures of Chapter 1 were safe, and it is why nobody linearizes apologetically near nominal voltage.

The linearization is not always safe, and the model says where it stops being so. Turn the generator into a load and walk the same feeder down its PV curve. The exact closure has a nose: the largest reactive draw bus 2 can take from the substation is $B_{\mathrm{ac}}V_s^2/4 = 0.4878$ per unit, and past it no voltage solves the row at all. Approaching that nose, $\partial Q_{2s}/\partial V_2 = B_{\mathrm{ac}}(2V_2 - V_s)$ falls toward zero, and a closure that pretends the slope is constant is wrong by more and more. Figure 6.4 plots the bias, the gap between the linearized model's estimate of the draw and the exact model's, in units of the exact posterior's own standard deviation. At a draw of 0.05 per unit, with bus 2 at 0.974, the bias is 0.44 standard deviations and a modeler would not notice it. At 0.10 it is 1.4; at 0.20, with the bus at 0.884, it is 5.8; at 0.45, close to the nose, it is 67. The linearized model does not become noisy, it becomes confidently wrong, which is the failure mode that matters: its error bar stays the same size while its answer walks away.

Table 6.4: Rows of $\mathbf{g}$ for the DC triangle with uncertain loads and a meter on line 1–2, and the variables each reads. The pattern of $\mathbf{J}_g$.

Row	Residual	θ_2	θ_3	F_{12}	F_{13}	F_{23}	P_2	P_3
b_2	$F_{12} - F_{23} + P_2$			•		•	•	
b_3	$F_{13} + F_{23} + P_3$				•	•		•
c_{12}	$F_{12} + B\theta_2$	•		•				
c_{13}	$F_{13} + B\theta_3$		•		•			
c_{23}	$F_{23} - B(\theta_2 - \theta_3)$	•	•			•		
π_{P_2}	$P_2 + 1.5$						•	
π_{P_3}	$P_3 + 0.5$							•
ℓ_{12}	$F_{12} - 1.10$			•				

One number in the table deserves a physical reading. With one sensor the injection's spread is 0.00583 here against 0.00582 in the linear model, a difference of a fifth of a percent. The exact closure has a slightly larger slope at the operating point, $B_{\mathrm{ac}}(2V_2 - V_s) = 2.005$ against $b_{2s} = 2$, so the sensor's lever arm on the injection is a shade longer and the same reading constrains it a shade better. The Laplace posterior reports that correctly because $\mathbf{\Lambda}$ at the mode carries the local slope; had the feeder been loaded instead of exporting, the same mechanism would have reported a much shorter lever arm, and the same table would have shown it.

6.6 Second worked example: the triangle in matrix form

The DC triangle with uncertain loads and one flow meter, the model of §3.5, in the same matrix form: seven unknowns, the two angles, three flows and two loads; eight rows of $\mathbf{g}$, five physics rows at width 0.001 per unit, two load priors and the meter. Table 6.4 gives the pattern of $\mathbf{J}_g$. The precision has eleven off-diagonal nonzeros, the eleven edges of the Markov graph: the seven of Figure 2.6 plus the four that the balances draw between the loads and the flows they balance. One solve gives the MAP, $P_2 = -1.421$, $P_3 = -0.460$, $F_{12} = 1.101$, $F_{13} = 0.780$, $F_{23} = -0.320$, which is the second column of Table 3.3, and selected inversion against the Cholesky factor gives the spreads 0.136, 0.269, 0.020, 0.135, 0.134, agreeing with the dense inverse to 10^{-12}. The condition number is 3.8×10^7, the value the sweep of Chapter 4 found at this width.

Fill. The lower triangle of $\mathbf{\Lambda}$ has 18 nonzeros. Angles first gives a factor with 22, four fill entries; flows first 23, five; the loads first, then the angles, then the flows, 20, two, which is what the minimum-degree heuristic also finds. The loads are the leaves of this graph, each read by one balance and its prior, and eliminating leaves first never fills; the angles then have two or three neighbors that are nearly cliques already. Flows first is again not the cheapest route, for the reason Chapter 5 counted: a flow's neighbors are the potentials at its ends and the other flows at both its buses, and they are not adjacent to one another until the flow joins them.

Observability. Remove the load priors and keep the one meter. The precision is then singular: its smallest eigenvalue is 10^{-18} of its largest, and the eigenvector belonging to it is the direction $\Delta P_2 = -0.5$, $\Delta P_3 = 1$, with the flows on lines 1–3 and 2–3 each moving by -0.5 and F_{12} not at all. That is the shift of load from bus 2 to bus 3 that leaves the metered flow unchanged, since $F_{12} = -(2P_2 + P_3)/3$; the meter cannot see it, no prior pins it, and the posterior is flat along it.

Table 6.5: Observability of the triangle's loads: the ratio of the smallest to the largest eigenvalue of the precision, and the posterior of the loads, with and without load priors, for one and two meters.

priors and meters	eigenvalue ratio	P_2	P_3
no priors; F_{12}	8.7×10^{-19}	unobservable	unobservable
no priors; F_{12}, F_{13}	6.9×10^{-7}	-1.260 ± 0.045	-0.780 ± 0.045
no priors; F_{12}, F_{23}	1.0×10^{-6}	-1.270 ± 0.028	-0.760 ± 0.045
load priors; F_{12}	2.6×10^{-8}	-1.421 ± 0.136	-0.460 ± 0.269

The factorization fails, and the vector that makes it fail names the unobservable combination. Table 6.5 lists the cures. A second meter on line 1–3 or on line 2–3 reads a different combination of the loads and restores a condition number near 10^6, with the loads known to 0.03 to 0.05 per unit from the meters alone; the load priors restore it too, at the price of a condition number of 4×10^7 and of loads known only to 0.14 and 0.27, because a prior pins the unseen direction with its own width and no more. Observability in the power engineer's sense is the nonsingularity of this matrix with the priors removed.

Sampling from the factor. Proposition 6.4 turns the Cholesky factor into a sampler: 20 000 draws of $\boldsymbol{\mu} + \mathbf{L}^{-\mathsf{T}}\boldsymbol{\xi}$, one backward solve each, have sample spreads 0.137 and 0.270 for the loads against the exact 0.136 and 0.269, and a sample correlation of -0.976, the exact value. Figure 6.5 draws four hundred of them with the exact ellipses. The samples answer questions that the mean and covariance answer only with more algebra, and any question, including a nonlinear one: the probability that the boundary flow exceeds 2.1 per unit is 0.0570 from the samples and 0.0574 exactly, and the probability that line 2–3 carries more than half a per unit toward bus 2 is 0.089. Chapter 10 uses exactly this sampler inside a region.

A nonlinear line closure. Replace the DC closure by the angle form of the power flow, $F = B \sin(\theta_i - \theta_j)$, and load the network heavily, with $B = 2.5$ so that the angles are large enough for the sine to matter. Gauss–Newton from a flat start, angles zero and loads at their priors, reduces the cost from 1.3×10^6 to 700 to 0.04 in three iterations, full steps throughout, and converges by the fifth. The mode has $\theta_2 = -0.456$ radians, where $\sin\theta_2 = -0.440$, a difference of four percent between the DC and the AC closures at this loading. The Laplace posterior of the loads is $P_2 = -1.431 \pm 0.139$ and $P_3 = -0.464 \pm 0.267$; the exact posterior, by quadrature over the two loads with the hard sine physics solved by Newton's method at each grid point, is -1.430 ± 0.138 and -0.463 ± 0.266. The linear model at the same susceptance gives -1.421 ± 0.136 and -0.460 ± 0.269. Four percent of nonlinearity in the closure moves the loads' posterior by a hundredth of a per unit and their spreads by two percent, and the Laplace posterior tracks the exact one to the same precision.

6.7 Filters, fields, and state estimators

Three things the reader may already know are this chapter under other names.

The *Kalman filter* is the case in which the variables are the state of a system at successive times, the physics rows are the transition from each time to the next, and the sensors read each time's state. The precision $\boldsymbol{\Lambda}$ is block-tridiagonal in time. Eliminating the blocks forward in

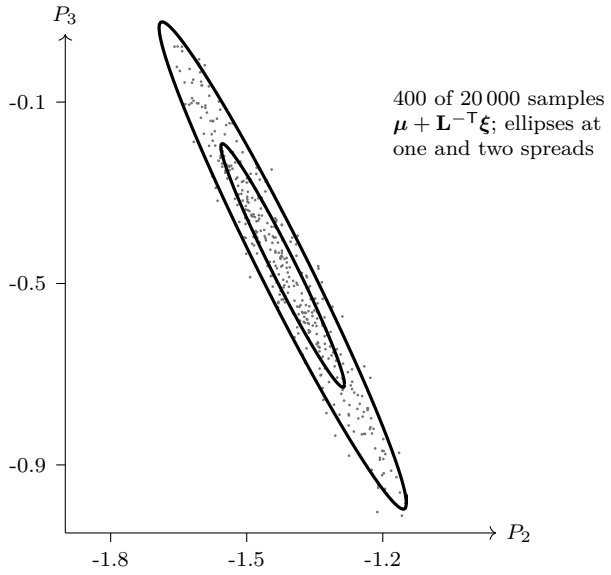


Figure 6.5: Four hundred posterior samples of the triangle's two loads, drawn from the Cholesky factor of the precision, with the exact one- and two-standard-deviation ellipses. The samples fill the ellipses in the proportions a Gaussian gives, because a triangular solve against the factor turns independent standard normals into draws with exactly the posterior covariance.

time is the filter's prediction and update; eliminating backward is the smoother. Chapter 16 unrolls the graph and says so at length.

A *Gaussian Markov random field* is a Gaussian whose precision is sparse, with the sparsity read as a graph. That is equation (6.3) with Proposition 6.1. The literature on such fields is the literature on this chapter's objects.

Power-system state estimation is this chapter with the AC power flow equations as the physics rows, the bus voltages and angles as the unknowns, meters as the sensors, and, in its classical form, no priors and hard physics. The weighted least-squares iteration it uses is equation (6.6), and its bad-data detection is the objective test that §4.5 previewed and Chapter 12 develops. The framework of this book adds soft physics, priors on the inputs, and the multi-domain graph; the linear algebra is the same.

6.8 In practice

Assemble in information form and never invert. Build $\boldsymbol{\Lambda}$ by adding one rank-one term per factor, factor it once, and answer every question by a solve against the factor: the mode, the variance of any variable by selected inversion, the summary of any region by a Schur complement. A dense inverse of a sparse precision is dense, and forming it is the commonest way to turn a fast model into a slow one.

Let the solver choose the order. A fill-reducing ordering found by a minimum-degree or nested-dissection heuristic is nearly always better than any order a modeler writes by hand, and the chapter's counts show why: on the feeder, inputs first has no fill and flows first has six; on

the triangle, the heuristic finds the least. The exception is when the order is chosen for reuse, region interiors before ports, which Chapter 8 explains.

Scale the rows. A precision assembled from rows in mixed units, per-unit power and per-unit voltage and radians, has a condition number set by the ratio of the largest weight to the smallest, and the tight physics rows dominate. Equilibrate the rows as Chapter 4 described, and check the condition number after assembly; a value near 10^{10} means the physics widths are too tight for the arithmetic.

Start Gauss–Newton from the physics. A cold start converges in the examples of this chapter, but a start from the deterministic solve at the prior means costs one solve and removes the backtracking. Watch the step length: a run of half-steps means the linearization is poor across the current iterate's neighborhood, and the Laplace posterior at the end deserves the checks of Chapter 12.

Report the Laplace posterior with its slope. The posterior spread of an input under a nonlinear closure is set by the closure's slope at the mode, as the exact-closure example showed. When the spread differs from a linear model's, the difference should be explained by that slope, and if it cannot be, the mode or the linearization is suspect.

Test Laplace where you can. On any model with two or three uncertain inputs, quadrature over the inputs in input-space form is cheap and exact. Run it once on a representative case before trusting the Laplace posterior on the full model; Figure 6.4 shows how far a linearized closure can drift once the operating point leaves the neighborhood it was linearized in.

6.9 What is missing

The chapter assumed every variable continuous and every row at worst mildly nonlinear. A discrete availability, a closure that switches regime, a bounded parameter, or a strongly curved law breaks one of the two, and Chapter 7 says which of the machinery survives each. And it took the elimination order as given; organizing it by physical regions and ports, so that each region's Schur complement is computed once and reused, is Chapter 8.

Notes and further reading

A Gaussian with a sparse precision matrix whose pattern is read as a graph is a Gaussian Markov random field, and Rue and Held [79] develop everything in §6.1 and §6.3, including the information form, the Schur complement as marginalization, and the use of a sparse Cholesky factor for sampling and for marginal variances. Selected inversion, computing chosen entries of the inverse from the factor without forming the inverse, is Erisman and Tinney's [21], and it was invented for power networks. Sparse Cholesky, fill-reducing orderings and their graph theory are in Davis [16] and Duff, Erisman and Reid [19]; the first fill-reducing ordering for network equations is Tinney and Walker's [91].

Gauss–Newton with backtracking for nonlinear least squares, its descent property, and its quadratic convergence under small residuals are in Nocedal and Wright [66]. The Laplace

approximation is classical; MacKay [57] and Bishop [9] present it in the form used here, as the Gaussian at the mode with the Hessian's curvature, and discuss when it is trusted. Proposition 6.2 is the book's statement of a fact that every implementation of weighted least squares over physics residuals relies on implicitly.

The three instances of §6.7 have their own literatures. The Kalman filter is Kalman's [39]; Särkkä and Svensson [81] give the modern treatment, including the view of filtering and smoothing as Gaussian inference on a chain, which is how Chapter 16 presents it. Power-system state estimation as weighted least squares over the power-flow equations is Schweppe's [82]; Monticelli [65] and Abur and Gómez Expósito [1] cover the Gauss–Newton iteration, sparse factorization, observability (the singular $\mathbf{\Lambda}$ of §6.2) and bad-data detection. Chavali and Nehorai [14] run the same estimation as message passing on a factor graph. Factor graphs for large sparse Gaussian and nearly-Gaussian estimation, with the same emphasis on the elimination order and the Cholesky factor, are the subject of Dellaert and Kaess [17].

Summary

- In information form, multiplying Gaussian factors is adding matrices. Every factor is a rank-one term on its scope; the posterior precision is $\mathbf{\Lambda} = \mathbf{J}_g^{\mathsf{T}} \mathbf{\Omega} \mathbf{J}_g$, and its sparsity is the Markov graph.
- The MAP solves $\mathbf{\Lambda z}^{\star} = \boldsymbol{\eta}$; for nonlinear rows, Gauss–Newton with backtracking. With no priors and no sensors the step is Newton's step for the physics: determinism is a corner of inference.
- Elimination is Cholesky; fill is fill; the Schur complement is the marginal. Marginal variances come from solves against the factor, never from a dense inverse.
- The Laplace posterior is the Gaussian at the mode with the assembled curvature: exact for linear rows, accurate when rows are nearly linear across the posterior's width.
- On the feeder: the precision pattern is the Markov graph by inspection; inputs-first elimination has zero fill, flows-first six; a the exact lossless closure converges in three Gauss–Newton steps from a cold start and its Laplace posterior matches quadrature to five decimals, because near nominal voltage the closure is almost its own linearization; loaded toward the nose of its PV curve, the linearized closure is biased by tens of standard deviations.
- On the triangle: eleven Markov edges, the MAP of Chapter 3 in one solve, selected inversion to 10^{-12}, leaves-first the cheapest order; a sine closure at heavy loading converges in five iterations and its Laplace posterior matches quadrature to a thousandth.

Exercises

Exercise 6.1. Show that the product of $\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{\Sigma}_2)$, as functions of $\mathbf{z}$, is proportional to a Gaussian with precision $\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}$ and information vector $\mathbf{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \mathbf{\Sigma}_2^{-1}\boldsymbol{\mu}_2$. Then derive the posterior of the worked example of §3.4 in one line from this rule.

Exercise 6.2. Proposition 6.3 gives the marginal in information form. Show that in covariance form the marginal of $\mathbf{u}$ is simply the block $\mathbf{\Sigma}_{uu}$ of the joint covariance, and hence that $(\mathbf{\Lambda}_{uu} - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu})^{-1} = \mathbf{\Sigma}_{uu}$: the Schur complement of the precision is the inverse of a block of the covariance. Which form is cheaper to compute when $\mathbf{u}$ is small and $\mathbf{x}$ is large and sparse?

Exercise 6.3. For the feeder's $\mathbf{\Lambda}$ in equation (6.10), carry out the inputs-first elimination symbolically, one variable at a time, and verify at each step that the eliminated variable's neighbors are already mutually adjacent. Then do the same for flows first and list the six fill entries.

Exercise 6.4. The DC triangle with uncertain loads (Figure 3.2, $a_{23} = 1$ fixed) and one flow meter is linear-Gaussian. Write its $\mathbf{g}$, its $\mathbf{J}_g$, and its $\mathbf{\Lambda}$, and find an elimination order with the least fill. Compare with the nodal form of Figure 2.5.

Exercise 6.5. Turn the generator of the worked example into a load and rerun the Laplace-versus-quadrature comparison with one sensor at draws of 0.10, 0.30 and 0.45 per unit. The Laplace spread stays accurate at all three; what does change, and why does the exact posterior stay nearly Gaussian even a hundredth of a per unit from the nose? Relate the answer to the slope of the closure and to how sharply the sensor reads it.

Solutions to selected exercises

Exercise 6.2. In covariance form the joint is $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ and the marginal of a block is obtained by dropping the other coordinates: $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \mathbf{\Sigma}_{uu})$, because a marginal of a Gaussian is Gaussian with the corresponding sub-vector and sub-block. Proposition 6.3 gives the same marginal with precision $\mathbf{\Lambda}_{uu} - \mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu}$, so that precision is $\mathbf{\Sigma}_{uu}^{-1}$: the Schur complement of the precision is the inverse of the covariance's block, which is the block-inverse identity read as a statement about marginals. When $\mathbf{u}$ is small and $\mathbf{x}$ is large and sparse, the Schur complement is the cheaper route: it needs one sparse factorization of $\mathbf{\Lambda}_{xx}$ and $|\mathbf{u}|$ solves against it, whereas $\mathbf{\Sigma}_{uu}$ by way of the full covariance needs the dense inverse of $\mathbf{\Lambda}$. For a region of a thousand interior variables and six ports that is six sparse solves against a dense million-entry inverse, and it is why Chapter 8 computes region messages as Schur complements.

Exercise 6.3. Inputs first: eliminate G, whose only neighbor is Q_{12}, trivially a clique; then V_s, whose neighbors V_2 and Q_{2s} are adjacent through c_{2a}; then V_1, neighbors V_2 and Q_{12}, adjacent through c_{12}; then V_2, neighbors Q_{12} and Q_{2s}, adjacent through b_2; then Q_{12}, neighbor Q_{2s}; then Q_{2s}. At every step the neighbors were already a clique, so no fill: a perfect elimination order, and the factor has the fourteen nonzeros of the lower triangle and no more. Flows first: eliminating Q_{12}, with neighbors V_1, V_2, Q_{2s}, G, fills V_1Q_{2s}, V_1G, V_2G and $Q_{2s}G$; eliminating Q_{2s}, now with neighbors V_1, V_2, G, V_s, fills V_1V_s and GV_s; the remaining four are then a clique and nothing more fills. Six fill entries, the number the script counts, and every one of them joins a pair that the physics never related directly: a flow's neighbors on both sides of it, tied together only because the flow between them was removed.

Exercise 6.1. As a function of $\mathbf{z}$, the logarithm of $\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{\Sigma}_i)$ is $-\frac{1}{2}\mathbf{z}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\mathbf{z} + \mathbf{z}^\mathsf{T}\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i$ plus a constant. The logarithm of the product is the sum, $-\frac{1}{2}\mathbf{z}^\mathsf{T}(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})\mathbf{z} + \mathbf{z}^\mathsf{T}(\mathbf{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \mathbf{\Sigma}_2^{-1}\boldsymbol{\mu}_2)$ plus a constant, which is the logarithm of a Gaussian with precision $\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}$ and information vector $\mathbf{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \mathbf{\Sigma}_2^{-1}\boldsymbol{\mu}_2$: in information form, multiplying Gaussians adds their parameters. For the worked example of §3.4, work in the plane of the two inputs. The prior has precision $\mathrm{diag}(1/0.02^2,\ 1/0.003^2)$ and information $(0.05/0.02^2,\ 1.000/0.003^2)$. The reading $V_2 = 1.027$ with accuracy 0.0005 reads $V_2 = V_s + G/2$, the linear function $\mathbf{h}^\mathsf{T}(G, V_s)$ with $\mathbf{h} = (\frac{1}{2}, 1)$, so it contributes precision $4 \times 10^6\,\mathbf{h}\mathbf{h}^\mathsf{T}$ and information $4 \times 10^6 \cdot 1.027\,\mathbf{h}$. In one line, the

posterior has precision $10^6\left(\begin{smallmatrix}1.0025 & 2\\ 2 & 4.1111\end{smallmatrix}\right)$ and information $10^6(2.054125,\ 4.219111)$, whose solution is $G = 0.05366 \pm 0.00582$ and $V_s = 1.00017 \pm 0.00287$, the second column of Table 3.2.

Exercise 6.4. In row form the variables are the two angles, the three flows and the two loads. The rows are the two balances, $F_{12} - F_{23} + P_2$ and $F_{13} + F_{23} + P_3$; the three closures, $F_{12} + B\theta_2$, $F_{13} + B\theta_3$ and $F_{23} - B(\theta_2 - \theta_3)$; the two load priors; and the meter on F_{12}. Each row's variables form a clique of $\mathbf{\Lambda}$'s pattern: $\{F_{12}, F_{23}, P_2\}$, $\{F_{13}, F_{23}, P_3\}$, $\{F_{12}, \theta_2\}$, $\{F_{13}, \theta_3\}$ and $\{F_{23}, \theta_2, \theta_3\}$, the priors and the meter adding only diagonal entries. Eliminate the loads first: each has two neighbours already joined by its balance row. Then eliminate F_{12}, whose neighbours θ_2 and F_{23} are joined by the closure of line 2–3, and then F_{13} likewise. What remains, θ_2, θ_3 and F_{23}, is one clique. No step creates a fill entry, so the order is perfect, and the largest clique has three variables. In the nodal form of Figure 2.5 the flows are gone. The variables are θ_2, θ_3, P_2 and P_3; the balances read $P_2 = B(2\theta_2 - \theta_3)$ and $P_3 = B(2\theta_3 - \theta_2)$, each touching its load and both angles; and the meter reads $F_{12} = -B\theta_2$, touching θ_2 alone. Eliminating the loads first is again perfect, and what remains is the pair of angles. The nodal form has four variables instead of seven and the same largest clique. The row form costs more variables for the same width, and buys a model in which every physical statement is a row that can be loosened or replaced on its own.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Letting the evidence pick the weight *the marginal likelihood as a function of a model's own trust; an interior maximum means the data identified the weight; too tight fits the wrong relation and too loose throws away a real one*
`foundation_electrical/physics_trust_calibration/problems/evidence_picks_the_weight`

Weighted least squares is the same estimate *the Gaussian posterior and weighted least squares are the same object; the normal equations as an information matrix; the prior as an extra row*
`subsystem_power_flow/state_estimation_diagnostics/problems/wls_equals_map_on_a_feeder`

What the width is made of *where a posterior width comes from; a measurement that is redundant against the physics rather than against another measurement; what a known injection buys*
`power_grid/state_estimation_ieee14/problems/what_the_width_is_made_of`

A fit that is too good *a reduced chi-square below one; a noise model that is too pessimistic; calibrating declared accuracy against observed residuals*
`power_grid/state_estimation_ieee14/problems/a_fit_that_is_too_good`

Chapter 7

Beyond Gaussian: hybrid states and nonlinearity

Chapter 6 assumed that every variable is continuous and every row is linear, or nearly so. Real systems break both assumptions, and they break them in a small number of ways: a component is in service or it is not; a device saturates or switches regime; a law curves strongly; a parameter has a bound. This chapter takes the four in turn, says what each does to the posterior, and says which parts of the machinery survive. The short answer is that the graph survives entirely, the Gaussian machinery survives conditionally on the discrete part, and what is lost is the single Gaussian: the posterior becomes a mixture, and the question of how many components it has is the question that Part III is organized around. The chapter's worked example is the smallest possible instance, a line that may be out of service, detected from one flow meter.

7.1 Four ways out of the Gaussian world

A discrete variable. The availability $a_{23} \in \{0, 1\}$ of Figure 3.2. Its prior is a probability mass, not a density; the closure that reads it is linear in the continuous variables for each fixed value of a_{23} and is not a single linear row across both values.

A piecewise closure. A compensator that regulates until it saturates, a tap changer at the end of its band, a limiter, a breaker: $y = f_b(x)$ with the regime b selected by the state itself. The closure is linear or smooth within each regime and has a kink at the boundary between regimes.

A strongly nonlinear closure. Radiation as the fourth power of temperature, AC power flow with its sines and cosines, a pump curve. The row is smooth but curves enough across the posterior's width that the Laplace posterior of §6.4 is in question.

A bounded parameter. A susceptance or an efficiency that must be positive, an availability that must lie in $[0, 1]$, a load that cannot be negative. A Gaussian prior puts mass where the parameter cannot be.

Nothing in this list changes the factor graph. A discrete variable is a circle; a piecewise closure is a square; a bound is a unary factor of a non-Gaussian shape. The scopes, the Markov graph, the separators and the treewidth are exactly as Chapters 2 and 5 described them. What changes is the arithmetic inside the squares, and therefore what the posterior looks like.

7.2 Hybrid states: the mixture over branches

Take a model with one discrete variable a and continuous variables $\mathbf{z}$. Write the joint as a sum over the values of a:

$$p(\mathbf{z}, a \mid \mathbf{y}) \;\propto\; \mathbb{P}(a)\; p(\mathbf{z} \mid a, \mathbf{y})\; Z_a, \qquad Z_a = \int p(\mathbf{y}, \mathbf{z} \mid a)\, \mathrm{d}\mathbf{z}. \tag{7.1}$$

For each fixed a the model is one of Chapter 6's: every row is linear in $\mathbf{z}$, so $p(\mathbf{z} \mid a, \mathbf{y})$ is Gaussian with a precision $\mathbf{\Lambda}_a$ and a mode $\mathbf{z}_a^\star$ obtained by one sparse solve. Call this the *branch* a. The number Z_a is the *evidence* of the branch: the probability of the readings under it, integrated over everything uncertain. For a linear-Gaussian branch it has a closed form,

$$\log Z_a \;=\; -\,C_a(\mathbf{z}_a^\star) \;-\; \tfrac{1}{2} \log\det \mathbf{\Lambda}_a \;+\; \text{const}, \tag{7.2}$$

where C_a is the branch's objective (6.4) and the constant collects the factors' normalizers, which are the same for every branch with the same rows and widths. The first term rewards a branch that fits the data and the priors closely; the second penalizes a branch whose posterior is sharply concentrated, since a sharp posterior means the branch predicted a narrow range of readings and was lucky to be right. Together they are the Gaussian form of the trade-off between fit and flexibility.

The log-determinant hides one more term, and it is the one that decides whether branches can be compared at all.

Proposition 7.1 (The evidence of a branch)**.** *Let a branch have inputs* $\mathbf{u}$ *with prior* $p(\mathbf{u})$*, solved variables* $\mathbf{x}$ *determined by* m *linear physics rows* $\mathbf{F}_a(\mathbf{x}, \mathbf{u})$ *with square Jacobian* $\mathbf{J}_{x,a}$ *with respect to* $\mathbf{x}$*, every physics width* σ*, and readings* $\mathbf{y}$ *with likelihood* $p(\mathbf{y} \mid \mathbf{x}, \mathbf{u})$*. As* $\sigma \to 0$ *the evidence* Z_a *of the branch's soft factors satisfies*

$$\frac{Z_a}{(\sqrt{2\pi}\,\sigma)^m} \;\longrightarrow\; \frac{1}{|\det \mathbf{J}_{x,a}|} \int p(\mathbf{u})\, p(\mathbf{y} \mid X_a(\mathbf{u}), \mathbf{u})\, \mathrm{d}\mathbf{u}. \tag{7.3}$$

The integral is the predictive density of the readings under the branch, and it does not depend on how the physics rows are written. Branches whose physics rows differ must therefore be weighted by $|\det \mathbf{J}_{x,a}|\, Z_a$*, or equivalently by the predictive density; weighting them by* Z_a *alone weights each by the reciprocal of its own physics determinant.*

Proof. Fix $\mathbf{u}$ and integrate the product of the physics factors and the likelihood over $\mathbf{x}$. Substitute $\mathbf{r} = \mathbf{F}_a(\mathbf{x}, \mathbf{u})$; for linear physics $\mathrm{d}\mathbf{x} = \mathrm{d}\mathbf{r}/|\det \mathbf{J}_{x,a}|$ with a constant Jacobian, and $\mathbf{x} = X_a(\mathbf{u}) + \mathbf{J}_{x,a}^{-1}\mathbf{r}$. The physics factors are $\prod_k \exp(-r_k^2/2\sigma^2)$, whose integral over $\mathbf{r}$ is $(\sqrt{2\pi}\sigma)^m$ and which, divided by that, tends to $\delta(\mathbf{r})$. So the inner integral tends to $(\sqrt{2\pi}\sigma)^m\, p(\mathbf{y} \mid X_a(\mathbf{u}), \mathbf{u})/|\det \mathbf{J}_{x,a}|$, and integrating over $\mathbf{u}$ against the prior gives the claim. For nonlinear physics $\mathbf{J}_{x,a}$ depends on $\mathbf{u}$ and stays inside the integral, which is Remark 4.1. □

The determinant is not a nuisance constant. Scaling one closure row by a factor, writing $F_{23}/B - (\theta_2 - \theta_3)$ instead of $F_{23} - B(\theta_2 - \theta_3)$, changes $|\det \mathbf{J}_x|$ and hence Z_a by that factor and changes nothing physical; the predictive density is unchanged. And a branch that changes a closure, as an outage does, changes the determinant with it. On the DC triangle the physics determinant with line 2–3 in service is three times its value with the line out, because the nodal matrices are $B\left(\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right)$ and $B\,\mathbf{I}$, with determinants $3B^2$ and B^2. The soft-row evidence

understates the in-service branch threefold, and the worked example below shows what that does to an alarm.

The posterior over the discrete variable follows by Bayes' rule,

$$\mathbb{P}(a \mid \mathbf{y}) \;\propto\; \mathbb{P}(a)\, Z_a, \tag{7.4}$$

and the posterior of any continuous variable is the *mixture*

$$p(z_j \mid \mathbf{y}) \;=\; \sum_a \mathbb{P}(a \mid \mathbf{y})\, \mathcal{N}\!\big(z_j;\; (\mathbf{z}_a^\star)_j,\; (\mathbf{\Lambda}_a^{-1})_{jj}\big), \tag{7.5}$$

one Gaussian per branch, weighted by the branch's posterior probability. Its mean and variance are those of a mixture: the mean is the weighted mean of the branch means, and the variance is the weighted variance within branches plus the variance of the branch means between them.

Principle 7.1. *A discrete variable splits the model into branches. Within a branch the model is linear-Gaussian and everything in Chapter 6 applies; across branches the posterior is a mixture weighted by prior times evidence. The evidence is one number per branch, and it comes from the same factorization that gives the branch's mode.*

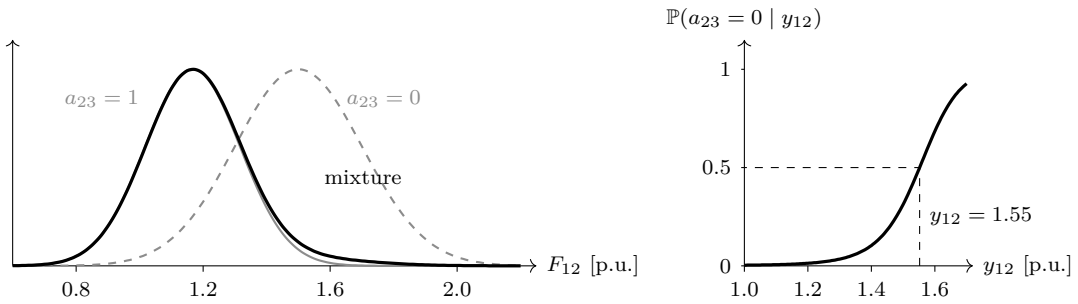
7.2.1 Worked example: a line that may be out

Return to the DC triangle of Figure 3.2. The loads are uncertain, $P_2 \sim \mathcal{N}(-1.5, 0.2^2)$ and $P_3 \sim \mathcal{N}(-0.5, 0.2^2)$ per unit, negative for load; the line susceptances are $B = 10$; the physics rows have width 0.01; the line from bus 2 to bus 3 is out of service with prior probability 0.05; and one meter reads the flow on line 1–2 with accuracy 0.02. Every number is from the script `ch07_hybrid_triangle.py` in the book's `scripts` directory.

Before the reading. With the line in service the metered flow is $F_{12} = -(2P_2 + P_3)/3$, which under the load priors is 1.167 ± 0.149; the loop shares bus 2's load with line 1–3 by way of line 2–3, which carries -0.333 ± 0.095. With the line out, bus 2 is served by line 1–2 alone, $F_{12} = -P_2 = 1.500 \pm 0.200$, and line 2–3 carries nothing. Panel (a) of Figure 7.1 draws the prior predictive of the metered flow: the two branch Gaussians, weighted 0.95 and 0.05, and their sum. The outage branch is a small shoulder on the right of the main peak.

After the reading. Table 7.1 sweeps the reading from 1.10 to 1.60. Two things happen at once. Within each branch the posterior is Gaussian and the meter pins the flow to ± 0.02; the branch's estimate of P_2 is then whatever load explains that flow under the branch's physics, -1.50 under the outage branch when the reading is 1.50, and -1.89 under the in-service branch, which needs a heavier load at bus 2 to push that much flow down line 1–2 when part of it is being shared around the loop. Across branches the posterior probability of the outage rises from 0.006 at a reading of 1.10, where the reading sits on the in-service peak, to 0.31 at 1.50 and 0.68 at 1.60, where it sits on the outage branch's. Panel (b) draws the whole curve; it crosses one half at a reading of 1.55.

The weights come from Proposition 7.1, and the script checks them against the predictive density of the reading computed directly. Had the soft-row evidence been used without the determinant, the in-service branch would have been understated threefold: the outage probability at 1.50 would have been reported as 0.58 instead of 0.31, at 1.60 as 0.87 instead of 0.68, and the crossover placed at 1.48 instead of 1.55. An alarm set at one half would have fired on readings between 1.48 and 1.55, which the correct posterior considers more likely normal than not.



(a) prior predictive of the metered flow

(b) outage probability after the reading

Figure 7.1: Line-outage detection from one meter. (a) The prior predictive of the metered flow F_{12}: the in-service branch (solid grey) and the outage branch (dashed grey), each drawn at unit peak so that both are visible, and the weighted mixture (black), scaled to its own peak; in the mixture the outage branch is a shoulder of height about a twentieth of the main peak. (b) The posterior probability that the line is out, as a function of the meter reading, with the branch evidence corrected by the physics determinant (Proposition 7.1); it crosses one half at $y_{12} = 1.55$.

Table 7.1: The triangle with an uncertain line and one meter: the outage posterior and the per-branch and mixture posteriors of the load at bus 2 as the meter reading is swept.

y_{12}	$\mathbb{P}(a_{23} = 0 \mid y)$	$P_2 \mid a_{23} = 1$	$P_2 \mid a_{23} = 0$	$\mathbb{E}[P_2 \mid y]$	$\mathrm{sd}[P_2 \mid y]$
1.10	0.006	-1.42 ± 0.09	-1.11 ± 0.02	-1.42	0.10
1.17	0.010	-1.50 ± 0.09	-1.18 ± 0.02	-1.50	0.10
1.30	0.034	-1.66 ± 0.09	-1.30 ± 0.02	-1.64	0.11
1.40	0.104	-1.77 ± 0.09	-1.40 ± 0.02	-1.74	0.14
1.50	0.312	-1.89 ± 0.09	-1.50 ± 0.02	-1.77	0.20
1.60	0.685	-2.01 ± 0.09	-1.60 ± 0.02	-1.73	0.20

What a single Gaussian would miss. Look at the last two columns at a reading of 1.50. The mixture's mean load is -1.77 with a spread of 0.20. No branch believes that. The in-service branch says -1.89 ± 0.09 and the outage branch says -1.50 ± 0.02; the mixture mean lies between two modes, in a region of low posterior density, and its spread is mostly the distance between the modes, not uncertainty within either. An operator told "the load is 1.67 give or take 0.2" would be told something that is not true under any hypothesis. The correct report is "either the line is out and the load is 1.50, or the line is in and the load is 1.89, with odds of about 69 to 31 that it is in". This is what a Laplace posterior, which fits one Gaussian at one mode, cannot say, and it is the first of the ways it fails.

7.2.2 Many discrete variables

With k discrete variables there are 2^k branches, each a sparse Gaussian solve. For k up to ten or so that is affordable and exact. Beyond that, three things help, and Part III develops each: most branches have negligible posterior weight and can be pruned by their evidence; discrete variables that are separated in the graph can be summed out independently, which is the region argument of §5.7 with discrete separators; and the sum over branches can be sampled rather than enumerated. Chapter 15 applies all three to the question of which k lines, out of hundreds,

are most likely to be out together.

7.3 Piecewise closures and regimes

A piecewise closure is a discrete variable in disguise. Write the regime as b and the closure as $r_b(\mathbf{z}) = 0$; then the model is a mixture over b exactly as in §7.2, with one difference. The regime is not free: it is determined by the state. A compensator is in its saturated regime when the demanded order exceeds one, and in its regulating regime otherwise. So a branch b is *consistent* only if its own posterior mode puts the state in regime b, and inconsistent branches have no weight.

The deterministic version of this is the *active-set loop*: guess the regimes, solve, check each closure's regime against the solved state, switch the inconsistent ones, and repeat until no regime changes. It usually converges in a few passes, and when it does the solution is on a single branch. The probabilistic version does the same with posteriors: for each candidate assignment of regimes, solve the branch, check consistency at its mode, and keep the branches whose modes are consistent, weighted by evidence. Usually one survives. Two survive when the posterior straddles a regime boundary, and the posterior is then a mixture whose components sit on either side of a kink.

The kink matters for one thing more than any other. A gradient computed across it is wrong: the sensitivity of the state to an input jumps at the boundary, and a finite difference that spans the boundary averages two slopes that belong to different physics. Chapter 14 returns to this when it computes sensitivities, and it is the reason that chapter insists on differentiating within the active regime.

When the regimes join continuously, as a compensator's do at saturation, the weight of a regime has an exact form that the active-set picture only approximates.

Proposition 7.2 (The weight of a regime)**.** *Let the physics be piecewise linear in the inputs* $\mathbf{u}$*: on region* $\mathcal{R}_b$ *of input space the solved variables are* $X_b(\mathbf{u})$*, the map of the linear branch* b*. Then the posterior probability of regime* b *is*

$$\mathbb{P}(b \mid \mathbf{y}) \;\propto\; p_b(\mathbf{y})\, \mathbb{P}_b(\mathbf{u} \in \mathcal{R}_b \mid \mathbf{y}), \tag{7.6}$$

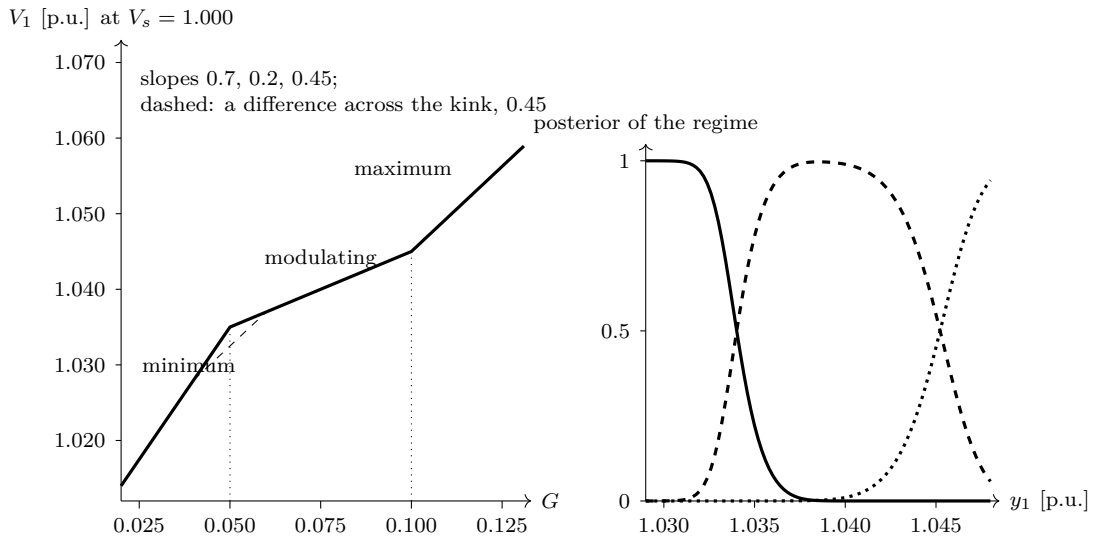
where $p_b(\mathbf{y})$ *is branch* b*'s predictive density of the readings and* $\mathbb{P}_b(\,\cdot \mid \mathbf{y})$ *is branch* b*'s Gaussian posterior over the inputs. The posterior of any quantity within regime* b *is branch* b*'s posterior restricted to* $\mathcal{R}_b$*.*

Proof. The posterior over the inputs is $p(\mathbf{u})\, p(\mathbf{y} \mid X(\mathbf{u}))/p(\mathbf{y})$, and on $\mathcal{R}_b$ the true map X equals X_b. So the mass of regime b is $\int_{\mathcal{R}_b} p(\mathbf{u})\, p(\mathbf{y} \mid X_b(\mathbf{u}))\, \mathrm{d}\mathbf{u}$ divided by $p(\mathbf{y})$. Branch b's posterior is $p(\mathbf{u})\, p(\mathbf{y} \mid X_b(\mathbf{u}))/p_b(\mathbf{y})$ over all of input space, so the integral is $p_b(\mathbf{y})$ times that posterior's mass on $\mathcal{R}_b$. □

The proposition replaces the active-set test "is the branch's mode in its region" by a mass, "how much of the branch's posterior is in its region". The two agree when the posterior is far from every boundary and disagree near one, and the worked example below shows by how much.

7.3.1 Worked example: a tap changer

Put an on-load tap changer on the export feeder. The transformer at the substation moves its tap to hold bus 2 at 1.025 per unit, and the susceptance the feeder sees through it ranges from 2 to 4



(a) the map from injection to generator-bus voltage (b) the regime after one reading

Figure 7.2: The tap changer. (a) The generator-bus voltage as a function of the injection at a substation voltage of 1.000: continuous, with kinks where the tap reaches each end of its band, and a dashed chord showing a central difference taken across the first kink. (b) The posterior probability of each regime as a function of the generator-bus reading.

per unit as the tap runs from one end of its band to the other. The branch closure then has three regimes. With the tap at its minimum, $Q_{2s} = 2(V_2 - V_s)$ and bus 2 is below the set point. With the tap modulating, $V_2 = 1.025$ and the susceptance is whatever the controller needs, between the limits. With the tap at its maximum, $Q_{2s} = 4(V_2 - V_s)$ and bus 2 is above the set point. The first boundary is where the minimum tap just holds bus 2 at 1.025, $G = 2(1.025 - V_s)$; the second is where the maximum tap does, $G = 4(1.025 - V_s)$. The priors of Chapter 3, $G \sim \mathcal{N}(0.05, 0.02^2)$ and $V_s \sim \mathcal{N}(1.000, 0.003^2)$, put the first boundary through the prior mean: the prior gives 0.500 to the tap at minimum, 0.484 to modulating, and 0.016 to the tap at maximum, which needs an injection of 0.10 per unit. A voltage sensor on the generator bus reads V_1 with accuracy 0.0005 per unit. Every number is from `ch07_regimes.py`.

The map and its kinks. In the three regimes the generator-bus voltage is $V_s + 0.7G$, $1.025 + 0.2G$ and $V_s + 0.45G$. Panel (a) of Figure 7.2 draws the map at $V_s = 1.000$: continuous, with slopes 0.7, 0.2 and 0.45 per unit of voltage per per unit of injection, and kinks at 0.05 and 0.10. A central difference across the first kink gives 0.45 for steps of 0.02, 0.005 and 0.0005. It converges as the step shrinks, but to the average of the two slopes, not to either, and nothing in the number says that it belongs to no physics.

The weights. Table 7.2 sweeps the reading and computes the regime probabilities three ways: exactly, by quadrature over the two inputs with the piecewise physics; by Proposition 7.2, with each branch's in-region mass estimated from 400 000 samples of its Gaussian posterior; and by the active-set shortcut, which keeps a branch only if its mode lies in its region. The proposition matches quadrature to 0.003 in every regime probability and to a hundred-thousandth of a per unit in the injection's mean and spread. The shortcut is right away from the boundaries and

Table 7.2: The tap changer after one reading of the generator-bus voltage: regime probabilities and the posterior of the injection, exact by quadrature and by Proposition 7.2, and the modulating probability under the active-set shortcut. All quantities in per unit.

	exact				proposition		shortcut
y_1	min	mod	max	G	mod	G	mod
1.0300	1.000	0.000	0.000	0.0432 ± 0.0043	0.000	0.0432 ± 0.0042	0.000
1.0325	0.919	0.081	0.000	0.0462 ± 0.0044	0.080	0.0462 ± 0.0044	0.000
1.0340	0.505	0.495	0.000	0.0482 ± 0.0035	0.498	0.0482 ± 0.0035	0.000
1.0350	0.221	0.779	0.000	0.0512 ± 0.0027	0.780	0.0512 ± 0.0027	0.000
1.0375	0.008	0.991	0.001	0.0624 ± 0.0025	0.991	0.0624 ± 0.0025	1.000
1.0400	0.000	0.990	0.010	0.0746 ± 0.0025	0.990	0.0746 ± 0.0025	1.000
1.0425	0.000	0.905	0.095	0.0865 ± 0.0028	0.906	0.0865 ± 0.0028	1.000
1.0450	0.000	0.548	0.452	0.0960 ± 0.0050	0.550	0.0960 ± 0.0050	0.642
1.0470	0.000	0.157	0.843	0.0999 ± 0.0066	0.159	0.0999 ± 0.0066	0.000

wrong near them. At a reading of 1.0350, where the exact posterior gives 0.22 to the tap at minimum and 0.78 to modulating, the shortcut reports the tap at minimum with certainty: the modulating branch's mode sits on the edge of its region and is discarded, though most of that branch's posterior lies inside. At 1.0450 it reports 0.64 modulating where the exact value is 0.55.

What the regime does to the sensor. In the modulating regime the generator-bus voltage is $1.025 + 0.2G$ whatever the substation does: the controller has absorbed the substation voltage, and the reading says nothing about it. At readings of 1.0375 and 1.0425 the posterior of the substation voltage is its prior, 1.0000 ± 0.0029 against 1.000 ± 0.003. The injection is known to 0.0025 there, which is the sensor's accuracy divided by the slope 0.2, while with the tap at its minimum the substation voltage's spread adds to the reading's and the injection is known only to 0.0043. What a sensor tells is a property of the regime, and the regime is a property of the state.

7.4 Strong nonlinearity: when the Laplace posterior holds and when it does not

Chapter 6 showed that the exact lossless closure left the Laplace posterior accurate to five decimals near nominal voltage, where the closure is barely curved. Curve it. Turn the generator into a load and walk the feeder down its PV curve, so that the closure's slope $\partial Q_{2s}/\partial V_2 = B_{\mathrm{ac}}(2V_2 - V_s)$ falls toward zero at the nose. Table 7.3 repeats the one-sensor comparison of Table 6.3 at draws of 0.05, 0.20 and 0.45 per unit, with the nose at 0.4878.

The result is not what a reader expecting nonlinearity to break the approximation would predict. At a draw of 0.45, a hundredth of a per unit from the nose, with bus 2 down at 0.639 and the closure's slope fallen from 1.85 to 0.54, the Laplace posterior still agrees with quadrature to four decimals and the exact posterior's skewness is under a thousandth. The reason is physical, and it is the mirror image of the one the thermal literature usually gives. A collapsing closure moves V_2 *more* per unit of draw: the slope $\partial V_2/\partial D$ grows from 0.54 at a draw of 0.05 to 1.84 at 0.45. The sensor's lever arm on the draw lengthens, the posterior narrows from ± 0.0056 to ± 0.0037, and a posterior that narrow sees almost none of the curvature that produced it. The nonlinearity that might have distorted the posterior has instead bought the information that

Table 7.3: The feeder loaded down its PV curve, with one sensor on V_2: Laplace posterior against exact quadrature as the closure's slope collapses. Skewness of the exact posterior of the draw D in the last column. All quantities in per unit.

D	V_2	Laplace D	exact D	exact V_s	skew
0.05	0.9737	0.0500 ± 0.0056	0.0500 ± 0.0056	1.0000 ± 0.0029	-0.000
0.20	0.8841	0.2000 ± 0.0051	0.2000 ± 0.0051	1.0000 ± 0.0029	-0.000
0.45	0.6392	0.4500 ± 0.0037	0.4500 ± 0.0037	1.0000 ± 0.0030	-0.000

pins it down. Curvature matters in proportion to how far the posterior extends across it, and here too the extension and the curvature moved in opposite directions.

So strong nonlinearity by itself is a weaker enemy of the Laplace posterior than folklore suggests, at least for closures of this kind. The approximation fails in three recognizable situations, and a modeler should look for these rather than for strongly curved closures.

Multiple modes. The outage example of §7.2.1: two branches, two modes, and a single Gaussian at either one misreports both the mean and the spread. Discrete structure is the usual cause; a closure that is symmetric in a variable, so that a sensor cannot distinguish a value from its negative, is another. Chapter 15's failure sets and the phase-angle ambiguities of AC power flow are the cases that matter in practice.

A mode at a bound. When the posterior mode sits on a constraint, §7.5 below, the curvature at the mode describes a Gaussian half of which lies where the variable cannot go.

Curvature that survives concentration. When the data are informative enough to concentrate the posterior and the closure still curves within that concentrated width. Periodic closures with wide priors on angles are the standard example; so is a posterior that straddles a regime boundary. Here the mode is right and the curvature is not, and the marginal computed from $\mathbf{\Lambda}^{-1}$ can be wrong by a factor.

The tests that detect these from the model itself, without exact quadrature, are in Chapter 12: a comparison of the Laplace posterior's predicted spread of each residual with the actual residuals, and a probe that evaluates the objective at a few points along each posterior axis and checks that it is quadratic. The remedies, in increasing cost, are to reparameterize so that the closure is nearer to linear in the new variable, to integrate exactly in the one or two dimensions that are nonlinear and use Laplace for the rest, and to sample. Chapter 9 takes them up.

7.5 Bounded parameters

A Gaussian prior on a susceptance $b \sim \mathcal{N}(5, 1^2)$ puts a probability of about 3×10^{-7} on $b < 0$, which is harmless. A prior $\mathcal{N}(1, 1^2)$ on a converter efficiency puts 0.16 on $\eta < 0$ and 0.5 on $\eta > 1$, which is not. Two remedies are standard.

Truncation. Multiply the Gaussian prior by the indicator of the admissible interval. The factor graph gains nothing; the prior's unary factor changes shape. The posterior mode is found by the same Gauss–Newton iteration with the step clipped at the bound, and if the unconstrained mode lies inside the interval nothing else changes. If the mode lies on the bound, the Laplace posterior is a Gaussian centered on the bound, half of it inadmissible; the honest report is the truncated Gaussian, whose mean is inside the interval and whose spread is smaller than $\mathbf{\Lambda}^{-1}$ says.

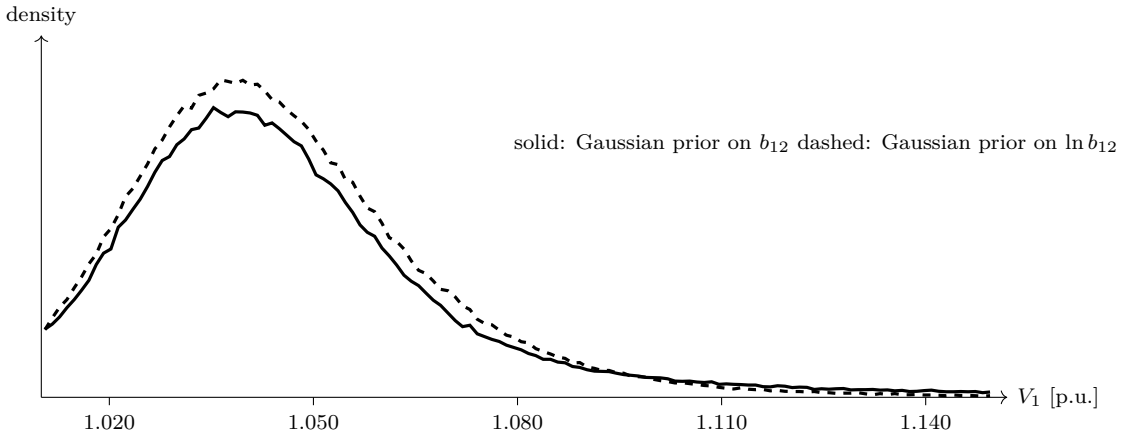


Figure 7.3: The prior predictive density of the generator-bus voltage under a Gaussian prior on the susceptance (solid) and a Gaussian prior on its logarithm (dashed), both with median 3 per unit. The solid curve's heavy right tail comes from susceptances near zero; its 6.6 percent of draws with negative susceptance, all with the generator bus below bus 2, lie off the left of the figure.

Reparameterization. Write the positive susceptance as $b = e^{\beta}$ and put the prior on β; write the efficiency as $\eta = 1/(1 + e^{-\lambda})$ and put the prior on λ. The closure that read b now reads e^{β}, which is one more nonlinearity in a row that is already treated by Gauss–Newton, and the bound has disappeared. This is usually the better choice: it keeps the machinery unchanged, and a Gaussian on β is often a more honest prior for a quantity known to within a factor than a Gaussian on b. Its cost is that β's posterior, mapped back to b, is skewed, and the Laplace posterior in β must be transformed rather than read off.

The two remedies differ where the data are thin, and the feeder shows where. Give the susceptance a Gaussian prior $\mathcal{N}(3, 2^2)$, a line susceptance known only to lie somewhere between one and five, or a Gaussian prior on $\ln b_{12}$ with the same median and a spread of 0.6. With the two voltage sensors of Chapter 3 the posterior of b_{12} is 4.82 ± 0.61 under the first and 4.81 ± 0.63 under the second: the data decide, and the bound never bites. Before the data the two priors differ sharply. Draw the prior predictive of the generator-bus voltage, $V_1 = V_2 + G/b_{12}$, by sampling the three inputs. Under the Gaussian prior 6.6 percent of the draws have $b_{12} \le 0$, and in 7.2 percent of all draws reactive power flows uphill, from bus 2 into the generator bus; the fifth percentile of V_1 is 0.987, below the substation itself, and 3.7 percent of the draws put the generator bus above 1.14 per unit because b_{12} is near zero. Under the log-normal prior almost no draw is unphysical, and the fifth to ninety-fifth percentiles are 1.014 to 1.085. Figure 7.3 draws both. A prior predictive is exactly what a Monte Carlo study of uncertain inputs computes, and a Gaussian prior on a positive parameter hands that study samples that are not physics.

7.6 What survives

It is worth listing what the four departures leave standing, because it is most of the book.

The factor graph, its Markov graph, its separators and its treewidth are untouched. Proposition 5.1 holds for any factors, and Proposition 5.3 is about the physics rows' scopes, which the departures do not change. Elimination and fill are the same graph operations; what changes is

that eliminating a discrete variable is a sum and eliminating a continuous one is an integral, and a factor that has both kinds of variable in its scope must be represented as a table of Gaussians, one per discrete configuration. Such *conditional Gaussian* factors have their own algebra, and junction-tree inference with them is exact when the discrete variables are eliminated after the continuous ones in each clique; the rule and its limits are in the notes.

The Gaussian machinery survives per branch. Every branch is a sparse solve, every branch's evidence is its objective plus a log-determinant from the same factorization, and every branch's marginals are selected inversions. The Laplace posterior survives within a branch under the conditions of §7.4.

What does not survive is the single Gaussian. The posterior is a mixture over branches and over regimes, and the honest summary of a mixture is its components and their weights, not its mean and variance. Where the number of branches is small the mixture is enumerated exactly; where it is large, Part III is about how to avoid enumerating it.

7.7 In practice

Compare branches by their predictive densities. When the branches differ in their physics, as an outage or a regime does, weight each by the predictive density of the readings, or by its soft-row evidence times its physics determinant. Never compare soft-row evidences alone; Proposition 7.1 says what that does, and the outage example shows it raising an alarm on a normal reading.

Report the components. A mixture's mean and variance are a summary that no branch believes when the weights are comparable. Report each branch's estimate and its weight, and collapse to one Gaussian only when one weight is near one.

Weight regimes by mass, not by mode. For a piecewise closure, Proposition 7.2 weights a regime by the part of its branch's posterior that lies in the regime's region. Keeping a branch because its mode is in its region is exact far from the boundaries and wrong near them, which is where the operator's questions usually are.

Differentiate within the regime. A finite difference across a kink converges to the average of two slopes. Take derivatives of the active regime's closure at the operating point, and if the operating point is near a boundary, report both slopes.

Parameterize positive quantities by their logarithms. A Gaussian prior on a susceptance, an efficiency or a load is harmless once the data are in and wrong before they are, and the prior predictive is what a Monte Carlo study and a planning study compute. Put the prior on the logarithm, or the logit for a quantity in an interval.

Test Laplace where it can fail. Multiple modes, a mode on a bound, and curvature that survives concentration are the three failures; a strongly curved closure is not one of them. Look for the three, and use the probes of Chapter 12 to check.

7.8 What is missing

Part II is complete. The model is a factor graph whose squares are soft physics, priors, sensors and failure factors; its posterior is a mixture of sparse Gaussians; its structure decides what is independent of what and what elimination costs. What has not been said is how to organize the computation when the graph is large: how to eliminate by regions and reuse the result (Chapter 8), what to do when exact elimination is too expensive (Chapter 9), how the factorized computation compares with sampling when both are costed in physics solves (Chapter 10), and how to fold in evidence as it arrives (Chapter 11). That is Part III.

Notes and further reading

Graphical models with both discrete and continuous variables, in which the continuous ones are Gaussian conditional on the discrete ones, are the conditional Gaussian models of Lauritzen and Wermuth [52]; exact propagation of probabilities, means and variances on a junction tree for such models, with the constraint on the elimination order mentioned in §7.6, is Lauritzen's [49]. The evidence of a branch as the integrated likelihood, its closed form for Gaussian models, and its use to compare hypotheses are treated by Kass and Raftery [42] under the name of Bayes factors, and by MacKay [57], whose account of the trade-off between fit and flexibility in equation (7.2) is the one followed here. That the soft-row evidences of branches with different physics must be corrected by the physics determinant, Proposition 7.1, is the coarea factor of Remark 4.1 appearing in model comparison; it is stated here because the soft formulation makes the uncorrected comparison easy to compute and wrong.

The accuracy of the Laplace approximation to posterior moments, with the error terms that explain why a nearly-Gaussian posterior is approximated well, is Tierney and Kadane's [90]; Bishop [9] gives the textbook account. The observation of §7.4, that a more strongly curved physical closure can make the posterior more Gaussian rather than less by changing the sensor's lever arm, is the book's, and the loading sweep is its evidence. evidence.

Line-outage detection from measurements is a standard problem in power-system operation, usually posed as a hypothesis test over a list of candidate outages; the treatment in §7.2.1 as a posterior over a discrete variable in the same graph as the state is the framework's, and Chapter 15 scales it up. The active-set loop of §7.3 is the standard treatment of piecewise constitutive laws in circuit and network simulators. Reparameterization of positive and bounded parameters by logarithms and logits is the common practice of Bayesian modeling [57].

Summary

- Four departures from the Gaussian world: a discrete variable, a piecewise closure, a strongly nonlinear closure, a bounded parameter. None changes the factor graph.
- A discrete variable splits the model into linear-Gaussian branches. Each has a mode, a precision and an evidence from one sparse factorization; the discrete posterior is prior times evidence; the continuous posterior is a mixture.
- Branches whose physics differ are compared by their predictive densities, which is the soft-row evidence times the physics determinant. On the triangle the uncorrected evidence understates the in-service branch threefold and would raise an outage alarm too early.

- One flow meter on the triangle detects a line outage: the outage probability rises from 0.006 to 0.68 across a reading sweep, and at the crossover the load posterior is bimodal, which no single Gaussian can report.
- A piecewise closure is a discrete variable determined by the state. A regime's weight is its branch's predictive density times the branch posterior's mass in the regime's region, which matches quadrature on the tap changer to 0.003; keeping branches whose modes are in their regions fails near the boundaries. A finite difference across a kink converges to the average of two slopes.
- In the tap changer's modulating regime the sensor says nothing about the substation voltage: what a reading tells depends on the regime.
- Loading the feeder to within a hundredth of the nose of its PV curve did not break the Laplace posterior, because the collapsing closure lengthened the sensor's lever arm and narrowed the posterior faster than it curved the map. Laplace fails at multiple modes, at a mode on a bound, and when curvature survives concentration.
- Bounded parameters: truncate, or reparameterize by a logarithm or a logit. Reparameterization is usually better: with data both give the same posterior, but a Gaussian prior on a positive susceptance sends 7.2 percent of prior-predictive draws uphill.

Exercises

Exercise 7.1. Derive equation (7.2) by completing the square in the branch's objective and integrating the Gaussian, and identify the constant. Then show that for two branches with the same rows and widths but different row coefficients the constant cancels and the log-determinant does not: it contains $\log|\det \mathbf{J}_{x,a}|$, which Proposition 7.1 says must be added back.

Exercise 7.2. In the outage example, move the meter to line 2–3 instead of line 1–2. Compute the prior predictive of its reading under both branches and explain why this meter detects the outage from a single reading with near certainty, whatever the loads. Then move it to line 1–3 and explain why it is exactly as informative as the meter on line 1–2.

Exercise 7.3. Add a second uncertain line to the triangle, line 1–3, so that there are four branches. Compute the prior probability of each, the predictive density of each at a reading $y_{12} = 1.50$, and the posterior. Which two branches can this meter not tell apart, and what does their posterior ratio equal?

Exercise 7.4. A shunt compensator delivers $Q = Q_{\max}\, u$ for an order $u \in [0, 1]$ and $Q = Q_{\max}$ for $u \geq 1$. Write the two regimes as rows, state the consistency condition for each, and construct a prior on u and a sensor on Q for which both regimes are consistent at their modes. Draw the resulting posterior of u and mark the kink.

Exercise 7.5. Repeat the loading sweep of Table 7.3 with the sensor accuracy coarsened from 0.0005 to 0.005 per unit, so that the posterior on D is no longer concentrated. Find the draw at which the Laplace standard deviation first errs by more than ten percent, and relate the answer to the third failure mode of §7.4.

Solutions to selected exercises

Exercise 7.1. For a fixed branch every factor is Gaussian in $\mathbf{z}$ with weight w_k, so the integrand of Z_a is $\prod_k (w_k/2\pi)^{1/2} \exp(-C_a(\mathbf{z}))$, with C_a the branch's quadratic objective. A quadratic

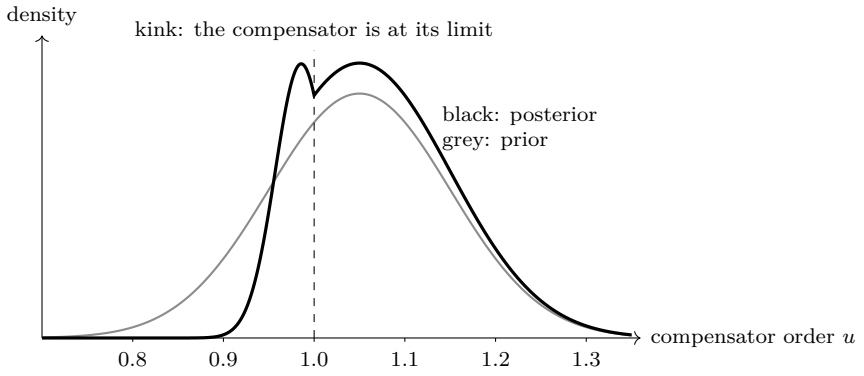


Figure 7.4: The posterior of the compensator order after a reading of 0.098 per unit with accuracy 0.003 (black), against the prior (grey). Below $u = 1$ the compensator regulates and the reading pins the order; above it the compensator is at its limit and the reading says nothing about the order. The posterior is continuous with a kink at $u = 1$, a notch between the two regimes' modes.

equals its value at its minimum plus half its Hessian form about the minimum: $C_a(\mathbf{z}) = C_a(\mathbf{z}_a^\star) + \frac{1}{2}(\mathbf{z} - \mathbf{z}_a^\star)^\mathsf{T}\mathbf{\Lambda}_a(\mathbf{z} - \mathbf{z}_a^\star)$. The Gaussian integral over d dimensions is $(2\pi)^{d/2}(\det \mathbf{\Lambda}_a)^{-1/2}$, so

$$\log Z_a = -C_a(\mathbf{z}_a^\star) - \tfrac{1}{2}\log\det\mathbf{\Lambda}_a + \tfrac{d}{2}\log 2\pi + \tfrac{1}{2}\sum_k \log\frac{w_k}{2\pi},$$

which is equation (7.2), the constant being the last two terms. Two branches with the same rows and widths share d and every w_k, so the constant cancels. The log-determinant does not. With m physics rows of width σ, $\mathbf{\Lambda}_a$ is dominated by $\sigma^{-2}\mathbf{J}_{g,a}^\mathsf{T}\mathbf{J}_{g,a}$ on the solved variables. As $\sigma \to 0$, $\log\det\mathbf{\Lambda}_a$ is $-2m\log\sigma + 2\log|\det\mathbf{J}_{x,a}|$, plus a term from the inputs' posterior, which the proof of Proposition 7.1 identifies with the predictive density. The first piece is common to the branches; the second is not whenever the branches' physics coefficients differ. So $\log Z_a$ carries $-\log|\det\mathbf{J}_{x,a}|$, which is exactly what the proposition adds back.

Exercise 7.4. The two regimes are two rows for the same closure, each with its consistency condition: regulating, $Q - Q_{\max}u = 0$ with $u \le 1$, and at its limit, $Q - Q_{\max} = 0$ with $u \ge 1$. The script `ch07_compensator.py` takes $Q_{\max} = 0.10$ per unit, a prior $u \sim \mathcal{N}(1.05, 0.1^2)$ for a compensator usually driven slightly past its limit, and a meter reading $Q = 0.098$ with accuracy 0.003. Each regime is a linear-Gaussian branch. The regulating branch reads u through the meter: its posterior is 0.986 ± 0.029, inside its region, and the predictive density of the reading under it is 30.5. The at-limit branch does not read u at all, since the output no longer depends on it. Its posterior is the prior, with mode 1.05, inside its region, and its predictive density is 106.5. Both regimes are consistent at their modes. By Proposition 7.2 each weight is the predictive density times the mass the branch's posterior puts in its own region, 0.690 for regulating and 0.692 for at-limit, so the probabilities are 0.222 and 0.778; direct integration of the exact posterior gives the same to four digits. Figure 7.4 draws that posterior. It is continuous, since both rows give $Q = Q_{\max}$ at $u = 1$, but its logarithm's slope jumps there from -17.2 just below to $+5.0$ just above. The kink is a notch between two local modes, one per regime, and a single Gaussian placed at either mode would miss the other regime entirely.

Exercise 7.5. The script `ch07_loading.py` computes the exact posterior of (D, V_s) by quadrature and the Laplace posterior from the curvature at the input-space mode. With the sensor coarsened to 0.005 per unit, the Laplace standard deviation of D is within 0.2 percent of the exact one at every draw from 0.05 to 0.45: at the heaviest, 0.00450 against 0.00451. No draw in the range makes the spread err by ten percent. What does move is the mean of D, 0.00017 low at the heaviest draw, a twenty-sixth of its spread, and the exact posterior's skewness, which grows from -0.018 to -0.037. The exercise's premise, that loosening the sensor would make the marginal spreads fail, is wrong here, and the reason is instructive. The sensor confines the posterior to a band about $V_2(D, V_s) = y$, and the third failure mode of §7.4 asks whether that band is curved enough to defeat a Gaussian. It is not. Evaluated two standard deviations along its widest axis, the objective is 1.98 and 2.03 against the quadratic prediction of 2.00 — about one percent out, at a draw a hundredth of a per unit from the nose. The map from the inputs to V_2 is steep there, but along the band that the reading pins it is nearly straight, and steepness is not curvature. In this domain the third failure mode has to be looked for somewhere other than in a single strongly loaded branch; Chapter 12's probes are how to look, and the check costs two objective evaluations.

Exercise 7.2. With the line in service the flow on line 2–3 is $F_{23} = (P_2 - P_3)/3$, so under the load priors its prediction is -0.333 ± 0.096 with the meter's accuracy added; with the line out it is exactly zero, 0.000 ± 0.020. The two predictions are separated by more than three of the wider spread, and a reading near either is decisive. A reading of zero gives $\mathbb{P}(\text{out}) = 0.990$ despite the prior of 0.05, and a reading of -0.167, halfway between the two, gives less than 10^{-4}: the out-of-service branch says the reading must be zero to within 0.02, so anything eight of its spreads away rules it out. The meter reads the quantity the outage changes most, the flow on the line itself. The meter on line 1–3 reads $F_{13} = -(P_2 + 2P_3)/3$, predicted 0.833 ± 0.150 in service and $-P_3 = 0.500 \pm 0.201$ out: the same separation and the same spreads as the meter on line 1–2, mirrored. At its out-branch prediction it gives $\mathbb{P}(\text{out}) = 0.315$, exactly what the line 1–2 meter gives at its own. It is as informative because the outage moves bus 2's load onto line 1–2 and bus 3's onto line 1–3, by amounts that the loads' symmetric priors make equal.

Exercise 7.3. Under each branch F_{12} is a linear combination of the loads: $-(2P_2 + P_3)/3$ with both lines in, $-P_2$ with line 2–3 out, $-(P_2 + P_3)$ with line 1–3 out, since bus 3 is then fed through bus 2, and $-P_2$ with both out, since bus 3 is islanded and its load shed. The predictive densities at 1.50 are 0.228, 1.985, 0.297 and 1.985; the priors are 0.9025, 0.0475, 0.0475 and 0.0025; the posterior is 0.644, 0.296, 0.044 and 0.016. The meter cannot tell "line 2–3 out" from "both out": in both, bus 2 is fed by line 1–2 alone and $F_{12} = -P_2$. Their predictive densities are identical, so their posterior ratio is their prior ratio, 19 to 1, whatever the reading. The second outage is invisible to this meter because it happens on the far side of the first; a meter on line 1–3 would see it at once. Pruning by weight alone would discard "both out" at a threshold of 0.02, and that is safe only because its consequence, bus 3 dark, is one an operator would see by other means.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

What a voltage-controlled bus actually fixes *what a bus kind declares and what it leaves free; the same physical device described by different boundary conditions; two conserved quantities with different conventions on the same node*
`subsystem_power_flow/ieee14_ac_pf/problems/what_a_pv_bus_fixes`

A reactive limit flips a bus *a regime change inside a model; the same physics with the boundary conditions swapped; a limit that binds turns a held quantity loose*
`subsystem_power_flow/ieee14_ac_pf/problems/a_q_limit_flips_a_bus`

Two ways a gradient expires *curvature against a kink; a derivative's reach in a smooth model and in a piecewise-linear one; which direction an extrapolation error runs*
`foundation_power_flow/load_uncertainty_queries_ac/problems/two_ways_a_gradient_expires`

Part III

Inference

Chapter 8

Exact inference: elimination, junction trees, Schur complements

Part II built the model and said what its posterior is. Part III is about computing it, and this chapter is about computing it exactly. The tools are the ones Chapter 5 introduced, elimination and separators, and Chapter 6 made concrete, the Schur complement. What is new is organization. Elimination can be arranged as messages passed along a tree, which gives every marginal from two sweeps; a loopy graph can be made into a tree of clusters; and, most usefully for a physical system, elimination can be organized by regions and ports, so that each region is summarized to its neighbors once and the summary is reused for every question that does not change the region. The chapter ends with two regions of a small network cut at one tie line, each region eliminated onto the port pair of the cut, and the exactness and the reuse checked to machine precision.

8.1 What exact means

A query is exact when its answer is the marginal, the mode, or the conditional of the posterior, computed without approximation beyond floating-point arithmetic. For the Gaussian branches of Chapter 7 that means the mode from a sparse solve, the marginal variances from selected inversion, and the marginal over any subset from a Schur complement. For the discrete part it means summing over every branch, or over every branch that the graph does not let one sum separately. Both are elimination: integrate or sum out the variables one does not want, in some order, and what remains is the marginal of the ones one does. This chapter takes the arithmetic of elimination as settled and asks only how to organize it.

8.2 Messages on a tree

Take a factor graph that is a tree and pick any variable as the root. Every other node has a unique path to the root, so every node has a parent and some children. Elimination from the leaves inward is then a local operation at each node, and the local objects it passes are called *messages*.

A variable x sends to a neighboring factor f the product of the messages it has received from its other neighboring factors:

$$m_{x \to f}(x) \;=\; \prod_{g \neq f} m_{g \to x}(x). \tag{8.1}$$

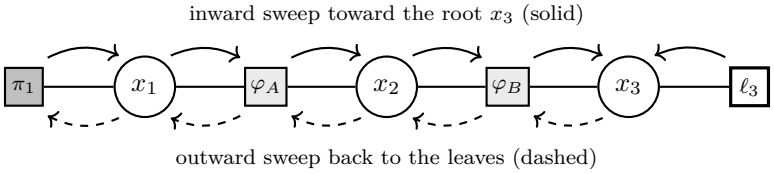


Figure 8.1: Sum-product on the three-variable chain with a prior on x_1 and a sensor on x_3. The inward sweep carries one message along each edge toward the root; the outward sweep carries one back. After both, every variable has received a message from every neighbor, and its marginal is their product (Proposition 8.1). Twelve messages, one per edge in each direction, give all three marginals.

A factor f with scope $\{x, y_1, \dots, y_k\}$ sends to x its own function times the messages from its other variables, integrated over those variables:

$$m_{f\to x}(x) \;=\; \int \varphi_f(x, y_1, \dots, y_k) \prod_i m_{y_i\to f}(y_i)\, \mathrm{d}y_1 \cdots \mathrm{d}y_k. \tag{8.2}$$

Equation (8.2) is one elimination step: it removes $y_1, \dots, y_k$ and leaves a function of x. A leaf factor, a prior or a sensor, has no other variables and sends itself. Run the messages from the leaves to the root, and the root's marginal is the product of what arrives. Run them back from the root to the leaves, and every variable's marginal is the product of what it has received from all sides:

$$p(x) \;\propto\; \prod_{f\ni x} m_{f\to x}(x). \tag{8.3}$$

Two sweeps, every marginal, no approximation. On a tree this is exact, and it is called the *sum-product* algorithm.

Proposition 8.1 (Sum-product is exact on a tree)**.** *If the factor graph is a tree, the messages* (8.1) *and* (8.2)*, computed from the leaves inward and then back outward, give every variable's marginal by equation* (8.3)*, exactly.*

Proof. Fix a variable x. Removing x from a tree splits it into one subtree per neighboring factor f, and because the graph is a tree no factor in one subtree reads a variable of another. So the joint is $\prod_{f\ni x} \Phi_f(x, V_f)$, where Φ_f is the product of the factors in f's subtree and V_f its variables other than x, and the marginal is $p(x) \propto \prod_{f\ni x} \int \Phi_f \,\mathrm{d}V_f$. It remains to show that $\int \Phi_f \,\mathrm{d}V_f = m_{f\to x}(x)$, which is an induction on the depth of the subtree. If f is a leaf, $\Phi_f = \varphi_f(x)$ and the message is φ_f itself. Otherwise f's subtree is f together with, for each of its other variables y_i, the subtrees hanging from y_i through its other factors g; these share no variables, so $\int \Phi_f \,\mathrm{d}V_f = \int \varphi_f(x, \mathbf{y}) \prod_i \big[\prod_{g\ni y_i,\, g\neq f} \int \Phi_g \,\mathrm{d}V_g\big] \mathrm{d}\mathbf{y}$. By the induction hypothesis each inner integral is $m_{g\to y_i}$, their product is $m_{y_i\to f}$ by equation (8.1), and the outer integral is equation (8.2). The inward sweep computes these messages for the root; the outward sweep computes, for every other variable, the messages from the side of the tree that the inward sweep did not visit. □

Figure 8.1 draws the two sweeps on the chain. The count is worth noticing: two messages per edge give every marginal, where computing each marginal separately by elimination would repeat most of the work once per variable. That saving is what the organization buys, and it is the same saving that selected inversion buys against a factorization in Chapter 6.

For Gaussian factors every message is a Gaussian in one variable, hence two numbers in information form, a precision λ and an information η. Equation (8.1) adds them. Equation (8.2) is a Schur complement: form the factor's precision on its scope, add the incoming precisions to the diagonal entries of $y_1, \dots, y_k$, and eliminate those variables. The whole computation is a sequence of small dense eliminations, one per factor, in an order fixed by the tree.

The chain of Figure 5.1 is the smallest instance. Root it at x_3. The message from φ_A to x_2 integrates x_1 out of φ_A times any prior on x_1; x_2 passes it to φ_B; φ_B integrates x_2 out and delivers a function of x_3. That is the reading of Chapter 3's worked example as a computation: the sensor's message enters at V_2, passes through the closures and balances, and arrives at G three factors away, attenuated by each.

8.3 Loops: the junction tree

Every factor graph in this book has loops, and on a loopy graph the messages (8.1)–(8.2) do not terminate and do not give exact marginals. Chapter 9 says what happens if one runs them anyway. The exact route is to make a tree.

Group the variables into overlapping *clusters* such that every factor's scope lies within some cluster, and arrange the clusters in a tree such that any variable shared by two clusters is contained in every cluster on the path between them. The second condition is the *running intersection property*, and a cluster tree with it is a *junction tree*. The intersection of two adjacent clusters is their separator. Assign each factor to a cluster containing its scope, and run the sum-product messages between clusters instead of between variables and factors: a cluster sends to its neighbor the product of its factors and its other incoming messages, integrated over everything not in the separator. Two sweeps give the marginal of every cluster, hence of every variable, exactly.

A junction tree is an elimination order in disguise, and the correspondence is the one Chapter 5 set up. Take any elimination order; each eliminated variable together with its neighbors at the time of elimination is a clique of the filled graph; the maximal cliques, arranged by the order, form a junction tree; and the largest cluster has one more variable than the order's width. The DC triangle in nodal form is a single cluster of two variables. The feeder in row form, with its six-cycle, has a junction tree whose largest cluster is three variables, and the inputs-first order of §6.5, which created no fill, is the order that builds it. For Gaussian factors, the junction tree's messages are the block Schur complements that a sparse Cholesky factorization performs in that order; the elimination tree of the factorization is the junction tree, and Chapter 6's statement that inference is one factorization is this section's statement with the clusters made explicit.

Figure 8.2 draws the junction tree of the DC triangle in row-per-factor form, built from the order F_{12}, F_{13}, F_{23}, θ_2, θ_3 that Chapter 5 found to have width two. Eliminating F_{12} forms the clique $\{F_{12}, \theta_2, F_{23}\}$; eliminating F_{13} forms $\{F_{13}, \theta_3, F_{23}\}$; eliminating F_{23} forms $\{F_{23}, \theta_2, \theta_3\}$; the rest are contained in these. Three clusters of three variables, joined through the third by two separators of two, and each of the five rows lies in a cluster that contains its scope: the balance b_2 and the closure c_{12} in the first, b_3 and c_{13} in the second, c_{23} in the third. The variable F_{23} is in all three clusters and on every path between them, which is the running intersection property.

Principle 8.1. *Exact inference on a loopy graph is exact inference on a tree of clusters. The clusters are the cliques of an elimination order, the largest cluster costs what treewidth says it costs, and for Gaussian models the cluster messages are the Schur complements of a sparse factorization.*

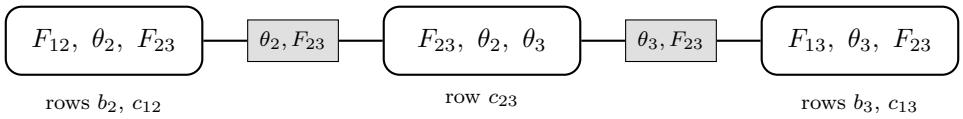


Figure 8.2: A junction tree for the DC triangle in row-per-factor form. Clusters (rounded) are the cliques of the elimination order $F_{12}, F_{13}, F_{23}, \theta_2, \theta_3$; separators (shaded) are their intersections. Every row is assigned to a cluster containing its scope, and the loop of the network has become a chain of clusters.

8.4 Regions and ports: the Schur complement as a message

For a physical system there is a natural coarse junction tree: the regions of a physical partition, with the ports of each cut as separators. Proposition 5.3 says the ports do separate, so the clusters are the regions' interiors together with their ports, and the tree is the tree of regions. This section writes out what the messages are.

Partition the variables into region interiors $I_1, \dots, I_m$ and the port set S, and partition the factors accordingly: those reading only $I_r \cup S$ belong to region r, and those reading only S, such as the cut closures once one potential per cut edge has been assigned to a side, belong to the cut. The precision matrix of the whole model, ordered with the interiors first and the ports last, is then bordered block-diagonal:

$$\boldsymbol{\Lambda} \;=\; \begin{pmatrix} \boldsymbol{\Lambda}_{11} & & & \boldsymbol{\Lambda}_{1S} \\ & \boldsymbol{\Lambda}_{22} & & \boldsymbol{\Lambda}_{2S} \\ & & \ddots & \vdots \\ \boldsymbol{\Lambda}_{S1} & \boldsymbol{\Lambda}_{S2} & \cdots & \boldsymbol{\Lambda}_{SS} \end{pmatrix}, \qquad \boldsymbol{\Lambda}_{SS} = \sum_r \boldsymbol{\Lambda}_{SS}^{(r)} + \boldsymbol{\Lambda}_{SS}^{\text{cut}}, \tag{8.4}$$

with no coupling between different interiors, because no factor reads two. Figure 8.4 draws the structure for the example below.

Proposition 8.2 (The region message)**.** *Let region r's factors have precision $\boldsymbol{\Lambda}^{(r)}$ and information $\boldsymbol{\eta}^{(r)}$ on $I_r \cup S$. Their product, marginalized over I_r, is a Gaussian on S with*

$$\mathbf{S}_r \;=\; \boldsymbol{\Lambda}_{SS}^{(r)} - \boldsymbol{\Lambda}_{SI}^{(r)} (\boldsymbol{\Lambda}_{II}^{(r)})^{-1} \boldsymbol{\Lambda}_{IS}^{(r)}, \qquad \mathbf{h}_r \;=\; \boldsymbol{\eta}_S^{(r)} - \boldsymbol{\Lambda}_{SI}^{(r)} (\boldsymbol{\Lambda}_{II}^{(r)})^{-1} \boldsymbol{\eta}_I^{(r)}. \tag{8.5}$$

This is the region's message to the rest of the system, and it is the Schur complement of the region's interior in the region's own precision.

Proposition 8.3 (Exactness of the reduced system)**.** *The posterior marginal on the ports is the Gaussian with precision $\sum_r \mathbf{S}_r + \boldsymbol{\Lambda}_{SS}^{cut}$ and information $\sum_r \mathbf{h}_r + \boldsymbol{\eta}_S^{cut}$. Given the port posterior $(\boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S)$, the posterior of region r's interior is Gaussian with*

$$\boldsymbol{\mu}_I = (\boldsymbol{\Lambda}_{II}^{(r)})^{-1} (\boldsymbol{\eta}_I^{(r)} - \boldsymbol{\Lambda}_{IS}^{(r)} \boldsymbol{\mu}_S), \qquad \boldsymbol{\Sigma}_I = (\boldsymbol{\Lambda}_{II}^{(r)})^{-1} + \mathbf{X}\, \boldsymbol{\Sigma}_S\, \mathbf{X}^{\mathsf{T}}, \quad \mathbf{X} = (\boldsymbol{\Lambda}_{II}^{(r)})^{-1} \boldsymbol{\Lambda}_{IS}^{(r)}. \tag{8.6}$$

Both are exact.

Proof. Proposition 8.2 is Proposition 6.3 applied to the product of region r's factors, which is a Gaussian on $I_r \cup S$. For Proposition 8.3, eliminate every interior from the whole precision at once. By the bordered structure (8.4), $\boldsymbol{\Lambda}_{I_r I_s} = \mathbf{0}$ for $r \neq s$, so the interior block is block diagonal,

its inverse is block diagonal, and the Schur complement $\mathbf{\Lambda}_{SS} - \sum_r \mathbf{\Lambda}_{SI_r}\mathbf{\Lambda}_{I_rI_r}^{-1}\mathbf{\Lambda}_{I_rS}$ splits into one term per region. The only factors that read I_r are region r's, so $\mathbf{\Lambda}_{I_rI_r} = \mathbf{\Lambda}_{II}^{(r)}$ and $\mathbf{\Lambda}_{SI_r} = \mathbf{\Lambda}_{SI}^{(r)}$; with $\mathbf{\Lambda}_{SS} = \sum_r \mathbf{\Lambda}_{SS}^{(r)} + \mathbf{\Lambda}_{SS}^{\text{cut}}$ the Schur complement is $\sum_r \mathbf{S}_r + \mathbf{\Lambda}_{SS}^{\text{cut}}$, and the same bookkeeping gives the information vector. For the interior, the conditional of I_r given the ports involves only the factors that read I_r, and by Proposition 6.5 it is Gaussian with precision $\mathbf{\Lambda}_{II}^{(r)}$ and information $\boldsymbol{\eta}_I^{(r)} - \mathbf{\Lambda}_{IS}^{(r)}\mathbf{x}_S$. Averaging its mean over the port posterior gives the first formula of equation (8.6), and the law of total covariance, the conditional covariance plus the covariance of the conditional mean $-\mathbf{X}\mathbf{x}_S + \text{const}$, gives the second. □

Nothing is approximated. The interior covariance in equation (8.6) has two terms: the region's own conditional uncertainty given its ports, and the port uncertainty carried into the interior by the sensitivity $\mathbf{X}$ of the interior to the ports. A region whose ports are known exactly has only the first term; a region cut off from all sensors has mostly the second.

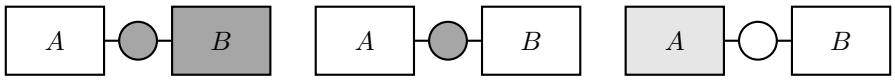
This organization has three names in three literatures. In sparse direct methods it is *nested dissection* or *substructuring*: eliminate the interiors, solve the reduced system on the separators, back-substitute. In electrical networks it is *Kron reduction*: eliminating the interior buses of a network leaves an equivalent network among the boundary buses, with equivalent lines and equivalent injections, and the reduced susceptance matrix is exactly the Schur complement. In graphical models it is the junction tree with regions as clusters. They are one computation, and the physical reading is the most useful: the region's message is the region as its neighbors see it, an equivalent network with error bars.

> **Principle 8.2.** *A region's Schur complement onto its ports is the region summarized to its neighbors, exactly. It is an equivalent network on the ports with uncertainty, and the rest of the system needs nothing else about the region.*

Proposition 8.4 (Messages compose)**.** *Split the interior variables into sets $I_1, \dots, I_R$, and call the rest S. The reduced system on S obtained by eliminating every interior at once equals the one obtained by eliminating them one set at a time, in any order. If, in addition, no factor involves variables of two different interiors, it equals the sum of the regions' messages, each computed from that region's factors alone, plus the factors that involve only S.*

Proof. The reduced system is the information form of the marginal of S, by Proposition 6.3. Marginalizing a density over $I_1 \cup I_2$ at once, or over I_1 and then over I_2, gives the same marginal, since the integrals can be taken in either order; so the reduced systems agree, and by induction they agree for any number of sets in any order. When no factor involves two interiors, the joint density is a product of one group of factors per region, each group involving only that region's interior and S, times the factors on S alone. Integrating over I_r touches only region r's group, which becomes its message on S. The product of the messages and the S-only factors is the marginal, and in information form a product is a sum. □

The script `ch08_compose.py` checks both parts on the two-region network of §8.6. Eliminating region A first, region B first, or both together gives the same 2×2 port system to fourteen digits relative to its entries, and adding the two regions' separately computed messages gives it too. Now add one factor that joins an angle of region A to an angle of region B directly, a comparison of θ_3 with θ_5, say, that does not pass through the port. Sequential elimination is still exact. The sum of separately computed messages is not. The new factor belongs to neither region's group



(a) a sensor inside B (b) a factor on the cut (c) a query inside A

Figure 8.3: What a change costs on the two-region network, by the table of §8.5. The circle is the port system. Dark: recomputed. Light: a back-substitution against a factorization already held. White: reused as it was. (a) A sensor inside region B changes B's message and the port solve. (b) A factor on the cut changes only the port solve. (c) A query about region A's interior is a back-substitution in A alone.

and touches no port, so neither message sees it, and the posterior mean of the tie flow comes out at 2.1496 instead of 2.1444, a quarter of its posterior spread. A factor across the cut that bypasses the ports makes a different partition, with a larger separator, and the regions have to be redrawn before their messages can be added.

Composition is what makes hierarchy possible. A region can itself be a union of subregions whose messages were computed separately and eliminated in any order, and Chapter 17 builds a building, a campus and a grid out of exactly this.

8.5 Reuse

The region message depends only on the region's own factors. That single fact decides what must be recomputed when something changes, and it is the structural basis for every claim of reuse in this book.

Change	Recompute	Reuse
sensor or prior inside region r	$\mathbf{S}_r$, $\mathbf{h}_r$; port solve; back-sub in r	every other region's message
factor on the cut	port solve; back-sub where asked	every region's message
query about region r's interior	back-substitution in r	everything else
region r replaced or re-modeled	$\mathbf{S}_r$, $\mathbf{h}_r$; port solve	every other region's message

Two things in the table deserve emphasis. A sensor inside region r changes $\mathbf{h}_r$ and, in general, $\mathbf{S}_r$; but it may change neither, if what it reads is a combination of interior variables that the ports do not depend on. The worked example has such a sensor: a meter inside a region that moves the region's interior estimates and leaves its message to the rest of the system untouched, because the region's total draw through its port is fixed by its loads whatever their split. And a query about one region's interior costs a back-substitution in that region alone, which is a solve against a factorization already held. Neither the port solve nor any other region is touched.

Figure 8.3 draws the three most frequent rows of the table on the two-region network. In operation the third case dominates. Most questions are about one region's interior given everything else, and each costs a solve. The first case is the one that makes distributed estimation practical: a region's own sensors change its own message and nothing else, so a region can fold in its readings and publish a new message without any other region repeating its work.

The cost model that follows is the one Chapter 10 uses to count. Let region r have n_r interior variables and k_r ports. Computing its message costs one sparse factorization of $\mathbf{\Lambda}_{II}^{(r)}$ plus k_r solves against it, once. The port solve costs a dense factorization of a $|S| \times |S|$ matrix, where $|S| = \sum_r k_r$. A query answered by back-substitution in one region costs a solve. A change

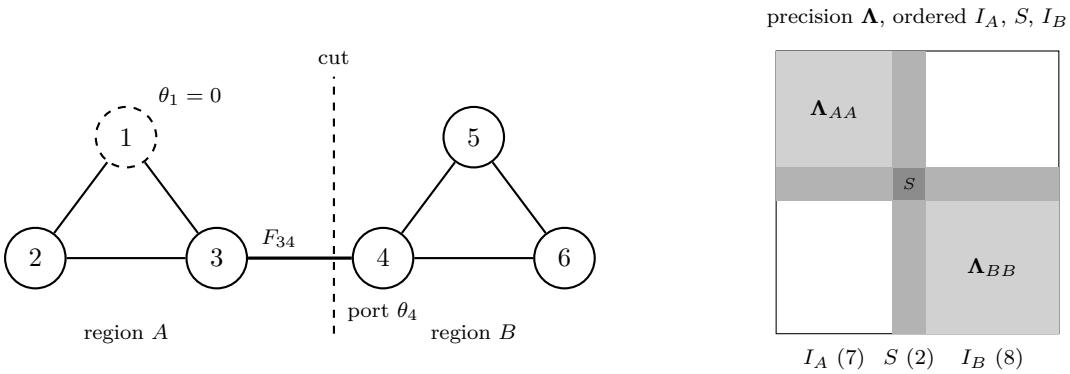


Figure 8.4: Two regions joined by one tie line. Left: the network; the dashed line is the cut, whose port pair is the tie flow F_{34} and the angle θ_4 on the B side. Right: the precision matrix with region A's seven interior variables first, the two ports in the middle, and region B's eight interior variables last. The white blocks are structurally zero: no factor reads variables from both interiors.

confined to one region costs that region's message again and the port solve again, and nothing else. Whether this beats re-solving the whole model depends on how many regions there are, how many ports, and how many questions are asked of one model; Chapter 10 puts numbers to it, and Chapter 18 asks how to cut so that the ports are few.

8.6 Worked example: two regions, one tie line

Two copies of the DC triangle are joined by a tie line. Region A is buses 1, 2, 3 with bus 1 the reference; region B is buses 4, 5, 6; the tie is line 3–4. Every line has susceptance $B = 10$. The five loads $P_2, \ldots, P_6$ are uncertain with Gaussian priors of spread 0.2 about -1.2, -0.8, -0.5, -1.0, -0.6 per unit. Physics rows have width 0.01. Three meters read F_{12}, F_{56} and the tie flow F_{34}, with accuracy 0.02, at 1.75, -0.15 and 2.15, values a little off the flows the nominal loads would produce. Figure 8.4 draws the network and the block structure of its precision. Every number is from the script `ch08_regions_schur.py` in the book's `scripts` directory.

The separator. In row-per-factor form the cut edge's port pair is $\{F_{34}, \theta_4\}$, with θ_4 assigned to region B and the cut closure c_{34}, which also reads θ_3, assigned to region A. The script confirms on the Markov graph of the assembled precision that no edge joins a variable of A's interior to one of B's. In nodal form, with the flows eliminated, the balance at bus 4 would read θ_3, θ_4, θ_5, θ_6 and P_4 together, making them a clique, and the smallest separator at the same cut would have three or four variables. The flow variable is worth keeping: it is the cheaper place to cut.

Exactness. The full model has 17 unknowns and 20 factors. Its posterior on the ports is $F_{34} = 2.1496 \pm 0.0199$ and $\theta_4 = -0.4490 \pm 0.0096$. Now eliminate each region separately. Region A's factors, including the cut closure, give a 2×2 message on the ports; region B's give another; their sum is the reduced port system. Solving it gives the same two numbers to fourteen decimal places, and back-substituting into region A by equation (8.6) reproduces every interior marginal

of the full model to thirteen: the load at bus 3, for instance, is -0.7805 ± 0.1794 either way. Proposition 8.3, checked.

The messages, read physically. Region A's message has precision

$$\mathbf{S}_A \approx \begin{pmatrix} 2738 & 1661 \\ 1661 & 11822 \end{pmatrix} \quad \text{on } (F_{34}, \theta_4) : \tag{8.7}$$

it constrains both the tie flow and the angle at bus 4, because bus 4 is tied through line 3–4 and the triangle to the reference, and because the tie meter sits on the cut side of the split and was assigned to A. Region B's message is

$$\mathbf{S}_B \approx \begin{pmatrix} 8.31 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbb{E}[F_{34} \mid \theta_4] = 2.100 + 0 \cdot \theta_4, \quad \text{sd } 0.347. \tag{8.8}$$

Region B says nothing about the angle at its own port, because nothing in B reaches the reference except the tie itself: its angles are defined only relative to θ_4. What it says about the flow is that the tie must carry 2.10 per unit into B with a spread of 0.347, which is the sum of its three load priors, $0.5 + 1.0 + 0.6$, with their spreads added in quadrature, $0.2\sqrt{3}$. That is Proposition 1.1, exactness at a cut, delivered as a message with an error bar: the flow across the boundary equals the net load inside, whatever the inside does, and the message is the inside's ignorance of its own total. Had B a second tie to the reference network, the message would acquire a slope in θ_4, the equivalent susceptance seen from bus 4, and the intercept would become an equivalent injection: the classical Ward equivalent, with the zero spread that classical equivalents assume replaced by the spread the priors imply.

Reuse, two ways. Move the meter on F_{56} from -0.15 to 0.00. Inside B the estimates shift: P_5 from -1.040 to -0.826 and P_6 from -0.593 to -0.807, the meter redistributing load between the two buses. Region B's message does not change at all, to twelve decimals: the meter reads a difference of loads, and the message carries only their sum. Nothing outside B needs to be told. Now instead change the prior mean of P_5 from -1.0 to -1.3. Region B's intercept moves from 2.100 to 2.400; region A's message is unchanged, to machine precision; and solving the 2×2 port system with A's old message and B's new one gives $F_{34} = 2.1506$, $\theta_4 = -0.4491$, which is what a full re-solve gives. Region A's seven-variable elimination was done once and not again.

On seventeen variables none of this is worth the bookkeeping. The point is the structure: what changed inside B was paid for inside B and at the two ports, and the whole of A was reused unread. That structure is what scales.

8.7 Second worked example: two areas with their own references

Region B of the previous example said nothing about the angle at its own port, because nothing in it reached the reference except the tie. A region that carries its own reference sends a richer message, and in a power system the common case is the voltage rather than the angle: a generator on automatic voltage regulation holds the magnitude at its bus, and holds it whether or not the tie is there.

Work in the decoupled linearization, in which a branch carries reactive power $Q_{ij} = b_{ij}(V_i - V_j)$ and a regulating generator pins V at its bus. Two areas of Figure 8.5 are each a triangle: area A

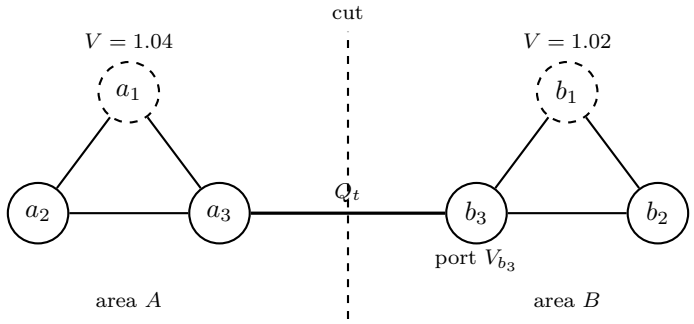


Figure 8.5: Two areas joined by one tie, each with a generator holding the voltage at one of its buses. The reservoir symbol marks the regulated bus. The cut is the tie; its port pair is the tie's reactive flow Q_t and the voltage at b_3. Unlike the previous example, area B's voltages are defined without the tie, so its message constrains the port voltage as well as the port flow.

Table 8.1: The two areas: the port posterior from the full model and from the reduced port system, and area B's message read as a Thevenin equivalent, $\mathbb{E}[V_{b_3} \mid Q_t] = V_{\text{eq}} + x_{\text{eq}} Q_t$, with and without the meter in area B, against the Kron reduction of area B's susceptance matrix.

	full model	reduced port system
Q_t [p.u.]	0.1072 ± 0.0077	0.1072 ± 0.0077
V_{b_3} [p.u.]	1.0087 ± 0.0010	1.0087 ± 0.0010
area B's message	V_{eq} [p.u.]	x_{eq} [p.u.]
with its meter	1.0064 ± 0.0011	0.02015
without its meter	1.0077 ± 0.0019	0.03333
Kron reduction at b_3	–	0.03333

is buses a_1, a_2, a_3 with a_1 regulating at 1.04 per unit, area B is b_1, b_2, b_3 with b_1 regulating at 1.02, and the tie joins a_3 to b_3. Every branch inside an area has susceptance 20; the tie is weaker, at 6. The four reactive loads are uncertain with Gaussian priors of spread 0.05 about 0.25, 0.18, 0.30 and 0.22 per unit. Physics rows have width 10^{-3}. One reactive-flow meter of accuracy 0.004 sits in each area, reading 0.2520 and 0.3050. The model has fifteen unknowns and seventeen factors. Every number is from `ch08_two_areas.py`.

Exactness. The port pair is $\{Q_t, V_{b_3}\}$. Area A's interior is its two free voltages, its three branch flows and its two loads; area B's is the same less the port voltage. The Markov graph has no edge between the two interiors. The full posterior gives $Q_t = 0.1072 \pm 0.0077$ per unit, flowing from the higher-voltage area A into B, and $V_{b_3} = 1.0087 \pm 0.0010$. The reduced port system, the sum of the two areas' messages, gives the same two numbers to 1×10^{-12} and their covariance to 8×10^{-18}, and back-substitution into area B reproduces its six interior marginals to 5×10^{-12}. Table 8.1 collects the numbers.

The message is a Thevenin equivalent. Area B has its own regulating generator, so its voltages are defined without the tie, and its message constrains the port voltage as well as the port flow. Read as the conditional of the port voltage given the tie flow, it is $\mathbb{E}[V_{b_3} \mid Q_t] = 1.0064 + 0.02015\, Q_t$ with a conditional spread of 0.0011 per unit: an equivalent voltage and

an equivalent reactance, with an error bar on the first. This is the Thevenin equivalent an engineer would draw at b_3, and the reactance is not quite the network's. Kron reduction of area B's susceptance matrix at b_3, with the regulated bus grounded, eliminates b_2 and leaves $2b - b^2/2b = 1.5b = 30$, a reactance of 0.03333; removing area B's meter from its message recovers that slope to 2×10^{-15}, with the equivalent voltage 1.0077 known only to 0.0019. The meter has done two things to the message. It has narrowed the equivalent voltage from 0.0019 to 0.0011 per unit, and it has stiffened the apparent reactance from 0.03333 to 0.02015, because a branch flow held near its reading pushes back against a change in the tie flow more than the bare network would. The message is the equivalent network with the region's readings folded in, which is more than Kron reduction gives and different from it; Chapter 17 finds the same three levels down.

Reuse. Raise area A's meter by 0.010, to 0.2620. Area B's message does not change at all — not to within a tolerance, but in the same floating-point bits, because no factor of area B was touched — and solving the two-variable port system with B's cached message gives $Q_t = 0.1054$ and $V_{b_3} = 1.0086$, the full re-solve's values to 1×10^{-12}.

8.8 Discrete variables by region

A region with d_r discrete variables has 2^{d_r} branches, and its message is a mixture of 2^{d_r} Gaussians on its ports, each with a weight from its branch's predictive density (Proposition 7.1). Regions combine by multiplying mixtures, and the product of mixtures with c_1 and c_2 components has $c_1 c_2$; across m regions with $d = \sum_r d_r$ discrete variables in all, the port posterior has $\prod_r 2^{d_r} = 2^d$ components, which is the full enumeration again. Region structure has not reduced the count. What it has done is localize the expensive part: each region's 2^{d_r} branches are eliminated on the region's own interior, once, and only the small mixtures on the ports are multiplied. When d_r is a handful per region that is exact and affordable; when it is not, pruning components by weight and sampling the remainder are the subject of Chapter 9, and Chapter 15 applies them to line outages by the hundred.

The two-region network shows the count at a size small enough to list. Make three lines uncertain, each out with probability 0.05: line 2–3 in region A, and lines 4–5 and 5–6 in region B. Region A has two branches, region B four, the network eight. Each branch is linear in the loads, so its weight is its prior times the predictive density of the three meters (Proposition 7.1), and Table 8.2 lists them, from `ch08_messages.py`. Before the meters the prior needs four components to reach ninety-nine percent of the weight: all lines in, at 0.857, and each single outage, at 0.045. The prior mixture on the port angle, drawn in Figure 8.6, is visibly spread: the outage of line 2–3 moves θ_4 from -0.443 to -0.500, and the loss of both lines at bus 5, which islands it and sheds its load, moves it to -0.277.

After the meters one component carries 0.99904 of the weight. The meter on line 5–6 reads -0.15, and a branch with that line out predicts a reading of zero to within the meter's accuracy; its predictive density is 10^{-11}, and every branch with line 5–6 out is gone. The outage of line 2–3 keeps 0.00096, which is the 0.001 that Chapter 9 finds for region A's mixture message on its own: the global enumeration and the region's message agree. The work divides as the text above said. Region A's interior is eliminated once for each of its two branches and region B's once for each of its four, six interior eliminations, and then eight two-variable port systems are combined; the global route solves eight full models. With m regions of b branches each that is mb interior eliminations against b^m full solves, while the port combinations remain b^m unless the weights

Table 8.2: The two-region network with three uncertain lines: the eight branches, their prior probabilities, the predictive density of the three meter readings under each, the posterior, and the port angle θ_4 before and after the readings. The two branches that island bus 5 shed its load.

lines out	prior	predictive density	posterior	θ_4 prior	θ_4 posterior
none	0.857	1.2×10^{1}	0.99904	-0.443 ± 0.060	-0.4490 ± 0.0095
2–3	0.045	2.2×10^{-1}	0.00096	-0.500 ± 0.072	-0.5100 ± 0.0204
4–5	0.045	6.9×10^{-6}	$< 10^{-8}$	-0.443 ± 0.060	-0.4484 ± 0.0095
5–6	0.045	3.6×10^{-11}	$< 10^{-8}$	-0.443 ± 0.060	-0.4490 ± 0.0095
2–3, 4–5	0.0024	1.3×10^{-7}	$< 10^{-8}$	-0.500 ± 0.072	-0.5091 ± 0.0204
2–3, 5–6	0.0024	6.5×10^{-13}	$< 10^{-8}$	-0.500 ± 0.072	-0.5100 ± 0.0204
4–5, 5–6	0.0024	4.7×10^{-14}	$< 10^{-8}$	-0.277 ± 0.049	-0.4483 ± 0.0095
all three	0.0001	8.5×10^{-16}	$< 10^{-8}$	-0.300 ± 0.060	-0.5090 ± 0.0204

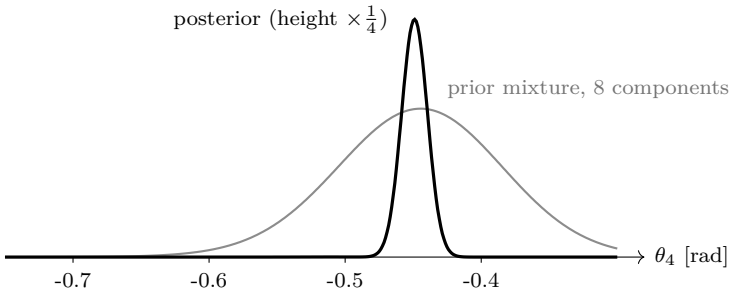


Figure 8.6: The port angle θ_4 of the two-region network with three uncertain lines. Grey: the prior mixture over the eight branches, with four components carrying ninety-nine percent of the weight. Black: the posterior after the three meters, one component, drawn at a quarter of its height.

prune them, which here they do to one.

8.9 In practice

Cut at a port pair, and assign each cut closure to one side. A region's factors are those that read only its interior and its ports. The closure of a cut edge reads potentials on both sides; give it to the side that owns the near potential, and keep the far potential as the port. Getting this wrong puts a factor in both regions or in neither, and the reduced system is then wrong without any error being raised.

Check exactness once. On a representative case, solve the full model and the reduced port system and compare to machine precision, as both examples of this chapter do. A discrepancy at the eighth digit is a factor assigned to the wrong region, not rounding.

Hold the factorizations. The value of the region message is reuse, and reuse needs each region's interior factorization kept between questions. A back-substitution is then one solve against a factor already held; a change in one region is that region's elimination and the port solve.

Read the message physically. A region without a reservoir sends a message on its port flow alone, its total with an error bar. A region with a reservoir sends a Thevenin equivalent: an equivalent potential with an error bar and an equivalent resistance. If the message has neither form, a factor is missing or misassigned.

Do not replace the message by a Kron reduction. The Kron reduction of the region's conductance matrix is the message with the region's sensors and priors removed. It is right for planning with no data and wrong for estimation with data; area B's meter stiffened the equivalent reactance by a factor of 1.65 and narrowed the equivalent voltage's spread by the same factor.

Keep discrete branches inside their regions. A region's branches are eliminated on its own interior, once; only the small mixtures on the ports are combined. Weight each branch by its predictive density, not by its soft-row evidence.

8.10 What is missing

Everything in this chapter was exact and everything was Gaussian per branch. Chapter 9 asks what to do when the exact route is too expensive: when the treewidth is large, when the branch count is large, or when a factor is not Gaussian and its message has no closed form. Chapter 10 then puts the exact route, the approximate route and plain sampling on one problem and counts what each costs in physics solves. And the reuse of §8.5 was stated for a change in one region; Chapter 11 treats the change that matters most in operation, one new sensor reading at a time, and shows it costs even less than a region message.

Notes and further reading

The sum-product algorithm on factor graphs, with the message equations (8.1)–(8.2) and the proof of exactness on trees, is Kschischang, Frey and Loeliger's [48]. The junction tree, the running intersection property and exact local propagation on loopy graphs are Lauritzen and Spiegelhalter's [51]; Koller and Friedman [46] give the correspondence between junction trees and elimination orders in full. For Gaussian models, Rue and Held [79] treat the same computation as sparse Cholesky, and the identification of the elimination tree with the junction tree is standard in both literatures.

Eliminating the interior of a subsystem onto its boundary is substructuring in structural mechanics and nested dissection in sparse direct methods [16, 25]; its iterative descendants are the domain decomposition methods of Toselli and Widlund [92]. In electrical networks the same Schur complement is Kron reduction [47], given its modern graph-theoretic treatment by Dörfler and Bullo [18], and the reduced network with equivalent injections is Ward's equivalent [95], still the standard external-system model in power-system analysis. Proposition 8.2 is Kron reduction of the posterior precision rather than of the susceptance matrix; the book's contribution is to read it as a message on the port separator of Proposition 5.3 and to carry the priors' and sensors' information along with it, so that the equivalent has error bars. Distributed power-system state estimation by message passing between areas [14] is the closest published instance of §8.4 in a power-system setting.

In water networks, Han, Eguchi, Mehrotra and Venkatasubramanian [33] estimate a damaged network component by component, which is the elimination of each biconnected component onto its articulation interface, and the block-Schur marginal of Proposition 8.2 is exact there whichever interface is chosen; what the interface decides is whether the interior factorizes (Chapter 5, Notes). Irofti, Romero-Ben, Stoican and Puig [36] tie the snapshots of a time window together with a persistence factor on the leak; a joint model of T network copies sharing one leak variable is the same construction, and its junction tree has the shared variable as every separator. The reuse of one factorization across many hypotheses in §8.5 is what makes a leak search affordable: a score test at the leak-free optimum costs one sparse solve per candidate junction against the cached factorization and shortlists the few hypotheses worth an exact evidence. Dellaert and Kaess [17] present the elimination algorithm, the Bayes tree and incremental re-elimination for the same sparse Gaussian computations in the language of robotics.

Summary

- On a tree, two sweeps of sum-product messages give every marginal exactly. Gaussian messages are precision-information pairs; a factor-to-variable message is a small Schur complement.
- On a loopy graph, exact inference runs on a junction tree of clusters, which is an elimination order made explicit; for Gaussian models it is the sparse Cholesky factorization.
- Physical regions with their ports as separators are a junction tree. A region's message is the Schur complement of its interior in its own precision: the region as its neighbors see it, an equivalent network with error bars. The reduced port system is exact and back-substitution recovers every interior exactly.
- The message depends only on the region's own factors. A change inside one region costs that region and the port solve; every other region is reused.
- On the two-triangle network the reduced system agrees with the full model to fourteen decimals; region B's message is its total load 2.10 ± 0.35 and nothing about the angle, which is exactness at a cut with an error bar; an interior meter moves B's loads and leaves its message untouched.
- With three uncertain lines the two-region network has eight branches; four carry ninety-nine percent of the prior and one carries 0.999 of the posterior. Regions eliminate their branches on their own interiors, mb eliminations for m regions of b branches, not b^m.
- On two areas joined by a tie, an area with its own regulating generator sends a Thevenin equivalent: an equivalent voltage with an error bar and an equivalent reactance. Without the area's meter the reactance is the Kron reduction, 0.03333 per unit; with it, 0.02015, and the equivalent voltage's spread falls from 0.0019 to 0.0011 per unit.

Exercises

Exercise 8.1. Write out the sum-product messages for the chain of Figure 2.1 rooted at G, with the priors and sensor of Figure 3.1 and hard physics, and show that they reproduce the second column of Table 3.2. Which message is the Schur complement of which block?

Exercise 8.2. Construct a junction tree for the DC triangle in row-per-factor form (Figure 2.3) from an elimination order of width two, verify the running intersection property, and identify

the separators. Then do the same with the availability a_{23} added and say which cluster holds the discrete variable.

Exercise 8.3. In the worked example, add a second tie line from bus 6 to bus 1. The cut now has two edges and four ports. Recompute region B's message and show that its precision on the angles is no longer zero; read off the equivalent susceptance between the two port buses and compare it with the Kron reduction of the susceptance matrix of region B alone.

Exercise 8.4. Prove the claim of §8.5 for the meter on F_{56}: show that, with B's only connection to the reference through the tie, the tie flow is a function of the sum of B's loads alone, and hence that any interior meter whose reading is a combination of loads with zero coefficient sum leaves the message unchanged. Find an interior meter that does change it.

Exercise 8.5. Give region B two uncertain lines, so that it has four branches, and region A one, so that it has two. Compute the eight-component mixture on the ports, its weights, and the number of components that carry more than 99 percent of the weight. At what number of uncertain lines per region does the mixture stop being enumerable on your machine?

Solutions to selected exercises

Exercise 8.1. With hard physics and one sensor on V_2, the only row that ties the uncertain quantities to the reading is $V_2 = V_s + G/b_{2s}$; the other rows determine V_1 and the flows, which nothing reads. The graph is then a star, one physics factor on (V_2, G, V_s) with three leaves: the prior on G, with information pair $(\lambda, \eta) = (2500,\ 125)$; the prior on V_s, $(1.111 \times 10^5,\ 1.111 \times 10^5)$; and the sensor on V_2, $(4 \times 10^6,\ 4.108 \times 10^6)$. Each leaf sends itself. The physics factor's message to G integrates V_2 and V_s against their incoming messages, $V_2 \sim \mathcal{N}(1.027, 0.0005^2)$ and $V_s \sim \mathcal{N}(1.000, 0.003^2)$: then $G/2 = V_2 - V_s \sim \mathcal{N}(0.027,\ 9.25 \times 10^{-6})$, so $G \sim \mathcal{N}(0.054, 0.006083^2)$, the pair $(2.703 \times 10^4,\ 1.459 \times 10^3)$. The marginal of G is the product of its two incoming messages, precision $2500 + 27027 = 29527$ and mean $(125 + 1459.5)/29527 = 0.053661$, a spread of 0.005820. The message to V_s is $V_s = V_2 - G/2$ with $G/2 \sim \mathcal{N}(0.025, 0.01^2)$ from G's prior message, that is $\mathcal{N}(1.002, 0.010012^2)$, and the marginal of V_s is 1.000165 ± 0.002874. Both are the second column of Table 3.2. The message to G is the Schur complement of the (V_2, V_s) block in the precision of the physics factor times the messages from V_2 and V_s: the two variables are eliminated onto G, which is equation (8.2) for Gaussians.

Exercise 8.3. With the second tie, region B has two port buses, 4 and 6, and four port variables: the tie flows F_{34} into bus 4 and F_{61} out of bus 6, and the two angles. Kron reduction of B's own susceptance matrix, the three lines 4–5, 4–6, 5–6 of susceptance 10, onto buses 4 and 6 eliminates bus 5: the 3×3 matrix $10\left(\begin{smallmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{smallmatrix}\right)$ loses its middle row and column through the Schur complement, leaving $\left(\begin{smallmatrix} 20 & -10 \\ -10 & 20 \end{smallmatrix}\right) - \frac{1}{20}\left(\begin{smallmatrix} 100 & 100 \\ 100 & 100 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 15 & -15 \\ -15 & 15 \end{smallmatrix}\right)$. The equivalent susceptance between the two port buses is 15: the direct line of 10 in parallel with the path through bus 5, two lines of 10 in series, which is 5. Region B's message now constrains the tie flows given the port angles, and the script finds the injections into B through its ports respond to (θ_4, θ_6) through exactly this matrix: the message's slope is the Kron-reduced susceptance, because B has no sensor and its load priors enter only the intercept. What B still cannot say is the level of its angles: the matrix has rows summing to zero, and only the tie to the reference through region A fixes the level. The message's precision on the angles is no longer zero, but it is singular along the direction that shifts both port angles together.

Exercise 8.4. With one tie, region B's three buses have balances $F_{34} - F_{45} - F_{46} + P_4 = 0$, $F_{45} - F_{56} + P_5 = 0$ and $F_{46} + F_{56} + P_6 = 0$; their sum is $F_{34} + P_4 + P_5 + P_6 = 0$, so the tie flow is minus the sum of the loads whatever the internal flows. The message on (F_{34}, θ_4) is therefore a statement about the sum $\Sigma = P_4 + P_5 + P_6$ alone, and a meter changes it only if it changes the posterior of Σ given the port. A meter reading a combination $\sum_i c_i P_i$ with $\sum_i c_i = 0$ is, under priors of equal spread, uncorrelated with Σ a priori, since $\mathrm{Cov}(\sum c_i P_i, \Sigma) = \sigma^2 \sum c_i = 0$, and for Gaussians uncorrelated is independent; conditioning on it leaves Σ's distribution unchanged. The flow on line 5–6 is such a combination: within B, $F_{56} = (P_6 - P_5)/3$ plus a term in F_{34} that the port already fixes. The script confirms it: a meter on F_{56} changes the message by 4×10^{-12}, and a meter on $P_5 - P_6$ by 3×10^{-11}. A meter on the sum changes it completely, the tie flow the message implies moving from 2.100 ± 0.347 to 2.299 ± 0.026; a meter on one load, P_5, changes it partly, to 2.199 ± 0.284, because one load's reading is correlated with the sum.

Exercise 8.2. The first part is Figure 8.2. In the row form of Figure 2.3, with the loads as fixed injections, the order F_{12}, F_{13}, F_{23}, θ_2, θ_3 has width two, and its cliques are the three clusters of the figure, with separators $\{\theta_2, F_{23}\}$ and $\{\theta_3, F_{23}\}$. The running intersection property holds: F_{23} is in all three clusters, and each angle is in two adjacent ones. If the loads are uncertain variables, as in the triangle of Exercise 5.3, eliminate them first. P_2 forms the cluster $\{P_2, F_{12}, F_{23}\}$ and P_3 the cluster $\{P_3, F_{13}, F_{23}\}$, and each joins the cluster holding its flow through the separator of that flow and F_{23}. The width stays two, since the new clusters also have three variables, and the property still holds: F_{23} is now in all five clusters, and each load is in one. With the availability a_{23} added, the closure of line 2–3 becomes $F_{23} - a_{23}B(\theta_2 - \theta_3)$, and a_{23} joins the cluster that holds that row, the middle one, which becomes $\{F_{23}, \theta_2, \theta_3, a_{23}\}$. The discrete variable lives in one cluster, and every message out of that cluster is a two-component mixture: Chapter 7's branch structure, seen as a junction tree.

Exercise 8.5. This is the configuration of §8.8: line 2–3 uncertain in region A, and lines 4–5 and 5–6 in region B, each out with probability 0.05. Table 8.2 lists the eight components with their weights. Before the meters, four components carry more than 99 percent of the weight; after them, one does, at 0.99904. The mixture's size is the product of the regions' branch counts, so with ℓ uncertain lines in each of m regions it has $2^{m\ell}$ components, of which the region-wise route eliminates only $m\,2^{\ell}$ interiors and combines the rest on the ports. Two regions of 20 uncertain lines each already give $2^{40} \approx 10^{12}$ port combinations, past what a workstation enumerates in an afternoon. Beyond that, the components have to be pruned by weight and by the readings, as Chapter 9 does, and the mixture carried as its few heavy components.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

The message is a Schur complement *marginalization by Schur complement; an interface marginal without forming the joint; a message as an exact summary of everything behind a cut*
`foundation_power_flow/local_to_global_poc/problems/the_message_is_a_schur_complement`

An area equivalent that is exact *decomposition as a solver strategy rather than an approximation; the Schur complement as an exact equivalent of everything behind the cut; a cut that separates against one that does not*

power_grid/area_decomposition/problems/an_area_equivalent_that_is_exact

Chapter 9

Approximate inference

Chapter 8 computed posteriors exactly, and said when that is affordable: when the treewidth of the factorization is small, when the number of discrete branches is small, and when every factor is Gaussian per branch. This chapter is about what to do when one of those fails. It has an unusual shape for a chapter on approximate inference, because its first and longest section is about a method that does not work on the graphs of this book. Loopy belief propagation, the standard approximate method for graphical models, fails to converge on every physical model the book has built, for a reason that is structural and worth understanding. The methods that do work are the ones that respect the structure the earlier chapters found: exact elimination within regions, approximate messages between them, and simulation inside a region that emits a compact summary to its ports. The chapter closes with the hierarchy of methods, from exact and analytical to simulation inside factors, and a rule for choosing among them.

9.1 When exact is too expensive

Three things make the exact route of Chapter 8 too expensive, and they call for different remedies.

Treewidth. The largest cluster of the junction tree, hence the largest dense block of the factorization, is too big. For the nearly-planar networks of this book this rarely happens in the Gaussian part; it happens for three-dimensional meshes and for factorizations chosen badly. The remedy is a better partition (Chapter 18) or an iterative solver in place of a direct one.

Branches. The number of discrete configurations is 2^k with k in the dozens or hundreds, as when every line of a network may be out. The remedy is to prune branches by their evidence, to exploit separation between discrete variables, and to sample the rest.

Non-Gaussian factors. A factor whose message has no closed form: a bounded prior, a saturating closure that the active-set loop cannot resolve, a region whose interior physics is strongly nonlinear. The remedy is to approximate the message, either by projecting it onto a Gaussian or by representing it by samples.

What follows treats the standard approximate methods in the order a graphical-model text would, but tested on the book's models, and the test changes the conclusions.

9.2 Loopy belief propagation

The sum-product messages (8.1)–(8.2) were derived for trees. Nothing stops one from running them on a loopy graph: initialize every message, update all of them from their current neighbors, repeat, and stop when they change by less than a tolerance. This is *loopy belief propagation*. It has been extraordinarily successful in decoding and in computer vision, where the graphs are loopy, the factors are weak, and approximate marginals are what is wanted. Its properties on Gaussian models are known precisely.

Proposition 9.1 (Gaussian loopy propagation)**.** *On a Gaussian model with precision $\mathbf{\Lambda}$, if loopy belief propagation converges, the means it returns are exact. Convergence is not guaranteed. A sufficient condition is* walk-summability*: the spectral radius of the matrix of absolute partial correlations, $|\mathbf{R}|$ with $\mathbf{R} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{\Lambda}\mathbf{D}^{-1/2}$ and $\mathbf{D} = \mathrm{diag}(\mathbf{\Lambda})$, is less than one. When it converges, the variances it returns are sums over a subset of the walks that the exact variances sum over, and are wrong in general; for models whose partial correlations are all non-negative they are underestimates.*

Proof of the claim about the means. Write the message from i to j as a precision P_{ij} and an information m_{ij}, and define for each edge the *cavity* precision and mean of i without j, $P_{i\setminus j} = \mathbf{\Lambda}_{ii} + \sum_{k\neq j} P_{ki}$ and $\mu_{i\setminus j} = (\eta_i + \sum_{k\neq j} m_{ki})/P_{i\setminus j}$. The iteration sets $P_{ij} = -\mathbf{\Lambda}_{ij}^2/P_{i\setminus j}$ and $m_{ij} = -\mathbf{\Lambda}_{ij}\,\mu_{i\setminus j}$, and the belief of i has precision $P_i = \mathbf{\Lambda}_{ii} + \sum_k P_{ki}$ and mean $\mu_i = (\eta_i + \sum_k m_{ki})/P_i$. At a fixed point, rearrange the belief of i: $\eta_i = P_i\mu_i - \sum_k m_{ki} = \mathbf{\Lambda}_{ii}\mu_i + \sum_k \big(P_{ki}\mu_i + \mathbf{\Lambda}_{ki}\mu_{k\setminus i}\big)$. So the i-th row of $\mathbf{\Lambda}\boldsymbol{\mu} = \boldsymbol{\eta}$ holds if, for every edge, $P_{ki}\mu_i + \mathbf{\Lambda}_{ki}\mu_{k\setminus i} = \mathbf{\Lambda}_{ik}\mu_k$. Write $a = \mathbf{\Lambda}_{ik}$, $p = P_{i\setminus k}$, $q = P_{k\setminus i}$, $u = \mu_{i\setminus k}$, $v = \mu_{k\setminus i}$. The beliefs are $\mu_i = (pu - av)/(p - a^2/q) = q(pu - av)/(pq - a^2)$ and, by symmetry, $\mu_k = p(qv - au)/(pq - a^2)$. Then $P_{ki}\mu_i + av = -\frac{a^2}{q}\,\frac{q(pu-av)}{pq-a^2} + av = \frac{ap(qv-au)}{pq-a^2} = a\mu_k$, which is the identity. □

The convergence condition and the characterization of the variances are Malioutov, Johnson and Willsky's walk-sum analysis, which expands the covariance as a sum over walks in the graph; the proof is longer than this book needs, and the notes give it. What the proof above shows is why a converged iteration is worth having: its means are the exact ones, whatever the loops. The condition has a physical reading. The partial correlation between two variables that share a factor measures how tightly that factor ties them given everything else. Walk-summability says the ties around every cycle must be weak enough, in a precise sense, that the influence circulating around the cycle dies out. A graph of weak, noisy pairwise factors, such as a decoding graph, satisfies it. A graph of tight physics does not.

9.2.1 On the book's models

Run Gaussian loopy propagation, in the standard pairwise form on the posterior precision, on three of the book's models: the soft export feeder with two sensors, the DC triangle with uncertain loads and one meter, and the two-region network of Chapter 8. Every number is from the script `ch09_approximate.py` in the book's `scripts` directory. Table 9.1 gives the result: the spectral radius that decides walk-summability, and whether the iteration converged, undamped and with damping of one half.

Not one of the physical graphs is walk-summable, and on not one does the iteration converge, damped or not. The only convergent cases are the two-variable reductions, which are trees, on which the messages are exact by construction. Even the angles-only nodal form of the two-region

Table 9.1: Gaussian loopy propagation on the book's models. $\rho(|\mathbf{R}|)$ is the walk-summability radius; below one guarantees convergence. Row form is the row-per-factor graph; nodal form has the flows eliminated; the last block has the loads eliminated as well. Physics widths as each chapter set them.

Model	Form	unknowns	$\rho(\lvert\mathbf{R}\rvert)$	converged
export feeder	row	6	1.64	no, with or without damping
DC triangle	row	7	1.70	no
two-region network	row	17	2.18	no
export feeder	nodal	4	1.80	no
DC triangle	nodal	4	1.50	no
two-region network	nodal	10	2.14	no
export feeder	voltages only	2	0.20	yes, exact (a tree)
DC triangle	angles only	2	0.18	yes, exact (a tree)
two-region network	angles only	5	1.01	no

network, five variables joined in two triangles and a tie, sits just above the threshold and does not converge.

Figure 9.1 follows the iteration sweep by sweep; the numbers are from the chapter's second script, `ch09_more.py`. On the triangle in row form the error of the belief means is 1.2 after the first sweep, 2.0 after the second and 1.8 after the thirtieth: the iteration neither settles nor explodes, it wanders at an error of order one hundred percent, which is worse than reporting the priors. On the feeder reduced to its two voltages and on the two-region network reduced to its port pair, both trees, the first sweep is exact to 10^{-11} and 10^{-14}, and nothing moves after it. The difference between the solid line and the other two is only whether the tight cycles were eliminated before the messages were passed.

Loosening the physics does not rescue it. Sweep the physics width on the triangle in row form from 0.01 to 2 per unit, the last being physics so loose as to be nearly absent: the radius falls from 1.70 to 1.14 and never reaches one, and the iteration never converges. The reason is visible in the structure of a physics row. The partial correlation between two variables is $-\mathbf{\Lambda}_{ij}/\sqrt{\mathbf{\Lambda}_{ii}\mathbf{\Lambda}_{jj}}$, and in the row form every entry of $\mathbf{\Lambda}$ is a sum over the few rows that read both variables. A physics variable sits in two or three rows, a flow in one closure and two balances, an angle in the closures of its lines, and nothing else; only the variables that carry a prior or a sensor have anything on their diagonal that the physics did not put there. So the tie between two physics variables that share a row is of order $1/\sqrt{d_i d_j}$, with d_i, d_j the number of rows each sits in: about one half. Every variable has several such ties, the row sums of $|\mathbf{R}|$ exceed one, and so does its spectral radius. Nor does the physics width matter: scaling every physics row by the same factor scales $\mathbf{\Lambda}_{ij}$, $\mathbf{\Lambda}_{ii}$ and $\mathbf{\Lambda}_{jj}$ together and leaves the partial correlations where they were, which is why the sweep moved the radius only as far as the priors and sensors, which did not scale, could move it. Eliminating the flows folds rows together and changes the ties a little, but a balance at a bus still ties that bus's angle to each neighbor's as tightly as the susceptances say, and the network's own cycles remain.

One pair shows the mechanism in numbers. The flow on line 1–2 and the angle at bus 2 share the closure c_{12} and nothing else, and each sits in two physics rows. Without the meter their partial correlation is -0.500 at every width, which is $-1/\sqrt{2 \cdot 2}$. The meter on F_{12} adds to

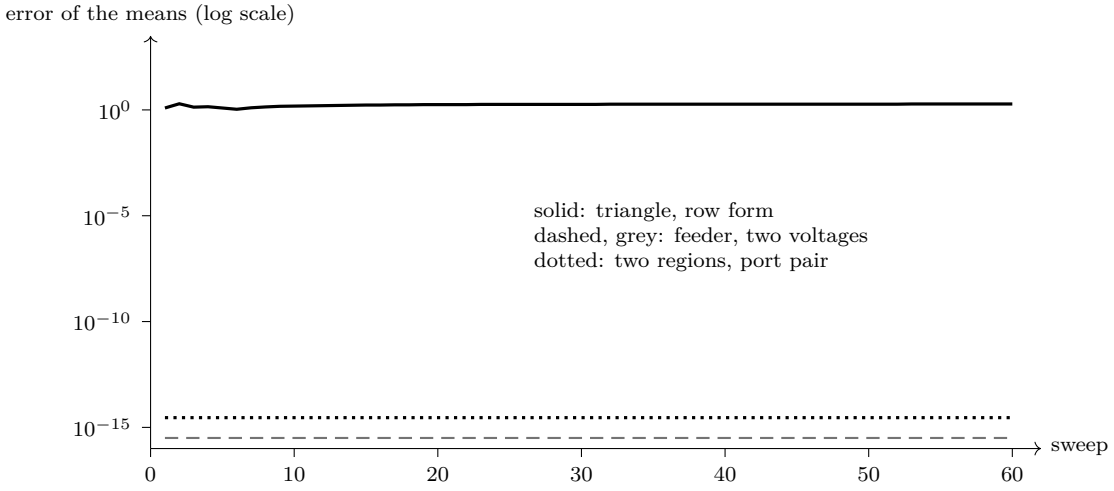


Figure 9.1: Gaussian belief propagation sweep by sweep: the largest error of the belief means against the exact posterior, on a logarithmic scale. Solid: the DC triangle in row form, radius 1.70; the error wanders between one and two, a hundred to two hundred percent, and never settles. Dashed: the feeder reduced to its two voltages, radius 0.20. Dotted: the two-region network reduced to its port pair, radius 0.29. Both reductions are trees and are exact after one sweep.

that variable's diagonal a weight the physics does not scale, and dilutes the tie to -0.471 at a width of 0.01 and to -0.001 at a width of ten; but the ties that no sensor touches stay at one half, and the radius of the whole never falls below 1.13.

Principle 9.1. *Loopy belief propagation is the wrong tool for a physical graph. Each physics variable is tied to a few others by factors that nothing dilutes, the partial correlations around every physical cycle are of order one half with several per variable, walk-summability fails, and the iteration does not converge. This is not a matter of tuning the widths; it is what conservation and closure do to the graph.*

The remedy is the one Chapter 8 already gave. Do not pass messages between variables around the cycles; eliminate the cycles. Group the variables into regions whose interiors contain the tight cycles, eliminate each interior exactly, and pass messages between regions. If the region graph is a tree, as it is for two regions or for a radial arrangement of regions, the region messages are exact and no iteration is needed. If the region graph has cycles, the messages between regions are loopy propagation at the region level, and the question of convergence returns, but on a graph whose factors are the region messages: dense on the ports, and, because each region's interior has absorbed its own tight ties, much weaker between regions than within them. Whether that graph is walk-summable is a property of the partition, and Chapter 18 measures it. The reader should expect it to hold when regions are large and ports are few, and to fail when the partition is fine.

9.3 Variational methods and expectation propagation

Loopy propagation is one of a family of methods that replace an intractable posterior by the nearest member of a tractable family. The family is chosen for tractability, a Gaussian or a

product of independent factors, and *nearest* is measured by a divergence.

Variational inference minimizes the divergence from the approximation to the posterior over a family that is usually a product of independent factors, one per variable or per block. It always converges, to a local optimum, and it always returns a distribution that is too narrow: the divergence it minimizes penalizes the approximation for putting mass where the posterior has none, so the approximation tucks itself inside the posterior's bulk. For a Gaussian posterior with correlated variables, a product-form approximation gets the means right and the variances too small by a factor that grows with the correlation, which for the ridges of Chapter 3 is a large factor. The Laplace approximation of §6.4 is also a member of this family, with the full Gaussian as the tractable class and the matching done at the mode rather than by a divergence; on the book's models it has served better than either, because a full Gaussian at the mode captures exactly the correlations that a product form throws away.

The size of the factor has a closed form for a Gaussian posterior.

Proposition 9.2 (Mean field on a Gaussian)**.** *Let the posterior be* $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$*. The product of independent Gaussians* $q = \prod_i \mathcal{N}(m_i, s_i^2)$ *that minimizes the divergence* $\mathrm{KL}(q \,\|\, p)$ *has* $m_i = \mu_i$ *and* $s_i^2 = 1/\boldsymbol{\Lambda}_{ii}$*. Since* $1/\boldsymbol{\Lambda}_{ii} \le (\boldsymbol{\Lambda}^{-1})_{ii}$*, with the ratio* $1 - R_i^2$ *where* R_i^2 *is the squared multiple correlation of variable* i *with all the others, mean field never overstates a variance and understates it by that factor.*

Proof. Up to a constant, $\mathrm{KL}(q\|p) = \frac{1}{2}\big[\sum_i \boldsymbol{\Lambda}_{ii} s_i^2 + (\mathbf{m}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Lambda}(\mathbf{m}-\boldsymbol{\mu}) - \sum_i \log s_i^2\big]$, since the expectation of $\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Lambda}(\mathbf{z}-\boldsymbol{\mu})$ under a product of Gaussians contributes only the diagonal of $\boldsymbol{\Lambda}$ from the variances. The quadratic form is minimized at $\mathbf{m} = \boldsymbol{\mu}$; setting the derivative in s_i^2 to zero gives $\boldsymbol{\Lambda}_{ii} = 1/s_i^2$. And $1/\boldsymbol{\Lambda}_{ii}$ is the conditional variance of z_i given every other variable, by Proposition 6.5, while $(\boldsymbol{\Lambda}^{-1})_{ii}$ is its marginal variance; conditioning on correlated variables removes the fraction R_i^2 of the variance. □

On the feeder after one reading the two inputs are correlated at -0.985. Mean field reports the injection to ± 0.00100 and the substation voltage to ± 0.00049, against the exact 0.00582 and 0.00287: the factor 0.172 on each, which Figure 9.2 draws. On all six variables of the soft feeder it is far worse. variable is pinned by its rows to within the physics width once the others are fixed, so the conditional variance of each is of the order of that width, and mean field reports the injection to ± 0.000010, six hundred times too narrow. Mean field on a physical graph in row form measures the physics widths and nothing else.

Expectation propagation keeps the message-passing structure of sum-product and replaces each message that has no closed form by the Gaussian that matches its first two moments, computed against the current approximation of everything else. It is the natural extension of the Gaussian machinery to the non-Gaussian factors of Chapter 7: a truncated prior, a discrete availability, a saturating closure each become a Gaussian message that is refined in place as the rest of the posterior sharpens. On a tree of regions it is exact for the Gaussian factors and approximate only at the non-Gaussian ones, and its fixed points are often accurate where variational inference is not, because moment matching does not tuck. Its cost is that it can fail to converge, and that the moment-matched Gaussian of a bimodal message is the mixture mean and variance that §7.2.1 showed to be a value no branch believes. §9.4.1 makes that precise for a region message.

9.4 Simulation inside a region

The most useful approximate method for a physical system is not a graphical-model method at all. It is to run an ordinary physical simulation, or an ordinary Monte Carlo, inside one region, and to emit the result as a message on the region's ports.

Take a region whose interior is expensive: nonlinear closures that need a Newton solve, a dozen discrete failure states, weather uncertainty that enters through many parameters at once. Its message to the rest of the system, Proposition 8.2, is the marginal of its factors over its interior, a function of the ports alone. Compute it by sampling: draw the region's uncertain inputs from their priors N times; for each draw, solve the region with the ports as free boundary variables, which for a linear-Gaussian interior conditional on the draw gives a Gaussian message on the ports by the Schur complement, and for a nonlinear interior gives a Laplace message at the solved mode. The region's message is then the equal-weight mixture of the N Gaussian components,

$$m_r(S) \;\approx\; \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}\big(S;\; \boldsymbol{\mu}_S^{(i)},\; \boldsymbol{\Sigma}_S^{(i)}\big), \tag{9.1}$$

a *particle message* whose particles are Gaussians. Compress it if the rest of the system wants a Gaussian, by moment matching; keep it as a mixture if it is multimodal; keep a few components with the most weight if it is nearly so.

Three things make this attractive. The samples are drawn once per region and reused for every query the rest of the system puts to that region, because the message is a function of the ports and the queries change only what is combined with it. The expensive solves happen inside the region, on the region's own variables, with the region's own sparse structure, and the number of solves is N times the number of regions, not N times the number of global scenarios. And nothing about the region's interior needs to be Gaussian, linear or even differentiable, so long as it can be solved; the framework's assumptions apply to the message, not to what produced it. This is the sense in which the framework does not replace conventional simulation but organizes it: a region can be a legacy simulator, and its output becomes a factor. Chapter 10 counts what this costs against global sampling.

9.4.1 What a mixture message loses when collapsed

Take region A of the two-region network and make line 2–3 uncertain, out with prior probability 0.05. Region A's message on its ports (F_{34}, θ_4) is a two-component mixture, one Gaussian per branch, weighted by the branch's evidence from the factors inside A, which include the meters on F_{12} and F_{34}, corrected for the physics determinant as Proposition 7.1 requires. The script computes both components. On the angle θ_4 they are -0.4490 ± 0.0096 with weight 0.999 and -0.5099 ± 0.0205 with weight 0.001: the meters inside A have all but ruled the outage out, and the two components are three standard deviations apart. Collapsing the mixture to one Gaussian by moment matching gives -0.4491 ± 0.0098, which is the dominant component with its spread inflated by two percent to cover the one-in-a-thousand chance of the other. That is a safe collapse: one component dominates, and the inflation is the honest price of the residual doubt. Had the weights been comparable, as they were at the crossover reading of §7.2.1, the collapsed Gaussian would have sat between two modes with a spread that is mostly their separation, and the rest of the system would have been told a falsehood. The rule is the one Chapter 7 stated: collapse when one component carries nearly all the weight; otherwise keep the components.

Table 9.2: Area 1 as a simulated region: the pocket's probability of a shortfall from two particle messages of N draws each, as an rms error over 200 repetitions and as a percentage of the exact probability, and from the Gaussian collapse of the exact messages.

	ambient 26.5°C		ambient 21°C	
	rms error	% of $\mathbb{P}$	rms error	% of $\mathbb{P}$
exact $\mathbb{P}$(shortfall)	0.557	—	0.0026	—
$N = 50$	0.058	10	0.0033	126
$N = 200$	0.032	6	0.0014	55
$N = 1{,}000$	0.013	2	0.0006	24
$N = 5{,}000$	0.006	1	0.0002	9
Gaussian collapse	+0.006	1	+0.0071	272

9.4.2 Worked example: a generation area as a simulated region

Take area 1 of the generation-adequacy example of Chapter 10 as a region. Its interior holds three unit ratings and a tie limit, its port is the power it delivers to the pocket, S_1, and with the ambient temperature fixed its message is the distribution of S_1. That message can be computed exactly on a fine grid, which is what makes the area a good test: simulate it with N draws, emit the draws as a particle message, and compare both the particles and their moment-matched Gaussian with the exact message. The pocket combines two such messages, one per area, and asks for the probability of a shortfall, $\mathbb{P}(S_1 + S_2 < 2{,}100)$. Every number is from the script `ch09_region.py`.

At the heat-wave ambient temperature of 26.5°C the message is nearly Gaussian, with skewness -0.08. The Gaussian collapse gives a shortfall probability of 0.563 against the exact 0.557, one percent off, and particle messages need about 5,000 draws each to do as well. On a mild afternoon at 21°C the picture reverses. The units bind in 55 percent of the draws and the tie in 45, so the message is the minimum of two comparable quantities, skewed by +0.33, and the pocket's shortfall is a tail event, with probability 0.0026. The Gaussian collapse puts it at 0.0097, 272 percent too high: a collapsed message is right in the middle of its distribution and wrong in exactly the tail the pocket asks about. The particle messages converge to the tail at the rate sampling allows, with an rms error of 24 percent of the probability at 1,000 draws and 9 percent at 5,000. Table 9.2 gives the convergence at both ambient temperatures, and Figure 9.3 draws the 21°C message.

The lesson is the one §9.4 began with, now in numbers. A region whose message is near Gaussian can be collapsed safely, and simulating it is wasted effort. A region whose message is skewed, bounded or multimodal should send its particles, or the questions downstream will be answered from the wrong tail. Which case holds can be read from the message itself, from its skewness or from a comparison of its tail with the Gaussian's, before the collapse is trusted.

Figure 9.4 sweeps the ambient temperature. Above about 23.5°C, where shortfalls are common, the collapse is within a few percent, better than 1,000 particles. Below it the shortfall becomes rare and the collapse's error grows without bound: 60 percent at 22°C, 270 at 21, 1,900 at 20, and at 19°C, where the exact probability is 5.5×10^{-6}, a factor of about eight hundred. Particles degrade too, since a rare event needs many draws to be seen at all; 1,000 draws give an rms error of 57 percent at 20°C. But they degrade as sampling does, and more draws cure them. No number of draws cures the collapse, whose error is in the shape it assumed.

Table 9.3: Pruning a hundred lines, each out with probability 0.01, by prior weight: the prior weight discarded by keeping every branch with at most k lines out, and the number of branches kept.

k	discarded weight	branches kept
1	2.6×10^{-1}	101
2	7.9×10^{-2}	5 051
3	1.8×10^{-2}	166 751
4	3.4×10^{-3}	4 087 976
5	5.3×10^{-4}	7.9×10^{7}
6	7.1×10^{-5}	1.3×10^{9}
7	8.2×10^{-6}	1.7×10^{10}
8	8.4×10^{-7}	2.0×10^{11}

9.5 Pruning by evidence

When the branches are many, most carry negligible weight, and the weights are known before any branch is fully processed: the evidence of a branch is one number from its factorization, and a branch's prior weight bounds its posterior weight from above whenever the evidence is bounded. Discard the branches whose weight falls below a threshold and renormalize the rest.

Proposition 9.3 (Pruning error)**.** *If the discarded branches carry total posterior weight ϵ, then for every event the probability computed from the retained branches, renormalized, differs from the exact probability by at most ϵ.*

Proof. Write the exact posterior as $p = (1 - \epsilon)\, p_R + \epsilon\, p_D$, with p_R the retained branches' mixture renormalized and p_D the discarded branches' mixture renormalized. For any event E, $p(E) - p_R(E) = \epsilon\,(p_D(E) - p_R(E))$, and both probabilities lie in $[0, 1]$, so their difference is at most one in magnitude. Hence $|p(E) - p_R(E)| \le \epsilon$. □

The consequence for practice is that one can process branches in order of prior weight, accumulate the retained posterior weight, and stop when the retained fraction is high enough for the accuracy required. How far that goes by prior weight alone is a count. Table 9.3 gives it for a hundred lines each out with probability 0.01. Keeping every branch with up to three lines out discards 1.8 percent of the prior weight and keeps 166 751 branches; to discard less than 10^{-6} requires every branch with up to eight lines out, 2×10^{11} of them, against $2^{100} \approx 1.3 \times 10^{30}$ in all. Pruning by prior weight removes nineteen orders of magnitude and still leaves two hundred billion solves. The rest must come from separation, which lets regions' branches be summed independently (Chapter 8), and from pruning by posterior weight once the readings are in, which Chapter 15 exploits.

Figure 9.5 draws the prior tail of Table 9.3 for both failure probabilities. On a logarithmic scale the tail falls faster than geometrically once k passes the mean number of failures, which is why a modest k reaches 10^{-6} at all. What the figure also shows is how much the answer depends on q: the same hundred lines cross the tolerance at $k = 8$ or at $k = 4$, and the branch count differs by a factor of fifty thousand.

Table 9.4: The hierarchy of inference methods for a physical factor graph.

Method	When	What it costs
Exact, analytical (tree messages)	tree-structured factorization, Gaussian factors	two sweeps of small Schur complements
Exact, computational (junction tree; regions and ports)	treewidth manageable; few discrete branches per region	one sparse factorization per region, a dense port solve
Laplace per branch	rows mildly nonlinear across the posterior width	a few Gauss–Newton iterations per branch
Expectation propagation	isolated non-Gaussian unary factors on a tree of regions	repeated moment matching; may not converge
Loopy propagation between regions	region graph has cycles; region messages weak	iteration; verify walk-summability first
Simulation inside a region	interior nonlinear, discrete-heavy, or a legacy simulator	N region solves, once; particle message on the ports
Pruning by evidence	many discrete branches, weights geometric	one evidence per retained branch
Global sampling (Chapter 10)	everything else; the baseline	N solves of the whole model per query

9.6 The hierarchy, and how to choose

Table 9.4 arranges the methods of this chapter and the last from exact and analytical to simulation inside factors, with the condition under which each is the right choice.

The rule that orders the table is: eliminate exactly wherever the structure allows, approximate only at the boundaries between what has been eliminated, and never pass loopy messages through tight physics. For the models of this book the first line that applies is usually the second, regions and ports with exact interiors, with the fourth or sixth used at the few places where a factor is not Gaussian. Loopy propagation between variables is never used. Global sampling is the comparison, not the method, and the next chapter costs it.

9.7 In practice

Compute the radius before running loopy propagation. The walk-summability radius of the precision takes one eigenvalue computation. If it is above one, do not run the iteration on that graph; on a physical graph in row form it always is.

Eliminate the tight cycles first. Group variables into regions whose interiors contain the physics cycles, eliminate each interior exactly, and pass messages between regions. When the region graph is a tree the messages are exact after one sweep, as the port-pair reduction of Figure 9.1 shows.

Do not trust a converged iteration's variances. Even where loopy propagation converges its means are exact and its variances are not. Report variances from a factorization, by selected inversion.

Do not use mean field on a physical graph. A product form over physics variables reports the physics widths, six hundred times too narrow on the feeder. If a cheap approximation is needed, use the Laplace posterior, which keeps the correlations.

Collapse mixtures only when one weight dominates. A two-component region message with weights 0.999 and 0.001 collapses safely; one with weights 0.7 and 0.3 does not, and the rest of the system must receive both components.

Prune with a stated error. Pruning by weight has the error bound of Proposition 9.3; state the discarded weight with every result, and expect prior weight alone to leave far too many branches for a network of a hundred uncertain components.

Collapse a message only where it is near Gaussian. Before replacing a region's particles or mixture by a Gaussian, compare the tail the downstream question needs with the Gaussian's tail. An area whose message was one percent off at one ambient temperature was off by a factor of nearly four at another.

9.8 What is missing

The chapter has said which approximate methods fit a physical graph and which do not, and it has counted their costs in words. Chapter 10 counts them in solves, on one problem, against enumeration and against global Monte Carlo, and states the conditions under which the factorized route wins and the conditions under which it does not. Chapter 11 then treats the approximation that operation needs most, folding in one reading at a time without re-solving, which turns out to be exact.

Notes and further reading

Loopy belief propagation on Gaussian models: Weiss and Freeman [96] proved that the means are exact at any fixed point and gave the first sufficient conditions for convergence; Malioutov, Johnson and Willsky [59] introduced walk-summability, showed that it guarantees convergence, and characterized the variances as sums over a subset of walks. The pairwise form of the iteration used in the script, as a solver for $\mathbf{\Lambda z} = \boldsymbol{\eta}$, is Shental, Siegel, Wolf, Bickson and Dolev's [83]. The observation that physics variables, sitting in only a few rows each with nothing to dilute the ties, carry partial correlations that no choice of physics width can reduce, and that physical row graphs are therefore not walk-summable, is the book's; it is elementary once stated, and it is consistent with the message-passing state estimators in the power-system literature [14] operating between areas rather than between variables.

Variational inference and its relation to belief propagation are set out by Wainwright and Jordan [94], with the divergence argument for why mean-field approximations are too narrow. Expectation propagation is Minka's [63]; Bishop [9] presents both in textbook form and MacKay [57] gives the Laplace approximation its place in the family. Particle representations of messages come from the sequential Monte Carlo literature; Robert and Casella [76] is the general reference and Särkkä and Svensson [81] the one closest to Chapter 16. The idea of a region as a black-box simulator that emits a summary to its boundary is co-simulation [26]; what §9.4 adds is that the summary is a probabilistic message on the port separator.

Summary

- Exact inference fails for large treewidth, many branches, or non-Gaussian factors; each has its own remedy.
- Gaussian loopy propagation has exact means when it converges, converges when walk-summable, and returns wrong variances. On every physical model of this book it is not walk-summable and does not converge, at any physics width, because each physics variable sits in only a few rows with nothing to dilute the ties, and the partial correlations around every cycle stay of order one half however the widths are scaled. Eliminate the cycles by regions instead.
- Variational inference is too narrow on correlated posteriors: mean field keeps the means and reports $1/\mathbf{\Lambda}_{ii}$, too narrow by $\sqrt{1-R_i^2}$, which is 0.172 on the feeder's two inputs and a factor of six hundred on its six physics variables. Expectation propagation extends Gaussian messages to non-Gaussian factors by moment matching, and collapsing a bimodal message is safe only when one component dominates.
- On the triangle in row form loopy propagation wanders at an error of one hundred percent; on tree reductions it is exact in one sweep.
- A region may be simulated and its result emitted as a particle message on its ports: solves stay inside the region, samples are reused across queries, and the interior may be anything solvable. A generation area's message collapses to a Gaussian within one percent on a hot afternoon and misstates the load pocket's rare shortfall by 272 percent on a mild one, where particles are needed.
- Pruning branches by evidence errs by at most the discarded weight. By prior weight alone, a hundred lines at one percent still leave 2×10^{11} branches for an error of 10^{-6}.
- Order of preference: exact by regions; Laplace per branch; EP or simulation at non-Gaussian places; loopy messages only between regions and only after checking walk-summability; global sampling as the baseline.

Exercises

Exercise 9.1. Show that two variables that appear in exactly one row each, the same row, have partial correlation of magnitude one whatever the row's weight, and that a variable appearing in d rows of equal weight and unit coefficients has partial correlation $1/\sqrt{d_i d_j}$ with a neighbor it shares one row with. Compute the partial correlation between F_{12} and θ_2 in the triangle row form with and without the sensor on F_{12}, and explain why the physics width drops out.

Exercise 9.2. Run the script's loopy propagation on the two-region network with the two regions replaced by their exact messages, so that the graph has only the two port variables and the cut factor. Show that it converges in one sweep and is exact, and explain why in terms of the region tree.

Exercise 9.3. Build a ring of m copies of region B from Chapter 8, each tied to the next, so that the region graph is a cycle. Compute the walk-summability radius of the port-level precision as m grows and as the tie susceptance varies, and find the regime in which loopy propagation between regions converges.

Exercise 9.4. Implement the particle message of equation (9.1) for region A with an uncertain line and uncertain susceptances, using $N = 200$ draws, and compare its moment-matched

Gaussian with the exact two-branch mixture of §9.4.1. How many draws are needed before the message's mean on θ_4 is within one percent of its spread?

Exercise 9.5. For L lines each out independently with probability q, find the number of lines out beyond which the discarded weight is below 10^{-6} for $L = 100$, $q = 0.01$, and count the retained branches. Repeat for $q = 0.001$, and say what the comparison implies for pruning by prior weight alone.

Solutions to selected exercises

Exercise 9.1. If variables i and j appear only in one row, with coefficients h_i, h_j and weight w, their block of $\mathbf{\Lambda}$ is $w\left(\begin{smallmatrix} h_i^2 & h_ih_j \\ h_ih_j & h_j^2 \end{smallmatrix}\right)$, and their partial correlation is $-wh_ih_j/\sqrt{wh_i^2 \cdot wh_j^2} = \mp 1$: magnitude one, whatever w. If i sits in d_i rows and j in d_j, all with unit coefficients and weight w, and they share one row, then $\mathbf{\Lambda}_{ii} = d_iw$, $\mathbf{\Lambda}_{jj} = d_jw$ and $\mathbf{\Lambda}_{ij} = w$, and the partial correlation is $-1/\sqrt{d_id_j}$. In the triangle, F_{12} sits in b_2 and c_{12} with unit coefficients, and θ_2 in c_{12} and c_{23} with coefficient B: $\mathbf{\Lambda}_{F_{12}F_{12}} = 2w$, $\mathbf{\Lambda}_{\theta_2\theta_2} = 2B^2w$, $\mathbf{\Lambda}_{F_{12}\theta_2} = Bw$, and the partial correlation is $-Bw/\sqrt{2w \cdot 2B^2w} = -\frac{1}{2}$, independent of w and of B, which is the script's -0.500 at every width. The width drops out because every entry scales with w. With the meter of weight w_m on F_{12}, $\mathbf{\Lambda}_{F_{12}F_{12}} = 2w + w_m$ and the partial correlation is $-1/\sqrt{2(2 + w_m/w)}$; at a physics width of 0.01 and a meter of 0.02, $w_m/w = 0.25$ and the value is $-1/\sqrt{4.5} = -0.471$. Now the width matters, because the meter's weight does not scale with it; but only the ties at metered variables are diluted.

Exercise 9.5. The number of lines out is binomial with $L = 100$ and $q = 0.01$, so the prior weight of having more than k out is the binomial tail, and the branches kept number $\sum_{j \le k} \binom{100}{j}$. Table 9.3 gives both: the tail first falls below 10^{-6} at $k = 8$, with 8.4×10^{-7} discarded and 2.0×10^{11} branches kept. At $q = 0.001$ the mean number out is 0.1 instead of 1, and the tail falls below 10^{-6} already at $k = 4$, with about 4.1×10^6 branches kept: fifty thousand times fewer, for components ten times more reliable. Pruning by prior weight works when the expected number of simultaneous failures is well below one, and fails as it approaches one; a large network of ordinary components is in the second regime, which is why the count must be cut further by separation and by the readings.

Exercise 9.2. With each region replaced by its exact message, Proposition 8.2, the graph left to the iteration has two port variables and the cut factor between them, with each region's message attached to its port as a factor on that port alone. That graph is a tree, and on a tree Gaussian propagation is sum-product, exact after one sweep from the leaves inward and back, by Proposition 8.1. The dotted curve of Figure 9.1 is this run: its error is 10^{-14} after the first sweep and nothing moves afterward. The walk-summability radius of the port-level precision is 0.29, well inside the unit circle, but it is not needed: exactness on a tree does not depend on convergence of an infinite walk sum. The region tree has done what the loopy iteration could not, by eliminating every cycle inside the regions before any message was passed.

Exercise 9.3. The script `ch09_region.py` builds the ring in nodal form: each copy of region B has buses 4, 5 and 6 and its three lines, bus 6 of each copy is tied to bus 4 of the next, bus 4 of the first copy is the reference, and every bus carries a load prior of spread 0.2. Eliminating

each region's interior bus leaves the port-level precision on the angles at buses 4 and 6. On the port variables one at a time, loopy propagation never converges: the walk-summability radius is between 1.07 and 1.83 for every ring from $m = 3$ to $m = 20$ and every tie from 1 to 30 per unit, and the iteration diverges in every case. A region's two ports are tied tightly through its own lines, and that alone breaks the scalar iteration. Propagation between regions should pass one message per region, over both its ports at once. Done that way it converges, and is exact, for the three-region ring at every tie, for four regions up to a tie of 10, and for six only at a tie of 1; from ten regions on it diverges at every tie. The reason is the region graph. In nodal form the load priors couple each angle to its neighbours' neighbours, so after the interiors are eliminated each region is joined to its second neighbours as well as its first. Three regions make a single loop, on which Gaussian propagation's means converge. From six regions on, every region has four neighbours, and the ring has become a ring of chords. The block walk-summability radius, a sufficient test, is below one only for the three-region ring with the weakest tie, 0.97. It certifies none of the other cases that converge, and it correctly refuses all the ones that do not. The regime in which propagation between regions converges is a coarse partition of a nearly tree-like region graph, which is the advice of §9.2 read backwards. A ring should be cut open, by merging two regions across one tie so that the region graph becomes a tree, or treated by a junction tree over regions, as Chapter 18 does.

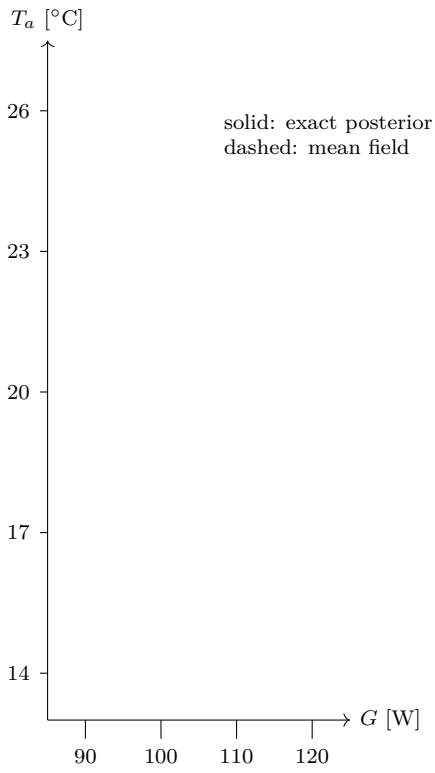
Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

A certificate on what you did not enumerate *enumerating a discrete space in probability order; a bound on the part you did not visit; a certificate that is valid but loose*
`foundation_power_flow/probabilistic_contingency_risk/problems/a_certificate_on_the_tail`

Probable is not risky *expected consequence against likelihood; a ranking that is flat within a size class; the cheap ordering and the expensive one*
`foundation_power_flow/probabilistic_contingency_risk/problems/probable_is_not_risky`

Truncation is not conservative *a quadratic search and the temptation to shorten its input; an answer that has stopped moving without having converged; a bound that runs the wrong way*
`foundation_power_flow/contingency_risk_at_scale/problems/truncation_is_not_conservative`



•

Figure 9.2: The feeder's posterior over the injection and the substation voltage after one reading (solid) and the mean-field approximation to it (dashed), both at one standard deviation. The exact posterior is a ridge at a correlation of -0.985; mean field is the axis-aligned ellipse that fits inside it, too narrow in each coordinate by $\sqrt{1-0.985^2} = 0.172$.

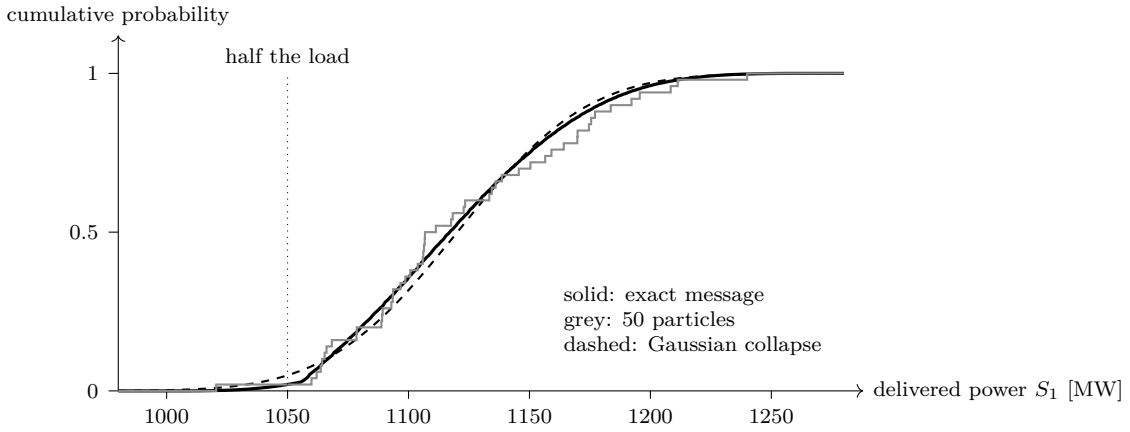


Figure 9.3: Area 1's message on its port at an ambient temperature of 21°C, as a cumulative distribution. Solid: the exact message. Grey: a particle message of 50 draws. Dashed: the Gaussian with the same mean and spread. The exact message is skewed by the competition between the unit and tie limits, and its lower tail, which decides whether two areas together fall below the load, is thinner than the Gaussian's.

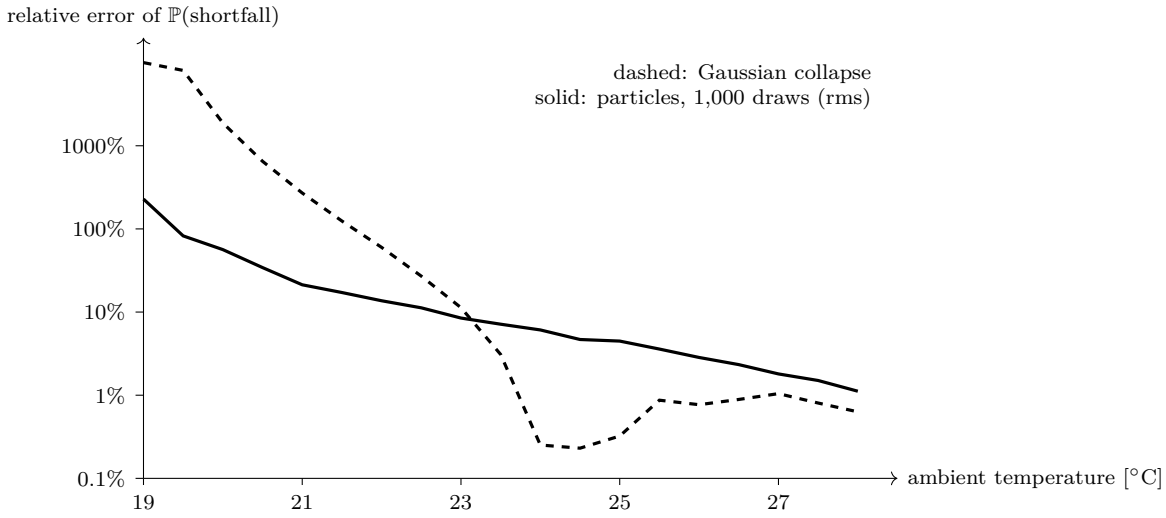


Figure 9.4: The relative error of the load pocket's shortfall probability across ambient temperatures, on a logarithmic scale clipped at 10,000 percent. Dashed: the Gaussian collapse of the two areas' exact messages. Solid: particle messages of 1,000 draws each, rms over 100 repetitions. The collapse is better where shortfalls are common and fails without bound where they are rare.

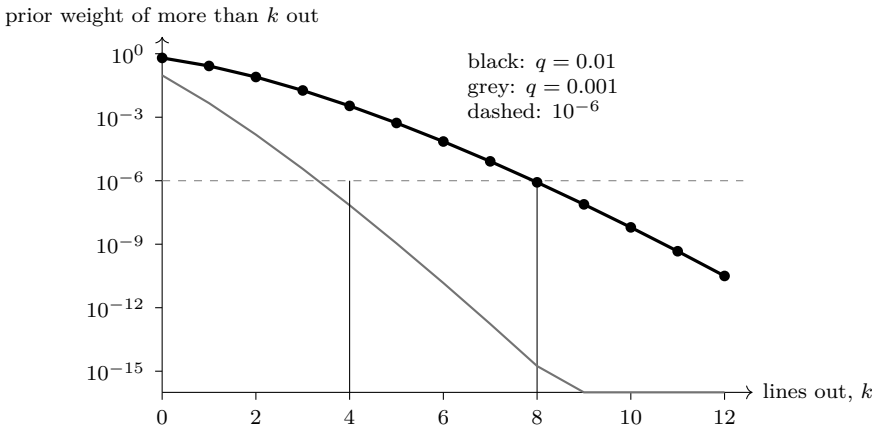


Figure 9.5: The prior weight of more than k of 100 lines out, each independently with probability q. Black: $q = 0.01$, which crosses 10^{-6} (dashed) at $k = 8$. Grey: $q = 0.001$, which crosses it at $k = 4$. Pruning by prior weight alone keeps every branch with k or fewer lines out.

Chapter 10

Enumeration, Monte Carlo, and factorized inference

Chapter 5 counted the cost of factorized inference in operations and warned that the count is not an argument against sampling. This chapter makes the comparison properly. It takes one small physical problem with a closed-form answer, computes the same two quantities by enumeration, by Monte Carlo, and by a factorized computation that exploits a separator of the model, and costs each in the one currency that matters for a physical system: the number of times the physics has to be solved. The result is not that sampling loses. It is that the two routes are good at different things, that the factorized route's advantage is reuse, and that reuse is worth a great deal when many questions are asked of one model. A second example, two generation areas that share the outdoor air, repeats the comparison on nine inputs and shows what a shared cause does to the cut. The chapter ends with the claim the book is prepared to defend, in three strengths, and with what has been established for each.

10.1 The question, stated honestly

A Monte Carlo method draws the uncertain inputs from their priors, solves the physics for each draw, and averages. It answers every question this book has posed. It computes expectations, tails, conditionals by filtering the draws, counterfactuals by re-drawing under a changed prior, and attributions by any of the standard variance decompositions. It does not care how many uncertain inputs there are: a hundred thousand draws from a space of 2^{60} states cost a hundred thousand solves. Nothing in Part II gives the factorized route an advantage on those grounds, and a claim that it "handles combinatorial uncertainty without enumeration" is a claim that sampling already makes good on.

What sampling pays for is the solve. Each draw is one evaluation of the physics, and for a real network that evaluation is a Newton iteration on thousands of unknowns, or an optimal power flow, at a cost of a fraction of a second to many seconds. A hundred thousand draws is hours. The question is therefore not which method can answer, but how many solves each needs for a given accuracy, and, since a model is rarely asked one question, how many solves the second and third and tenth questions need once the first has been answered. That is the comparison this chapter runs.

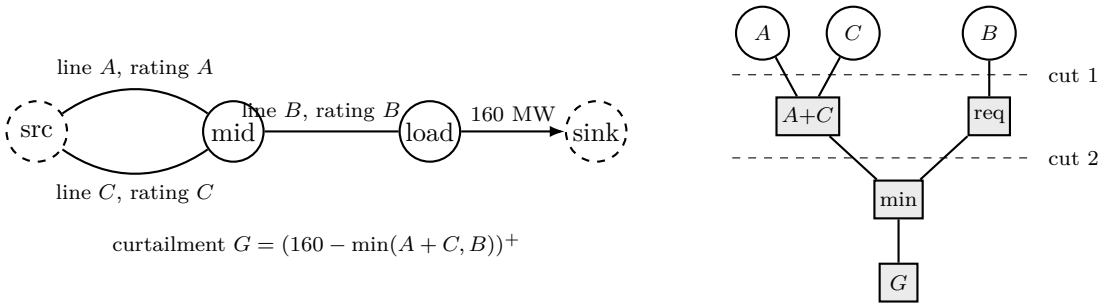


Figure 10.1: Left: the three-line network. Two lines in parallel feed a middle bus; one line feeds the load. Right: the dependency graph of the compiled model, inputs at the top and the curtailment at the bottom. Cut 1 separates the three ratings from everything below; cut 2 separates (A+C, req), two quantities, from the curtailment. Each cut is a separator in the sense of Chapter 5, and §10.6 counts what each saves.

10.2 The problem

Two lines, A and C, run in parallel from a source to a middle bus; a third line, B, runs from the middle bus to a load of 160 megawatts. Each line has a rating, the most it may carry. The ratings are uncertain and independent: A and C uniform on $[70, 120]$ megawatts, B uniform on $[140, 180]$. What can be delivered to the load is the smaller of what the parallel pair can carry, $A + C$, and what line B can carry; whatever the load needs beyond that is curtailed. The curtailment is

$$G \;=\; \big(160 - \min(A + C,\; B)\big)^+, \tag{10.1}$$

and because it has this closed form its expectation, variance and probability of being zero are known exactly, by the derivation in the solution to Exercise 10.1: $\mathbb{E}[G] = 16/3$ megawatts, $\mathrm{Var}[G] = 41.956$, $\mathbb{P}(G = 0) = 0.46$. The Shapley shares of the variance, a standard attribution of $\mathrm{Var}[G]$ to the three ratings that Chapter 14 defines, have no such short closed form. Their reference values are computed with the closed-form G on a grid of 241 points per rating, far finer than any grid scored below; on that grid the same sums return $\mathbb{E}[G]$ within 4×10^{-5} of $16/3$. Line B accounts for 93.6 percent and A and C for 3.2 each. Every method is scored against these values.

Figure 10.1 draws the network and the dependency graph of the model as compiled, which is the factor graph of the closures with the balance rows absorbed. Every number of the three-line comparison is read from the committed results of the study that ran it; the script `ch10_three_way.py` in the book's `scripts` directory reads them and prints them, and computes nothing itself. The checks of those numbers, and the second example of §10.7, are computed by `ch10_more.py`.

10.3 Enumeration: the product grid

Discretize each rating on n points and take the product: n^3 combinations, each with a known probability, each solved once. Expectation is the weighted sum of the n^3 curtailments; so is any other functional. This is enumeration, called product-grid quadrature when the inputs are continuous, and it is the exact marginalization of the discretized model. Table 10.1 gives its

Table 10.1: Product-grid quadrature on the three-line problem: grid size, solves, and the error of the mean, the variance, and line B's Shapley share against the closed form. The share costs no solves beyond the grid.

n	solves	$\mathbb{E}[G]$ error [MW]	$\mathrm{Var}[G]$ error	share B [%]	share error [pp]
9	729	0.045	2.47	92.47	1.12
17	4,913	0.012	0.60	93.30	0.29
25	15,625	0.0054	0.27	93.47	0.13
33	35,937	0.0026	0.16	93.52	0.075
41	68,921	0.0012	0.11	93.55	0.048

convergence. At $n = 9$, which is 729 solves, the mean is within 0.045 megawatts of the closed form; at $n = 41$, 68,921 solves, within 0.0012. The error falls roughly as n^{-2}, as quadrature of a piecewise-smooth integrand should, and the cost rises as n^3.

That the error falls as n^{-2} deserves a proof, because the familiar error bound of the trapezoid rule assumes a smooth integrand and G is not smooth: it bends where $A + C = B$, where $A + C = 160$ and where $B = 160$. The study's grid is the trapezoid rule on n evenly spaced points per rating, with step h and half weights at the two ends.

Proposition 10.1 (Trapezoid rule across kinks)**.** *Let f be continuous on $[a, b]$ and twice differentiable with $|f''| \le M$ except at K points, at each of which its slope jumps by at most J. The trapezoid rule with step h has error at most $(b-a)Mh^2/12 + KJh^2/8$. On a product grid in d coordinates, for a function whose one-dimensional sections all obey these bounds, the error is at most d times the one-dimensional bound.*

Proof. On a cell $[x_0, x_0+h]$ with no kink, the rule integrates the chord through the two end values, and f differs from its chord by $\frac{1}{2}f''(\xi)(x - x_0)(x - x_0 - h)$ for some ξ in the cell. Integrating over the cell gives an error of $h^3|f''(\xi)|/12$ at most, and the $(b-a)/h$ cells contribute at most $(b-a)Mh^2/12$. On a cell with a kink at $x_0 + s$, write $f = \ell + k$, where ℓ continues the left-hand piece smoothly across the kink and $k(x) = j\,(x - x_0 - s)^+$ carries the jump j in slope. The part ℓ is covered by the first bound. The part k has integral $j(h-s)^2/2$ over the cell and trapezoid value $\frac{h}{2}\,j(h-s)$, and their difference is $\frac{1}{2}js(h-s)$, at most $jh^2/8$, reached at $s = h/2$. There are K such cells. On a product grid the rule is the one-dimensional rule applied coordinate by coordinate, $Q_1 \cdots Q_d$ in place of $I_1 \cdots I_d$, and the difference telescopes: $Q_1 \cdots Q_d - I_1 \cdots I_d = \sum_k Q_1 \cdots Q_{k-1}(Q_k - I_k)I_{k+1} \cdots I_d$. Term k is the one-dimensional error for a function of x_k alone, the section integrated over the later coordinates, which has no more kinks or curvature than the sections themselves, averaged with the positive weights of the earlier coordinates, which sum to one. □

The curtailment is piecewise linear, so $M = 0$ and only the kink term remains: the error is of order h^2 with a constant set by the jumps in slope. The ratio of the error to h^2 bears this out. It is 0.0012, 0.0013 and 0.0008 per square megawatt at $n = 9$, 17 and 41. It does not settle to one constant, because the kinks fall at different places in their cells as n changes, and the $s(h-s)$ of the proof moves with them.

Table 10.2: Monte Carlo on the three-line problem. Top: the mean, rms error over eight seeds. Bottom: line B's Shapley share by pick-freeze, which needs six further sample sets, $7N$ solves in all, rms over five seeds.

N	solves	rms error of $\mathbb{E}[G]$ [MW]	rms error of $\mathrm{Var}[G]$
300	300	0.50	2.34
1,000	1,000	0.13	1.53
3,000	3,000	0.11	0.68
10,000	10,000	0.059	0.38
N	solves	share B [%]	rms error [pp]
300	2,100	91.9	7.05
1,000	7,000	96.2	3.11
3,000	21,000	87.9	2.64

10.4 Monte Carlo

Draw N triples of ratings, solve each, average. Table 10.2 gives the root-mean-square error of the mean over eight independent seeds: 0.50 megawatts at 300 draws, 0.059 at 10,000. The error falls as $N^{-1/2}$, as it must.

Proposition 10.2 (Monte Carlo error)**.** *Let $G_1, \dots, G_N$ be independent draws of a quantity with mean μ and variance σ^2. The sample mean $\bar{G}_N$ has expectation μ and root-mean-square error $\sigma/\sqrt{N}$, whatever the number of uncertain inputs behind each draw.*

Proof. $\mathbb{E}[\bar{G}_N] = \frac{1}{N}\sum_i \mathbb{E}[G_i] = \mu$. The variance of a sum of independent terms is the sum of their variances, so $\mathrm{Var}[\bar{G}_N] = N\sigma^2/N^2 = \sigma^2/N$, and the rms error of an unbiased estimator is the square root of its variance. Nothing in the argument refers to the inputs. □

The constant is the standard deviation of the quantity itself, $\sigma = \sqrt{41.956} = 6.48$ megawatts. The proposition predicts 0.374, 0.205, 0.118 and 0.065 megawatts at the four sample sizes of Table 10.2. The study measured 0.50, 0.13, 0.11 and 0.059, each an rms over eight seeds. An rms over eight seeds is itself uncertain by about a quarter, $1/\sqrt{2 \cdot 8}$, so the agreement is as close as eight seeds allow.

Set the two side by side on the mean. The grid at 729 solves has a smaller error than Monte Carlo at 10,000, and Proposition 10.2 says how many draws would match it: $(6.48/0.045)^2$, about 20,600, a factor of twenty-eight in solves. The two figures bound the same gap: at least fourteen as measured, since 729 grid solves beat 10,000 draws, and about twenty-eight by the $\sigma/\sqrt{N}$ law. At $n = 17$ the grid's error of 0.012 would take about 275,000 draws, fifty-six times its 4,913 solves. This is real, and it is not a property of graphical models. It is the usual advantage of quadrature over sampling in low dimension. In solves the grid's error falls as $(n^d)^{-2/d}$, the $-2/d$ power of the solves, against sampling's $-1/2$; the two exponents cross at $d = 4$, and beyond four inputs the grid loses for all but the smallest budgets. The book does not claim this factor as its own.

10.5 Attribution: where the structural difference appears

Now ask the second question: how much of the variance of G is due to each rating? Chapter 14 defines the Shapley share; for now it is enough that computing it needs conditional expectations of G given each subset of the inputs, $\mathbb{E}[G \mid A]$, $\mathbb{E}[G \mid A, B]$, and so on, for every subset.

On the grid, every such conditional expectation is a partial sum over solves already made: fix A at a grid value, sum over the B and C grid values with their weights. The joint distribution is held on a product set, and marginalizing over some coordinates is summing over them. The attribution costs no solves beyond the grid, and at $n = 17$, 4,913 solves, line B's share is within 0.29 percentage points of the reference.

Proposition 10.3 (Conditional expectations on a product grid)**.** *Let the inputs be independent and discretized on a product grid $x = (x_1, \dots, x_d)$ with weights $w(x) = \prod_i w_i(x_i)$, and let G be known at every grid point. For any subset u of the inputs, the conditional expectation of G given $X_u = x_u$ in the discretized model is*

$$\mathbb{E}[G \mid X_u = x_u] \;=\; \sum_{x_{-u}} \Big(\prod_{i \notin u} w_i(x_i) \Big)\, G(x_u, x_{-u}), \tag{10.2}$$

a partial sum over values already solved, and $\mathrm{Var}(\mathbb{E}[G \mid X_u])$ for all 2^d subsets needs no further solve.

Proof. The conditional probability of x_{-u} given x_u is $w(x) / \sum_{x'_{-u}} w(x_u, x'_{-u})$. The denominator is $\prod_{i \in u} w_i(x_i) \prod_{i \notin u} \sum_{x_i} w_i(x_i) = \prod_{i \in u} w_i(x_i)$, since each marginal sums to one, so the ratio is $\prod_{i \notin u} w_i(x_i)$. The variance of the conditional expectation is the sum over x_u of $\prod_{i \in u} w_i(x_i)$ times the square of (10.2), less the square of the mean. Every term is a value of G at a grid point. □

The proposition leans on the product form twice: the inputs are independent, and the grid is a product set. A cloud of random draws lacks the second property. No two draws share a value of A, so there is nothing to sum over with A held fixed, and a conditional expectation has to be manufactured by pairing draws, which is what pick-freeze does.

By sampling, a conditional expectation given a subset needs a sample designed for it. The standard construction, pick-freeze, draws a second independent sample and recombines the two so that some inputs are held fixed across a pair of draws; for three inputs it needs six further sample sets, $7N$ solves in all. At $N = 3{,}000$, which is 21,000 solves, line B's share is 2.64 points off, and the error is not falling smoothly with N because the estimator's variance is large. To match the grid's 0.29 points would take, on the $N^{-1/2}$ law, some eighty times the samples.

Figure 10.2 draws all four curves on one plot of error against solves. The two mean curves are the quadrature-versus-sampling story: parallel on the log-log plot, offset by that factor. The two attribution curves are the structural story: the grid's attribution error is its mean error's twin, at the same solves, because it costs no more; the sampling attribution sits an order of magnitude above the sampling mean, because every conditional expectation is a new sample.

> **Principle 10.1.** *The factorized route does not win by avoiding a large state space. It wins when a computation held on a product set answers the second question from the solves that answered the first. Its advantage is reuse, and it is measured in solves per question, not in solves per answer.*

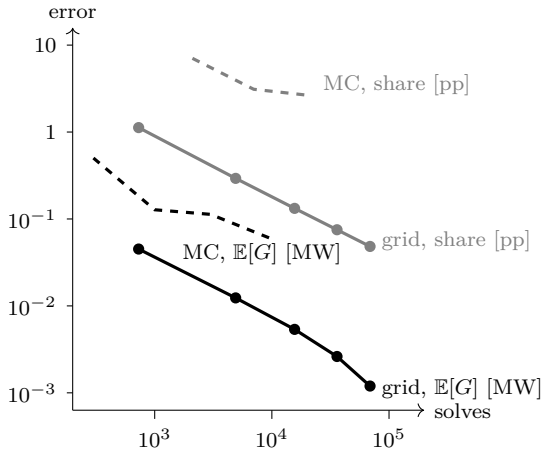


Figure 10.2: Error against solves on the three-line problem, from the committed study results. Solid: product grid; dashed: Monte Carlo. Black: error of the mean in megawatts; grey: error of line B's Shapley share in percentage points. The grid's attribution rides on its mean because it costs no further solves; the sampling attribution needs $7N$ solves and its error is an order of magnitude above the sampling mean's.

10.6 The factorized route: cuts of the dependency graph

The grid of §10.3 enumerated n^3 combinations of the ratings. It did not have to. The dependency graph of Figure 10.1 shows that the curtailment depends on the ratings only through two intermediate quantities, $A+C$ and the requirement at the middle bus, and on those only through their minimum. Each level of the graph is a separator in the sense of Chapter 5: given the quantities on a cut, what lies below is independent of what lies above.

Enumerate on a cut instead of on the inputs. The pair $(A+C,\ B)$ takes $(2n-1)\,n$ distinct values on the grid, not n^3, because $A+C$ takes $2n-1$ distinct values and each carries a known probability from the convolution of A's and C's. Solving the physics once per distinct value of the cut and weighting by the cut's distribution gives the same expectation as the full grid, exactly, because the expectation of a function of the cut is the same whether one sums over the inputs or over the cut with its induced distribution. Table 10.3 gives the ladder of cuts and the distinct solves each needs. At $n = 17$ the cut $(A+C,\ B)$ needs 561 solves against the grid's 4,913, and the study verified that the two means agree to 10^{-13} megawatts. The cut below it, the single quantity that G depends on, needs fewer than n, 33 at $n = 41$: the whole problem is one-dimensional at that level, and G's distribution is the pushforward of a one-dimensional distribution through a scalar function.

Proposition 10.4 (Enumeration on a cut)**.** *Let X take values on a finite grid with probabilities $p(x)$, and let the quantity of interest depend on X only through a cut $Z = c(X)$, so that $G = h(Z)$. Then for any function ϕ*

$$\mathbb{E}[\phi(G)] \;=\; \sum_{z} \phi(h(z))\, q(z), \qquad q(z) = \sum_{x:\, c(x)=z} p(x), \tag{10.3}$$

which needs one evaluation of h per distinct value of the cut on the grid. If a component of the cut is a sum of independent inputs, its distribution is the convolution of theirs.

Table 10.3: The cut ladder of the three-line model: for each separator of the dependency graph, its dimension and the number of distinct physics solves needed to compute $\mathbb{E}[G]$ exactly on an n-point grid. The bottom row is the full product grid.

cut (frontier)	dim	$n = 9$	$n = 17$	$n = 25$	$n = 33$	$n = 41$
$\{f_B\}$: what G reads	1	8	14	20	26	33
$\{A{+}C,\ \text{req}\}$	2	85	297	637	1,105	1,701
$\{A{+}C,\ B\}$	2	153	561	1,225	2,145	3,321
$\{A,\ C,\ \text{req}\}$	3	405	2,601	8,125	18,513	35,301
$\{A,\ B,\ C\}$: the product grid	3	729	4,913	15,625	35,937	68,921

Proof. Group the terms of $\sum_x \phi(h(c(x)))\, p(x)$ by the value of $c(x)$. Within a group the factor $\phi(h(z))$ is common and the probabilities add to $q(z)$. For a component $Z_1 = X_1 + X_2$ with X_1 and X_2 independent, $q_1(z_1) = \sum_{x_1} p_1(x_1)\, p_2(z_1 - x_1)$, which is the convolution. □

On the three-line grid with trapezoid weights, $n = 5$ puts weights $(2, 4, 4, 4, 2)/16$ on the ratings of A and of C. The convolution puts $(4, 16, 32, 48, 56, 48, 32, 16, 4)/256$ on the nine values of $A + C$, a discrete triangle, which is what the sum of two uniforms is. Recomputing the grid means through the cut reproduces the full-grid means to 2×10^{-15} at every n of Table 10.1. The evaluations of h are the physics. The convolution is arithmetic on weights and costs no solve, because in this model two parallel lines of ratings A and C act on the rest of the network exactly as one line of rating $A + C$ does, which is what equation (10.1) says.

This is Chapter 5's count made physical. The exponent of the cost is the dimension of the cut, not the number of inputs; the cut is a separator of the model's own graph; and the model exposes it, because the compiled dependency graph is the factor graph with the balances folded in. Nothing was approximated, and every functional of G, not just its mean, comes from the same 561 solves, since they determine G's distribution.

Two honest qualifications. A sampler can exploit the same cut: draw $A + C$ from its convolution instead of drawing A and C, and the draws of the two inputs collapse to one. The error still falls as $N^{-1/2}$, so the cut helps the sampler less than it helps the grid, but it helps. And a cut of dimension two on a three-input problem is a modest saving; the ladder matters because on a large model the cuts are the ports of Chapter 8, their dimension is the port count, and the input count is in the hundreds.

10.7 Second worked example: two generation areas

A load pocket draws 2,100 megawatts, supplied by two generation areas. Each area has three units in parallel, each rated between 350 and 450 megawatts, uniform and independent. The units are gas turbines, whose output falls as the air they breathe warms, here by 1.5 percent per degree above 18°C. The ambient temperature T is uncertain, uniform on $[18, 28]$°C, and it is the same for both areas, which sit in one weather region. Each area delivers through its own export path, whose rating L_k is uncertain, uniform on [1,056, 1,257] megawatts. An area delivers the smaller of its derated unit capacity and its export rating, and the pocket's balance leaves the

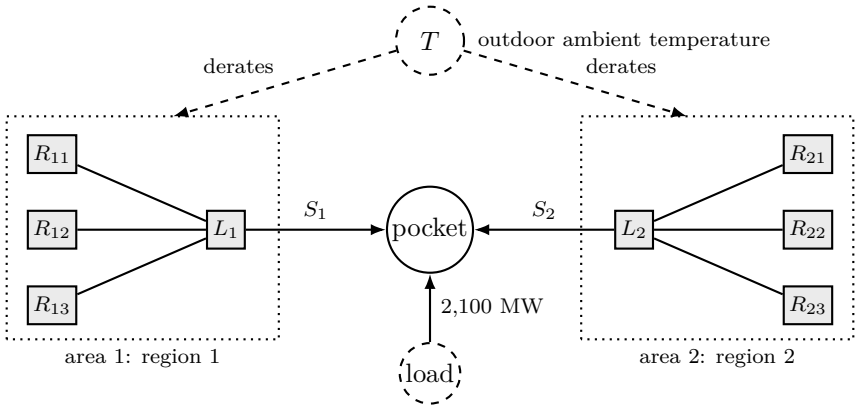


Figure 10.3: Two generation areas serving one load pocket. Each area has three units in parallel, ratings R_{kj}, and an export path whose rating L_k caps what it delivers; S_k is the area's delivered power and its port to the pocket. The ambient temperature T derates every unit in both areas. Given T the areas are independent, so the separator between them is $\{T, S_1, S_2\}$, not the ports alone.

rest unserved:

$$S_k = \min\Big(\big(1-\alpha(T-18)\big)\sum_{j=1}^{3} R_{kj},\ L_k\Big), \qquad U = \big(2{,}100 - S_1 - S_2\big)^+. \tag{10.4}$$

The unserved load U is demand the pocket cannot meet, and it shows up as load shedding. There are nine uncertain inputs: six ratings, two export ratings and the ambient temperature. Figure 10.3 draws the areas and the cut.

The dependency graph has the shape of the three-line model's, twice over, with one node more. Inside each area the unit ratings enter only through their sum, a cut of one dimension. The area's delivered power is its port to the pocket, and the ambient temperature feeds both areas. Given the ambient temperature the areas are independent, and each area's message to the pocket is the distribution of its delivered power.

Proposition 10.5 (Region messages with a shared input)**.** *Let the inputs be (W, X_1, X_2), mutually independent, let region k's port be $S_k = s_k(W, X_k)$, and let $G = h(S_1, S_2)$. Then*

$$\mathbb{E}[\phi(G)] \;=\; \sum_{w} p(w) \sum_{s_1, s_2} \phi\big(h(s_1,s_2)\big)\, q_1(s_1 \mid w)\, q_2(s_2 \mid w), \qquad q_k(s \mid w) = \sum_{x_k:\ s_k(w,x_k)=s} p_k(x_k). \tag{10.5}$$

The cost is one region solve per distinct value of each region's port for each value of W, and the combination is an evaluation of h. If the two regions are identical in law, $\mathrm{Cov}(S_1, S_2) = \mathrm{Var}(\mathbb{E}[S_1 \mid W])$*. The product of the marginal messages, $q_1(s_1)\, q_2(s_2)$, which is what the computation gives when W is left out of the separator, implies a covariance of zero, and is wrong whenever $\mathbb{E}[S_1 \mid W]$ varies with W.*

Proof. Condition on $W = w$. The inputs X_1 and X_2 are independent of each other and of W, so given $W = w$ the ports $s_1(w, X_1)$ and $s_2(w, X_2)$ are functions of independent variables and are independent. Their distributions are $q_1(\cdot \mid w)$ and $q_2(\cdot \mid w)$ by Proposition 10.4 applied inside each region, which gives $\mathbb{E}[\phi(G) \mid W = w]$ as the inner double sum; averaging over w

Table 10.4: The two generation areas. For each grid size: the solves of the full product grid on nine inputs; the area solves of the region route of Proposition 10.5; the errors of $\mathbb{E}[U]$ and $\mathbb{P}(U > 0)$ against the reference; and the rms error of Monte Carlo on $\mathbb{E}[U]$ at as many full solves as the region route's area solves, by Proposition 10.2.

n	full grid	area solves	$\mathbb{E}[U]$ error [MW]	$\mathbb{P}(U > 0)$ error	Monte Carlo rms [MW]
5	2.0×10^6	650	2.34	0.0118	1.08
9	3.9×10^8	4,050	0.593	0.0068	0.433
17	1.2×10^{11}	28,322	0.146	0.0011	0.164
33	4.6×10^{13}	211,266	0.035	0.0007	0.060

gives (10.5). For the covariance, the law of total covariance gives $\mathrm{Cov}(S_1, S_2) = \mathbb{E}[\mathrm{Cov}(S_1, S_2 \mid W)] + \mathrm{Cov}(\mathbb{E}[S_1 \mid W], \mathbb{E}[S_2 \mid W])$. The first term is zero by conditional independence. When the regions are identical in law, $\mathbb{E}[S_1 \mid W] = \mathbb{E}[S_2 \mid W]$, and the second term is the variance of that function of W. □

Table 10.4 compares the routes. The full product grid on nine inputs is n^9 solves, 3.9×10^8 at $n = 9$ and 1.2×10^{11} at $n = 17$, and nobody runs it. Proposition 10.5 replaces it by $2n^2(3n - 2)$ area solves, 4,050 at $n = 9$. For each of the n ambient values each area is enumerated on its own cut: $3n - 2$ values of the rating sum, from two convolutions, times n export ratings. The combination $U = (2{,}100 - S_1 - S_2)^+$ is the pocket's energy balance, evaluated by sorting one message against the other, with no physics in it. At $n = 5$ the full grid is still affordable, 1,953,125 solves, and it gives $\mathbb{E}[U] = 12.588677026$ megawatts; the region route gives the same number from 650 area solves, to 2.5×10^{-14}. The reference is a region computation at $n = 161$: $\mathbb{E}[U] = 10.248$ megawatts, $\mathbb{P}(U > 0) = 0.195$ and a standard deviation of 27.6 megawatts. Twenty million Monte Carlo draws agree with it to 0.008 megawatts.

Read the last two columns of the table together, and the region route does not beat sampling on the mean. At $n = 9$ its error of 0.59 megawatts is what Monte Carlo reaches in about 2,200 draws. At $n = 17$ its 0.146 would take Monte Carlo about 36,000 draws, against 28,322 area solves, and an area solve is about half a full solve. The probability of a shortfall tells the same story: at 4,050 draws its Monte Carlo standard error is 0.0062, against the region route's 0.0068. The reason is Proposition 10.1 in five effective dimensions, the ambient temperature and two per area once the ratings are summed. That is past the crossover at four, and the unserved load is a tail quantity, zero over four fifths of the input space and bending sharply at its edge. The error is spread over the directions: nine ambient nodes with 33 points on the other inputs still leave 0.40 megawatts, and 33 ambient nodes with nine points on the others leave 0.23. The cut made the grid possible at all, 4,050 solves in place of 3.9×10^8. It did not make the grid better than sampling for one number. That is the three-line lesson read the other way: quadrature's advantage in three dimensions was not the book's, and its absence in five is not the book's failure.

Now ask the question a site engineer asks: how bad is it on a hot day? The region computation already holds the answer. Its outer sum runs over the ambient temperature, and before that sum is taken each term is $\mathbb{E}[U \mid T = w]$, Proposition 10.3 with $u = \{T\}$. Figure 10.4 draws it. Below 21°C the areas never fall short. At 24.25°C the expected shortfall is 4.5 megawatts with a 15 percent chance of any; at 28°C it is 66 megawatts with an 84 percent chance. The nine values cost nothing beyond the 4,050 area solves, and they are within 0.27 megawatts rms of the fine computation. A Monte Carlo estimate of the same curve sorts its draws into nine ambient bins

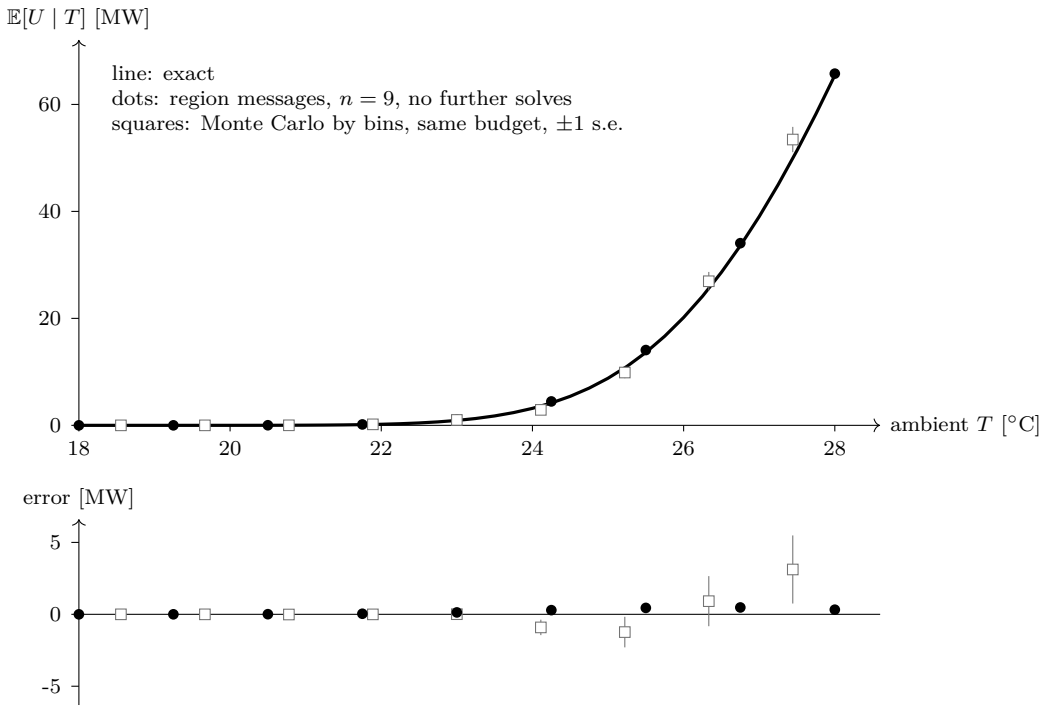


Figure 10.4: The expected unserved load given the outdoor ambient temperature, for the two generation areas. Top: the line is exact, from a region computation at $n = 33$. The dots are the nine values held by the region computation at $n = 9$, which cost nothing beyond its 4,050 area solves. The squares are Monte Carlo at the same 4,050 draws sorted into nine ambient bins, with one standard error, for one seed of the eight behind the rms error quoted in the text. Bottom: the errors of both against the exact curve, on a scale ten times finer.

and averages each bin. At the same 4,050 draws each bin holds about 450, and the rms error of the nine bin means over eight seeds is 0.97 megawatts, three and a half times the region route's. Since the error falls as the inverse square root of the draws, matching the region route would take about thirteen times the solves. The weather alone accounts for 39 percent of the variance of U, a first-order attribution in the sense of Chapter 14, and it is again a partial sum.

Tails are a harder test. The probability that the shortfall exceeds t is $\mathbb{P}(S_1 + S_2 < 2{,}100 - t)$: the same messages with the balance evaluated at a lower load, so it too costs no further solve. Its accuracy is another matter. At $t = 100$ megawatts the reference probability is 0.0269. The region route gives 0.0325 at $n = 9$ and 0.0281 at $n = 17$, errors of 21 and 4 percent, while Monte Carlo at 4,050 draws has a relative standard error of 9.5 percent. At $t = 150$ megawatts, a probability of 0.005, the region route is off by 25 and 16 percent and Monte Carlo by 22 percent at 4,050 draws and 8 percent at 28,322. On a tail probability the grid does no better than sampling, and the reason is Proposition 10.1 again, this time by its failure. A probability is the expectation of an indicator, which jumps rather than bends. On a cell containing a jump the trapezoid rule is off by up to half the jump times h, so the error is of order h, not h^2. The remedy is to integrate one input in closed form inside the last region, which turns the jump back into a kink (Exercise 10.7).

Last, the separator. Leave the ambient temperature out of it: compute each area's message averaged over the weather, and combine the two as independent regions. Every number comes

out too small. At $n = 33$ the expected shortfall is 5.8 megawatts against 10.3, the probability of one is 0.145 against 0.195, and the standard deviation is 19.1 against 27.6. Proposition 10.5 says why. One area's delivered power has mean 1,092 megawatts and standard deviation 56.0; the part of its variance that the ambient temperature explains is 1,324 square megawatts, so the two areas' deliveries are correlated at 0.42. The standard deviation of their sum is 94.5 megawatts with the shared ambient temperature and 79.2 without it. The unserved load is the tail of that sum below 2,100 megawatts, so a thinner tail means a smaller shortfall. A hot afternoon derates both areas at once. A model that lets one area's bad day be offset by the other's good one has forgotten that there is only one afternoon.

> **Principle 10.2.** *A common cause belongs in the separator. An input that feeds two regions, such as the weather, a shared supply or a common-mode failure, must be conditioned on before the regions' messages are combined. Combining their marginal messages treats the regions as independent and understates every joint risk.*

10.8 Beyond three inputs

The product grid dies at n^d. With fourteen uncertain inputs and nine points each it is 2×10^{13} solves, and there is no cut of dimension two to save it. What survives is the structure, and it survives in two forms.

The first is Chapter 9's simulation inside a region: sample the inputs of each region, solve each region on its own variables, and combine the regions' particle messages exactly on their ports. The samples are per region, the combination is on the separator, and the solves are of regions rather than of the whole. Chapter 18 measures what that saves on a real network.

The second is to make each solve return more than a number. When the physics is solved once and returns, along with the curtailment, its gradient with respect to every uncertain input, the solves form a census of value-and-gradient pairs from which many questions can be answered without further solves. Chapter 14 develops this, and Chapter 21 runs it on a network with 73 buses and 14 uncertain line ratings, every rating set at 50 percent of its nominal value so that the network is stressed, and its committed results give the count. A census of 3,000 solves, certified on 2,000 further holdout draws, plus 512 for the mean and 512 for each of two questions that needed a small residual sample, answered the mean and nine planning questions: the marginal value of each line's capacity, the attribution of variance to each rating, the value of a measurement, a screening of which ratings matter, the mean under a changed weather belief, the mean under correlated ratings, a what-if, the tail, and the worst case. The total was 6,536 solves. The same nine questions answered by the standard sampling method for each cost 113,876; each has its own sample, except the value of a measurement, which reads the attribution's sample at no further solves. Seven of the nine cost the census route nothing at all beyond the census. That is Principle 10.1 at scale: one computation held in a form that the next question can read.

10.9 The claim

There are three claims one could make for the factorized route, in increasing strength, and the book makes the third.

Weak. The factorized route handles combinatorial uncertainty without enumerating every state.
True, and worthless: sampling does too.

Strong. The factorized route exploits the physical factorization and its separators to reuse expensive physics solves across many uncertain states and many queries. True, and established in this chapter: the cut ladder reuses solves across states, the product set reuses them across queries, and the census reuses them across nine questions.

Strongest. For equivalent accuracy, the number of expensive physics solves the factorized route needs is much smaller than the number sampling needs, and the ratio grows with the number of questions asked. Measured: about twenty-eight to one for one question in three dimensions, which is quadrature's doing and which disappears on the nine inputs of the generation areas; no further solves against $6N$ for the second question; about thirteen to one for the areas' shortfall given the weather; seventeen to one over the mean and nine questions on a real network. The last three are reuse's doing.

The strongest claim carries its conditions. The reuse across states needs a cut of small dimension, which a physical system usually has and a generic function of many inputs usually does not. The reuse across queries needs the computation to be held in a form the next query can read, a product set, a region message, a census with gradients, rather than a single number. And the comparison is in solves; where solves are cheap, sampling's simplicity may be worth more than its solve count. Where they are not, which is every network of operational size, the count is what matters, and it is the count the book keeps.

10.10 In practice

Count solves, at a stated accuracy. Wall-clock time depends on the machine and the implementation, and state counts depend on the discretization. The number of physics solves needed to reach a stated error on a stated question is the comparison that transfers. Plot error against solves, as Figure 10.2 does, not error against grid size or sample size.

Read the cuts from the compiled graph. The dependency graph of the compiled model records what each quantity reads. Walk it upward from the quantity of interest and note the frontier at each level. The smallest frontier is the cheapest exact enumeration, and a frontier component that is a sum of independent inputs is a convolution, which costs no solve.

Put every shared input in the separator. Before combining region messages, list the inputs that feed more than one region: weather, a shared supply voltage or temperature, a common maintenance crew, a common-mode failure. Condition on each. The test is mechanical: an input with edges into two regions is a separator variable, whatever its physical scale.

Check exactness once, on a grid small enough to enumerate. The generation-area example checked the region route against 1.95 million full solves at $n = 5$ before trusting it at $n = 33$. The check is cheap at small n and catches a misassigned input, which is the usual error.

Match the route to the question. For one mean in many effective dimensions, sampling is simple and competitive, and it should be used. For conditional means, attributions and many questions of one model, hold the computation in a form the next question can read: the product set, the region messages, the census.

Refine where the error is. A grid's error has a part per coordinate, and refining one coordinate costs in proportion to its nodes, not to the whole grid. In the region route the outer coordinate is the cheapest to refine: refining only the ambient temperature, from nine nodes to 33, cuts the areas' error of the mean from 0.59 to 0.23 megawatts at 14,850 area solves.

10.11 What is missing

Every query in this chapter was a forward one: uncertain inputs pushed through the physics to a consequence. Chapter 11 turns to the inverse direction under operation, where readings arrive one at a time and the question is how cheaply each can be folded into the posterior; the answer, a rank-one update, is the cheapest reuse in the book. Part IV then asks the questions an operator asks, and Chapter 14 in particular develops the census with gradients that §10.8 previewed.

Notes and further reading

The three-line comparison and the cut ladder are the case study of Chapter 20, and this chapter's numbers are its committed results read verbatim. The advantage of quadrature over sampling in low dimension, and the curse of dimensionality that ends it, are standard; Robert and Casella [76] treat both sides. Variance-based attribution and the Shapley share are Sobol's [84], Owen's [68] and Song, Nelson and Staum's [85]; the pick-freeze estimator and its $(d+4)N$ or $7N$ solve count for $d = 3$ are Saltelli's [80]. That every conditional expectation is a partial sum on a product grid is Fubini's theorem and needs no citation; that the model's compiled dependency graph exposes the cuts on which the sum can be taken is the framework's. The generation areas of §10.7 were built for this chapter; their constants are typical of a mid-size data-center area rather than taken from one. The law of total covariance behind Proposition 10.5 and the trapezoid error bound behind Proposition 10.1 are standard, and both are proved on the page.

The three-strength statement of the claim in §10.9 is the book's, and it was arrived at by conceding the weak claim to sampling first. The reader who has met the literature on "probabilistic power flow" will recognize the weak claim as the one usually made there, and the strong one as the one that point-estimate, cumulant and surrogate methods make implicitly; the census with gradients of §10.8 is a surrogate method, and Chapter 14 says which.

Summary

- Sampling answers every question and does not care about the size of the state space; what it pays for is the solve. The comparison is in solves per question at equal accuracy.
- On three uncertain ratings, the product grid reaches at 729 solves an accuracy that sampling needs about 20,700 draws for: quadrature's advantage in low dimension, not the book's.
- The trapezoid grid's error is of order h^2 even across kinks, and Monte Carlo's is $\sigma/\sqrt{N}$ whatever the dimension. In solves the grid's exponent is $2/d$ against sampling's $1/2$.
- The attribution costs the grid nothing beyond the grid, because conditional expectations are partial sums on a product set; sampling needs $7N$ solves and is an order of magnitude less accurate at 21,000.
- The model's dependency graph has cuts; enumerating on the cut $(A{+}C, B)$ needs 561 solves for the same mean to 10^{-13} as the 4,913-solve grid. The exponent is the cut's dimension.

- Beyond three inputs the grid dies but the structure survives, as region messages and as a census with gradients: 6,536 solves for the mean and nine questions against 113,876 on a 73-bus network.
- On two generation areas with nine inputs, region messages conditioned on the shared ambient temperature replace n^9 solves by $2n^2(3n-2)$, exactly. They match sampling on the mean and beat it by about thirteen in solves on the shortfall given the weather.
- A common cause belongs in the separator: combining the areas' marginal messages understates the expected shortfall by more than forty percent.
- The book's claim is the strongest one, with its conditions: a small cut, a reusable form, and expensive solves.

Exercises

Exercise 10.1. Derive the closed form of $\mathbb{E}[G]$ for the three-line problem from equation (10.1) and the uniform priors, and confirm 16/3. Then derive $\mathbb{P}(G = 0)$ and confirm 0.46.

Exercise 10.2. Show that on an n-point uniform grid for A and for C, the sum $A + C$ takes $2n - 1$ distinct values, and derive their probabilities. Confirm the $(2n-1)\,n$ entries of the third row of Table 10.3.

Exercise 10.3. Implement pick-freeze for the three-line problem and reproduce the $7N$ solve count. Then implement the sampler that draws $A + C$ from its convolution and show how many of the seven sample sets it saves.

Exercise 10.4. The grid's mean error falls as n^{-2} and Monte Carlo's as $N^{-1/2}$. For d uncertain inputs the grid costs n^d solves. Find the dimension at which, for an error of 0.01 megawatts on a problem with the same constants as this one, sampling first needs fewer solves than the product grid.

Exercise 10.5. For the two-region network of Chapter 8 with all five loads uncertain, list the cuts of its dependency graph in nodal form, their dimensions, and the distinct solves each needs on an n-point grid. Which cut is the port pair, and what does enumerating on it cost?

Exercise 10.6. For the generation areas of §10.7, derive $\mathbb{E}[S_k \mid T]$ in closed form when the export limit never binds, and from it $\mathrm{Var}(\mathbb{E}[S_k \mid T])$. Compare with the 1,324 square megawatts computed with the export limit, and explain the sign of the difference.

Exercise 10.7. In the region computation for $\mathbb{P}(U > t)$, replace area 2's grid over its export rating by the exact uniform distribution, so that given the ambient, area 2's rating sum and area 1's delivery, the probability of a shortfall above t is a continuous, piecewise-linear function of the remaining inputs. Show that the error in $\mathbb{P}(U > 100)$ then falls as h^2, and find the n at which it beats Monte Carlo at the same number of area solves.

Solutions to selected exercises

Exercise 10.1. Write $S = A + C$. Because S and B are both at least 140, $\min(S, B) \geq 140$ and G never exceeds 20. For $0 \leq t < 20$, $G > t$ exactly when $\min(S, B) < 160 - t$, which is the complement of the event that both $S \geq 160 - t$ and $B \geq 160 - t$. The ratings are independent, so

$$\mathbb{P}(G > t) = 1 - \mathbb{P}(S \geq 160 - t)\,\mathbb{P}(B \geq 160 - t).$$

B is uniform on $[140,180]$, so $\mathbb{P}(B \ge 160 - t) = (20+t)/40$. The sum of two independent uniforms on $[70,120]$ has the triangular density $(s-140)/2500$ on $[140,190]$, so $\mathbb{P}(S < 160 - t) = (20-t)^2/5000$. The mean of a nonnegative variable is the integral of its tail:

$$\mathbb{E}[G] = \int_0^{20} \Big[1 - \frac{20+t}{40} + \frac{(20-t)^2(20+t)}{200{,}000}\Big]\,dt = 20 - 15 + \tfrac{1}{3} = \tfrac{16}{3},$$

the last integral by $u = 20 - t$: $\int_0^{20} u^2(40-u)\,du/200{,}000 = (320{,}000/3 - 40{,}000)/200{,}000 = 1/3$. At $t = 0$ the tail is $1 - 0.92 \times 0.5 = 0.54$, so $\mathbb{P}(G = 0) = 0.46$. The same route gives the second moment, $\mathbb{E}[G^2] = \int_0^{20} 2t\,\mathbb{P}(G > t)\,dt = 400 - 1000/3 + 56/15 = 70.4$, and the variance $70.4 - (16/3)^2 = 41.956$ quoted in §10.2.

Exercise 10.2. On the n-point grid the ratings of A and C take the values $70 + ih$, $i = 0,\dots,n-1$, with $h = 50/(n-1)$. A sum $a_i + c_j = 140 + (i+j)h$ depends only on $i+j$, which runs over $0,\dots,2n-2$: $2n-1$ distinct values. By Proposition 10.4 the probability of the k-th is $\sum_{i+j=k} w_i w_j$, the discrete convolution of the weight vector with itself. With the trapezoid weights, $w_0 = w_{n-1} = 1/(2(n-1))$ and $w_i = 1/(n-1)$ otherwise, the result is a discrete triangle with ramps at its ends, $(4, 16, 32, 48, 56, 48, 32, 16, 4)/256$ at $n = 5$. The rating of B takes n values independently of $A + C$, so the pair $(A + C, B)$ takes $(2n-1)n$ values: 153, 561, 1,225, 2,145 and 3,321 at $n = 9$, 17, 25, 33 and 41, the third row of Table 10.3. Computing $\mathbb{E}[G]$ on those pairs, with the convolved weights times B's weights, reproduces the full-grid means of Table 10.1 to 2×10^{-15}.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

One expectation, two ways of getting it *exact marginalization against sampling; quadrature error falling with grid spacing while sampling error falls with the root of the sample count; a realized error against a standard error*
`foundation_power_flow/terra_vs_monte_carlo_three_line/problems/three_routes_one_expectation`

Attribution for free *Shapley attribution of a variance; a conditional expectation as a partial sum over a product grid; re-using solves rather than repeating them*
`foundation_power_flow/terra_vs_monte_carlo_three_line/problems/attribution_for_free`

Chapter 11

Sequential evidence

Every posterior so far was computed with all the readings in hand. In operation they are not. A sensor reports, then another, then the first again a minute later, and the question after each is what the system now believes. Re-solving the whole model after every reading is correct and wasteful: the reading is one row, and one row changes the posterior by a rank-one term. This chapter shows that folding a reading into a Gaussian posterior costs one solve against a factorization already held, that the same solve yields the number by which the reading surprised the model, that the value of a reading can be computed before it is taken, and that at the scale of a real model the update is tens of times cheaper than a re-solve. It closes Part III.

11.1 Readings arrive one at a time

Take the posterior of a linear-Gaussian branch after the readings so far: precision $\mathbf{\Lambda}$, information $\boldsymbol{\eta}$, mean $\boldsymbol{\mu} = \mathbf{\Lambda}^{-1}\boldsymbol{\eta}$, covariance $\mathbf{\Sigma} = \mathbf{\Lambda}^{-1}$, with $\mathbf{\Lambda}$ held as a sparse factorization. A new reading y of the linear combination $\mathbf{h}^{\mathsf{T}}\mathbf{z}$ with accuracy σ_y is one more row of $\mathbf{g}$, and by §6.1 it adds $w\,\mathbf{h}\mathbf{h}^{\mathsf{T}}$ to the precision and $w\,y\,\mathbf{h}$ to the information, $w = 1/\sigma_y^2$. For a sensor on one variable, $\mathbf{h}$ is a unit vector and the change is one diagonal entry. The new posterior could be had by refactoring the new precision. It need not be.

11.2 One reading, one rank-one update

Proposition 11.1 (Rank-one conditioning)**.** *Let* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ *and let* $y = \mathbf{h}^{\mathsf{T}}\mathbf{z} + \nu$ *with* $\nu \sim \mathcal{N}(0, \sigma_y^2)$ *independent of* $\mathbf{z}$*. Then* $\mathbf{z} \mid y$ *is Gaussian with*

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{k}\,(y - \mathbf{h}^{\mathsf{T}}\boldsymbol{\mu}), \qquad \mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{k}\,\mathbf{s}^{\mathsf{T}}, \qquad \mathbf{s} = \mathbf{\Sigma}\mathbf{h}, \quad \mathbf{k} = \frac{\mathbf{s}}{\sigma_y^2 + \mathbf{h}^{\mathsf{T}}\mathbf{s}}. \tag{11.1}$$

Proof. In information form the reading adds $w\mathbf{h}\mathbf{h}^{\mathsf{T}}$ to the precision and $wy\mathbf{h}$ to the information, $w = 1/\sigma_y^2$. Write $S = \sigma_y^2 + \mathbf{h}^{\mathsf{T}}\mathbf{s}$. The claim for the covariance is that $\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{s}\mathbf{s}^{\mathsf{T}}/S$ inverts $\mathbf{\Lambda}' = \mathbf{\Lambda} + w\mathbf{h}\mathbf{h}^{\mathsf{T}}$. Multiply, using $\mathbf{\Lambda}\mathbf{\Sigma} = \mathbf{I}$ and $\mathbf{\Lambda}\mathbf{s} = \mathbf{h}$:

$$\mathbf{\Lambda}'\mathbf{\Sigma}' \;=\; \mathbf{I} + w\,\mathbf{h}\mathbf{s}^{\mathsf{T}} - \frac{(1 + w\,\mathbf{h}^{\mathsf{T}}\mathbf{s})}{S}\,\mathbf{h}\mathbf{s}^{\mathsf{T}} \;=\; \mathbf{I} + w\,\mathbf{h}\mathbf{s}^{\mathsf{T}} - w\,\mathbf{h}\mathbf{s}^{\mathsf{T}} \;=\; \mathbf{I},$$

since $(1 + w\,\mathbf{h}^{\mathsf{T}}\mathbf{s})/S = (\sigma_y^2 + \mathbf{h}^{\mathsf{T}}\mathbf{s})/(\sigma_y^2 S) = w$. This is the Sherman–Morrison identity. For the mean, $\boldsymbol{\mu}' = \mathbf{\Sigma}'(\boldsymbol{\eta} + wy\mathbf{h})$. The first part is $\mathbf{\Sigma}'\boldsymbol{\eta} = \boldsymbol{\mu} - \mathbf{s}\,(\mathbf{h}^{\mathsf{T}}\boldsymbol{\mu})/S$. The second uses

$\boldsymbol{\Sigma}'\mathbf{h} = \mathbf{s} - \mathbf{s}\,(\mathbf{h}^\mathsf{T}\mathbf{s})/S = \mathbf{s}\,\sigma_y^2/S$, so $wy\,\boldsymbol{\Sigma}'\mathbf{h} = y\,\mathbf{s}/S$. Adding, $\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{s}\,(y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu})/S$, which is $\boldsymbol{\mu} + \mathbf{k}\,(y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu})$. □

Three things about equation (11.1) carry the chapter.

The only expensive object is $\mathbf{s} = \boldsymbol{\Sigma}\mathbf{h}$, and it is not computed by forming $\boldsymbol{\Sigma}$. It is the solution of $\boldsymbol{\Lambda}\mathbf{s} = \mathbf{h}$ against the factorization already held: two triangular solves, the same operation as one selected inversion in §6.3. For a sensor on a single variable, $\mathbf{s}$ is one column of the posterior covariance, the covariance of that variable with every other, which is what a reading of that variable needs to know in order to move every other.

The vector $\mathbf{k}$ is the *gain*. It says how far each variable moves per unit of surprise, and it is the covariance column scaled by the total uncertainty of the reading, prior plus sensor. A variable tightly correlated with the measured one moves a lot; an independent one does not move.

And the covariance update is a subtraction of a rank-one matrix that is never formed. Keep $\boldsymbol{\Sigma}$ implicit: hold the factorization of the original $\boldsymbol{\Lambda}$ and a list of the pairs $(\mathbf{k}_i, \mathbf{s}_i)$ from the updates so far. Then any product $\boldsymbol{\Sigma}'\mathbf{v}$ is a solve against the factorization followed by a subtraction per pair, and any marginal variance is a solve followed by a few inner products. After r readings each solve costs r extra inner products, which is nothing until r approaches the ratio of a factorization's cost to a solve's, at which point one refactors and empties the list.

Principle 11.1. *A reading is a rank-one change to the posterior. Folding it in costs one solve against the factorization already held, and the same solve yields the covariance column the reading needed and the gain it produced. Nothing is refactored.*

11.3 The innovation

The scalar $y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}$ in equation (11.1) is the *innovation*: the reading minus what the model predicted for it. Its predicted spread, $\sqrt{\sigma_y^2 + \mathbf{h}^\mathsf{T}\mathbf{s}}$, is the sensor's accuracy combined with the posterior's own uncertainty about the measured quantity, and the ratio of the two, the *standardized innovation*, is a number the model should see distributed as a standard Gaussian if the model is adequate. A reading that lands three or four spreads from its prediction is either a faulty sensor or a faulty model, and it is flagged before it is folded in. Chapter 12 builds detection on this; here it is enough that the number costs nothing beyond the update, because $\mathbf{h}^\mathsf{T}\mathbf{s}$ was computed for the gain.

Proposition 11.2 (Innovations)**.** *Let readings $y_1, \dots, y_r$ of an adequate linear-Gaussian model be folded in one at a time, and let e_i and s_i be the innovation of reading i and its predicted variance, both computed before reading i is folded in. Then the standardized innovations $e_i/\sqrt{s_i}$ are independent standard Gaussians, $\sum_i e_i^2/s_i$ is chi-squared with r degrees of freedom, and the log predictive density of all the readings is*

$$\log p(y_1, \dots, y_r) \;=\; -\tfrac{1}{2}\sum_{i=1}^{r}\Big(\log 2\pi s_i + \frac{e_i^2}{s_i}\Big). \tag{11.2}$$

Proof. By the chain rule of probability, $p(y_1, \dots, y_r) = \prod_i p(y_i \mid y_1, \dots, y_{i-1})$. Under the model, y_i given the earlier readings is Gaussian with mean $\mathbf{h}_i^\mathsf{T}\boldsymbol{\mu}_{i-1}$ and variance s_i: the prediction that Proposition 11.1 makes before it updates. So each factor is the Gaussian density of e_i with

variance s_i, and taking logs gives (11.2). For independence, $e_i/\sqrt{s_i}$ given $y_1, \ldots, y_{i-1}$ is standard Gaussian whatever those readings are. The earlier standardized innovations are functions of them, so $e_i/\sqrt{s_i}$ is standard Gaussian given the earlier innovations too. A sequence each member of which is standard Gaussian given all the earlier ones has a joint density that factors into standard Gaussians, which is independence. The sum of squares of r independent standard Gaussians is chi-squared with r degrees of freedom by definition. □

Equation (11.2) is the prediction-error decomposition. The predictive density that Proposition 7.1 used to weigh branches accumulates one reading at a time, from the two numbers each update computes anyway.

The feeder, one reading at a time. Return to the export feeder of Figure 3.1 with hard physics. Every number is from the script `ch11_sequential.py` in the book's `scripts` directory. Under the prior, V_2 is predicted at 1.02500 ± 0.01045. The reading is 1.027: innovation +0.00200, standardized +0.19, unremarkable. One rank-one update gives $G = 0.05366 \pm 0.00582$ and $V_s = 1.00016 \pm 0.00287$, the second column of Table 3.2. Now V_1 is predicted at 1.03773 ± 0.00143, the virtual sensor of §3.4 with its spread now including the instrument's. The reading is 1.038: innovation +0.00027, standardized +0.19 again. The second update gives $G = 0.05463 \pm 0.00286$, $V_s = 0.99973 \pm 0.00173$, the third column. The batch posterior with both readings agrees with the two sequential updates to eight decimals in the mean and seven in the covariance, the residual being the conditioning of the tight physics rows and not the method.

Figure 11.1 adds a third reading, $V_s = 0.9995$ with accuracy 0.0005, and folds the three in three orders. In every order the injection ends at 0.05497 ± 0.00097, the batch posterior with all three readings. The paths differ. Read the substation busbar first and the injection does not move at all, 0.05000 ± 0.02000, because under the prior the substation voltage and the injection are independent and the reading carries nothing about G. Read V_2 next and the injection drops to ± 0.00140, far better than the ± 0.00582 that V_2 gave alone, because the busbar reading has pinned the one quantity that V_2 confounds with the injection. Read V_1 first and the injection is at 0.05409 ± 0.00425, after which the busbar reading removes 94 percent of what remains. No innovation in any of the three orders exceeds a quarter of its spread, as it should not for readings consistent with the model.

11.4 Many readings

Readings in sequence are updates in sequence, and for a linear-Gaussian model the result does not depend on the order, since each update is exact and the final posterior is the one with all rows present. A block of r readings taken together is the Woodbury identity in place of Sherman–Morrison: r solves, an $r \times r$ dense system, the same result.

This is the Kalman filter with a static state. The filter of Chapter 16 adds a prediction step between updates, in which the state moves and its uncertainty grows; with no dynamics the prediction is the identity and only the update remains, and the update is equation (11.1). The reader who knows the filter will recognize $\mathbf{k}$ as the Kalman gain and $\sigma_y^2 + \mathbf{h}^\mathsf{T}\mathbf{s}$ as the innovation covariance. What the sparse factorization adds is that the state may have a hundred thousand components and the update still costs one solve, because $\boldsymbol{\Sigma}$ is never formed. The block form is derived in the solution to Exercise 11.1.

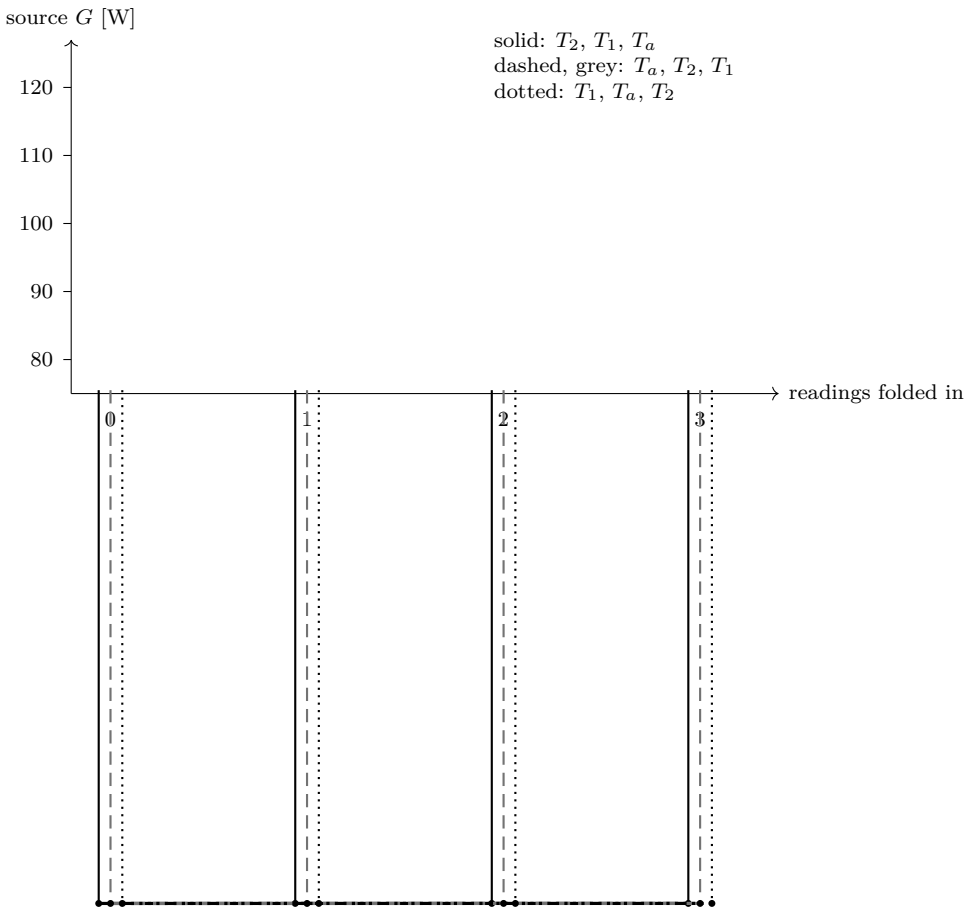


Figure 11.1: The injection of the export feeder as three readings are folded in, in three orders: $V_2 = 1.027$, $V_1 = 1.038$ and $V_s = 0.9995$, each of accuracy 0.0005. Dots are posterior means and bars one posterior standard deviation; the three orders are offset slightly for legibility. Every order ends at the batch posterior. Read first, the busbar moves nothing, because under the prior the substation voltage and the injection are independent.

11.5 Worked example: a stream of flow readings

The two-region network of Chapter 8 carries six flow meters, on lines 1–2, 2–3, 3–4, 4–5, 4–6 and 5–6, each of accuracy 0.02 per unit. They report in rounds, one after another, and the loads do not change between rounds: twenty-four readings of a static state. The state is a draw from the prior, with loads at buses 2 through 6 of -1.407, -0.601, -0.297, -1.315 and -0.275. From the third round the meter on line 4–5 reads 0.10 high, a calibration drift of five times its accuracy. Every number is from the script `ch11_more.py`.

Figure 11.2 plots the standardized innovation of each reading and, below it, their running sum of squares against the 99 percent point of the chi-squared distribution with as many degrees of freedom as readings. In the first round the innovations are large in the units of the readings, up to 0.20, and ordinary in their own, none beyond 1.7 spreads, because the predicted spread of a first reading is mostly the prior's uncertainty about the flow: 0.19 for line 1–2. By the second round every flow is known to about the meters' accuracy, and the predicted spreads have fallen

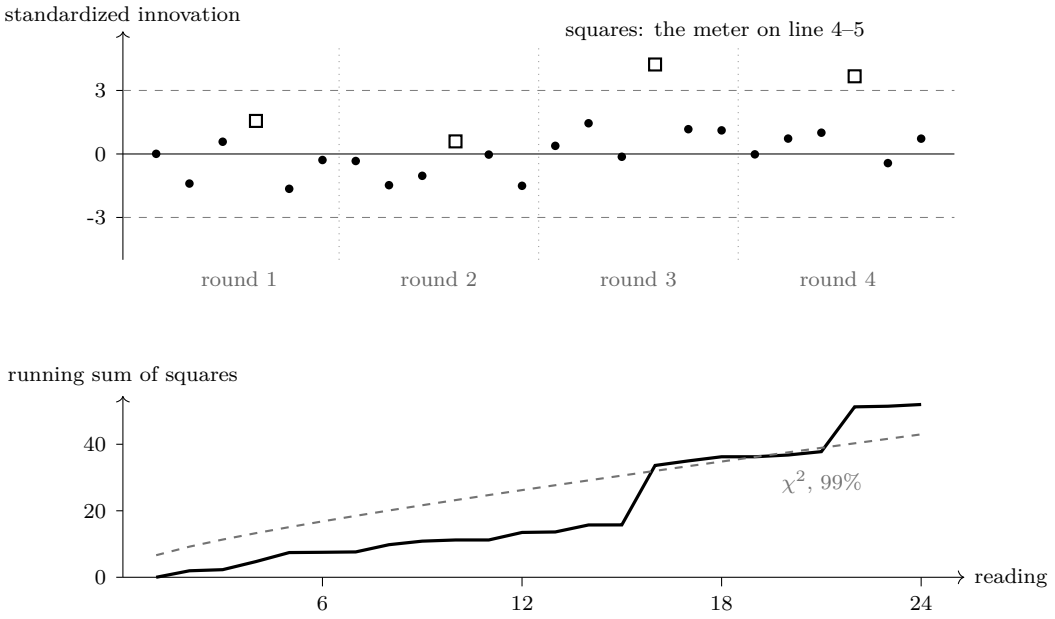


Figure 11.2: The two-region network's six flow meters read in four rounds, a static state. Top: the standardized innovation of each reading; squares mark the meter on line 4–5, which reads 0.10 high from the third round, and the dashed lines are at ±3. Bottom: the running sum of squared standardized innovations (solid) against the 99 percent point of the chi-squared distribution with as many degrees of freedom as readings (dashed).

to between 0.026 and 0.028. The drifting meter's first biased reading, the sixteenth, lands 4.2 spreads high, and its second, the twenty-second, 3.7. No other reading in the stream exceeds 1.7.

The running sum tells the same story less sharply. It crosses the quantile at the sixteenth reading, falls back below it at the twentieth as clean readings dilute the one bad one, and crosses again for good at the twenty-second. A cumulative test spreads a sudden fault over every reading that follows; a per-reading test sees it at once. The cumulative test is the right one for a slow drift that no single reading reveals.

Detection matters because a folded-in fault corrupts everything the reading touches. Folding all twenty-four readings in puts the load at bus 5 at -1.355 ± 0.017, 2.3 standard deviations from the true -1.315. Withholding the two flagged readings gives -1.304 ± 0.020, 0.6 from it. The rule that did the withholding, to hold aside any reading whose standardized innovation exceeds three, used nothing the update had not already computed. Chapter 12 turns the rule into a test with a stated false-alarm rate and asks which fault explains a flag.

Last, the evidence. The sum over the stream of the per-reading terms of equation (11.2) is 32.146284. The log predictive density of all twenty-four readings at once, a twenty-four-dimensional Gaussian with its own determinant and quadratic form, is the same to the six decimals shown.

11.6 The Laplace posterior and re-linearization

For a nonlinear model the posterior is Gaussian only in the Laplace sense of §6.4: the Gaussian at the mode with the curvature there. Proposition 11.1 conditions that Gaussian exactly. What

it does not do is move the linearization point: the new mode of the true posterior differs from $\boldsymbol{\mu}'$ by the amount the nonlinear rows curve between the old mode and the new mean.

The right procedure is the cheap one first. Fold the reading in by the rank-one update; check the standardized innovation and the size of the shift $\boldsymbol{\mu}' - \boldsymbol{\mu}$ against the posterior width; if both are modest, the linearization is still good and nothing more is needed. If either is large, take one Gauss–Newton step from $\boldsymbol{\mu}'$, which lands on the exact new mode when the rows are nearly linear across the shift, and refactor there. For a model whose rows are linear, which the DC and decoupled models of this book are, the check always passes and the update is always exact. For an AC model the check fails only when a reading moves the state by a sizable fraction of the region over which the sines and cosines curve, which under normal operation is rarely.

A worked example on a nonlinear line. Take one line whose flow is $P = B \sin\theta$ with $B = 1.5$ per unit, a prior of 1.0 ± 0.3 on the power it carries, a physics row of width 0.01, and a phasor measurement of the angle θ with accuracy 0.01 radians. Before any reading the Laplace posterior sits at $\theta = 0.7297$, $P = 1.000 \pm 0.300$, θ to within ± 0.269, and the two are correlated at 0.9994: the Gaussian lies along the tangent to the sine at the mode. Figure 11.3 draws three readings of the angle, 0.76, 0.95 and 1.20. For the first, the rank-one update gives $P = 1.0338$ against the exact new mode 1.0333 ± 0.0148, an error of a thirtieth of a posterior standard deviation. For the second it gives 1.2457 against 1.2197 ± 0.0133, two standard deviations off. For the third it gives 1.5245 against 1.3975 ± 0.0114, eleven off. The update moves along the tangent, and the sine has bent away from it. In every case one Gauss–Newton step from the updated mean lands on the exact mode to 2×10^{-8}.

The two checks of the procedure coincide here, because the reading is of θ itself and much sharper than the prior: the standardized innovations are 0.11, 0.82 and 1.75, and so are the shifts in units of the prior's standard deviation of θ. Neither innovation of the bad cases would raise an alarm as a fault. The shift is what matters, and a shift of 0.82 prior standard deviations already cost two posterior ones. Re-linearize whenever the shift exceeds about a quarter of the prior's standard deviation along the reading; the step costs one factorization.

11.7 The value of a reading before it is taken

Look at the covariance update in equation (11.1): it does not contain y. The reduction in uncertainty that a reading brings is known before the reading is taken. For a target quantity $g = \mathbf{c}^\mathsf{T}\mathbf{z}$, the posterior variance after a reading of $\mathbf{h}^\mathsf{T}\mathbf{z}$ with accuracy σ_y is

$$\mathrm{Var}'(g) \;=\; \mathrm{Var}(g) - \frac{(\mathbf{c}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h})^2}{\sigma_y^2 + \mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h}}, \tag{11.3}$$

whatever the reading turns out to be. It follows from the covariance update of equation (11.1) by multiplying on both sides by $\mathbf{c}$: $\mathbf{c}^\mathsf{T}\boldsymbol{\Sigma}'\mathbf{c} = \mathbf{c}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{c} - (\mathbf{c}^\mathsf{T}\mathbf{k})(\mathbf{s}^\mathsf{T}\mathbf{c})$, and $\mathbf{c}^\mathsf{T}\mathbf{k} = \mathbf{c}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h}/(\sigma_y^2 + \mathbf{h}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{h})$. The reduction is the squared covariance between the target and the measured quantity, scaled by the measured quantity's total uncertainty. A sensor is valuable to a target exactly insofar as the two are correlated under the current posterior, and its value changes as the posterior does.

When the target is the whole state rather than one quantity, the natural measure is the reduction in entropy, which for a Gaussian is half the log of the ratio of determinants of the

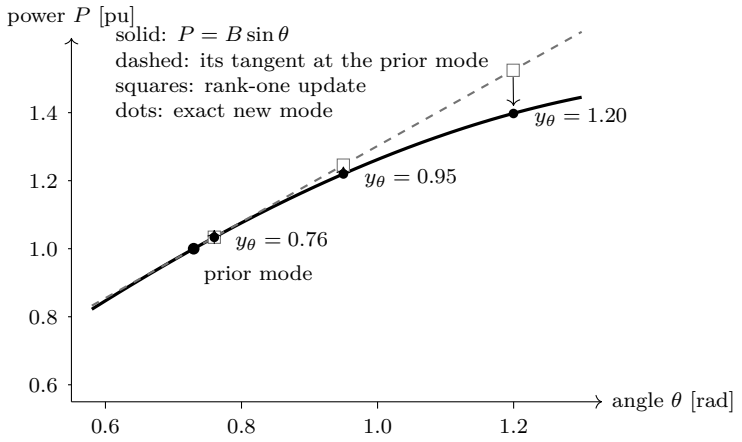


Figure 11.3: A line with flow $P = B \sin\theta$, $B = 1.5$, and three phasor readings of its angle. The Laplace posterior before the readings lies along the tangent at the prior mode (dashed). The rank-one update of that Gaussian moves along the tangent (squares), while the exact new modes lie on the curve (dots). The error grows with the shift, from a thirtieth to two and then eleven posterior standard deviations. One Gauss–Newton step from each square reaches its dot.

covariances before and after. A single reading gives

$$\tfrac{1}{2}\log\frac{\det\boldsymbol{\Sigma}}{\det\boldsymbol{\Sigma}'} \;=\; \tfrac{1}{2}\log\frac{\det\boldsymbol{\Lambda}'}{\det\boldsymbol{\Lambda}} \;=\; \tfrac{1}{2}\log\Big(1+\frac{\mathbf{h}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{h}}{\sigma_y^2}\Big). \tag{11.4}$$

The last step is the matrix determinant lemma: $\det(\boldsymbol{\Lambda} + w\mathbf{h}\mathbf{h}^{\mathsf{T}}) = \det\boldsymbol{\Lambda}\,\det(\mathbf{I} + w\boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^{\mathsf{T}})$, and $\mathbf{I} + w\,\mathbf{s}\mathbf{h}^{\mathsf{T}}$ has the eigenvalue $1 + w\,\mathbf{h}^{\mathsf{T}}\mathbf{s}$ on $\mathbf{s}$ and the eigenvalue 1 on every vector orthogonal to $\mathbf{h}$, so its determinant is $1 + w\,\mathbf{h}^{\mathsf{T}}\mathbf{s}$. A reading is worth most to the whole state where the state's own uncertainty about the measured quantity is largest compared with the sensor's accuracy.

Table 11.1 applies it. On the feeder, a sensor on the substation busbar is worth nothing to the injection under the prior: the two are independent, as Chapter 5 said, so their covariance is zero and equation (11.3) subtracts nothing. A sensor on either bus voltage is worth most of the injection's variance, and a direct flow meter, even a poor one, nearly all of it. Now read V_2 and ask again. The injection's spread is 0.00582, and the sensor that would reduce it most is the one on the busbar, by 94 percent, because the reading has correlated the injection with the substation voltage at -0.985 and a reading of one is now a reading of the other. The sensor that was worthless has become the best. That is the ridge of §3.4 read as a decision: the next sensor should be placed where the posterior's remaining uncertainty is concentrated, and the posterior says where.

On the two-region network the target is the load at bus 5, and the candidates are flow meters on each line. The meter on the line into bus 5 from bus 4 is worth 78 percent of the load's variance; the meter on the tie line, two buses away, a third; the meter on line 1–2 in the other region, an eighth. The ordering follows the graph, and equation (11.3) computes it from one solve per candidate against the factorization already held.

Figure 11.4 shows how the values change once the best meter has been read. After the meter on line 4–5, the meter on line 5–6, worth 48 percent of the prior variance, is worth 90 percent of what remains, and the meter on line 4–6 is worth 75 percent; the meters in region A fall to a few percent. The balance at bus 5 makes the load the difference of the two flows at the bus. With

Table 11.1: The value of the next sensor, computed before any reading. Top: the feeder, target G, prior spread 0.02. Bottom: the two-region network, target P_5, prior spread 0.200, flow meters of accuracy 0.02.

feeder, sensor on	sd of G after	reduced by
under the prior		
V_1 (acc. 0.0005)	0.00425	95%
V_2 (acc. 0.0005)	0.00582	92%
V_s (acc. 0.0005)	0.02000	0%
Q_{2s} (acc. 0.002)	0.00199	99%
after V_2 is read (sd 0.00582)		
V_1	0.00286	76%
V_s	0.00140	94%
Q_{2s}	0.00189	89%

network, meter on	sd of P_5 after	reduced by
F_{12}	0.187	12%
F_{23}	0.179	20%
F_{34}	0.164	33%
F_{45}	0.093	78%
F_{46}	0.179	20%
F_{56}	0.145	48%

one flow known, the other is the load, and the meter that reads it directly becomes decisive. The whole-state measure of equation (11.4) orders the meters differently. The meter on the tie line is worth 2.86 nats to the whole state, the most of the six, and the meter on line 4–5 only 2.02. The target chooses the sensor; there is no best sensor in the abstract.

Choosing sensors by this criterion, one at a time, each chosen given the last, is greedy sensor placement, and the criterion is the expected reduction in a target's variance. Chapter 12 generalizes the criterion to the expected information gain about the whole state, which is the classical measure of an experiment's value, and shows that for a Gaussian it is the log of a determinant ratio, computed by the same solves.

Principle 11.2. *The uncertainty a reading removes is known before the reading is taken. It is the squared covariance between target and measurement, divided by the measurement's total spread, and it is one solve per candidate.*

11.8 Cost at scale

The feeder has six unknowns and the comparison is not worth timing. Take instead a mesh of $m \times m$ buses, each tied to its four neighbours through a susceptance and to ground through a shunt, with an uncertain reactive injection at every bus; eliminating the injections leaves one soft row per bus on the voltages, and the precision is sparse with the mesh's own structure, which in two dimensions has fill. Ten voltage meters at random buses report in turn. Table 11.2 times the two ways of folding them in.

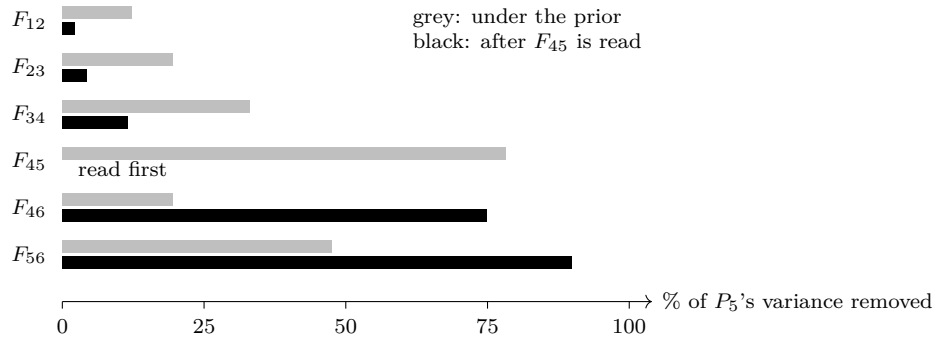


Figure 11.4: The value of each flow meter to the load at bus 5, as the percentage of its current variance the meter would remove, computed before any reading. Grey: under the prior. Black: after the meter on line 4–5 has been read. The meter on line 5–6 goes from second to decisive, because the load at bus 5 is the difference of the two flows at the bus.

Table 11.2: A mesh of $m \times m$ buses: one sparse factorization against one rank-one update, and ten readings folded in by refactoring each time against factoring once and updating ten times. Means agree to 10^{-13}. The matrix is the two-dimensional grid Laplacian plus a diagonal, so these costs are a fact about that structure rather than about the quantity flowing on it.

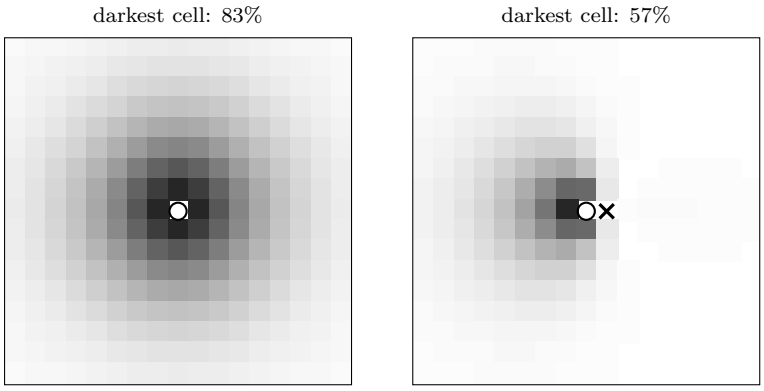
m	unknowns	factor nonzeros	one factorization	one update	ratio	ten readings, ratio
30	900	7.0×10^4	6.2 ms	0.13 ms	48	6.9
100	10,000	2.0×10^6	117 ms	3.5 ms	33	10.7
300	90,000	3.3×10^7	3.55 s	49 ms	72	9.1

One rank-one update costs between a thirtieth and a seventieth of a factorization across three decades of size, the ratio being that of a solve to a factorization, which for a two-dimensional sparse structure grows slowly with size. Over ten readings the factor-once route is about ten times faster than refactoring, the first factorization being paid either way; over a hundred it would be some fifty times faster, and the limit is the per-update ratio. The means agree to thirteen decimals. At ninety thousand unknowns a reading is folded in fifty milliseconds, which is faster than SCADA reports.

11.9 Worked example: where to put the next meter

A substation bus sits behind a transformer with no instrument transformer of its own, and its voltage is the target. Model the surrounding network as the mesh of §11.8 at 30×30 buses with stronger coupling: susceptance 4 per unit between neighbouring buses, a shunt of 0.25 per unit from each bus to ground, and an uncertain reactive injection of standard deviation 0.02 per unit at every bus. The unmetered bus is at the centre. Voltage meters of accuracy 0.001 per unit can go on any other bus. Under the prior the centre's voltage has a standard deviation of 0.00576 per unit, and it is correlated with each neighbour at 0.92.

Figure 11.5(a) maps the value of a meter at each bus near the centre, by equation (11.3). The whole map needs one column of the covariance, the centre's, and its diagonal. A neighbour removes 82.6 percent of the centre's variance, a bus two steps away 74.0, three steps 59.1, and



(a) under the prior (b) after the first meter

Figure 11.5: The value of a voltage meter at each bus near the centre of a 30×30 mesh, for the voltage at the centre (ringed), where no meter can go. Darker cells remove more of the centre's current variance; each panel is scaled to its own largest value. (a) Under the prior: the four neighbours are worth 82.6 percent each, and the value falls off slowly with distance. (b) After a meter on the east neighbour (cross), whose own cell is now worth nothing: the opposite neighbour is now worth most, 56.6 percent of what remains.

seven steps 23.3. The mesh spreads the correlation over several buses, so the value falls off slowly with distance. Greedy placement takes a neighbour first, and the centre's standard deviation drops to 0.00240 per unit.

Panel (b) is the map after that reading. The best next position is the opposite neighbour, now worth 56.6 percent of what remains. The buses beside the first meter have lost most of their value, and those on the far side have kept it. The first reading told the model the voltage on one side of the centre, and what remains uncertain is the gradient across it, which the opposite side measures. The second meter brings the centre to 0.00158 per unit. The third, on a side neighbour, removes 14.5 percent of what remains, and the fourth, on the last neighbour, 20.5 percent, more than the third did. A meter's value can rise as others are placed: readings on opposite sides of a bus are worth more together than apart, the explaining away of Proposition 5.2 again.

The four meters leave the centre at 0.00130 per unit, and no meter outside the centre can do much better. With its four neighbours known exactly the centre would still be uncertain by 0.00120 per unit, and with every other bus of the mesh known exactly, by 0.00110. What remains is the centre's own injection. It enters only the centre's balance, where it is divided by the total susceptance at the bus, $4 \times 4 + 0.25$ per unit, which sets the scale: 0.02/16.25 is 0.00123 per unit. The neighbours' balances recover a little more, through the coupling between them and the centre, and that is all there is. A virtual sensor at an unmetered bus is limited not by where the meters go but by what the physics lets through.

11.10 In practice

Hold one factorization and a correction list. Factor the precision once, keep the pairs $(\mathbf{k}_i, \mathbf{s}_i)$ of the updates, and refactor when the list costs more to apply than a factorization costs to redo. Table 11.2 puts the break-even between thirty and seventy updates for the mesh.

Test every reading before folding it in. Compute the standardized innovation from the solve the update needs anyway, and hold aside a reading beyond three spreads. Count the holds per sensor. A sensor that is held once may have glitched; a sensor held repeatedly is drifting, as the meter on line 4–5 was.

Run a cumulative test as well, and know what it dilutes. The running sum of squared innovations catches a slow drift that no single reading reveals, and it blurs a sudden fault over every reading that follows. In the stream of §11.5 it dipped back below its quantile between the two biased readings. Use both tests.

Re-linearize by shift, not by surprise. On a nonlinear model the error of the rank-one update grows with how far the reading moves the state, measured in prior standard deviations along the reading. A modest innovation does not certify a small error. One Gauss–Newton step from the updated mean restores the mode.

Choose the next sensor for the question. Compute the value of each candidate for the target that matters, by equation (11.3), and recompute after every reading, since the values change. The whole-state measure of equation (11.4) is for when there is no single target.

Order does not change the answer; it changes what is known early. Every order ends at the batch posterior. When decisions are made between readings, read first the sensor worth most now.

11.11 What is missing

Part III has organized the computation: exactly by regions, approximately where the structure runs out, costed against sampling in solves, and updated one reading at a time without re-solving. What it has not done is ask a question. Every computation so far returned a posterior, and a posterior is not an answer until someone asks it something: what is the voltage where there is no sensor, is this reading a fault or a bad meter, what would happen if that line were taken out, which input drives the risk, which lines are most likely to fail together. Those are the operator's questions, and Part IV asks them in turn, beginning with the first.

Notes and further reading

The Sherman–Morrison and Woodbury identities and their history are reviewed by Hager [31]. The update (11.1) is the measurement update of the Kalman filter [39], whose gain and innovation are the same objects; Särkkä and Svensson [81] give the modern treatment, including the information form used here and the relation to Gaussian inference on a graph. Computing the covariance column by a solve against a cached sparse factorization rather than by forming the covariance is selected inversion again [21]; the correction-list scheme of §11.2 is the standard way to apply low-rank updates to a factorized matrix without refactoring, and is what a Kalman filter on a sparse state amounts to.

That the covariance reduction from a Gaussian measurement is independent of the measurement's value, and can therefore be used to choose measurements before they are made,

is the basis of Bayesian experimental design. Lindley [53] defined the value of an experiment as its expected information; Chaloner and Verdinelli [13] review the Bayesian theory, and the greedy variance-reduction criterion of §11.7 is its simplest instance. The reading of the criterion on the feeder, that the busbar sensor's value goes from nothing to nearly everything once one temperature is known, is the book's illustration of a general fact: a measurement's value is a property of the current posterior, not of the sensor.

The prediction-error decomposition of equation (11.2) is standard in filtering, where it is how the likelihood of a model's parameters is computed from a run of the filter; Särkkä and Svensson [81] derive it in that setting. The counterexample to greedy placement in the solution to Exercise 11.4 is the explaining-away of Proposition 5.2 read as a design problem.

Summary

- A reading adds a rank-one term to the precision. Conditioning on it is one solve against the held factorization for the covariance column, a gain, and a subtraction never formed. Exact for a linear-Gaussian branch.
- The same solve gives the innovation's predicted spread; the standardized innovation is the running adequacy test. Both of the feeder's readings surprised the model by a fifth of a spread.
- Readings in sequence commute; blocks use Woodbury; the whole is the static Kalman filter on a sparse state. For a Laplace posterior, update first and re-linearize only when the innovation or the shift is large.
- The variance a reading removes from a target is known before the reading: squared covariance over total spread. On the feeder the busbar sensor is worth nothing before V_2 is read and 94 percent of the remaining variance after; on the network the meter nearest the target is worth most.
- On a mesh of ninety thousand unknowns one update costs a seventieth of a factorization and takes fifty milliseconds.
- Standardized innovations of an adequate model are independent standard Gaussians, and their log densities sum to the evidence. On the network stream a meter drifting by five times its accuracy was flagged at 4.2 spreads on its first biased reading; folding it in would have moved the load at bus 5 by 2.3 standard deviations.
- On a nonlinear line the rank-one update errs by up to eleven posterior standard deviations when a reading shifts the state by 1.75 prior ones. One Gauss–Newton step repairs it.
- On a mesh of buses the best place for a voltage meter near an unmetered bus is a neighbour, and after it the opposite neighbour. Four meters bring the unmetered bus to 0.00130 per unit, against a floor of 0.00110 that no outside meter can pass.
- A sensor's value depends on the target and on what has been read: after the meter on line 4–5, the meter on line 5–6 goes from 48 to 90 percent of the load's remaining variance.

Exercises

Exercise 11.1. Derive equation (11.1) from the information-form update $\mathbf{\Lambda}' = \mathbf{\Lambda} + w\mathbf{h}\mathbf{h}^\mathsf{T}$, $\boldsymbol{\eta}' = \boldsymbol{\eta} + wy\mathbf{h}$ and the Sherman–Morrison identity. Then derive the Woodbury form for r simultaneous readings and show it reduces to r sequential applications of the rank-one form.

Exercise 11.2. Show that the standardized innovations of a sequence of readings on an adequate linear-Gaussian model are independent standard Gaussians, so that their sum of squares over r readings is chi-squared with r degrees of freedom. Relate this to the misfit statistic of §4.5.

Exercise 11.3. On the feeder, read V_s first (say 0.9995 with accuracy 0.0005), then V_2, then V_1, and confirm that the final posterior equals the batch posterior with all three readings. Then compute the value of each of the three sensors under the prior and after each reading, and show the order in which greedy placement would choose them.

Exercise 11.4. For the two-region network, choose two flow meters greedily to minimize the posterior variance of P_5, then choose the best pair by exhaustive search over the fifteen pairs, and compare. Repeat with the target changed to the tie flow F_{34}.

Exercise 11.5. Rerun the mesh timing with a hundred readings instead of ten, and with the correction list refreshed by a refactorization every r readings for $r \in \{10, 30, 100\}$. Find the r that minimizes total time at $m = 100$ and relate it to the per-update ratio of Table 11.2.

Solutions to selected exercises

Exercise 11.1. The rank-one form is the proof of Proposition 11.1. For r readings at once, stack their rows into $\mathbf{H}$, $r \times n$, with noise covariance $\mathbf{R}$, and write $\mathbf{S} = \mathbf{R} + \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}$. The claim is that $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}\mathbf{S}^{-1}\mathbf{H}\boldsymbol{\Sigma}$ inverts $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$. Multiplying,

$$\begin{aligned}\boldsymbol{\Lambda}'\boldsymbol{\Sigma}' &= \mathbf{I} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\boldsymbol{\Sigma} - \mathbf{H}^{\mathsf{T}}\mathbf{S}^{-1}\mathbf{H}\boldsymbol{\Sigma} - \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}\mathbf{S}^{-1}\mathbf{H}\boldsymbol{\Sigma} \\ &= \mathbf{I} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{S} - \mathbf{R} - \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}})\mathbf{S}^{-1}\mathbf{H}\boldsymbol{\Sigma} \;=\; \mathbf{I},\end{aligned}$$

where the second line inserts $\mathbf{S}\mathbf{S}^{-1}$ into the second term of the first and $\mathbf{R}^{-1}\mathbf{R}$ into the third. The mean follows as in the rank-one case: $\boldsymbol{\mu}' = \boldsymbol{\mu} + \boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}\mathbf{S}^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu})$. The cost is r solves for $\boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}$ and one dense $r \times r$ factorization. To see that r rank-one updates give the same result when $\mathbf{R}$ is diagonal, look at the information form. The batch posterior has precision $\boldsymbol{\Lambda} + \sum_i w_i \mathbf{h}_i \mathbf{h}_i^{\mathsf{T}}$ and information $\boldsymbol{\eta} + \sum_i w_i y_i \mathbf{h}_i$. Each rank-one update is exact, so it produces the posterior whose precision and information have exactly one more term of each sum. After r updates both sums are complete, in whatever order they were taken, and a Gaussian is determined by its precision and information. The stream of §11.5 confirms it to machine precision: its sequential evidence equals the batch predictive density.

Exercise 11.3. In every order the injection ends at 0.05497 ± 0.00097 and the substation voltage at 0.99952 ± 0.00048, the batch posterior with all three rows; Figure 11.1 draws the three paths. The values for the injection, as percentages of its current variance, are as follows. Under the prior, V_1 is worth 95.5, V_2 91.5 and V_s nothing, so greedy placement takes V_1. After it, V_s is worth 94.4 and V_2 54.5, so greedy takes V_s. After both, V_2 removes 6.1 percent. The greedy order is V_1, V_s, V_2, and the sensor worth nothing at first is chosen second, because the first reading made the substation voltage the largest remaining uncertainty about the injection. In the order the exercise prescribes, V_s first, the injection's variance is unchanged by the first reading, as its value of zero said it would be.

Exercise 11.4. For the load at bus 5, greedy placement takes the meter on line 4–5 first, worth 78 percent, and then the meter on line 5–6, which leaves a standard deviation of 0.0296. Exhaustive search over the fifteen pairs finds the same pair best; the next are 4–5 with 4–6 at 0.0468 and 4–6 with 5–6 at 0.0475. For the tie flow F_{34} greedy takes the meter on the tie line itself and then any other: every pair containing it leaves 0.0199 or 0.0200, the meter's own accuracy, and the second meter hardly matters. On this network greedy placement of two meters is optimal for all seventeen variables as targets, for sums and differences of loads, and for the whole-state measure on the five loads.

It is not optimal in general, and the failure is explaining away. Let the target be $t \sim \mathcal{N}(0,1)$ and let a nuisance $u \sim \mathcal{N}(0,1)$ be independent of it. Offer three sensors: A reads $t+u$ and B reads u, both with accuracy 0.01, and C reads t with accuracy 0.5. Alone, C removes $1-1/(1+4) = 80$ percent of the target's variance, A removes $1/(2+10^{-4})$, about half, and B nothing, since u is independent of t. Greedy takes C, leaving a variance of 0.2. Then A has covariance 0.2 with the target and variance 1.2, and removes 0.04/1.2, leaving 0.167, while B is still worth nothing. The pair A and B, which greedy never considers, leaves about 2×10^{-4}: their difference reads t to within 0.014. A meter on a flow that carries none of the target can be worthless alone and decisive in a pair. When candidates come in such pairs, search over pairs.

Part IV

The operator's questions

Chapter 12

Conditioning, diagnosis, and virtual sensors

A posterior is not an answer. It is a distribution over every variable of the system, and an operator asks it something specific: what is the temperature at the node that has no sensor, is this reading a fault or an unmodeled draw, can this parameter be recovered from these sensors at all, and which sensor should be added next. This chapter shows that each of those questions is a readout of the one posterior, computed from the factorization already held, and that the posterior can also be asked the question that matters most before any other: whether it should be believed. Adequacy, localization, hypothesis comparison, identifiability, experiment design and the validity of the Gaussian approximation are each one section, each with a number on the feeder, and each is a solve or a determinant against $\mathbf{\Lambda}$.

12.1 Everything is a readout

Table 12.1 lists the questions of this chapter and the next three against what each reads from the posterior. The point of the table is its right-hand column: nothing in it is a new model. Adding a question does not add a computation to the model; it adds a readout of a posterior that was computed once. That is the practical consequence of building the probabilistic model on the physics' own rows, and it is why the book insisted in Chapter 3 that sensors and priors be added to the graph rather than the graph be built around them.

12.2 Virtual sensors

The marginal of a variable that no sensor reads is a *virtual sensor*: the physics and the sensors that do exist determine it, and the posterior says how well. It is the same marginal as for a measured variable, read the same way; the only difference is whether a likelihood factor happens to touch it. Table 12.2 gives the full posterior of the feeder with both voltage readings, from the script `ch12_diagnosis.py` in the book's `scripts` directory, which supplies every number in the chapter. The two measured voltages are known to five ten-thousandths, their sensors' accuracy less a little for the physics' corroboration. The two reactive flows, which no instrument reads, are known to 0.0029 per unit, and the injection and the substation voltage likewise, all from two voltage sensors and four rows of physics.

Table 12.1: Operational questions as readouts of one posterior.

	Question	Readout
Q1	estimate a measured quantity	its marginal mean and spread
Q2	estimate an unmeasured one (virtual sensor)	the same marginal, on a variable with no sensor
Q3	is the model adequate?	the misfit against its chi-squared law
Q4	where is it inadequate?	studentized residuals, ranked
Q5	which explanation?	log-evidence of each hypothesis
Q6	can this parameter be recovered?	the parameter block's Schur complement and its flattest direction
Q7	which sensor next?	expected information gain, one solve per candidate
Q8	should the Gaussian be believed?	curvature probes and reweighting efficiency
Q9	what if? (Chapter 13)	clamp, re-infer
Q10	which input drives the risk? (Chapter 14)	gradients from the solve

Table 12.2: The feeder's posterior with both readings: every variable, with whether a sensor reads it.

variable	posterior	sensor
V_1	1.03797 ± 0.00047	yes
V_2	1.02704 ± 0.00044	yes
Q_{12}	0.05463 ± 0.00286	no
Q_{2s}	0.05463 ± 0.00286	no
G	0.05463 ± 0.00286	no
V_s	0.99973 ± 0.00173	no

A virtual sensor is only as good as the model behind it, which is why the rest of the chapter is about testing the model.

12.3 Is the model adequate?

Chapter 4 named the misfit, the sum of squared standardized residuals over every factor at the posterior mode, and stated its law with its assumptions. Restated:

Proposition 12.1 (Adequacy)**.** *For a linear-Gaussian model with m_g factors (physics rows, priors and sensors, each a Gaussian pseudo-observation) and n unknowns, if the model is correctly specified, the misfit $\chi^2 = \mathbf{g}(\mathbf{z}^\star)^\mathsf{T}\boldsymbol{\Omega}\,\mathbf{g}(\mathbf{z}^\star)$ is chi-squared distributed with $m_g - n$ degrees of freedom. Its expected value is $m_g - n$, and the probability of a misfit at least as large as the one observed is the model's p-value.*

Proof. Scale each factor by the square root of its weight, so that the factors read $\tilde{\mathbf{c}} = \tilde{\mathbf{H}}\mathbf{z} + \tilde{\mathbf{e}}$, with $\tilde{\mathbf{e}}$ standard Gaussian in m_g dimensions when the model is correctly specified. The mode is the least-squares solution $\mathbf{z}^\star = (\tilde{\mathbf{H}}^\mathsf{T}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^\mathsf{T}\tilde{\mathbf{c}}$, and the standardized residual vector is $\tilde{\mathbf{c}} - \tilde{\mathbf{H}}\mathbf{z}^\star = (\mathbf{I} - \mathbf{P})\tilde{\mathbf{c}} = (\mathbf{I} - \mathbf{P})\tilde{\mathbf{e}}$, where $\mathbf{P} = \tilde{\mathbf{H}}(\tilde{\mathbf{H}}^\mathsf{T}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^\mathsf{T}$ is the orthogonal projection onto the columns of $\tilde{\mathbf{H}}$. The second equality holds because $(\mathbf{I} - \mathbf{P})\tilde{\mathbf{H}} = \mathbf{0}$. The misfit is $\tilde{\mathbf{e}}^\mathsf{T}(\mathbf{I} - \mathbf{P})\tilde{\mathbf{e}}$, and $\mathbf{I} - \mathbf{P}$ is

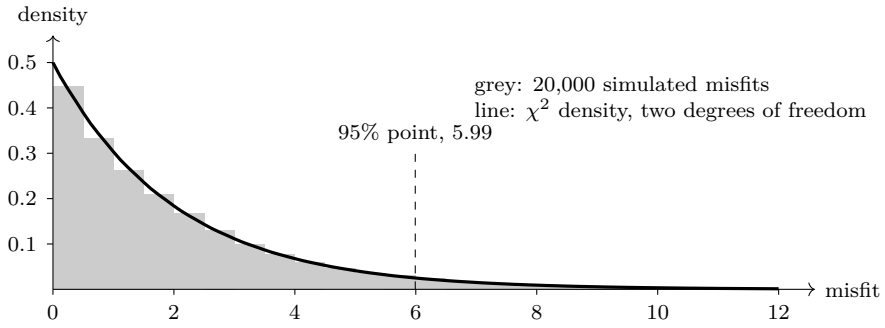


Figure 12.1: Twenty thousand misfits of the feeder's model on readings generated from the model itself, against the chi-squared density with two degrees of freedom that Proposition 12.1 predicts. The dashed line is the 95 percent point, above which 4.9 percent of the simulated misfits fall.

a symmetric projection of rank $m_g - n$. In a basis of its eigenvectors the misfit is the sum of squares of $m_g - n$ independent standard Gaussians, which is the chi-squared law. □

A prior counts as a pseudo-observation of its variable at its mean, and the proof treats it as one; that is where "correctly specified" bites. A broad prior that only keeps a variable identifiable, and from which the truth is not drawn, is not a pseudo-observation in this sense. It belongs in neither the misfit nor the count, and Chapter 25 measures what counting it costs on a real network. The script `ch12_more.py` checks the law by simulation. Twenty thousand reading pairs generated from the feeder's own priors and physics give a mean misfit of 2.00 against the law's 2, a variance of 3.89 against 4, and 4.9 percent of misfits above the 95 percent point. The physics in those draws was exact, not perturbed at its stated width of 10^{-5} per unit. Perturbing it at that width changes the three numbers to 2.01, 4.12 and 5.2 percent, the same law within simulation error. Figure 12.1 draws the simulated misfits against the density the proposition predicts.

When $m_g - n \leq 0$ the model has no redundancy and adequacy cannot be assessed; a model that exactly fits its data by construction has not been tested by it, and the honest report is that verdict rather than a comfortable fit. On the feeder with two readings there are eight factors and six unknowns: two degrees of freedom.

With the consistent readings 1.027 and 1.038 the misfit is 0.07 against an expectation of 2; the p-value is 0.97. With the inconsistent pair 1.027 and 1.050 of Chapter 4, and the physics rows at a diagnostic width of 10^{-3} per unit so that they may be violated, the misfit is 70 and the p-value is 5×10^{-16}. The model is wrong, and the test says so with no knowledge of how.

12.4 Where is it wrong?

The natural next step is to look at the residuals one factor at a time. The raw residual of a sensor, $y - \mathbb{E}[z]$, has a variance smaller than the sensor's, because the fitted mean was pulled toward the reading; the right statistic divides by $\sqrt{\sigma_y^2 - \mathrm{Var}[z]}$, the *studentized* residual, and the same correction applies to any factor. This is the classical bad-data statistic of state estimation and the gross-error statistic of data reconciliation, and it is one selected inversion per factor. The variance $\sigma_y^2 - \mathrm{Var}[z]$ is derived in the solution to Exercise 12.1.

For the inconsistent readings the studentized residuals are, in order of the factors: balance at bus 1, +2.4; balance at bus 2, −7.8; the upper line's closure, −8.4; the substation line's closure,

Table 12.3: Redundancy of each meter on the two-region network, and the smallest gross error that produces a studentized residual of three. Every meter has accuracy 0.02 per unit.

meter	redundancy r_k	smallest error flagged [pu]
F_{12}, F_{23}, F_{45}, F_{56}	0.58	0.079
F_{13}, F_{46}	0.28	0.11
P_2, P_5	0.31	0.107
F_{34} (tie line)	0.019	0.43

+7.8; injection prior, +2.4; substation-voltage prior, −7.8; the two sensors, +8.3 and −8.4. Six of the eight factors are violated by eight standard deviations, and the pattern does not point at one of them. It cannot. The two readings imply an injection of 0.115 per unit and a substation voltage of 0.9695, and a model with tight physics distributes that contradiction across every factor that could absorb a piece of it. Ranking residuals localizes a fault when one factor is far more violated than the rest; when the contradiction is global, as here, it says only that something is wrong everywhere downstream of the readings.

Principle 12.1. *Studentized residuals localize a fault when it is local. A global contradiction spreads across every factor that can absorb it, and ranking residuals then says where the model bends, not what broke.*

12.5 Worked example: bad data on a power network

Localization works when the fault is local and the model has redundancy around it. The two-region network shows both conditions, and what happens when the second fails. Give it seven flow meters, one on every line, and injection meters at buses 2 and 5, all of accuracy 0.02 per unit, with the load priors of Chapter 8. The misfit has nine degrees of freedom. With clean readings it is 6.35, a p-value of 0.70, and no studentized residual exceeds 0.8.

A meter's *redundancy* is the fraction of its variance that the rest of the model leaves unexplained, $r_k = 1 - \mathrm{Var}[\mathbf{h}_k^{\mathsf{T}}\mathbf{z}]/\sigma_k^2$, with the variance taken under the posterior. A gross error b on meter k moves its residual by $b\,r_k$, and the studentized residual divides by $\sigma_k\sqrt{r_k}$, so the error shows up as $b\sqrt{r_k}/\sigma_k$. The smallest error flagged at three spreads is therefore $3\sigma_k/\sqrt{r_k}$. Table 12.3 lists both. The meters in the two loops have redundancy 0.58 on lines 1–2, 2–3, 4–5 and 5–6 and 0.28 on lines 1–3 and 4–6, and the injection meters 0.31. The meter on the tie line has 0.019. An error there must reach 0.43 per unit, twenty-two times the meter's accuracy, before the test flags it.

The reason is the graph. The tie line is a bridge, the only line between the regions, so nothing else in the model measures the flow it carries except the load priors, which are ten times less precise than the meter. A meter on a bridge is what state estimation calls a critical measurement. It is also the port of Chapter 8 seen from the other side: a separator of size one, across which the two regions can only be reconciled through the one number the meter reads.

Now make one meter read 0.15 high, seven and a half of its accuracies. Figure 12.2 draws the studentized residuals. With the fault on the meter on line 2–3, the misfit jumps to 38.3, a p-value of 1.5×10^{-5}, and the ranking puts the faulty meter first at 5.7, ahead of the injection meter at bus 2 at −4.1 and the meter on line 1–3 at −3.1. The meters around a fault share its

(a) meter on 2–3 reads 0.15 high (b) meter on 3–4 reads 0.15 high

Figure 12.2: Studentized residuals on the two-region network when one meter reads 0.15 per unit high, seven and a half times its accuracy. Black marks the faulty meter, and the dashed line is at three. (a) A meter inside a loop: the fault leads the ranking. (b) The meter on the tie line, a bridge: no residual reaches two, and the error is absorbed into the loads at the two ends of the line.

residual, but the fault itself leads. With the fault on the tie line, the misfit is 8.37, a p-value of 0.50, and no residual exceeds 1.5. The test sees nothing. Meanwhile the estimate of the tie flow has moved 7.4 posterior standard deviations, and the loads at buses 3 and 4 have moved 4.3 each in opposite directions. The model has absorbed the error by trading load between the two ends of the bridge, which is the only explanation the graph allows, and no residual can contradict it.

> **Principle 12.2.** *A measurement is only as checkable as it is redundant. A meter on a bridge of the network, a separator of size one, has nothing to be checked against, and a gross error on it moves the state without moving the residuals. Redundancy is computed before any reading, from the same diagonal as the virtual sensors.*

12.6 Which explanation?

When the residuals cannot say what broke, name the candidates and let the evidence choose. Each candidate is the model plus one latent variable with a zero-centered prior: an unmodeled draw at bus 2, which adds a term to that balance; a draw at bus 1; a bias on the V_1 sensor, which adds an offset to its likelihood; a bias on the V_2 sensor. Each is a linear-Gaussian model with one more variable, and each has an evidence, the integral of its factors over all its variables, which Chapter 7 gave in closed form:

$$\log Z \;=\; -C(\mathbf{z}^\star) - \tfrac{1}{2}\log\det\mathbf{\Lambda} + \tfrac{d}{2}\log 2\pi + \sum_k \log\frac{1}{\sqrt{2\pi}\,\sigma_k}, \tag{12.1}$$

with d the number of variables and the last sum over every factor's normalizer. The constants matter here because the hypotheses have different numbers of variables. One factor that does not matter here is the determinant of the physics rows' Jacobian with respect to the solved variables, which Chapter 7 had to restore when its branches changed a closure (Proposition 7.1): every hypothesis in this section adds its latent variable to a balance or a sensor row as an input with a

Table 12.4: Five explanations of the inconsistent readings 1.027 and 1.050, scored by log-evidence. Posterior probabilities at equal prior odds. Physics width 10^{-5} per unit; all quantities in per unit.

hypothesis	fault prior	$\log Z$	G	fault estimate	probability
nothing wrong	–	-30.0	0.0972	–	0.000
draw at bus 2	0.05	-0.75	0.1127	0.0574 ± 0.0073	0.010
draw at bus 1	0.05	-28.6	0.0566	-0.0414 ± 0.0187	0.000
bias on V_1 sensor	0.005	$+2.88$	0.0569	0.0114 ± 0.0014	0.387
bias on V_2 sensor	0.005	$+3.32$	0.0716	-0.0085 ± 0.0010	0.603

prior, and leaves the physics Jacobian as it was, so that factor is common to all of them and cancels. The determinant term that remains is the *Occam factor*: it charges each hypothesis for the volume of states its extra variable makes plausible, so that a hypothesis with a wide prior on its fault variable is not rewarded merely for being able to fit anything.

Table 12.4 gives the result, and it is not the one Chapter 4 assumed. The null hypothesis and the draw at bus 1 are ruled out by thirty nats. The draw at bus 2, which Chapter 4 loosened a balance to find, is a possible explanation, at one percent. The two sensor biases are the probable ones, sixty percent to thirty-nine, a bias of eleven thousandths on one sensor or eight and a half on the other. The reason is in the columns: the draw explanation needs a draw of 0.057 per unit, comfortable under its 0.05 prior, but also an injection of 0.113, three spreads above its nominal 0.05; the bias explanations need a bias of two spreads and an injection within one. The evidence weighs every factor, and the injection prior tips it.

Two things follow. First, the verdict depends on the priors given to the fault variables, and it should: a modeler who knows the sensors are freshly calibrated would put a one-thousandth prior on the biases, and the draw would win. The computation makes that judgment explicit and prices it. Second, the computation is one factorization per hypothesis, each on a model one variable larger than the base, and the hypotheses can be as many as the operator can name. Chapter 15 uses the same evidence to rank hundreds of them.

Figure 12.3 prices the first judgment. It holds the draw's prior at 0.05 per unit and varies the width of the bias priors. Below 0.0023 the draw at bus 2 is the better explanation. Above it the bias on V_2 is, and above 0.0063 the bias on V_1 takes over. The evidence for a bias rises steeply as its prior widens from half a thousandth, because a narrow prior cannot reach the eight- or eleven-thousandth bias the readings need, and the fit pays for the shortfall. It peaks near a hundredth of a per unit, at 3.8 for the bias on V_2 and 4.3 for the bias on V_1, and then falls slowly. A prior far wider than needed spends its probability on biases that did not happen, which is the Occam factor at work.

12.7 Can this parameter be recovered?

Whether a parameter can be recovered from the sensors at hand is a property of $\mathbf{\Lambda}$ alone, before any reading. Partition the variables into the parameters of interest $\mathbf{p}$ and everything else, and take the Schur complement onto the parameters as in Chapter 8. By block elimination the $(\mathbf{p}, \mathbf{p})$ block of $\mathbf{\Lambda}^{-1}$ is the inverse of $\mathbf{\Lambda_{pp}} - \mathbf{\Lambda_{ps}}\mathbf{\Lambda_{ss}^{-1}}\mathbf{\Lambda_{sp}}$, so the Schur complement's inverse is the marginal parameter covariance. Its eigenvectors are the directions in parameter space the data constrain, and the eigenvector of the smallest eigenvalue is the *flattest* direction, the combination

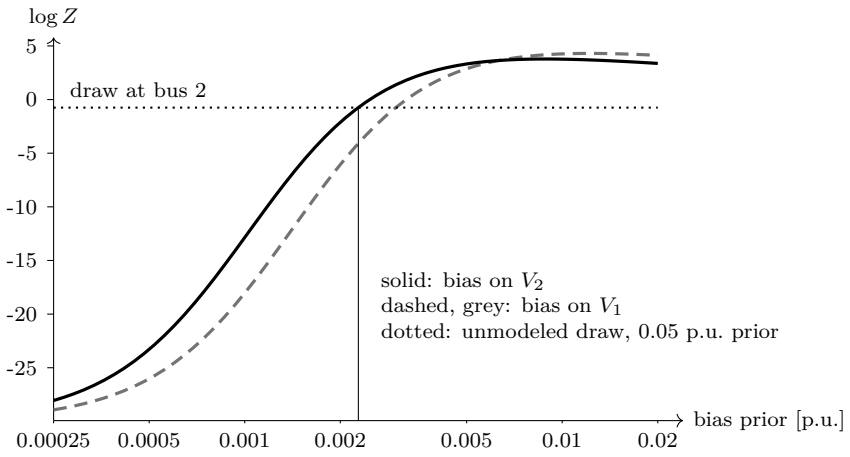


Figure 12.3: Log-evidence of the fault hypotheses for the inconsistent feeder readings, as the width of the sensor-bias priors varies with the draw's prior held at 0.05 per unit. The thin vertical line marks the crossover at 0.0023, below which the draw is the better explanation.

of parameters the data cannot separate. A scale-free summary per parameter is the ratio of its posterior spread to its prior spread, ρ_i: near zero, the data pinned it; near one, the data said nothing.

Free the susceptance b_{12} on the feeder with a prior so wide as to be no prior, and ask what the sensors can do for it. With only the V_2 reading, $\rho_b = 1.000$ and the flattest direction is b_{12} itself: a single voltage a bus beyond the line says nothing about it, because V_2 depends on the injection and the substation voltage and not on how the reactive power gets from the generator to bus 2. Add the V_1 reading and ρ_b falls to a thousandth, with b_{12} known to ± 0.65 per unit: the voltage drop across the line, $V_1 - V_2$, is the injection over the susceptance, and the injection is known from V_2. The flattest direction is then the injection–substation ridge, as it has been since Chapter 3. None of this needed a reading; it needed the sensors' positions and the model.

12.8 Worked example: can a susceptance drift be seen?

A substation busbar feeds a bus through a line, and from there two branches carry reactive power to two buses that draw it. The line's susceptance b falls as the conductor ages and sags, and an operator wants to know when to reconductor. The busbar is held at 1.000 per unit and measured. The two draws Q_a and Q_c are known only from load forecasts, 0.20 ± 0.03 and 0.25 ± 0.04 per unit, and the susceptance has a prior of 12 ± 3 per unit. Voltage sensors of accuracy 0.0005 per unit read both drawing buses. With all three branches at the same susceptance, the voltages fall by flow over susceptance along each one,

$$V_h = 1.000 - \frac{Q_a + Q_c}{b}, \qquad V_a = V_h - \frac{Q_a}{b}, \qquad V_c = V_h - \frac{Q_c}{b},$$

and the feeder carries $Q_a + Q_c$. At the nominal values the two measured voltages are 0.945833 and 0.941667 and the feeder carries 0.4500 per unit. Every number is from `ch12_more.py`.

Proposition 12.2 (A symmetry the voltage sensors cannot see)**.** *The bus voltages are unchanged when b, Q_a and Q_c are multiplied by a common factor. The Fisher information that the voltage*

Table 12.5: Can a susceptance drift be seen? The posterior spread of the line's susceptance for four sensor sets, its ratio to the prior spread, and the shift of the susceptance's estimate caused by a 20 percent drift, in units of its own posterior spread. Flow meters have accuracy 2 percent. All quantities in per unit.

sensors	sd of b	ρ_b	drift seen at
bus voltages	1.21	0.40	2.1 sd
+ meter on the draw at a	0.35	0.12	8.6 sd
+ meter on the draw at c	0.31	0.10	9.6 sd
+ meter on the feeder flow	0.25	0.08	12.1 sd

sensors carry about (b, Q_a, Q_c) therefore has the vector (b, Q_a, Q_c) in its null space, and no number of voltage sensors can resolve that direction.

Proof. Under the scaling by λ, every flow and every susceptance scales by λ, so every ratio of a flow to a susceptance is unchanged. Each voltage is the busbar's value less a sum of such ratios, so the voltages are constant along the ray $\lambda(b, Q_a, Q_c)$, their derivative along it is zero, and the vector lies in the null space of the voltages' Jacobian $\mathbf{J}$ and hence of the Fisher information $\mathbf{J}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{J}$. □

The script confirms it. Scaling all three by 0.8 or by 1.3 leaves the voltages unchanged to six decimals, while the feeder flow moves to 0.3600 or 0.5850 per unit. The smallest eigenvalue of the sensors' Fisher information is 5×10^{-14}, along the scaling ray to three decimals. What the sensors do fix is the ratio of the susceptance to either draw: b/Q_a is known to 2.2 percent. The susceptance itself is known only as well as the draw priors allow, to ± 1.21 per unit, 40 percent of its prior spread (Table 12.5).

Now let the line drift by 20 percent, b from 12 to 9.6 per unit, with the draws unchanged. The voltages move by -0.0135 and -0.0146 per unit, twenty-seven and twenty-nine times the sensors' accuracy, so the change is plainly seen. What the sensors alone cannot say is that it is the line. A 25 percent rise in both draws at the old susceptance gives the same voltages, by the proposition, and the posterior splits the change between the two explanations: the susceptance falls by 2.1 of its own standard deviations and both draws rise a little. Figure 12.4 draws the two posteriors. One meter on either draw breaks the symmetry. The susceptance is then known to ± 0.35 per unit, and the same drift moves its estimate by nearly nine standard deviations with the draws left where they were. A meter on the feeder flow does better still. The question the operator asked, a line that has aged or a load that has grown, is answered by a flow meter and by nothing else.

Principle 12.3. *A transformation of the inputs that leaves every reading unchanged is a direction no data can resolve. Find the symmetries of the physics before choosing sensors; each one needs a sensor that breaks it.*

12.9 Which sensor next?

Chapter 11 chose the next sensor by the variance it would remove from one target. The general criterion is the information it would add about the whole state, and for a Gaussian model it has

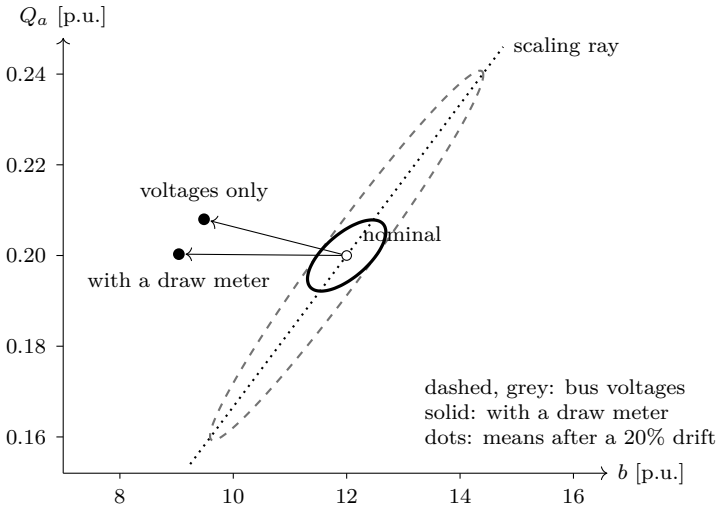


Figure 12.4: The feeder's posterior over the line's susceptance and the draw at bus a, at two standard deviations. Dashed grey: bus voltages only, an ellipse along the scaling ray (dotted), the direction the sensors cannot see. Solid: with a meter on the draw at a. Dots: the posterior means after a 20 percent drift. With voltages only, the change is split between the susceptance and the draws; with the draw meter, it goes to the susceptance.

Table 12.6: Four candidate sensors on the feeder under the prior: expected information gain about the whole state, and the spread of the injection after the reading (the criterion of Chapter 11).

sensor on	accuracy	EIG [nats]	sd of G after
V_1	0.0005	3.36	0.00425
V_2	0.0005	3.04	0.00582
V_s	0.0005	1.81	0.02000
Q_{2s}	0.002	2.31	0.00199

a closed form that, like the variance reduction, does not depend on the reading:

$$\text{EIG} \;=\; \tfrac{1}{2}\log\det(\mathbf{I}+\boldsymbol{\Sigma}\,\mathbf{H}), \qquad \text{EIG} = \tfrac{1}{2}\log\Big(1+\frac{\mathbf{h}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{h}}{\sigma_y^2}\Big) \text{ for one sensor,} \tag{12.2}$$

with $\mathbf{H}$ the Fisher information the candidate contributes. This is the expected reduction in the posterior's entropy, in nats, and choosing the sensor that maximizes it is D-optimal design. For one sensor it is one solve. The one-sensor form is equation (11.4) of Chapter 11; the block form follows the same way, from $\det(\boldsymbol{\Lambda}+\mathbf{H}) = \det\boldsymbol{\Lambda}\,\det(\mathbf{I}+\boldsymbol{\Sigma}\mathbf{H})$.

Table 12.6 sets the two criteria side by side under the prior. They agree on the best single sensor, V_1, and disagree elsewhere: the busbar sensor carries 1.8 nats about the state, since it pins one of the two uncertain inputs outright, and no information at all about the source, since under the prior the two are independent; the flow meter, poor as it is, carries less information in total but nearly all that there is about the source. Information gain is the right criterion when the whole state matters, and the target criterion when one quantity does, and the posterior supplies both from the same solve. For discrete hypotheses, where the value of a reading depends on which

Table 12.7: Laplace validity probes on the tap changer of Chapter 7 with one V_1 reading, in input space. Curvature: objective increment over its quadratic prediction along the widest posterior axis. Regime: the one the mode lies in, and how near the reading puts it to a boundary.

reading y_1	regime at the mode	mode G	curvature at ± 1 sd	at ± 2 sd
1.0300	minimum, far inside	0.04318	1.00, 1.00	1.00, 1.00
1.0375	modulating, far inside	0.06231	1.00, 1.00	1.00, 1.00
1.0340	minimum, near a boundary	0.04864	1.05, 1.00	1.93, 1.00
1.0450	modulating, near a boundary	0.09488	1.00, 1.26	1.00, 3.35

hypothesis it favors, the corresponding quantity is the mutual information between hypothesis and reading, and it is evaluated by quadrature over the predictive mixture of Chapter 7.

12.10 Should the Gaussian be believed?

Every readout above assumed the posterior is Gaussian, exactly for linear rows and by the Laplace approximation otherwise. Chapter 7 named the three ways the approximation fails and promised tests that detect them from the model itself. Two tests, both built from what the mode solve already produced, do it.

Curvature. Along an eigenvector of the posterior covariance, the Laplace posterior predicts that the objective rises by exactly $\frac{1}{2}k^2$ at k standard deviations from the mode, whatever the eigenvalue. Evaluate the true objective there and take the ratio to the prediction: one for a quadratic objective, far from one where a row curves within the posterior's width. Probing the widest directions is enough, since they are where the posterior extends farthest.

Reweighting. Draw from the Laplace Gaussian and weight each draw by the ratio of the true posterior density to the Gaussian's. If the two agree the weights are uniform and the *effective sample size*, $(\sum w)^2 / \sum w^2$, equals the number of draws; where they disagree the weights concentrate and the ratio collapses.

Table 12.7 runs the curvature probe on the tap changer of Chapter 7, at four readings: two that put the mode far inside a regime and two that put it near a boundary. Far inside, the ratio is 1.00 in every direction and at every distance, and it is exactly one rather than nearly one, because within a regime the model is linear and Gaussian and the Laplace posterior is not an approximation at all. That is worth seeing once: the probe's null value is attainable.

Near a boundary it is not. At $y_1 = 1.0340$ the mode sits in the minimum-tap regime with the modulating regime a little above it, and two spreads in that direction the objective is 1.93 times its quadratic prediction, because the map's slope halves across the kink and the true cost rises faster than the Gaussian expects. At $y_1 = 1.0450$ the mode is modulating with the maximum-tap regime a little below, and two spreads down the objective is 3.35 times predicted. Both are one-sided: the ratio stays at 1.00 in the direction that stays inside the regime. That asymmetry is the signature, and a probe that reports only a symmetric width would miss it.

This is the second and third failure modes of §7.4 together, and in this domain they arrive through the regimes rather than through smooth curvature. Chapter 7 measured the smooth kind and found it weak: the exact branch closure, even a hundredth of a per unit from the nose

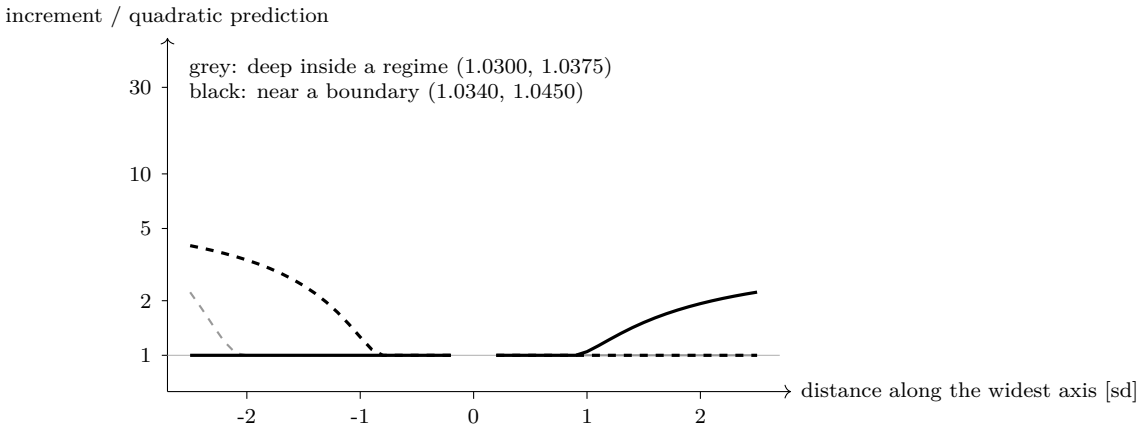


Figure 12.5: The curvature probe on the tap changer as a function of distance along the widest posterior axis: the increment of the objective over its quadratic prediction, on a logarithmic scale, for the four cases of Table 12.7. A Gaussian posterior would give one everywhere, and inside a regime it does.

of its PV curve, keeps the objective quadratic to about one percent. What bends a per-unit posterior out of shape is a controller that changes regime, and the reading that puts the state near that change is exactly the reading an operator is most interested in.

Escalation. When the probes fail the answer is not to return the Gaussian anyway. The cheap correction is importance reweighting itself: the reweighted draws are samples from the true posterior, and their weighted moments correct skew and curvature near the mode, at the cost of the draws and with the effective sample size as the measure of how well. It cannot find a mode it never visits. It also inherits the scale of the Gaussian it draws from, so where that scale is itself wrong the reweighting cannot repair it. Genuine multimodality, the first failure mode, escalates to a sampler that does not start from the Laplace Gaussian at all: a particle filter that draws the parameters from the prior and integrates the variables the physics determines out of each particle, at the cost of one solve per particle. All three return the same readouts, so what changes downstream is only how much they cost.

> **Principle 12.4.** *A posterior is a claim, and the model carries the instruments to test it: the misfit for adequacy, studentized residuals for location, evidence for explanation, the parameter Schur complement for recoverability, and curvature and reweighting for the Gaussian itself. None needs anything the mode solve did not produce.*

12.11 In practice

Run the misfit test first, and report its degrees of freedom. A p-value on zero degrees of freedom does not exist, and a model with none has not been tested by its data. Report the verdict together with the count.

Compute every sensor's redundancy before trusting its residual. Redundancy comes from the posterior's diagonal and needs no reading. A meter with redundancy below about a tenth cannot be checked. The fix is another measurement that covers the same quantity: a second meter across the bridge, or an injection meter at one of its ends.

Look for the physics' symmetries before placing sensors. A common scaling of a susceptance and the flows through it, a common offset of every angle in a model driven by differences, a change of a bridge's susceptance that only its angles feel: each leaves some set of sensors blind. Test each candidate sensor set against them. The flattest direction of the parameter Schur complement finds the symmetries that nobody thought of.

Rank studentized residuals only when the misfit fails, and expect company. The meters around a local fault share its residual. The fault leads the ranking when it is local and redundant; when the ranking is flat, the contradiction is global.

When residuals smear, name hypotheses and write down their priors. Score each by its evidence. The verdict depends on the fault priors, so report them, and report where the verdict would change, as Figure 12.3 does.

Probe the Gaussian where sensors are precise and closures curve. Evaluate the objective at one and two spreads along the widest axes. A ratio beyond about two, or an effective sample fraction below a half, means the readouts need reweighting or a sampler.

A virtual sensor is valid only while the misfit passes. Put the misfit next to every virtual reading an operator sees. When it fails, the virtual readings are the first casualty.

12.12 What is missing

Every question here was asked of the world as it is. The next two chapters ask about the world as it might be: what would happen if a line were taken out or a setpoint changed (Chapter 13), and which uncertain input the answer depends on most (Chapter 14). The first needs an idea the factor graph does not contain, which is what it means to intervene; the second needs the solve to return its gradient.

Notes and further reading

The misfit test and the studentized residual are the chi-squared test and the normalized residual of power-system state estimation, developed for bad-data detection and identification and treated in full by Monticelli [65] and Abur and Gómez Expósito [1]; the correction $\sigma_y^2 - \mathrm{Var}[z]$ in the denominator is theirs. The same statistics are the global test and the measurement test of process data reconciliation [58]. The observation that a global contradiction spreads across every factor, so that residual ranking fails exactly when it is most needed, is old in both fields under the name of smearing, and hypothesis comparison by evidence is the standard remedy in the reconciliation literature as "serial elimination" of suspected gross errors. Measurement redundancy and critical measurements, those whose removal leaves part of the state unobservable,

belong to the observability analysis of state estimation [1, 65]; the meter on the tie line of §12.5 is the textbook case.

The evidence with its Occam factor, equation (12.1), and its use to compare models of different dimension are MacKay's [57]; Kass and Raftery [42] review Bayes factors. Identifiability through the Fisher information's eigenstructure is classical; the parameter Schur complement is its Gaussian-posterior form. Expected information gain and D-optimal design are Lindley's [53] and are reviewed by Chaloner and Verdinelli [13]. The curvature probe is the book's simplest form of a check on the Laplace approximation; the effective sample size under importance reweighting is standard [76], and the escalation to a particle filter over the prior is Gordon, Salmond and Smith's bootstrap filter [27].

Summary

- Every operational question is a readout of one posterior; adding a question adds no model.
- A virtual sensor is the marginal of an unmeasured variable. Two voltage sensors and four rows of physics give the feeder's flows to 0.0029 per unit.
- The misfit is chi-squared with factors minus unknowns degrees of freedom, because the standardized residuals are a projection of rank $m_g - n$ of standard Gaussian noise; 0.07 on two for consistent readings, 70 for inconsistent. Studentized residuals localize local faults; a global contradiction smears across every factor.
- A meter on a bridge of the network, the tie line, has redundancy 0.019. An error of 0.15 per unit on it passes the misfit test at a p-value of 0.50 and moves the loads at the line's ends by 4.3 standard deviations; the same error inside a loop is flagged at 5.7.
- Named hypotheses scored by evidence decide what residuals cannot: for the inconsistent readings, a biased sensor at 99 percent over a draw at one, and the verdict is priced by the fault priors. Below a bias prior of 0.0023 per unit, the draw wins.
- Symmetries of the physics are blind spots of the sensors. Scaling a line's susceptance and the draws through it together leaves the bus voltages unchanged, so voltage sensors alone see a 20 percent drift at 2.1 standard deviations; one flow meter makes it nine. No flow meter sees the susceptance of a bridge; a phasor measurement across it does.
- Recoverability is the parameter Schur complement's flattest direction: b_{12} is invisible to one voltage sensor and known to 0.65 with two.
- Expected information gain is a log-determinant, one solve per sensor; it and the target criterion agree on the best sensor and disagree on the busbar sensor, for a reason the graph explains.
- Curvature probes and reweighting detect a failed Gaussian from the model alone: inside a regime the curvature ratio is exactly one, and a reading that puts the mode near a tap changer's boundary drives it to 1.9 on one side and 3.4 on the other while leaving it at one on the other side.

Exercises

Exercise 12.1. Derive the variance of a sensor's raw residual $y - \mathbb{E}[z]$ under the posterior, $\sigma_y^2 - \mathrm{Var}[z]$, from the rank-one update of Chapter 11, and show it is always positive.

Exercise 12.2. Re-run the hypothesis comparison of Table 12.4 with the sensor-bias priors tightened to one degree. Find the prior width at which the leak at node 2 overtakes the biases, and explain the crossover in terms of equation (12.1)'s terms.

Exercise 12.3. Add a hypothesis to Table 12.4: both sensors biased, each with a five-degree prior. Compute its evidence, split it into the fit term and the rest, and explain how it scores against the single-bias hypotheses.

Exercise 12.4. For the two-region network of Chapter 8 with the tie susceptance freed, compute the parameter Schur complement onto the five loads and the susceptance for each single flow meter, and find the meter for which the susceptance has the smallest ρ. Compare with the expected information gain of each meter.

Exercise 12.5. Implement the profiled curvature probe: instead of moving the full state along an eigenvector, clamp one variable at $\pm k$ spreads and re-minimize the rest. Apply it to the $p = 4$, precise-sensor case of Table 12.7 and compare the ratios with the unprofiled ones; explain why the profiled probe stays on the physics manifold and the unprofiled one need not.

Solutions to selected exercises

Exercise 12.1. Let v be the variance of z under the posterior without the sensor, and $S = \sigma_y^2 + v$ the predicted variance of the reading. By Proposition 11.1 the posterior mean with the sensor is $\mu' = \mu + (v/S)(y - \mu)$, so the raw residual is $y - \mu' = (y - \mu)\,\sigma_y^2/S$. Under the model $y - \mu$ has variance S, so the raw residual has variance σ_y^4/S. The posterior variance with the sensor is $\mathrm{Var}[z] = v - v^2/S = v\sigma_y^2/S$, and $\sigma_y^2 - \mathrm{Var}[z] = \sigma_y^2(S - v)/S = \sigma_y^4/S$, the same number. It is positive whenever the sensor has any noise at all. It falls toward zero as v/σ_y^2 grows, which is a sensor whose quantity nothing else in the model knows: the fit follows the reading, the residual cannot move, and a gross error on it is invisible. Divided by σ_y^2 it is the redundancy $r_k = \sigma_y^2/S$ of §12.5.

Exercise 12.2. At a bias prior of 0.001 per unit the bias on V_2 has $\log Z = -12.86$ against the draw's -0.75, and the draw wins by twelve nats. The crossover is at 0.00228 (Figure 12.3). In the terms of equation (12.1), the fit term $-C(\mathbf{z}^\star)$ of the bias hypothesis is -19.50 at 0.001 and -2.06 at 0.005, because a 0.001 prior can afford only a 0.0043 bias, half of the 0.0085 the readings need, and the rest of the contradiction is paid by the injection prior and the sensors. The volume and normalizer terms together move by much less, from 6.64 at 0.001 to 5.38 at 0.005. Tightening the prior makes the bias cheaper in Occam terms by 1.3 nats and dearer in fit by 17.4, and the fit wins. The draw's terms do not change.

Exercise 12.3. The two-bias hypothesis has $\log Z = 3.87$, against 3.32 for the bias on V_2 alone and 2.88 for the bias on V_1 alone: it wins, by 0.55 nats over the better single bias, odds of 1.7. Split against the single bias on V_2, its fit term is -1.27 against -2.06, better by 0.79, and its volume and normalizer terms are 5.14 against 5.38, worse by 0.24. The Occam penalty for the second fault variable is real but small. The fit improves because two moderate biases, $+0.0049$ and -0.0051 per unit, sit comfortably inside their 0.005 priors where one bias of -0.0085 does not, and because they let the injection return to 0.064 from 0.072. The biases are poorly determined one by one, ± 0.0039 and ± 0.0029, and correlated at 0.94: the readings fix their

difference to ± 0.0016 and leave their sum to the priors. More faults are not always penalized; two small faults can be more probable than one large one, and the evidence says when.

Exercise 12.4. With the tie susceptance freed around 10 ± 2 per unit, every single flow meter leaves its ratio ρ at 1.000: no flow meter says anything about it. The expected information gains of the meters about the whole state still differ, from 1.58 nats for the meter on line 5–6 to 2.86 for the meter on the tie line, and none of that information is about the susceptance. The reason is that the tie is a bridge. The flow on a bridge is the total load beyond it, whatever the bridge's susceptance, and the flows elsewhere are set by the loads and by the susceptances of the loops they lie in, which do not include the tie. The susceptance moves only the angles across the tie, and flow meters do not read angles. A phasor measurement of the angle difference $\theta_3 - \theta_4$, of accuracy 0.001 radians, does. Alone it brings ρ to 0.64, since the flow it would need to compare with is itself uncertain through the loads; together with the tie flow meter it brings ρ to 0.058, since the pair reads the line's own Ohm's law, flow against angle. The meter worth most to the state is worth nothing to this parameter, which is Principle 12.3 in another form.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Where did the kilowatts go? *competing explanations scored by evidence; a freed parameter buys fit and pays Occam; the base rate matters only when the evidence is close*
`foundation_electrical/power_chain_loss_localization/problems/where_did_the_kilowatts_go`

Offset, or loss? *identifiability as a property of the measurement set; a posterior that has collapsed back onto its prior has learned nothing; which meter carries which parameter*
`foundation_electrical/power_chain_loss_localization/problems/offset_or_loss`

A virtual sensor on the unmetered leg *a quantity nobody measures; estimated from conservation; the width of a recovered estimate as the propagated width of its neighbours*
`foundation_electrical/power_chain_loss_localization/problems/virtual_sensor_on_the_unmetered_leg`

Three explanations for one feeder *model comparison over a shared physics; a hypothesis that survives only by positing an implausible value; the base rate against the evidence*
`subsystem_power_flow/state_estimation_diagnostics/problems/feeder_explanations_ranked`

The next measurement to buy *the value of a measurement before it is taken; expected entropy reduction rather than accuracy; measuring the quantity the hypotheses actually disagree about*
`subsystem_power_flow/state_estimation_diagnostics/problems/the_next_meter_to_buy`

The meter that reads wrong *detection against identification; the chi-square of a pooled fit; the normalized residual of a single reading*
`power_grid/bad_data_ieee14/problems/the_meter_that_reads_wrong`

The test that does not see it *a pooled statistic dilutes a single outlier; detection power against identification power; the order in which to run two tests*
`power_grid/bad_data_ieee14/problems/the_test_that_does_not_see_it`

Dropping the suspect *what removing a reading buys and what it costs; redundancy measured rather than assumed; a meter set so redundant that losing one costs almost nothing*
`power_grid/bad_data_ieee14/problems/dropping_the_suspect`

Which branch drifted? *a fault as a free parameter rather than a discrete event; ranking branches by evidence; shrinkage of a maximum-a-posteriori estimate toward its prior*
`power_grid/branch_fault_localization/problems/which_branch_drifted`

Chapter 13

Interventions and counterfactuals

An operator does not only ask what is. An operator asks what would happen: if that line were opened, if this load were curtailed, if the setpoint were raised. Those are questions about a world that has been acted on, and a factor graph, which encodes compatibility among variables, does not by itself say what acting on the world does to it. This chapter supplies what is missing. The physics rows of Chapter 1 are mechanisms; the inputs of Chapter 4 are the quantities nothing in the model determines; and an intervention is either the imposition of an input or the replacement of a named mechanism by another. With that semantics, seeing and doing become different operations on the same graph, a counterfactual becomes three steps of which two are already familiar, and a decision becomes a tree of branches that share one posterior up to the point where they diverge. Each is worked on the book's models, and the difference between seeing and doing comes out in numbers.

13.1 Why the graph is not enough

Chapter 2 chose the factor graph over the directed graph because a closure has no direction: $Q = UA\,(T_h - T_c)$ relates three quantities and computes none of them. That choice served every chapter since. It has a price, and the price falls due here.

Conditioning on a reading is well defined on any factor graph: multiply by the likelihood, renormalize. It answers "given that I have seen $V_2 = 1.027$, what do I believe?" It does not answer "if I make $V_2 = 1.027$, what will happen?" The two differ because seeing $V_2 = 1.027$ is evidence about everything that could have caused it, the injection and the substation voltage among them, while making $V_2 = 1.027$ causes it by some means and leaves the injection and the substation voltage as they were. To say what making does, one must say what mechanism was used to make it, and that is a statement about which relation in the model was overridden. The undirected graph does not contain that statement. Pearl's directed models contain it as the arrows: intervening on a variable severs the arrows into it and leaves the rest. The book's models have no arrows to sever, and something has to stand in for them.

13.2 Rows are mechanisms; inputs are exogenous

What stands in for them is the distinction Chapter 4 already drew. The variables of a model split into *inputs* $\mathbf{u}$, the quantities that carry priors because nothing in the model determines them (parameters, boundary conditions, sources, discrete availabilities), and *solved variables* $\mathbf{x}$,

which the physics determines from the inputs. Each physics row is a *mechanism*: a relation the system enforces, and one that could be enforced differently, by a different device, a different law, a controller. The well-posedness of §1.5 says that the rows, taken together, determine the solved variables from the inputs; no single row determines any single variable, but the set determines the set.

Definition 13.1 (Intervention)**.** An intervention on a physical model is one of two operations:

1. *Impose an input*: replace the prior on an input u_j by the fixed value $u_j = v$. The physics rows are untouched.
2. *Replace a mechanism*: remove a named physics row r_k and add in its place a new row r'_k that the intervening device enforces. The inputs and every other row are untouched.

The posterior after an intervention is the posterior of the modified model, with the evidence that was conditioned on before the intervention retained only where the intervention has not made it inapplicable.

The first operation needs no further choice; an input is exogenous by construction, and fixing it is what "do" means. The second needs a choice, and the definition makes the modeler state it: which row is replaced, and by what. "Make $V_2 = 1.027$" is not an intervention until it says whether bus 2 is held at 1.027 by a compensator that supplies whatever reactive power is needed, which replaces the branch closure, or by a tap changer that moves until bus 2 reads 1.027, which replaces the substation voltage's prior. These are different devices, they leave different things unchanged, and they lead to different posteriors. A directed graph would have committed to one of them when its arrows were drawn. The factor graph defers the commitment to the moment of intervention, which is where it belongs.

> **Principle 13.1.** *Seeing is conditioning. Doing is replacing: an input's prior by a value, or a named mechanism by another. A value alone does not specify an intervention; the mechanism that imposes it does, and the modeler must name it.*

This is a structural-causal-model layer laid over the acausal graph. In Pearl's terms the physics rows are the structural equations, the inputs are the exogenous variables, and the replacement of a row is the surgery that "do" performs; what is different is that the equations are not written with an output variable, so the surgery must name the equation rather than the variable. That is not a limitation. It is the honest form of the operation for a physical system, in which the same variable can be set by different means.

13.3 Seeing versus doing, on the feeder

Take the export feeder with hard physics and the priors of Chapter 3. Every number is from the script `ch13_interventions.py` in the book's `scripts` directory. Table 13.1 sets four posteriors side by side.

Seeing. A voltage sensor on bus 2 reads 1.027. The injection's belief tightens from ± 0.020 to ± 0.0058 and moves up; the substation voltage's moves a little; the two become correlated; V_1 is predicted to ± 0.0013. The reading is evidence about its causes, and the posterior distributes it among them in proportion to how much each could have contributed. This is Chapter 3's result.

Table 13.1: The feeder: prior predictive; conditioning on a reading of V_2; two interventions that both make $V_2 = 1.027$ by different mechanisms; and doing versus seeing on an input. All quantities in per unit.

	Q_{2s}	V_1	V_2	G	V_s
prior predictive	0.050 ± 0.020	1.0350 ± 0.0143	1.0250 ± 0.0104	0.050 ± 0.020	1.0000 ± 0.0030
see V_2 (acc. 0.0005)	0.0537 ± 0.0058	1.0377 ± 0.0013	1.0270 ± 0.0005	0.0537 ± 0.0058	1.0002 ± 0.0029
do, compensator replaces c_{2s}	0.050 ± 0.020	1.0370 ± 0.0040	1.0270	0.050 ± 0.020	1.0000 ± 0.0030
do, tap changer replaces π_{V_s}	0.050 ± 0.020	1.0370 ± 0.0040	1.0270	0.050 ± 0.020	1.0020 ± 0.0100
do $G = 0.06$		1.0420 ± 0.0030	1.0300 ± 0.0030	0.060	1.0000 ± 0.0030
see $G = 0.06$ (acc. 0.002)		1.0419 ± 0.0033	1.0300 ± 0.0032	0.0600 ± 0.0020	1.0000 ± 0.0030

Doing, one way. A compensator at bus 2 holds it at 1.027, absorbing or injecting whatever reactive power it must. The mechanism replaced is the branch closure, which no longer describes how reactive power leaves bus 2. The injection is untouched: 0.050 ± 0.020, as under the prior. So is the substation voltage. The flow down the upper line is still the injection, by the balances, and the generator-bus voltage is 1.027 plus the injection over the susceptance: 1.0370 ± 0.0040, its spread now the injection's spread through the susceptance rather than the reading's. Nothing upstream of bus 2 has learned anything, because nothing was observed.

Doing, another way. The tap changer moves until bus 2 reads 1.027. The mechanism replaced is the substation voltage's prior: it is no longer a set point with a deadband but a control variable set to whatever bus 2 needs, $V_s = 1.027 - G/b_{2s}$, which is 1.0020 ± 0.0100 with the injection's uncertainty passed through. The generator-bus voltage and the injection are exactly as under the first intervention, since bus 2 is at 1.027 and the injection is untouched either way; the substation voltage is entirely different, and so is the physical device. Both are "do $V_2 = 1.027$". Neither is "see $V_2 = 1.027$".

Doing on an input. Set the injection to 0.06 per unit. There is no mechanism to name: the injection is an input and setting it replaces its prior. The voltages follow: $V_2 = 1.0300 \pm 0.0030$, the substation voltage's spread passed through. Now instead *observe* the injection at 0.06 with a meter accurate to 0.002, and condition. The posteriors are the same to within the meter's accuracy. For an exogenous input, seeing and doing coincide, because an input has no causes in the model for a reading of it to be evidence about. That is the sense in which the inputs are exogenous, and it is why the first operation of Definition 13.1 needs no choice.

Proposition 13.1 (Seeing and doing on an input)**.** *Let u_j be an input whose prior is a factor on u_j alone. Whatever other evidence the model holds, the posterior over every other variable after a noiseless reading $u_j = v$ equals the posterior after imposing $u_j = v$.*

Proof. Conditioning on the reading multiplies the joint density by $\delta(u_j - v)$ and renormalizes. Under the delta, the prior factor $p_j(u_j)$ becomes the constant $p_j(v)$, which cancels in the normalization. Imposing the value replaces $p_j(u_j)$ by $\delta(u_j - v)$ directly. The two unnormalized products differ by the constant factor $p_j(v)$, so their normalized versions are the same. □

For a solved variable the argument fails, because conditioning keeps every row and doing removes one. The removed row carried information about the other variables, and the feeder's two interventions show how much: each removes a different row and each leaves a different

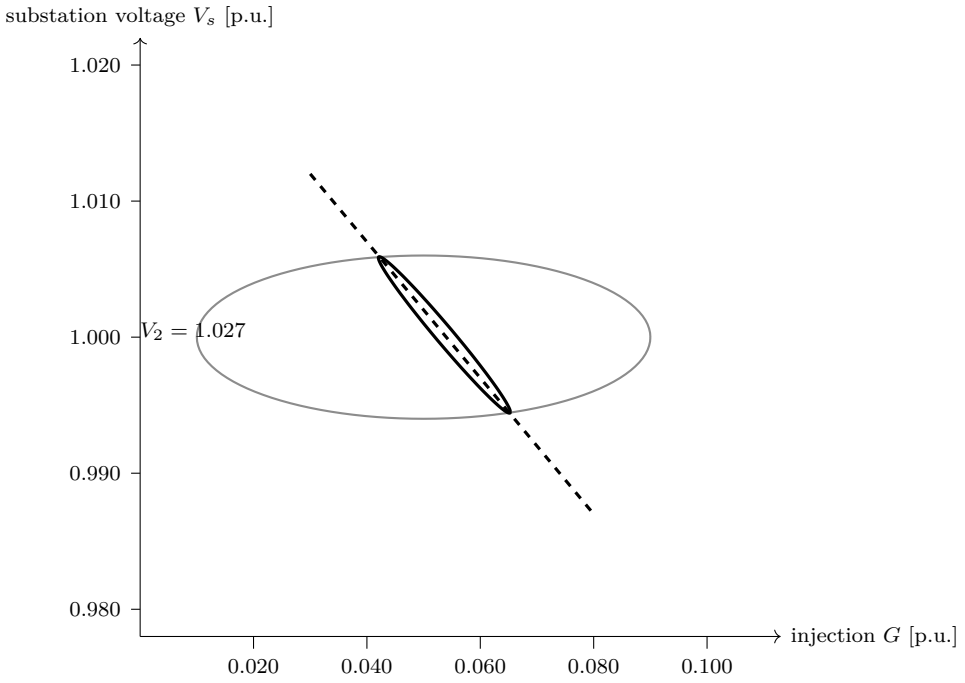


Figure 13.1: Seeing and doing on the feeder, in the plane of the two inputs, at two standard deviations. Grey: the prior, which is also the input posterior after a compensator holds $V_2 = 1.027$. Black: after seeing $V_2 = 1.027$, a thin ridge. Dashed: after the tap changer holds $V_2 = 1.027$, the line $V_s = 1.027 - G/2$, along which the injection keeps its prior.

posterior. Figure 13.1 draws the three in the plane of the two inputs, which are what the worlds share. Seeing bus 2 at 1.027 confines the inputs to a thin ridge, $G/2 + V_s \approx 1.027$, correlated at -0.985: the reading is evidence, and it moves the belief about the causes. Doing it with a compensator leaves the inputs at their prior, since nothing was observed. Doing it with the tap changer puts them on the line $V_s = 1.027 - G/2$ exactly, since the substation voltage is now whatever the injection requires. The injection keeps its prior spread along the line, which is why the substation voltage's spread becomes ± 0.010.

13.4 Counterfactuals: three steps

A counterfactual asks what would have happened, given what did. It combines evidence about the world as it is with an intervention on the world as it might be, and the combination has a definite order.

1. *Abduction.* Condition on the evidence in the factual model and read off the posterior of the inputs. This is what the evidence says about the exogenous quantities, which are what the two worlds share.
2. *Action.* Apply the intervention: impose the input or replace the mechanism.
3. *Prediction.* Push the abduced posterior of the inputs through the intervened model, with the evidence removed, since it was evidence about the factual world.

The evidence is used once, in step one, to learn about the inputs. It is not used in step three, because in the counterfactual world the readings would have been different; what carries over is

Table 13.2: The triangle: factual posterior with the meter reading; the counterfactual outage by abduction, action and prediction; and the naive re-conditioning of the outage model on the same reading.

	F_{12}	F_{23}	P_2	P_3
factual: line in, $y_{12} = 1.25$	1.25 ± 0.02	-0.35 ± 0.09	-1.60 ± 0.09	-0.55 ± 0.18
counterfactual: line out, loads abduced	1.60 ± 0.09	0	-1.60 ± 0.09	-0.55 ± 0.18
naive: line out, re-conditioned on y_{12}	1.25 ± 0.02	0	-1.25 ± 0.02	-0.50 ± 0.20

not the readings but what they revealed about the loads, the parameters, the sources.

Proposition 13.2 (Counterfactuals in a linear model)**.** *Let the physics rows be linear and hard, let the factual posterior of the inputs be $\mathcal{N}(\mathbf{m}_u, \mathbf{\Sigma}_u)$, and let the intervened rows determine the solved variables as $\mathbf{x}' = \mathbf{A}'\mathbf{u} + \mathbf{b}'$. Then the counterfactual distribution of $\mathbf{x}'$ is $\mathcal{N}(\mathbf{A}'\mathbf{m}_u + \mathbf{b}',\ \mathbf{A}'\mathbf{\Sigma}_u\mathbf{A}'^{\mathsf{T}})$. The naive computation conditions the prior of the inputs on the readings through the intervened map in place of the factual one, and it agrees with the counterfactual only when the intervention leaves the readings' dependence on the inputs unchanged.*

Proof. Abduction conditions the factual model on the readings. The inputs' posterior is Gaussian, since the factual model is linear-Gaussian, and it is $\mathcal{N}(\mathbf{m}_u, \mathbf{\Sigma}_u)$. After the action the solved variables are the linear function $\mathbf{A}'\mathbf{u} + \mathbf{b}'$ of the inputs, by the well-posedness of the intervened rows, and a linear image of a Gaussian is Gaussian with the mean and covariance stated. The naive computation treats a reading $y = \mathbf{h}^{\mathsf{T}}\mathbf{x}$ as $\mathbf{h}^{\mathsf{T}}(\mathbf{A}'\mathbf{u} + \mathbf{b}')$ where the factual world had $\mathbf{h}^{\mathsf{T}}(\mathbf{A}\mathbf{u} + \mathbf{b})$. The two conditionings coincide exactly when $\mathbf{h}^{\mathsf{T}}\mathbf{A}' = \mathbf{h}^{\mathsf{T}}\mathbf{A}$ and $\mathbf{h}^{\mathsf{T}}\mathbf{b}' = \mathbf{h}^{\mathsf{T}}\mathbf{b}$. □

The triangle with a line out. Line 1–2 is metered at 1.25 with the loop intact and the loads uncertain as in Chapter 7. What would the flows have been if line 2–3 had been out? Table 13.2 gives the three answers one could compute.

Abduction: the reading, with the loop intact, says the load at bus 2 is -1.60 ± 0.09 and the load at bus 3 -0.55 ± 0.18, correlated at -0.94 since the meter sees a combination of them. Action: replace the closure of line 2–3 by $F_{23} = 0$. Prediction: with the loop open, bus 2 is served by line 1–2 alone, so the metered line would carry the whole of bus 2's load, 1.60 ± 0.09, a third more than it does, with the load's uncertainty passed through. That is the counterfactual, and it is what an operator wants to know before opening the line.

The naive computation conditions the outage model on the reading, as if the meter had read 1.25 with the line already out. It concludes that the metered line would carry 1.25, which is what it does now, and that bus 2's load is -1.25, which contradicts what the factual world established. It has used the reading as evidence in a world where the reading would not have occurred. The difference between the two rows is the whole content of the word "counterfactual", and it is a third of a per-unit on the line in question. In the terms of Proposition 13.2, the outage makes bus 2 radial, so the row of $\mathbf{A}'$ for the metered flow picks out the load at bus 2 alone, and the counterfactual flow is that load's abduced posterior with its sign changed. The meter's row under the outage reads $-P_2$ where the factual one read a mixture of both loads, so the naive computation is conditioning on a different functional of the inputs. Figure 13.2 draws the counterfactual flow against the limit: 49 percent of it lies beyond. The naive answer, a spike at the factual flow, would have told the operator the line was safe.

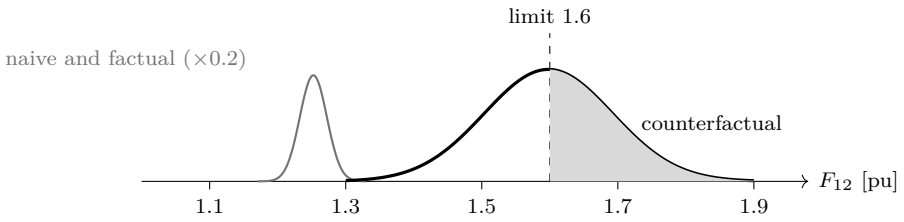
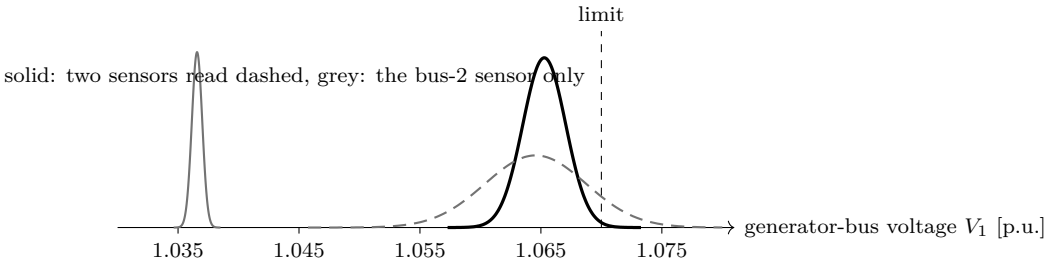


Figure 13.2: The flow on line 1–2 had line 2–3 been out, given the factual reading of 1.25 with it in. Black: the counterfactual, 1.60 ± 0.09, with the 49 percent beyond the limit shaded. Grey, drawn at a fifth of its height: the naive re-conditioning, which coincides with the factual flow.



*imes*0.25

Figure 13.3: The feeder's generator-bus voltage had one circuit of the substation line been out. Solid: the counterfactual with both sensors in the factual world, 1.0653 ± 0.0018. Dashed: with the bus-2 sensor only, 1.0646 ± 0.0042, of which 9.7 percent lies beyond the 1.070 limit. Grey, drawn at a quarter of its height: the naive re-conditioning at 1.0366, which bends the inputs to fit the old readings.

> **Principle 13.2.** *A counterfactual learns the inputs from the factual evidence, changes the mechanism, and predicts with the evidence removed. Re-conditioning the changed model on the old evidence answers a different question, and usually the wrong one.*

The feeder with a circuit out. The same distinction with a closure in place of a line. Both voltage sensors of the feeder read, 1.027 and 1.038, with the substation line's susceptance at 2 per unit. What would the voltages have been had one of its two circuits been out, the susceptance at 1? Abduction gives the inputs the feeder's posterior: the injection at 0.05463 ± 0.00286 and the substation voltage at 0.99973 ± 0.00173, correlated at -0.979. Action replaces the branch closure by $Q_{2s} = 1.0\,(V_2 - V_s)$. Prediction pushes the abduced inputs through: bus 2 would have been at 1.0544 ± 0.0012 and the generator bus at 1.0653 ± 0.0018, with a probability of 0.004 of exceeding 1.070 per unit. The naive computation re-conditions the one-circuit model on the two readings and concludes that bus 2 would have read 1.0287, the generator bus 1.0366, the injection 0.039 and the substation voltage 0.989. It has bent the inputs until the old readings fit the weaker line, which answers the question of what inputs would have produced these readings with a circuit out, and nobody asked it. With only the bus-2 sensor in the factual world the counterfactual generator-bus voltage is 1.0646 ± 0.0042 and the probability of exceeding 1.070 is 0.097. The second sensor, by pinning the ridge, cuts that probability by a factor of about twenty-four: a counterfactual is only as sharp as its abduction. Figure 13.3 draws the three answers.

Table 13.3: Four actions on the triangle, evaluated on the shared abduced posterior of the loads. Limit on line 1–2: 1.6 per unit.

action	F_{12}	$\mathbb{P}(F_{12} > 1.6)$	F_{13}
do nothing	1.25 ± 0.02	0.000	0.90
open line 2–3	1.60 ± 0.09	0.49	0.55
curtail 0.3 at bus 2	1.05 ± 0.02	0.000	0.80
open 2–3 and curtail 0.3	1.30 ± 0.09	0.001	0.55

13.5 Decisions as branches

An operator choosing among actions asks the counterfactual question once per action, and the three steps have a structure that makes this cheap. Abduction is done once: the posterior of the inputs, which every action shares, is the one computation on the factual evidence. Each action is then a branch: an intervened model, evaluated with the shared input posterior. The branches form a tree whose root is the factual posterior, whose edges are interventions, and whose leaves are the worlds that result; a second intervention under a first is a child branch, and a contingency that may or may not occur under an action is a pair of children weighted by its probability. The readouts of Chapter 12 apply at every leaf, and a weighted reduction over leaves, a mean, a tail quantile, an expected shortfall, gives the action's risk.

The do-nothing baseline. One branch is always the identity: no intervention, the factual posterior carried forward. It is the baseline against which every action is judged, and its inclusion is not a formality. An action is worth taking only if its leaf is better than the baseline's, and "better" is a reduction over the leaf's posterior that must be computed for the baseline too, with the same inputs and the same evidence. Recommending an action without the do-nothing leaf beside it is recommending it against nothing.

Four branches on the triangle. The quantity at risk is the flow on line 1–2 against a limit of 1.6 per unit. Four actions: nothing; open line 2–3; curtail bus 2 by 0.3; both. Table 13.3 gives each leaf's flow and the probability of exceeding the limit, all from the one abduced posterior of the loads.

Opening the line alone puts the metered line at its limit with even odds of exceeding it; the loop was carrying a third of bus 2's load around the other way. Curtailment alone is safe and unnecessary. Opening the line with curtailment is safe, at a tenth of a percent, and it is the branch an operator who needs the line out for maintenance would choose. The four leaves cost four solves on a shared prior; on a large network they cost four region messages, with the abduction and every region the actions do not touch reused across the tree, as Chapter 8 arranged.

Figure 13.4 draws the tree, with a contingency added under both actions that open line 2–3 (Exercise 13.4): line 1–3 fails with probability 0.02. With both lines out, bus 3 is islanded and its load, 0.55 ± 0.18 per unit, is shed. The flow on line 1–2 is unchanged, since bus 2 was already fed by that line alone. The contingency leaves the probability of an overload where it was and adds an expected shed load of 0.011 per unit to both branches. A leaf must report every quantity that matters, not only the one that motivated the tree.

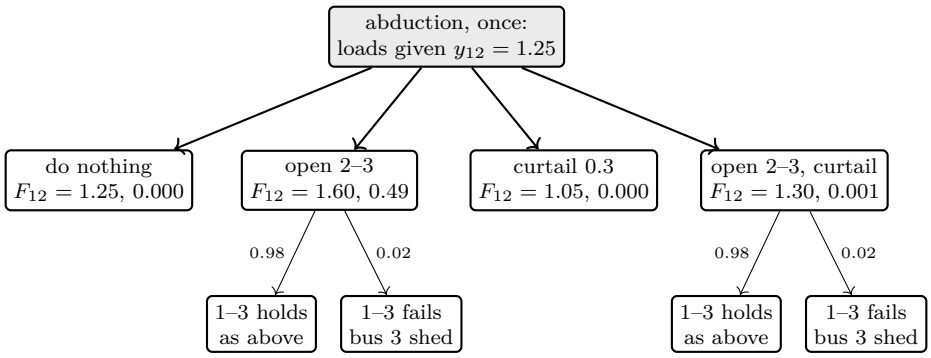


Figure 13.4: The decision tree of Table 13.3. The root is the abduction, done once; each edge below it is an intervention; each leaf is evaluated on the shared input posterior and shows the flow on line 1–2 and the probability that it exceeds its limit of 1.6. Under the two actions that open line 2–3, a contingency splits the leaf: line 1–3 fails with probability 0.02, islanding bus 3.

Table 13.4: Five branches on the generation areas in a heat wave, on 200,000 shared draws from the abduced inputs: the expected unserved load, the probabilities of any shortfall and of one above 50 megawatts, the change against doing nothing, and that change's standard error computed on the shared draws and on independent ones.

action	$\mathbb{E}[U]$ [kW]	$\mathbb{P}(U > 0)$	$\mathbb{P}(U > 50)$	change [kW]	s.e., shared	s.e., independent
do nothing	30.45	0.554	0.263	—	—	—
standby unit	2.30	0.085	0.013	−28.15	0.086	0.093
export rating +10%	30.14	0.545	0.261	−0.31	0.007	0.128
shed 100 kW	2.27	0.081	0.014	−28.17	0.078	0.094
standby unit and rating	2.30	0.085	0.013	−28.15	0.086	0.093

13.6 Worked example: an afternoon heat wave

The two generation areas of Chapter 10 face a hot afternoon. The forecast puts the ambient temperature at 26.5 ± 0.7°C, and that is the abduction here: the one piece of evidence, folded into the inputs' posterior, while the unit ratings and export ratings keep their priors. The pocket's load is 2,100 megawatts. Besides doing nothing, the operator has four actions: start the standby unit in area 1, a fourth machine rated like the others; raise area 2's export rating by 10 percent; shed 100 megawatts of load; or start the standby unit and raise the export rating. Each is a branch, and each is evaluated on the same 200,000 draws from the abduced inputs, which is the Monte Carlo form of a shared posterior. Table 13.4 gives the leaves.

Doing nothing leaves an even chance of a shortfall and a one-in-four chance of more than 50 megawatts. The standby unit cuts the expected shortfall from 30.5 to 2.3 megawatts and the chance of any to 8.5 percent, as much as shedding 100 megawatts of load does, without shedding anything. Raising the export rating does almost nothing, 0.31 megawatts, and adding it to the standby unit does nothing at all. The reason is in the do-nothing leaf: at this ambient temperature both areas are limited by their units in 93 percent of the draws, and a wider export path out of a unit-limited area carries no more power. The model has answered a question the operator did not ask, which constraint binds, and that answer decides between two actions that sound equally plausible.

The last two columns are about the arithmetic of branches. The decision rests on the

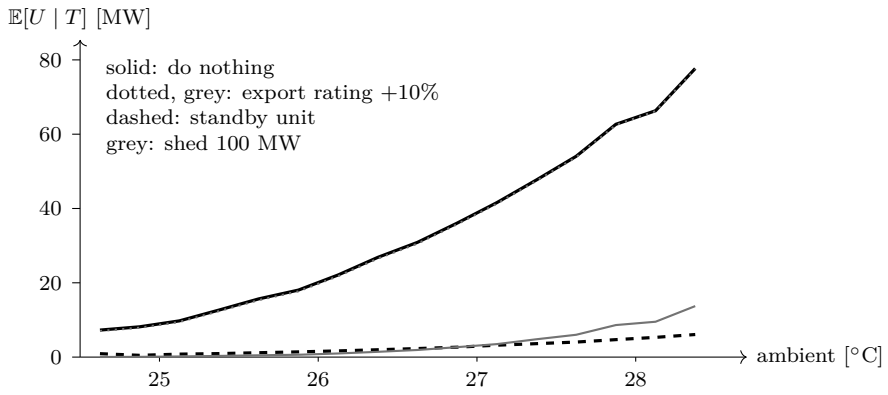


Figure 13.5: The expected unserved load given the ambient temperature, for four branches of Table 13.4, read from the shared draws in bins of a quarter degree. The export rating's curve lies on the do-nothing curve. The standby unit and the load shed cross: shedding does slightly better on milder afternoons and the standby unit on the hottest.

differences between leaves, and computing every leaf on the same draws makes the noise of a difference small where the actions are similar. The export rating's effect, -0.31 megawatts, has a standard error of 0.007 on the shared draws and 0.128 on independent ones: the independent route would need 388 times the draws to resolve it as well. For the standby unit, whose leaf is far from the baseline, sharing the draws helps little, since the difference is almost all of the baseline's own variability. This is the sampling counterpart of sharing the abduction: the branches share not only the posterior but the samples of it, and the comparison is sharper for it.

Proposition 13.3 (Shared draws)**.** *Let two branches be evaluated on N draws each, giving estimates $\bar{U}_a$ and $\bar{U}_0$ of their means, and let σ_a and σ_0 be the spreads of their outcomes. With independent draws,* $\mathrm{Var}(\bar{U}_a - \bar{U}_0) = (\sigma_a^2 + \sigma_0^2)/N$*. With shared draws it is $(\sigma_a^2 + \sigma_0^2 - 2\rho\,\sigma_a\sigma_0)/N$, where ρ is the correlation of the two outcomes across draws. Sharing helps exactly when $\rho > 0$.*

Proof. With shared draws the difference of the means is the mean of the N differences $U_a(\mathbf{x}_n) - U_0(\mathbf{x}_n)$, whose variance is $\mathrm{Var}(U_a - U_0)/N = (\sigma_a^2 + \sigma_0^2 - 2\,\mathrm{Cov}(U_a, U_0))/N$. With independent draws the two means are independent and their variances add. □

For the export rating, the two leaves differ by a third of a megawatt on draws whose shortfall varies by tens of megawatts, so ρ is nearly one and sharing is worth a factor of 388. For the standby unit the leaves are nearly uncorrelated, since the unit removes most of the shortfall wherever it occurs, and the factor is close to one.

Figure 13.5 splits the leaves by the ambient temperature, a partial sum on the shared draws. The export rating's curve lies on the do-nothing curve at every ambient temperature. The standby unit and the load shed are close overall, but they are not the same action. On a milder afternoon a shortfall happens when some area's units are weak, and area 1's loop caps what a standby machine there can add; shedding load helps wherever the shortfall comes from, and it does slightly better. On the hottest afternoons both areas are derated together, the standby machine adds as much as area 1's export path allows, about 130 megawatts at the mean rating, more than the 100 shed, and it does better.

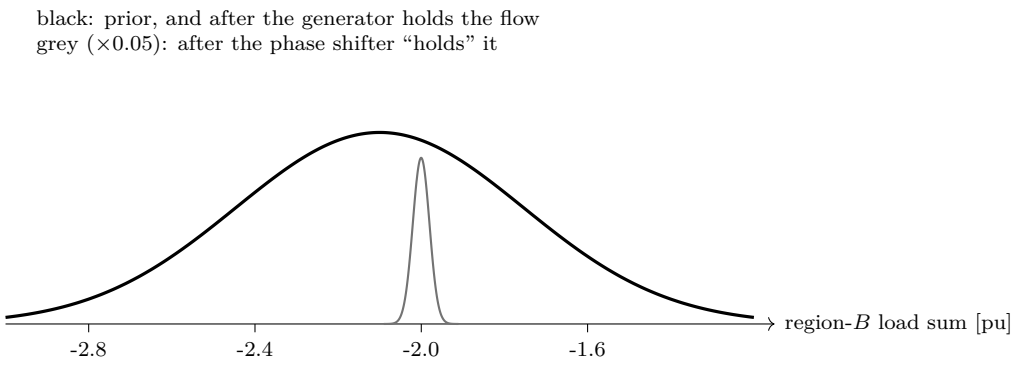


Figure 13.6: The sum of the region-B loads, with nothing observed. Black: the prior, which is also the posterior after the generator at bus 4 holds the tie flow at 2.0. Grey, drawn at a twentieth of its height: after the phase shifter is told to hold it, a spike at -2.000. The device has conditioned the exogenous loads, which no physical device can do.

13.7 Worked example: holding the tie flow

An operator wants the tie line of the two-region network held at 2.0 per unit, below its expected 2.10. Definition 13.1 requires a device, and two are candidates. A phase-shifting transformer on the tie adds an adjustable angle ϕ to the tie's closure, $F_{34} = B(\theta_3 - \theta_4 + \phi)$, and its controller sets ϕ to hold the flow. A controllable generator at bus 4 adds an injection g_4 to that bus's balance, and its controller sets g_4 to hold the flow. Each intervention adds the controller's target, $F_{34} = 2.0$, and frees the control variable it acts through. Every number is from `ch13_more.py`.

The generator does what was asked. The loads are untouched: the sum of the region-B loads is -2.10 ± 0.35, as before the intervention. The generator's output is 0.10 ± 0.35 per unit; it absorbs whatever the loads do. The phase shifter does something else. With nothing observed, the posterior of the region-B loads moves: their sum goes from -2.10 ± 0.35 to -2.000 ± 0.020, and each load rises by 0.033 with its spread falling from 0.20 to 0.16. The phase angle, meanwhile, is left wherever its prior put it, ± 1000 radians: the model has no use for it. The loads are exogenous, and an intervention has conditioned them. The reason is the bridge. The flow on a bridge is the total load beyond it, fixed by the balances whatever the angles, so no phase shift can change it. The model says so twice, by leaving ϕ undetermined and by bending the loads to meet the target. Figure 13.6 draws the region-B load sum under the two devices.

Proposition 13.4 (Realizable interventions)**.** *Let the physics rows be linear. Let an intervention replace mechanism rows, possibly adding control variables. If the modified rows determine the solved and control variables uniquely for every value of the inputs, then, with no evidence, the posterior of the inputs after the intervention is their prior. If they determine them only for inputs on a proper subspace, the posterior of the inputs is confined to that subspace.*

Proof. Write the modified rows as $\mathbf{G}_x\mathbf{x} + \mathbf{G}_u\mathbf{u} = \mathbf{c}$, with $\mathbf{x}$ now including the control variables. If $\mathbf{G}_x$ is square and nonsingular, then $\mathbf{x} = \mathbf{G}_x^{-1}(\mathbf{c} - \mathbf{G}_u\mathbf{u})$ for every $\mathbf{u}$, and the joint density is the prior of $\mathbf{u}$ times a delta on $\mathbf{x}$ whose integral over $\mathbf{x}$ is the constant $1/|\det \mathbf{G}_x|$. The marginal of $\mathbf{u}$ is its prior. If $\mathbf{G}_x$ has a left null vector $\boldsymbol{\ell}$, then $\boldsymbol{\ell}^\mathsf{T}(\mathbf{c} - \mathbf{G}_u\mathbf{u}) = 0$ is a condition on the inputs alone, which every solution requires, and the marginal of $\mathbf{u}$ is the prior restricted to that hyperplane. With soft rows the restriction is a Gaussian of the rows' width about the hyperplane. □

For the phase shifter, the left null vector sums the balances of region B and the target row: the balances say $F_{34} + P_4 + P_5 + P_6 = 0$, the target says $F_{34} = 2.0$, and together they say $P_4 + P_5 + P_6 = -2.0$, a condition on inputs alone. That is the hyperplane to which the loads were confined, at the width of the physics rows.

Principle 13.3. *An intervention that moves the posterior of an exogenous input, with nothing observed, is not realizable by the named device. Check it before reading any other number: after an intervention, and before any evidence, the inputs must be where their priors put them.*

The generator's intervention sits inside region B: region A's message to the tie is unchanged and is reused, and only region B's is recomputed, with g_4 in it. That is the reuse of Chapter 8 applied to decisions: an action changes the messages of the regions it touches and no others.

13.8 What the semantics does not decide

Three things are outside what Definition 13.1 settles, and the reader should know where the line is.

It does not decide which row a real device replaces. A compensator at bus 2 replaces the branch closure in the example because the example said so; a real compensator also has losses, a limited range, its own dynamics, and a failure rate. The definition requires the modeler to state the replacement; it does not validate it.

It does not decide what evidence survives an intervention. The rule given, that evidence is retained where the intervention has not made it inapplicable, is a rule about which readings still describe the counterfactual world, and applying it requires judgment: a meter on a line that is opened reads nothing afterward; a meter on a line elsewhere reads something, but not what it read before. The three-step recipe sidesteps the question by using the evidence only in abduction, which is the safe default, and the modeler who wants to retain a reading in the counterfactual world should say why it would still hold.

And it does not, by itself, quantify how well the model's mechanisms correspond to the world's. An intervention computed on a wrong model is a wrong prediction with an honest error bar; the diagnostics of Chapter 12 say whether the factual model fits, and nothing in this chapter says whether the counterfactual one would.

13.9 In practice

Name the device, then the row it replaces. "Set V_2 to 1.027" is a wish. "A compensator at bus 2 replaces the branch closure" is an intervention. Write the second form down for every action, including the control variable the device acts through.

Check that the inputs did not move. Before reading any other number, compare the inputs' posterior after the intervention, with no evidence, against their prior. If it moved, the device cannot do what the row claims, as the phase shifter on a bridge could not.

Abduce once, branch many, and keep the do-nothing leaf. The abduction is the only step that sees the evidence, and every branch shares it. The do-nothing branch is the baseline every recommendation is judged against.

Drop the evidence in prediction. A reading describes the factual world. Carry it into a counterfactual only when you can say why the device leaves it unchanged, and say so in the report.

Evaluate every branch on the same draws. When the leaves are computed by sampling, share the samples across branches, as the branches share the abduction. For two actions whose leaves are close, this is worth hundreds of times the draws; for actions far apart it costs nothing.

Report every quantity at every leaf. A contingency may leave the quantity that motivated the tree unchanged and move another, as line 1–3's failure left the overload probability alone and shed the load at bus 3.

Sharpen the abduction to sharpen the counterfactual. A counterfactual inherits the spread of the inputs' posterior. The second sensor on the feeder cut the probability of the one-circuit generator-bus voltage exceeding its limit from 0.097 to 0.004.

13.10 What is missing

Every branch here reported a mean, a spread and a probability. What it did not report is why: which uncertain input the risk depends on most, how much a better forecast of the load at bus 3 would change the verdict, and whether the probability of 0.49 is a floor, a ceiling or a point estimate. Those are questions of sensitivity and attribution, and they need the solve to return its gradient. That is Chapter 14. And every intervention here was one line or one load; Chapter 15 asks about the failures the operator did not choose, many at once.

Notes and further reading

The distinction between seeing and doing, the "do" operator, the surgery on structural equations that defines it, and the three-step abduction-action-prediction account of counterfactuals are Pearl's [71]; Peters, Janzing and Schölkopf [72] give a compact modern treatment with structural causal models as the primary object. In those accounts each structural equation has a designated output and the intervention severs the arrows into it. The version in Definition 13.1, with the equation named rather than the variable, is the book's adaptation to models whose equations are acausal, and it is what a physical intervention amounts to: a device that enforces a different relation. The reading of a Chapter 1 model as a structural causal model whose exogenous variables are the inputs and whose mechanisms are the rows is the book's, and it is the reason the input/solved-variable split of Chapter 4 was drawn where it was.

Decision trees over interventions and contingencies, evaluated by weighted reductions over leaves, are standard in reliability and risk analysis; expected shortfall as the tail reduction is Rockafellar and Uryasev's [77], and Part VI uses it. The do-nothing baseline as a mandatory branch is a rule of this book's case-study practice, adopted after enough reports had recommended actions against no comparison.

Summary

- A factor graph encodes compatibility, not mechanism; it cannot say what acting does until the mechanism is named.
- Physics rows are mechanisms, inputs are exogenous. An intervention imposes an input or replaces a named row. "Do $V_2 = 1.027$" by a compensator and by the tap changer are different interventions with different posteriors; seeing $V_2 = 1.027$ is a third. On an input, seeing and doing coincide.
- A counterfactual is abduction, action, prediction, with the evidence used only in the first. On the triangle the counterfactual outage flow is 1.60 ± 0.09; naive re-conditioning gives 1.25, the factual flow, and is wrong.
- Decisions are branches on a shared abduced posterior, with the do-nothing leaf always present. Opening the line alone: even odds of exceeding the limit; with curtailment: a tenth of a percent.
- In a linear model a counterfactual is the abduced input posterior pushed through the intervened map. Had a circuit of the feeder's substation line been out, the generator bus would have been at 1.0653 ± 0.0018; the naive computation says 1.0366 and bends the substation voltage to 0.989.
- An intervention is realizable only if it leaves the exogenous inputs where their priors put them. A phase shifter on a bridge cannot hold its flow, and the model shows it by moving the loads; a generator in region B can.
- On the generation areas in a heat wave the standby unit is worth as much as shedding 100 megawatts and the export rating nothing, because both areas are unit-limited. Evaluating the branches on shared draws resolves the rating's 0.3-megawatt effect with 388 times fewer samples.
- A leaf reports every quantity: a contingency under "open line 2–3" leaves the overload probability at 0.49 and adds an expected shed load of 0.011.
- The semantics requires the modeler to name the replaced mechanism and to decide what evidence survives; it does not do either for them.

Exercises

Exercise 13.1. On the feeder, define a third intervention that makes $V_2 = 1.027$: a shunt device at bus 2 that supplies whatever reactive power is needed, which adds a source term to the balance at bus 2 and leaves both closures intact. Compute its posterior and compare V_1, Q_{2s} and V_s with the two interventions of Table 13.1.

Exercise 13.2. Show that for an input with an independent prior, conditioning on a noiseless reading of it and imposing it give identical posteriors over every other variable, and that for a solved variable they do not in general. State the condition on the graph under which they coincide for a solved variable.

Exercise 13.3. Repeat the counterfactual of Table 13.2 with a second meter on line 1–3 in the factual world. Show that the abduced posterior of the loads is now tighter and less correlated, and that the counterfactual outage flow's spread falls accordingly. What does the naive computation give with two meters?

Exercise 13.4. Add to the branch tree of Table 13.3 a contingency: under the "open line 2–3" action, line 1–3 also fails with probability 0.02. Compute the leaf for that contingency, the weighted probability of exceeding the limit on line 1–2 over the two leaves, and the expected shortfall of F_{12} at the 95 percent level.

Exercise 13.5. For the two-region network of Chapter 8, formulate the intervention "hold the tie flow at 2.0 per unit" in two ways: by a phase-shifting device that replaces the tie closure, and by curtailment in region B that shifts the load priors. Compute both posteriors, identify which region's message each intervention changes, and say what is reused.

Solutions to selected exercises

Exercise 13.1. The shunt device adds its output s to the balance at bus 2, $Q_{2s} = Q_{12} + s$, and its controller sets s so that $V_2 = 1.027$. The replaced mechanism is the free choice of s, and both closures stay. The injection is untouched at 0.050 ± 0.020, and so is the substation voltage at 1.000 ± 0.003. The upper line's closure gives $V_1 = 1.027 + G/5 = 1.0370 \pm 0.0040$, as in both interventions of Table 13.1. The substation line's closure gives $Q_{2s} = 2(1.027 - V_s) = 0.054 \pm 0.006$, whose spread is now the substation voltage's rather than the injection's, and the device's output $s = Q_{2s} - G = 0.004 \pm 0.021$ carries both. Against the compensator, V_1 is the same and Q_{2s} differs, 0.054 ± 0.006 against 0.050 ± 0.020. Against the tap changer, the substation voltage differs, 1.000 ± 0.003 against 1.002 ± 0.010. Three devices, one value, three posteriors.

Exercise 13.2. The first part is Proposition 13.1. For a solved variable x_i, seeing keeps every row and adds $\delta(x_i - v)$, while doing by a device removes the row the device overrides and adds the same delta. The two coincide when the removed row, with the delta and every other factor in place, contributes a constant. That happens when the row contains a variable that appears in no other factor and has a flat prior: integrating the row over that variable gives the same number whatever the other variables are. A controller's actuator is such a variable. The shunt device of Exercise 13.1 is the example. Its output s appears only in the balance at bus 2, with a flat prior, and holding $V_2 = 1.027$ with the device gives exactly the posterior of seeing $V_2 = 1.027$ in the model where the device's output is free, since the rows are the same. The compensator and the tap changer are not of this form in the original model. The branch closure ties the substation voltage to the injection, and the substation voltage's prior carries a set point, so removing either changes what the other variables may be, and the posteriors of Table 13.1 differ from seeing.

Exercise 13.3. The second meter reads 0.90, where the one-meter posterior predicted 0.899 ± 0.092. With both meters the abduced loads are $P_2 = -1.597 \pm 0.047$ and $P_3 = -0.552 \pm 0.047$, correlated at -0.77 instead of -0.94: each meter reads a different combination of the loads, and together they pin both. The counterfactual outage flow on line 1–2 is $-P_2$, 1.597 ± 0.049, half the spread of the one-meter answer. Its probability of exceeding 1.6 is 0.47, barely moved, because the mean sits almost exactly at the limit. The naive computation conditions the outage model on both readings, which under the outage read $-P_2$ and $-P_3$ directly. It concludes $F_{12} = 1.25$, $P_2 = -1.25$ and $P_3 = -0.89$, now wrong for both loads. More factual evidence sharpens the counterfactual and leaves the naive answer as wrong as before.

Exercise 13.4. Under "open line 2–3" the leaf without the contingency has $F_{12} = 1.60 \pm 0.09$ and a probability of 0.49 of exceeding 1.6. With line 1–3 also out, bus 3 is islanded: its load of

0.55 ± 0.18 per unit is shed, and line 1–2 still carries bus 2's load alone, the same 1.60 ± 0.09. The weighted probability over the two leaves is therefore still 0.49. For a Gaussian the mean of the worst five percent is $\mu + \sigma\,\varphi(z_{0.95})/0.05$, with φ the standard normal density and $z_{0.95} = 1.645$, so the expected shortfall of F_{12} at 95 percent is 1.79 per unit, the same in both leaves. What the contingency adds is a shed load, expected $0.02 \times 0.55 = 0.011$ per unit. Under "open 2–3 and curtail 0.3" the same contingency sheds the same load, and the flow's expected shortfall is 1.49, below the limit.

Exercise 13.5. The phase shifter is worked in §13.7. It replaces the tie closure, so it changes the relation on the port itself, and with nothing observed it moves the region-B loads to a sum of -2.000 ± 0.020 while leaving its own angle undetermined: by Proposition 13.4 it is not realizable, because the flow on a bridge does not depend on the angle across it. Curtailment that shifts the load priors, by 0.1 per unit of expected region-B load, changes only region B's factors, so only region B's message is recomputed and region A's is reused. It leaves the tie flow at 2.00 ± 0.35: the mean is where it was asked to be, but the flow is not held, since the loads still vary. Holding it exactly takes a controller with an actuator inside region B, the generator at bus 4 of §13.7, which again changes only region B's message and absorbs the loads' variability in its own output, 0.10 ± 0.35.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

What is worst next depends on what is already gone *a contingency ranking conditional on the current state; evidence changing what to worry about next; rank correlation as a measure of how stale a list is*
`foundation_power_flow/inference_conditioned_nk_interdiction/problems/conditioning_moves_the_worst_set`

Chapter 14

Attribution, sensitivity, and certificates

The previous chapter left three questions unanswered at every leaf of its decision tree: which uncertain input the risk depends on most, how much a unit change in each would move it, and whether a reported number is a floor, a ceiling or a guess. All three are questions about how a consequence of the physics varies with the inputs, and all three become cheap when the solve returns, along with the consequence, its gradient with respect to every input. This chapter shows where that gradient comes from, what attribution means and how it differs from sensitivity, and how a census of solves with gradients, harvested once, answers the three questions and several more at no further cost, with a certificate, whenever the consequence is convex in the inputs. It ends with the one trap that a certificate cannot survive: a gradient taken across a kink.

14.1 Three questions, one derivative

Let $G(\mathbf{u})$ be a scalar consequence of the physics, the curtailment, the tie flow, the voltage at a bus, as a function of the uncertain inputs $\mathbf{u}$. The operator's three questions are, in order of how much they ask of G:

1. *Sensitivity*: by how much does G change per unit change in u_i at the current state? That is $\partial G/\partial u_i$, one number per input, at one point.
2. *Attribution*: how much of the uncertainty in G is due to the uncertainty in u_i? That is a property of G over the whole prior, not at a point, and it needs the joint behavior of G and the inputs.
3. *Certification*: is the reported expectation, tail or worst case of G a bound, and in which direction? That needs a global property of G, and convexity is the one that delivers.

Sensitivity is the derivative. Attribution can be approximated by it and is exactly it only for a linear G. Certification uses it as a supporting plane. All three start from the same object, and the first task is to obtain it without differencing.

14.2 Sensitivities from the solve

At a solved state $\mathbf{F}(\mathbf{z}^\star, \mathbf{u}) = \mathbf{0}$ the solved variables are an implicit function of the inputs, $\mathbf{z} = X(\mathbf{u})$, and the implicit function theorem gives their derivative from the two Jacobians the solve already

assembled:

$$\frac{\partial X}{\partial \mathbf{u}} \;=\; -\,\mathbf{J}_z^{-1}\,\mathbf{J}_u, \qquad \mathbf{J}_z = \frac{\partial \mathbf{F}}{\partial \mathbf{z}}, \quad \mathbf{J}_u = \frac{\partial \mathbf{F}}{\partial \mathbf{u}}. \tag{14.1}$$

$\mathbf{J}_z$ is the Jacobian of Chapter 1, square and already factored; $\mathbf{J}_u$ is the same assembly with the input columns kept rather than dropped, which costs nothing extra. For n_u inputs the forward form is n_u solves against the held factorization, one per column of $\mathbf{J}_u$. For one consequence $G = \mathbf{c}^\mathsf{T}\mathbf{z}$ the *adjoint* form is a single solve, $\mathbf{J}_z^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}$, after which $\partial G/\partial \mathbf{u} = -\boldsymbol{\lambda}^\mathsf{T}\mathbf{J}_u$ for every input at once. A model with a hundred inputs and one consequence pays one solve; with one input and a hundred consequences, one solve the other way.

Derivation. Differentiate the identity $\mathbf{F}(X(\mathbf{u}), \mathbf{u}) = \mathbf{0}$ with respect to $\mathbf{u}$: $\mathbf{J}_z\,\partial X/\partial \mathbf{u} + \mathbf{J}_u = \mathbf{0}$. At a well-posed solution $\mathbf{J}_z$ is nonsingular, which gives equation (14.1). For $G = \mathbf{c}^\mathsf{T}\mathbf{z}$ the derivative is $\partial G/\partial \mathbf{u} = -\mathbf{c}^\mathsf{T}\mathbf{J}_z^{-1}\mathbf{J}_u$. Grouped from the left, $\boldsymbol{\lambda}^\mathsf{T} = \mathbf{c}^\mathsf{T}\mathbf{J}_z^{-1}$ is the solution of $\mathbf{J}_z^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}$, one solve, after which $-\boldsymbol{\lambda}^\mathsf{T}\mathbf{J}_u$ is a sparse product. Grouped from the right, $\mathbf{J}_z^{-1}\mathbf{J}_u$ is n_u solves, one per input column. □

On the export feeder with the susceptance b_{12} freed, the inputs are (G, V_s, b_{12}); here G is the feeder's reactive injection, not the consequence $G(\mathbf{u})$ of §14.1. The script `ch14_attribution.py` in the book's `scripts` directory, which supplies every number in the chapter, evaluates equation (14.1) at the nominal state:

$$\frac{\partial (V_1, V_2)}{\partial (G, V_s, b_{12})} \;=\; \begin{pmatrix} 0.700 & 1.000 & -0.002 \\ 0.500 & 1.000 & 0 \end{pmatrix}, \tag{14.2}$$

agreeing with central differences to 10^{-12}. The generator-bus voltage rises seven tenths of a per unit for each per unit of injection, one for one with the substation voltage, and falls two thousandths per unit of susceptance; bus 2 does not see the susceptance at all. The same numbers are the lever arms that Chapters 3 and 11 read off by hand, now read off the Jacobian for any model.

At scale. On the feeder, with three inputs, the choice between the forward and adjoint forms hardly matters. Take instead the mesh of buses of Chapter 11 at 100×100, with an uncertain reactive injection at every bus: ten thousand unknowns, ten thousand inputs, and one consequence, the voltage at the centre, where a load sits. The forward form would solve once per injection, ten thousand solves, about 7 seconds against the held factorization. The adjoint form solves once, in 0.7 milliseconds after a factorization of 20 milliseconds, and returns all ten thousand sensitivities; at the buses checked they agree with finite differences to six digits. The centre rises by 0.123 per unit for each per unit of its own injection, 0.063 for a neighbour's and 0.0025 for an injection ten buses away. The sensitivities sum to 4.000, which is $1/h$: a uniform extra injection at every bus must raise every bus by $1/h$ for the shunts to absorb it. And the centre follows the reference voltage one for one. Figure 14.1 draws the sensitivities around the centre. The map is the mesh's Green's function seen from the centre, and since the mesh is linear it is also the attribution: with independent injections of equal spread, each injection's share of the centre's variance is proportional to the square of its sensitivity, by Proposition 14.1.

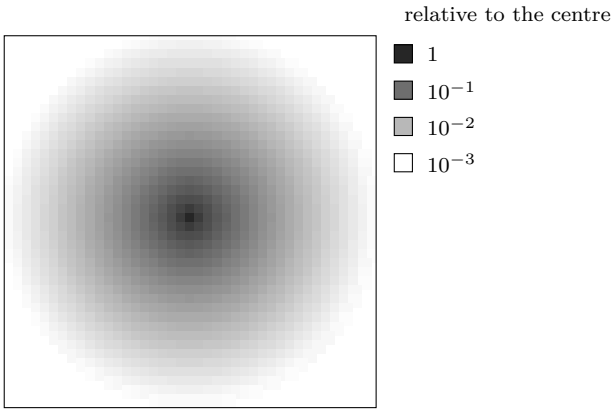


Figure 14.1: The sensitivity of the centre voltage of a 100×100 mesh of buses to the reactive injection at each bus within twenty buses of the centre, relative to the centre's own, on a logarithmic grey scale. One adjoint solve computes all ten thousand values.

14.3 Attribution

Attribution asks for a decomposition of $\mathrm{Var}[G]$ into shares due to each input. Two definitions are standard.

The *Sobol* first-order index of input i is the fraction of the variance explained by the conditional expectation given u_i alone, $S_i = \mathrm{Var}[\mathbb{E}[G \mid u_i]]/\mathrm{Var}[G]$; the *total* index adds every interaction involving u_i. When G is additive in independent inputs the first-order indices sum to one and equal the totals; when it is not, the difference measures interaction, and the indices do not by themselves say how to divide the interaction among the inputs involved.

The *Shapley* share does. It averages, over every ordering of the inputs, the increase in explained variance that input i brings when added to the inputs before it. Shares sum to one, respect symmetry, and give an input that matters only jointly with another half the joint credit. For d inputs the average runs over $d!$ orderings and 2^d subsets, each subset needing a conditional expectation; that is the computation Chapter 10 found to cost the sampling route $(d+4)N$ solves and the grid route nothing beyond the grid.

The cheap proxy is the gradient, and for a linear consequence it is exact.

Proposition 14.1 (Attribution of a linear consequence)**.** *Let $G = g_0 + \sum_i g_i u_i$ with independent inputs of variances σ_i^2. Then $\mathrm{Var}[\mathbb{E}[G \mid u_S]] = \sum_{i \in S} g_i^2 \sigma_i^2$ for every subset S of the inputs. The Sobol first-order and total indices of input i are both $g_i^2\sigma_i^2/\mathrm{Var}[G]$, and so is its Shapley share.*

Proof. By independence, $\mathbb{E}[G \mid u_S] = g_0 + \sum_{i \in S} g_i u_i + \sum_{i \notin S} g_i \mathbb{E}[u_i]$, whose variance is $\sum_{i \in S} g_i^2 \sigma_i^2$. The first-order index takes $S = \{i\}$. The total index is one minus the explained fraction of the complement of $\{i\}$, which leaves the same term. In the Shapley average every marginal contribution of input i, $v(S \cup \{i\}) - v(S)$, equals $g_i^2\sigma_i^2$ whatever S is, so the average is that value too. □

On the feeder, V_1 has variance $2.05 times 10^{-4}$, of which the injection contributes $1.96 times 10^{-4}$, 95.6 percent, and the substation voltage $9 times 10^{-6}$. For a nonlinear G the same formula with the gradient averaged over the prior, the *activity* $\mathbb{E}[(\partial G/\partial u_i)^2]\,\mathrm{Var}[u_i]$, is a first-order proxy that a census of gradients provides at no cost. It can be wrong, and the three-line problem shows how:

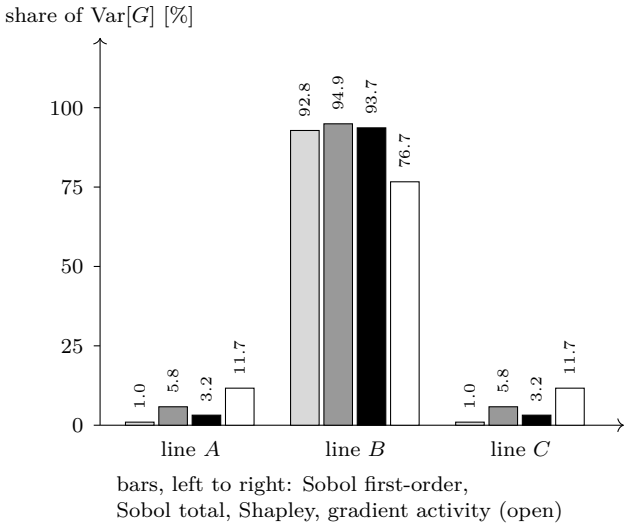


Figure 14.2: Four attributions of the three-line curtailment's variance, in percent: Sobol first-order and total indices and Shapley shares on the 25^3 grid, and the gradient activity from the 300-point census. For lines A and C the Shapley share lies between the first-order and total indices; the activity lies outside them.

the activity shares are 11.7, 76.7 and 11.7 percent for the ratings of lines A, B and C, against Shapley shares of 3.2, 93.6 and 3.2. The parallel lines' ratings matter only when their sum is the binding constraint, which is rarely, and only jointly; the gradient is nonzero whenever they bind and does not see that they bind together. Activity is a screening tool, useful for saying which inputs can be ignored; it is not an attribution.

Figure 14.2 sets four measures side by side, from the script `ch14_more.py`. The Sobol first-order indices of the parallel lines are 1.0 percent each and their totals 5.8: each line matters mostly through its interaction with the other, and 5.2 percent of the variance is interaction, which the first-order indices credit to no one. The Shapley share splits it, 3.2 each, between the first-order and total indices. The activity proxy, 11.7 each, lies outside that bracket: the gradient counts every draw on which the pair binds, as if each line bound alone.

Principle 14.1. *Sensitivity is a derivative at a point; attribution is a decomposition over the prior. The gradient gives the first exactly and the second only for a linear consequence. Where inputs interact, attribution needs conditional expectations, and the question is where to get them cheaply.*

14.4 A census with gradients

Suppose each solve returns G and ∇G at the drawn input, and suppose G is convex in the inputs. Then every solve yields a *supporting plane*, $G(\mathbf{u}) \geq G(\mathbf{u}_j) + \nabla G(\mathbf{u}_j)^{\mathsf{T}}(\mathbf{u} - \mathbf{u}_j)$ for all $\mathbf{u}$, and a census of C solves yields C of them. Their pointwise maximum, floored at whatever lower bound G is known to obey,

$$L(\mathbf{u}) \;=\; \max\Big\{\,0,\; \max_{j \leq C}\big[G(\mathbf{u}_j) + \nabla G(\mathbf{u}_j)^{\mathsf{T}}(\mathbf{u} - \mathbf{u}_j)\big]\Big\} \;\leq\; G(\mathbf{u}) \quad \text{for every } \mathbf{u}, \tag{14.3}$$

is a piecewise-linear convex *surrogate* that lies below G everywhere, not only at the census points and not only under the prior the census was drawn from. This is the cutting-plane construction of convex optimization, used here not to optimize but to bound.

Proposition 14.2 (Supporting planes bound a convex consequence)**.** *Let $G \geq 0$ be convex, and let $\mathbf{s}_j$ be a subgradient of G at $\mathbf{u}_j$ for each census point. Then L of equation* (14.3)*, with $\mathbf{s}_j$ in place of $\nabla G(\mathbf{u}_j)$, satisfies $L \leq G$ everywhere. Consequently, under any distribution of the inputs, $\mathbb{E}[L] \leq \mathbb{E}[G]$ and $\mathbb{P}(L > g) \leq \mathbb{P}(G > g)$ for every g. The surrogate L is convex and piecewise linear, and its maximum over a box is attained at a vertex.*

Proof. A subgradient satisfies $G(\mathbf{u}) \geq G(\mathbf{u}_j) + \mathbf{s}_j^{\mathsf{T}}(\mathbf{u} - \mathbf{u}_j)$ for every $\mathbf{u}$, by definition, and the gradient of a differentiable convex function is one. The maximum of quantities each at most $G(\mathbf{u})$, together with $0 \leq G(\mathbf{u})$, is at most $G(\mathbf{u})$. A pointwise inequality survives any expectation, and the event $\{L > g\}$ is contained in $\{G > g\}$. A maximum of affine functions is convex. A convex function on a box attains its maximum at a vertex, because every point of the box is a convex combination of the vertices and the function there is at most the same combination of the vertex values. □

The inequality is a theorem given convexity. The *certificate* is its empirical check: evaluate G on a holdout set of draws, and verify $L \leq G$ on every one. A violation is not a statistical fluctuation; it is proof that either G is not convex or a gradient is wrong, and either invalidates every bound below.

14.4.1 What the census answers

Once L is built, it is a function that costs nothing to evaluate, and everything asked of G can be asked of L instead, with a known direction of error:

- $\mathbb{E}[L]$ under any belief is a *floor* on $\mathbb{E}[G]$ under that belief, including beliefs other than the one the census was drawn from: a changed forecast, a correlated prior, a what-if on the box.
- $\mathbb{P}(L > g)$ is a floor on $\mathbb{P}(G > g)$: a tail floor at every level.
- $\max L$ over a box is a floor on the worst case, and since L is convex the maximum over a box is at a vertex.
- The mean census gradient is the mean sensitivity, the shadow price of each input's capacity.
- Attribution is computed on L by any method, at zero solves, since L is cheap; where L is close to G the shares are close.

14.4.2 The three-line problem

The curtailment $G = (160 - \min(A + C, B))^+$ is convex in the ratings: the minimum of linear functions is concave, its negative convex, and the positive part preserves convexity. Its gradient at any point is the gradient of the active piece: $(-1, 0, -1)$ when the parallel pair binds, $(0, -1, 0)$ when line B does, zero when nothing is curtailed. A census of 300 draws with these gradients gives the surrogate of equation (14.3). Table 14.1 certifies it on 20,000 holdout draws and answers the questions.

On this problem the surrogate is not merely a floor; it is exact, and the reason is instructive. The curtailment has three linear pieces, and once the census has landed at least one point on each, the three supporting planes reproduce the function everywhere. The gap between L and G

Table 14.1: A census of 300 evaluations with active-set gradients on the three-line curtailment: the certificate on 20,000 holdout draws and the answers read from the surrogate at zero further evaluations, against the truth.

question	from the surrogate L	truth	direction
certificate: $\max(L - G)$ on holdout	3×10^{-14}	≤ 0 required	—
$\mathbb{E}[G]$ under the census belief	5.322	5.333 (holdout 5.322)	floor
$\mathbb{E}[G]$, A, C shifted to $[60, 110]$	8.274	8.274	floor
worst case over the box	20.0	20.0 at $(70, 140, 70)$	floor
$\mathbb{P}(G > 5)$, $\mathbb{P}(G > 10)$, $\mathbb{P}(G > 15)$	0.400, 0.266, 0.131	same	floors
shadow prices $\mathbb{E}[\nabla G]$	$(-0.05, -0.51, -0.05)$	—	—
Shapley shares on a 25^3 midpoint grid of L	3.2, 93.7, 3.2	3.2, 93.7, 3.2	—

on the holdout is zero to machine precision, the floors are the values, and the Shapley shares computed on a midpoint grid of L, at zero further evaluations of G, are the reference shares of Chapter 10 to a tenth of a point. On a real load-shedding model, a linear program whose value has as many pieces as the program has bases, the gap is finite and the floors are floors; Chapter 21 runs exactly this construction on a 73-bus network with fourteen uncertain ratings and nine questions, and its solve counts were the ones Chapter 10 quoted: 6,536 against 113,876.

The floors under a shifted belief deserve a second look. The census was drawn with the parallel lines' ratings on $[70, 120]$; the question asked what the curtailment would be if they were on $[60, 110]$, a worse world. No new evaluation of G was made. The surrogate, being a bound everywhere, is a bound there too, and in this case it is the answer. A sampler would have drawn a new sample.

How many census points does exactness take? The surrogate is exact once the census has a point on each curtailed piece, the one where the parallel pair binds and the one where line B binds; the zero piece is the floor. A draw lands on the first with probability 0.054 and on the second with 0.486, so a census of C random points is exact with probability $1 - (1 - 0.054)^C - (1 - 0.486)^C + (1 - 0.054 - 0.486)^C$. Figure 14.3 draws it against 1,000 random censuses at each size. The rare piece sets the size: 55 points for 95 percent and 84 for 99, against two placed deliberately, one on each piece. A census drawn from the prior spends its points where the prior puts its mass, and a rare regime needs either many points or one placed on purpose.

Principle 14.2. *A convex consequence with gradients from the solve yields supporting planes, and their maximum is a surrogate that bounds the consequence everywhere. Every question asked of the surrogate is answered at zero further solves with a known direction of error, and the certificate is a holdout check that fails loudly.*

14.5 The kink

Everything in the previous section rested on the gradient being a subgradient: a slope that supports the function from below. For a convex function that is smooth the gradient is the only subgradient and any correct computation of it will do. For a convex function with a kink, the minimum of the three-line problem, the positive part, the maximum in a limiter, the active-set gradient is a subgradient and the finite difference is not.

Replace the active-set gradients of the census by central differences of G with a step of one megawatt, and re-certify. Table 14.2 gives the result. Eighteen of the three hundred census points

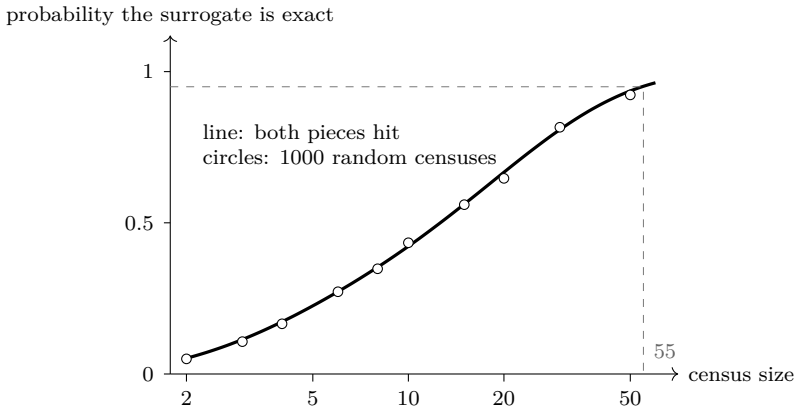


Figure 14.3: The probability that a census of random draws gives an exact surrogate of the three-line curtailment, against the census size. Line: the probability that the census hits both curtailed pieces. Circles: the fraction of 1,000 random censuses whose surrogate is exact on the holdout. The dashed line marks 95 percent, reached at 55 points.

Table 14.2: The certificate with active-set gradients and with central finite differences of G (step 1 MW) on the same 300 census points and 20,000 holdout draws.

gradient	census points at a kink	holdout violations	$\max(L - G)$
active set	0 of 300	0 of 20,000	3×10^{-14} MW
central differences	18 of 300	984 of 20,000	+0.124 MW (1.9% of sd)

straddle a kink, in the sense that the difference step crosses from one linear piece to another; at each of them the central difference averages the two slopes and the resulting plane crosses the kink and sits above G nearby. On the holdout the surrogate exceeds G at 984 of 20,000 points, by up to 0.12 megawatts, two percent of the curtailment's spread. The certificate fails. Nothing about the function has changed; only the gradient, and only where the function bends.

Proposition 14.3 (A central difference across a kink)**.** *Let f be convex and piecewise linear in one variable, with a single kink at which its slope rises by J. A central difference with step h, taken at a centre within h of the kink, gives a line that lies above f at the kink by up to $Jh/8$, the largest value reached when the centre is $h/2$ from the kink. A central difference whose centre is farther than h from every kink returns the slope of its piece, which is a subgradient.*

Proof. Put the kink at 0 and subtract the left-hand line, which changes neither the difference quotient's error nor the violation, so that $f(x) = J\,x^+$. For a centre $x_0 \in (-h, h)$ the central difference is $s = (f(x_0 + h) - f(x_0 - h))/(2h) = J(x_0 + h)/(2h)$. The line through $(x_0, f(x_0))$ with slope s has height $f(x_0) - s\,x_0$ at the kink, where f is zero. For $x_0 < 0$ that height is $-Jx_0(x_0 + h)/(2h)$, and for $x_0 > 0$ it is $Jx_0(h - x_0)/(2h)$. Both are positive, and each is largest, $Jh/8$, at $|x_0| = h/2$. A centre farther than h from the kink differences a single linear piece and returns its slope. □

The census violation of 0.124 megawatts in Table 14.2 matches this bound: the slope along the rating of line B jumps by one where B takes over from the parallel pair, and the step was one megawatt, so $Jh/8 = 0.125$. Figure 14.4 draws it on a line through the kink. In several variables

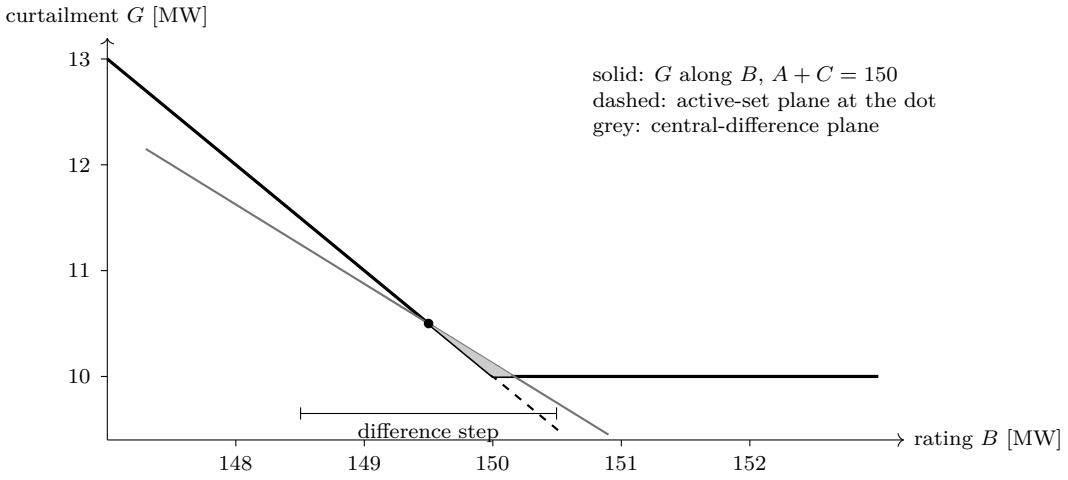


Figure 14.4: The three-line curtailment along the rating of line B with $A + C = 150$ (solid), which has a kink at $B = 150$ where its slope rises from -1 to 0. At the dot, 149.5, the active-set plane (dashed) has the slope of its piece and touches G without crossing it. The central-difference plane (grey), whose step of one megawatt reaches across the kink, has slope -0.75 and lies above G by up to 0.125 megawatts, the shaded sliver: an eighth of the step times the jump in slope.

the bound is only a lower estimate. Each coordinate's difference errs by its own amount, so the plane is tilted along the kink as well as across it, and along the kink the violation grows with distance; on the overload of Exercise 14.5 it reaches fifteen times the one-dimensional bound.

The fix is not a smaller step; a smaller step straddles fewer points and each straddled point is just as wrong. The fix is to difference what is smooth and compose with what is not: the flows and voltages that the physics solves for are smooth in the inputs, and the kink is in the consequence built from them, a minimum, a positive part, a limit exceeded; take the gradient of the smooth inner map, by equation (14.1) or by differences, and apply the active piece of the outer function at the center point. That is the active-set gradient of Table 14.1, it costs no extra solves, and it is what Chapter 7 meant by differentiating within the regime. The rule is the one that a finite-difference adapter must never break: it may return a gradient, but it must never declare the consequence convex, because its gradients across kinks are not subgradients and the certificate they produce is worthless.

Principle 14.3. *A finite difference across a kink is not a subgradient, even of a convex function. Difference the smooth inner map and apply the active set at the center. Never let a differenced gradient carry a certificate.*

14.6 Worked example: convexity lost and recovered

The generation areas of Chapter 10 test the census on a consequence with a different shape. Their unserved load $U = (2{,}100 - S_1 - S_2)^+$, with $S_k = \min\big((1 - \alpha(T - 18)) \sum_j R_{kj},\ L_k\big)$, has an active-set gradient everywhere, by the same reasoning as the curtailment's. On the unit-limited piece an area's delivery rises with each rating at the derated rate and falls with the ambient temperature at α times the rating sum; on the export-limited piece it rises with the export path's

Table 14.3: Attribution of the generation areas' shortfall variance to three groups of inputs, in percent: Sobol first-order and total indices and Shapley shares by pick-freeze on 400,000 draws per sample set, and the gradient activity summed over each group.

group	first-order	total	Shapley	gradient activity
weather	36.6	76.9	55.9	71.0
area 1	10.2	35.8	22.2	14.5
area 2	10.2	35.4	22.0	14.5

rating. A census of 500 draws over all nine inputs builds the surrogate, and the holdout refuses to certify it: 3,863 of 20,000 holdout points lie below the surrogate, by up to 5.5 megawatts, a fifth of the shortfall's spread. Figure 14.6 shows where. The violations sit at hot ambients, 26.4°C on average.

The gradients are right; the function is not convex. The derated capacity $(1 - \alpha(T - 18))R$ is bilinear in the ambient temperature and a rating, and its Hessian in those two inputs has eigenvalues $\pm\alpha$: a saddle. The shortfall inherits the saddle wherever the units bind, which at hot ambient temperatures is almost everywhere.

The cure is the separator of Chapter 10. At a fixed ambient temperature the derating is a number, each area's capacity is linear in its ratings, its delivery is the minimum of two linear functions and so concave, and the shortfall, the positive part of a constant minus two concave functions, is convex in the other eight inputs. Stratify the census by the ambient temperature: at each of nine ambient nodes, 120 census points in the other eight inputs and 5,000 holdout points. The certificate holds at every node, with a worst violation of 10^{-12} megawatts, rounding, in 45,000 holdout points. The floors on the conditional mean are exact at the four hottest nodes, 3.91, 14.23, 33.02 and 64.81 megawatts. At 23°C the floor is 0.92 against 1.03, and at the two cooler curtailed nodes it is zero, because the census found little or none of the small curtailed region there.

Attribution by groups. The nine inputs fall into three groups an operator would name: the weather, area 1 with its three units and its export path, and area 2. Table 14.3 gives four measures for the groups, from seven sample sets of 400,000 draws each by pick-freeze; the areas' model costs almost nothing to evaluate, so the sampling route is affordable here. The weather's first-order index is 37 percent, the share of the shortfall's variance that the ambient temperature alone explains; the nine-node grid of Chapter 10 put it at 39. Each area's is 10. The first-order indices sum to 57 percent, so 43 percent of the variance is interaction: a shortfall needs a hot afternoon and a weak area together. The totals, 77, 36 and 35, count the interaction in full for every group that takes part in it. Shapley splits it, 56 percent to the weather and 22 to each area. The gradient activity says 71 and 15. It weighs each input by its slope where the shortfall is positive, and the ambient temperature's slope, the derating of six units at once, is large wherever there is a shortfall. It does not see that the shortfall exists only when an area is also weak, which is where the areas earn their Shapley share.

A floor under a forecast. The stratified census is a set of surrogates, one per ambient, and it can answer questions about any belief over the ambient at no further cost, provided the step between ambient nodes is bounded too. Extend it to 29 nodes from 24 to 31°C, a quarter degree

apart, with 120 census points each: 3,480 evaluations in all. The census floors the conditional mean at each node. Between nodes the physics supplies the rest of the argument. The shortfall never falls as the ambient rises, since a hotter afternoon derates every unit and changes nothing else, so on each interval between nodes the conditional mean is at least its value at the lower node. Weighting each node's floor by the forecast's probability of the interval above it therefore gives a certified floor, a lower Riemann sum.

Proposition 14.4 (A floor under any belief over a monotone input)**.** *Let $G(t, \mathbf{x}) \geq 0$ be nondecreasing in a scalar input t for every $\mathbf{x}$, let $t_1 < \cdots < t_m$ be nodes, and let $L_i(\mathbf{x}) \leq G(t_i, \mathbf{x})$ for each i. For any distribution of t independent of $\mathbf{x}$, with $t_{m+1} = \infty$,*

$$\mathbb{E}[G] \;\geq\; \sum_{i=1}^{m} \mathbb{P}(t_i \leq t < t_{i+1})\, \mathbb{E}[L_i(\mathbf{x})]. \tag{14.4}$$

Proof. For t in $[t_i, t_{i+1})$, monotonicity gives $G(t, \mathbf{x}) \geq G(t_i, \mathbf{x}) \geq L_i(\mathbf{x})$, and for $t < t_1$, $G(t, \mathbf{x}) \geq 0$. Take the expectation over $\mathbf{x}$ with t fixed, which by independence does not change the distribution of $\mathbf{x}$, and then over t, splitting its range at the nodes. □

Under the heat-wave forecast of Chapter 13, 26.5 ± 0.7°C, it is 28.18 megawatts, against 30.58 ± 0.09 from 200,000 direct draws. Under a hotter forecast of 27.5 ± 0.7°C it is 49.53, against 53.05 ± 0.11. Figure 14.5 sweeps the forecast.

The floor is loose by roughly half a node spacing times the slope of the conditional mean, two or three megawatts here, and halving the spacing halves it, at twice the census. The tempting alternative is to weight each node by the forecast's density instead. That gives 30.44 under the heat wave, much closer, but it is not a bound: under a mild forecast of 25 ± 0.7°C it gives 10.55 megawatts where direct draws give 10.14 ± 0.05. The conditional mean is convex in the ambient temperature, and a density-weighted sum over nodes overstates a convex function. A floor at each node becomes a floor over the forecast only through an argument about what happens between the nodes, and here monotonicity is that argument. Each new forecast costs no evaluation of the areas at all.

> **Principle 14.4.** *Convexity is a property of a consequence in chosen inputs. When one input spoils it, condition on that input and certify the rest; the input to condition on is usually a separator already.*

14.7 Conditions

The chapter's method rests on two conditions, and it is worth stating where each holds.

The solve must return its gradient. Equation (14.1) does this for any model whose residuals are differentiable in the inputs, at the cost of keeping the input columns of the Jacobian. A load-shedding linear program returns its gradient as the dual variables on the rating constraints, for free. An iterative solver that returns only a value must be differenced, with the caveat of §14.5.

The consequence must be convex in the inputs. The value of a linear program is convex in its right-hand sides, so any consequence that is the optimal value of a linear allocation, load shed, redispatch cost, is convex in the capacities and loads that bound it. A DC power flow's line flows are linear in the injections, so any convex function of them, an overload measured

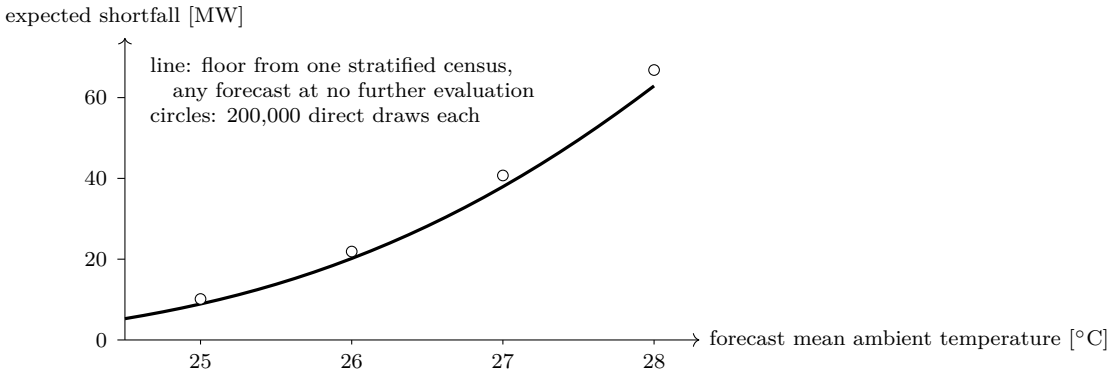


Figure 14.5: The expected unserved load of the generation areas under an ambient forecast of mean μ and spread 0.7°C, as μ sweeps from 24.5 to 28°C. Line: the certified floor from one stratified census of 3,480 evaluations, a lower sum over ambient nodes that uses the shortfall's monotonicity in the ambient temperature; every forecast costs no further evaluation. Circles: 200,000 direct draws at each of four forecasts.

as a positive part, is convex. An AC power flow is not linear, its flows are not convex in the injections, and a surrogate built on it is a surrogate without a certificate: still useful for screening, mean sensitivities and variance reduction, none of which need the bound, but no longer a floor. Chapter 22 puts the nine questions to an AC model, the IEEE 14-bus network with eight uncertain loads and total line overload as the consequence, in a congested regime and an intermittent one. The shadow prices, the total Sobol indices and the first-order indices survive as estimates in both. The screening bound stays certified in both, because it never needed the inequality. The five questions that read a floor from the surrogate, a new belief, correlated loads, a what-if curve, the tail floors and the worst case, lose the certificate. In the congested regime they are empirical: $L \leq G$ held on all 2,000 holdout and 10,000 reference draws, but nothing guarantees that it holds elsewhere. In the intermittent regime they are estimates: $L \leq G$ fails on 1,149 of the 10,000 reference draws, by at most 7×10^{-4} of the standard deviation of G.

14.8 In practice

Take gradients from the solve. Keep the input columns of the Jacobian when assembling it. For one consequence and many inputs, solve the adjoint system once; for one input and many consequences, solve forward once. Difference only what the solver cannot differentiate, and only the smooth part of it.

Screen with the gradient, attribute with conditional expectations. The activity proxy says which inputs can be dropped. The shares that go into a report come from Shapley or Sobol on a grid or on a certified surrogate, where they cost no further solves.

Certify before using any floor. Check $L \leq G$ on a holdout drawn independently of the census. One violation invalidates every bound built from that surrogate, and it points at either a wrong gradient or a lost convexity.

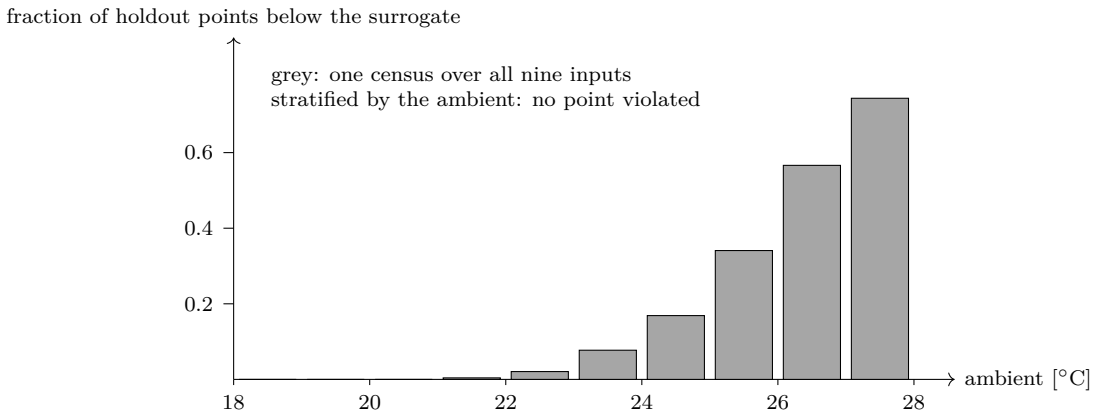


Figure 14.6: The generation areas' shortfall and a census with active-set gradients over all nine inputs: the fraction of holdout points at which the surrogate exceeds the shortfall, by ambient bin. The violations are at hot ambient temperatures, where the derating's saddle matters. Stratified by the ambient temperature, the same construction violates at no holdout point.

Place census points on purpose. A census drawn from the prior finds a rare regime late: 55 random points for what two placed points do on the three-line problem. Seed the census at the vertices of the box and at the boundaries of the regimes you know about.

Turn floors at nodes into a floor over a belief with an argument. A surrogate stratified on one input bounds the consequence at each node and nowhere between. A monotone input gives the argument that carries the bound between nodes, Proposition 14.4; without one, a density-weighted sum over the nodes is an estimate, and it can overstate.

Check convexity in the coordinates you use. A product of inputs, a derating, an efficiency that varies with load: each can make a consequence non-convex. Stratify on the input that causes it, and certify within each stratum.

14.9 What is missing

Every input in this chapter was continuous. The inputs that fail discretely, the availabilities of Chapter 7, have no gradient, and the question of which of them matter most, alone and together, needs the branch machinery instead: which combinations of failures are most probable, which most damaging, and how a common cause ties them together. That is Chapter 15, the last of Part IV.

Notes and further reading

Sensitivities by the implicit function theorem and their adjoint form are standard in design optimization and in data assimilation; Nocedal and Wright [66] treat the linear algebra, and the observation that a linear program's duals are the derivatives of its value with respect to its right-hand sides is the sensitivity theorem of linear programming, found there and in Boyd and Vandenberghe [11], who also give the convexity of the optimal value in the constraint data that

§14.7 relies on. Variance-based attribution is Sobol's [84]; the Shapley share as an attribution that resolves interactions is Owen's [68], with estimation by Song, Nelson and Staum [85]; the pick-freeze estimator is Saltelli's [80]. The gradient-based activity as a screening proxy is common in sensitivity analysis under various names; its failure on the three-line problem is the book's illustration.

Supporting planes of a convex function and their pointwise maximum are the cutting-plane method of Kelley [44]; the use of that construction as a certified surrogate, evaluated under beliefs other than the census's and checked on a holdout, is the framework's, and the case study cited in §14.4 is its first full instance. The kink caveat of §14.5 was found the hard way, on an AC model whose consequence was a sum of positive parts. In the AC part of the surrogate case study, in its intermittent regime, differenced gradients produced certificate violations of 1.09 megawatts, about a sixth of the consequence's spread of 6.38. The active-set correction cut them by more than two orders of magnitude, to 0.0046 megawatts; the residue, under a thousandth of the spread, is the model's non-convexity itself. The caveat is recorded here so that no reader repeats it.

Summary

- Sensitivity, attribution and certification all start from $\partial G/\partial \mathbf{u}$, which the implicit function theorem gives from the two Jacobians the solve assembled: n_u forward solves or one adjoint solve. On a 100×100 mesh of buses one adjoint solve, 0.7 milliseconds, gives the centre voltage's sensitivity to all ten thousand injections, against 7 seconds forward.
- Attribution is a variance decomposition over the prior; Sobol indices and Shapley shares define it; the gradient gives it exactly only for a linear consequence. On the three-line problem the gradient proxy says 12/77/12 and the truth is 3/94/3, with the parallel lines' Shapley shares between their first-order and total Sobol indices.
- A convex consequence with gradients yields supporting planes whose maximum bounds it everywhere. On the three-line problem 300 solves give an exact surrogate; floors on the mean, the tail, the worst case and a shifted belief, and the Shapley shares, follow at zero further solves.
- A finite difference across a kink is not a subgradient: 18 straddled census points broke the certificate at 984 holdout draws, by up to $Jh/8$, an eighth of the step times the jump in slope. Difference the smooth inner map and apply the active set at the center.
- The surrogate is exact once the census hits every piece; a random census needs 55 points for that with probability 0.95, two placed points always suffice.
- On the generation areas a census over all nine inputs fails its certificate at hot ambient temperatures, because the derating is bilinear; stratified by the ambient temperature it holds at every holdout point. With the shortfall's monotonicity in the ambient temperature it floors the expected shortfall under any forecast at no further evaluations: 28.2 megawatts under the heat wave, against 30.6 by direct sampling. A density-weighted sum over the nodes is closer but is not a bound.
- Over three groups the areas' shortfall is 56 percent weather and 22 percent each area by Shapley. 43 percent of its variance is interaction, and the gradient proxy overstates the weather at 71.
- Conditions: the gradient must come from the solve, and the consequence must be convex. Linear programs and DC flows qualify; AC flows do not, and lose the bound but not the

screening.

Exercises

Exercise 14.1. Derive equation (14.1) by differentiating $\mathbf{F}(X(\mathbf{u}), \mathbf{u}) = \mathbf{0}$, and derive the adjoint form for $G = \mathbf{c}^{\mathsf{T}}\mathbf{z}$. Count the solves for n_u inputs and n_c consequences in each form and state when each is preferable.

Exercise 14.2. For the three-line problem, compute the Sobol first-order and total indices of the three ratings on the 25^3 grid, and show that the totals exceed the first-order indices for lines A and C. Relate the difference to the interaction the activity proxy missed.

Exercise 14.3. Prove that the positive part of a convex function is convex, that the minimum of two linear functions is concave, and hence that the three-line curtailment is convex. Then give a two-line network whose curtailment is not convex in its ratings.

Exercise 14.4. Reduce the census of Table 14.1 to 30 points and re-certify. Find the smallest census for which the surrogate is still exact on the holdout, and explain the number in terms of the pieces of G.

Exercise 14.5. Implement the kink fix for the three-line problem with finite differences: difference the smooth quantities $A + C$ and B (trivially) and apply the active piece of G at the center, and show the certificate holds. Then apply the same construction to the DC triangle with an overload consequence $\max(0, |F_{12}| - 1.4)$ and uncertain loads, differencing the flows and applying the active set.

Solutions to selected exercises

Exercise 14.1. The derivation is given after equation (14.1). For n_u inputs and n_c consequences the forward form needs n_u solves, each giving one column of $\partial X/\partial \mathbf{u}$ and hence the sensitivity of every consequence to that input. The adjoint form needs n_c solves, each giving the gradient of one consequence with respect to every input. With the factorization held, each solve is two triangular solves, so the cheaper form is the one with the smaller count: adjoint when there are fewer consequences than inputs, forward otherwise. On the feeder the generator-bus voltage's sensitivities, $(0.700, 1.000, -0.002)$ to the injection, the substation voltage and the susceptance, come from one adjoint solve with $\mathbf{c}$ the unit vector on V_1.

Exercise 14.2. On the 25^3 grid the variance is 41.92. The first-order indices are 1.0, 92.8 and 1.0 percent for lines A, B and C; the totals are 5.8, 94.9 and 5.8. The first-order indices sum to 94.8 percent, so 5.2 percent of the variance is interaction, and for each parallel line the total exceeds the first-order index by 4.8 points: almost all of what the parallel lines contribute, they contribute jointly. Line B's total exceeds its first-order index by only 2.1 points. The activity proxy missed exactly this: the gradient $(-1, 0, -1)$ on the piece where the pair binds credits each line with the full slope whenever the pair binds, as though either alone could be varied to relieve the curtailment, which is the interaction counted twice.

Exercise 14.4. A 30-point census still certifies, as every census with active-set gradients must, since its planes are subgradients; what it may lose is exactness. It is exact with probability 0.81 (Figure 14.3), and when it is not, its mean gap is small, 0.06 megawatts on average over random 30-point censuses. The smallest census that is exact is two points, one on the piece where the parallel pair binds and one on the piece where line B binds, since the curtailment has three linear pieces and the floor at zero supplies the third. Drawn at random, the rare piece, hit with probability 0.054, sets the size: 55 points for 95 percent.

Exercise 14.3. If f is convex, $\max(f, 0)$ is the maximum of two convex functions. A maximum of convex functions is convex, because at a convex combination of points each function is at most the same combination of its values, and so is their maximum. The minimum of two linear functions is concave by the same argument with the inequality reversed, and its negative is convex. The curtailment is the positive part of 160 minus that minimum, so it is convex. For a two-line network whose curtailment is not convex, feed a load of D through a transfer switch from either of two lines, one at a time, whichever has the larger rating. The deliverable power is $\max(A, B)$, which is convex, and the curtailment $(D - \max(A, B))^+$ is the positive part of a concave function, which need not be convex: on the segment from $(A, B) = (D, 0)$ to $(0, D)$ it is zero at both ends and $D/2$ in the middle. A census on it would produce planes that cut above it, and the certificate would say so.

Exercise 14.5. On the DC triangle with equal susceptances the flow on line 1–2 is linear in the loads, $\partial F_{12}/\partial(P_2, P_3) = (-0.667, -0.333)$, and the overload $\max(0, |F_{12}| - 1.4)$ is convex, positive on 19.8 percent of the box $P_2 \in [-2, -1]$, $P_3 \in [-1, -0.2]$. With central differences of the overload, step 0.05 per unit, 18 of 200 census points straddle the kink, and the surrogate exceeds the overload at 7,445 of 20,000 holdout points, by up to 0.065 per unit. That is fifteen times the one-dimensional bound $Jh/8 = 0.0042$ along P_2: each coordinate's difference errs differently, the plane tilts along the kink line, and the violation grows with distance along it. Differencing the flow instead, which is exact here because the flow is linear, and applying the active piece of the overload at the centre gives no violations, the largest gap being 2.6×10^{-15}. The three-line construction of the exercise's first part is the same: difference $A + C$ and B, which are the inputs themselves, and apply the active piece.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

A dual is a derivative, not a plan *a linear program's dual as an exact derivative; additivity of the gradient in the small; where the piecewise-linear function changes slope*
`foundation_power_flow/rating_uncertainty_queries/problems/a_dual_is_a_derivative`

Which ratings matter *a sparse gradient; constraints that do not bind contributing nothing; ranking candidates by sensitivity*
`foundation_power_flow/rating_uncertainty_queries/problems/which_ratings_matter`

A cheap model that pays for itself *a control variate built from a model that is known to be wrong; correlation rather than accuracy as the thing that pays; an exactly known expectation as the enabling condition*
`foundation_power_flow/rating_uncertainty_surrogate/problems/a_cheap_model_that_pays_for_itself`

What a wind sensor buys *uncertainty as a cost measured in amperes; the value of a measurement expressed in the units of the decision; diminishing returns to precision*
`transmission/dlr_rating_uncertainty/problems/what_a_wind_sensor_buys`

What the sensors are worth on a real corridor *three rating schemes on one circuit; an uplift that can be negative; a confidence penalty against a conservative assumption*
`transmission/wirewarrior_deployed_value/problems/what_the_sensors_are_worth`

Chapter 15

Combinatorial failure

The interventions of Chapter 13 were chosen by the operator. The failures of this chapter are not. Lines trip, transformers trip, breakers stick, and they do so in combinations that nobody selected, at rates that depend on the weather, and sometimes for a shared reason. The operator's questions are then combinatorial: which components are most likely to fail together, which combinations would hurt most, how much of the total risk the plausible combinations carry, and what it costs to leave the rest unexamined. This chapter treats those questions as inference on the hybrid model of Chapter 7, one branch per failure set, and shows that the machinery already built answers them: a prior over failure sets that the pruning proposition of Chapter 9 makes enumerable, a consequence per branch that is an intervention of Chapter 13 and a rank-one update of Chapter 11, and a mixture over the branches whose weights and tails are the risk. Two worked examples carry it, an network of six buses and a load bus with four transformers, and the second shows what the first cannot: that a shared cause defeats redundancy in a way independent failures never do.

15.1 The question

A system with L components that can fail has 2^L possible failure sets. Let $S \subseteq \{1, \dots, L\}$ be the set of failed components, let $\mathbb{P}(S)$ be its prior probability, and let $C(S) \geq 0$ be the consequence of that failure set: the load that cannot be served, the overload beyond ratings, the depth by which a bus voltage falls below its limit, or whatever the operator counts. The four questions are:

1. *Risk*: what is $\mathbb{E}[C] = \sum_S \mathbb{P}(S)\, C(S)$, and what is the probability $\mathbb{P}(C > c)$ of a consequence beyond some level?
2. *Ranking*: which failure sets contribute most to the risk, $\mathbb{P}(S)\, C(S)$ in decreasing order?
3. *Worst case*: among sets of size k, which has the largest consequence, $\max_{|S|=k} C(S)$? This is the $N - k$ question in its deterministic form, and the set that attains it is the one an adversary would choose.
4. *Coverage*: if only sets up to size k are examined, how much of the risk has been captured, and how much probability has been left unexamined?

The deterministic tradition answers only the third, usually for $k = 1$: a system is $N - 1$ *secure* if no single failure produces an unacceptable consequence, and the check is to try each of the L single failures in turn. That is a valuable check and an incomplete one. It weights every single failure equally, though some are far more likely than others; it says nothing about pairs, though pairs happen; and it reports a verdict, secure or not, rather than a risk. The probabilistic

Table 15.1: Coverage by enumeration depth under independent equal-rate failures: the number of failure sets of size exactly k, the probability that at most k components fail, and the probability left unexamined.

	$L=7$, $q=0.02$			$L=100$, $q=0.01$			$L=500$, $q=0.005$		
k	sets	$\mathbb{P}(\lvert S\rvert \le k)$	left	sets	$\mathbb{P}(\lvert S\rvert \le k)$	left	sets	$\mathbb{P}(\lvert S\rvert \le k)$	left
0	1	0.868	0.13	1	0.366	0.63	1	0.082	0.92
1	7	0.992	$7.9\,10^{-3}$	100	0.736	0.26	500	0.287	0.71
2	21	0.9997	$2.6\,10^{-4}$	4,950	0.921	0.079	$1.2\,10^{5}$	0.543	0.46
3	35	$1-5\,10^{-6}$	$5.3\,10^{-6}$	$1.6\,10^{5}$	0.982	0.018	$2.1\,10^{7}$	0.758	0.24
4	35	$1-7\,10^{-8}$	$6.5\,10^{-8}$	$3.9\,10^{6}$	0.997	$3.4\,10^{-3}$	$2.6\,10^{9}$	0.892	0.11
5	21	1	$4\,10^{-10}$	$7.5\,10^{7}$	0.9995	$5.4\,10^{-4}$	$2.6\,10^{11}$	0.958	0.042

version answers all four, and the $N-1$ check is its $k=1$ slice.

15.2 The prior over failure sets

15.2.1 Independent failures

The simplest prior takes each component to fail independently with its own probability q_i over the horizon of interest. Then

$$\mathbb{P}(S) \;=\; \prod_{i\in S} q_i \prod_{i\notin S}(1-q_i), \tag{15.1}$$

and when the rates are equal, $q_i = q$, the probability depends only on the size of the set: $\mathbb{P}(S) = q^{|S|}(1-q)^{L-|S|}$, and the number of failures is binomial with mean Lq. The prior decays geometrically in the size of the set, by a factor $q/(1-q)$ per additional failure, and this decay is what makes the combinatorial question tractable.

Proposition 15.1 (Coverage by size)**.** *Under independent equal-rate failures, the total probability of all failure sets of size greater than k is the binomial tail $\mathbb{P}(|S|>k) = \sum_{j>k}\binom{L}{j}q^j(1-q)^{L-j}$. By Proposition 9.3, examining only the sets of size at most k and renormalizing errs in any probability by at most that tail; the risk it misses is at most the tail times the largest consequence.*

Proof. The first statement is the definition of the binomial distribution applied to $|S|$, which is a sum of L independent Bernoulli variables with success probability q. The second is Proposition 9.3 with the discarded set being every S with $|S|>k$ and the discarded weight being its total probability. For the risk, $\sum_{|S|>k}\mathbb{P}(S)C(S) \le \max_S C(S)\sum_{|S|>k}\mathbb{P}(S)$. □

Table 15.1 evaluates the tail for three systems, from the script `ch15_combinatorial.py` in the book's `scripts` directory, which supplies every number in the chapter. Seven lines at two percent, the small network below, are covered to 2.6×10^{-4} by 29 sets. A hundred lines at one percent, a real transmission region, need the 161,700 sets of size three to reach 98 percent and the 3.9 million of size four to reach 99.7. Five hundred components at half a percent, a large network or a data center's equipment list, are not covered to 96 percent until size five, at 2.6×10^{11} sets, which no one enumerates. The decay that makes seven lines trivial makes five hundred hard, because the expected number of failures, Lq, has grown from 0.14 to 2.5, and the sets that carry the probability are the ones with two or three failures, of which there are millions.

Three things follow from the table. On a small system, enumerate everything; there is no reason not to. On a medium system, enumerate to the depth the tail requires for the accuracy required, which for a risk accurate to one percent is $k = 3$ at a hundred lines, and report the tail as the unexamined probability. On a large system, enumeration to sufficient depth is impossible, and one of three things must give: the consequence must be cheap enough to sample the prior instead of enumerating it, the structure must separate the components into regions whose failure sets can be enumerated independently, or the question must be narrowed from "all sets" to "the sets that matter", which is the ranking and worst-case questions with bounds. The rest of the chapter takes the first two systems and the last option.

15.2.2 Failure as an intervention

A failed component is an intervention in the sense of Chapter 13: the closure of a failed line is replaced by "no flow", the closure of a failed transformer by "no power carried", and in the hybrid model of Chapter 7 the replacement is carried by an availability variable a_i in the closure's scope, $F_{ij} = a_{ij} B(\theta_i - \theta_j)$. A failure set S is the branch with $a_i = 0$ for $i \in S$ and $a_i = 1$ otherwise. The consequence $C(S)$ is a readout of that branch: the overload of the surviving lines, the load at buses the failures have cut off, the voltage of a bus whose transformers are out. The prior $\mathbb{P}(S)$ is the product of the availability priors. And the risk $\mathbb{E}[C]$ is the mixture of the branch consequences by their weights, which is what Chapter 7 said a discrete variable does to a posterior, now with L of them.

15.3 The consequence per branch, and how to get it cheaply

For a network in the DC approximation, the consequence of a failure set needs the flows with the failed lines removed, and two things can happen. The surviving network can remain connected to the reference, in which case the flows redistribute and some may exceed their ratings; or a failure can cut a set of buses off entirely, an *island*, in which case those buses can be served by nothing and their load is shed. Both are visible from the graph before any solve: connectivity from the reference is a graph search, and the islanded buses' load is a sum.

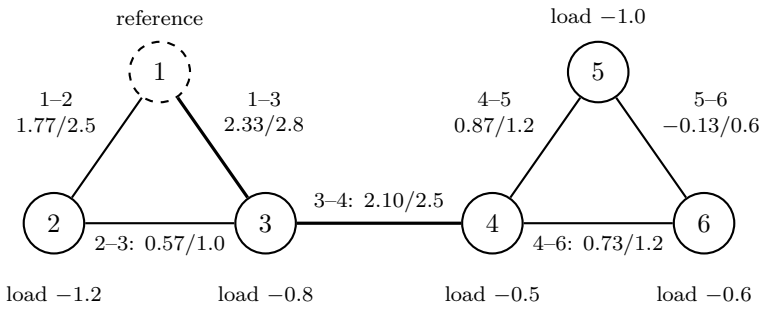
For the connected part, the flows after an outage are a rank-one update of the flows before it, and the update has a name.

Proposition 15.2 (Line outage distribution factors)**.** *In a DC network, let F_e be the pre-outage flow on line e, and let line o be removed without islanding any bus. Then the post-outage flows are*

$$F'_e \;=\; F_e + \mathrm{LODF}_{e,o}\, F_o, \qquad \mathrm{LODF}_{e,o} \;=\; \frac{\mathrm{PTDF}_{e,o}}{1 - \mathrm{PTDF}_{o,o}}, \tag{15.2}$$

where $\mathrm{PTDF}_{e,o}$ is the change in the flow on line e when one unit of power is injected at the sending end of line o and withdrawn at its receiving end, with all lines in service.

Proof. Write the nodal equations of Chapter 2 as $\mathbf{B}'\boldsymbol{\theta} = \mathbf{P}$, with $\mathbf{B}'$ the reduced susceptance matrix, and let $\mathbf{a}_o$ be the incidence column of line o (entries $+1$ at its sending bus, -1 at its receiving bus, zero elsewhere, reference row dropped). Removing line o subtracts its contribution: $\mathbf{B}'' = \mathbf{B}' - B_o \mathbf{a}_o \mathbf{a}_o^{\mathsf{T}}$, a rank-one change. Rather than invert $\mathbf{B}''$, observe that the network with line o removed and the same injections carries the same flows as the network with line o present, the same injections, and an additional transfer of F'_o units injected at o's receiving end and



labels: base-case flow / rating, per unit; thick lines are the two whose loss costs most

Figure 15.1: The six-bus network of Chapter 8 with line ratings. Every line fails independently with probability 0.02. The tie line is the only path from region B to the reference: its loss islands buses 4 to 6.

withdrawn at its sending end, chosen so that line o carries zero net flow. (The transfer cancels the line's flow, and a line carrying zero flow can be removed without changing anything else.) With line o present the flows are linear in the injections, so the transfer of t units along o's path changes every flow by $t\,\mathrm{PTDF}_{e,o}$, by definition of the factor. The line's own flow becomes $F_o + t\,\mathrm{PTDF}_{o,o}$, and setting the net flow to zero after subtracting the transfer itself, $F_o + t\,\mathrm{PTDF}_{o,o} - t = 0$, gives $t = F_o/(1 - \mathrm{PTDF}_{o,o})$. Every other flow is then $F_e + t\,\mathrm{PTDF}_{e,o}$, which is equation (15.2). The denominator vanishes only when $\mathrm{PTDF}_{o,o} = 1$, which is when line o is the only path between its ends, which is when removing it islands, the case excluded. □

The proposition is the Sherman–Morrison identity of Chapter 11 in physical clothing: a line outage is a rank-one change to the precision of the network, and its effect on every flow is one column of the pre-outage sensitivity, scaled. The column $\mathrm{PTDF}_{\cdot,o}$ is one solve against the base-case factorization, so the consequences of all L single outages cost L solves against one factorization, and pairs cost two applications of the same formula. On the six-bus network the script checks the proposition on the outage of line 1–3: the predicted flows agree with a full re-solve to 5×10^{-15}.

15.4 Worked example: a six-bus network

Take the two-region network of Chapter 8: buses 1 to 3 in a triangle with bus 1 the reference, buses 4 to 6 in a triangle, one tie line 3–4, seven lines in all, every susceptance 10 per unit, and the nominal loads of that chapter. Give each line a rating, chosen so that the base case is within limits and single outages are not: Figure 15.1 shows the network with its base-case flows and ratings. Each line fails independently with probability 0.02 over the horizon. The consequence of a failure set is the total overload of the surviving lines beyond their ratings plus the load shed at islanded buses, in per unit.

All 128 **sets, exactly.** Seven lines give 128 failure sets, and the script solves every one: a connectivity check, a DC solve on the surviving connected part, the overload and the shed. Table 15.2 gives the seven single outages and the eight sets that contribute most to the risk.

Table 15.2: The six-bus network: single outages, and the eight failure sets that contribute most to the risk $\mathbb{E}[C] = 0.165$ per unit. C is overload plus shed load.

line out	overload	shed	C	$\mathbb{P}(S)\,C$
1–2	1.50	0	1.50	0.0266
1–3	3.50	0	3.50	0.0620
2–3	0.10	0	0.10	0.0018
3–4	0	2.10	2.10	0.0372
4–5	0.80	0	0.80	0.0142
4–6	0.40	0	0.40	0.0071
5–6	0	0	0	0
failure set	$\mathbb{P}(S)$	C	$\mathbb{P}(S)\,C$	share of risk
{1–3}	1.8×10^{-2}	3.50	6.2×10^{-2}	37.6%
{3–4}	1.8×10^{-2}	2.10	3.7×10^{-2}	22.5%
{1–2}	1.8×10^{-2}	1.50	2.7×10^{-2}	16.1%
{4–5}	1.8×10^{-2}	0.80	1.4×10^{-2}	8.6%
{4–6}	1.8×10^{-2}	0.40	7.1×10^{-3}	4.3%
{2–3}	1.8×10^{-2}	0.10	1.8×10^{-3}	1.1%
{1–3, 4–5}	3.6×10^{-4}	4.30	1.6×10^{-3}	0.9%
{1–2, 1–3}	3.6×10^{-4}	4.10	1.5×10^{-3}	0.9%

The network is not $N-1$ secure: six of the seven single outages produce an overload or a shed. The loss of line 1–3 is the worst, because it carries the most and its flow must then go around through line 1–2, which it overloads by 1.6, and through line 2–3, which it overloads by 1.9; the two overloads sum to 3.5. The loss of the tie line is next: nothing overloads, but region B is islanded and its 2.1 per unit of load is shed entirely. The loss of line 5–6 costs nothing, since the two loads it serves are reached from bus 4 either way. The ranking by $\mathbb{P}(S)C(S)$ is the ranking by consequence among the single outages, because they share a probability, and the first pair enters at seventh place with less than one percent of the risk.

Risk and coverage. The risk over all 128 sets is $\mathbb{E}[C] = 0.165$ per unit, and the probability of any consequence at all is 0.114. Figure 15.2 draws the two coverage curves against enumeration depth: the probability retained by the sets of size at most k, and the fraction of the risk they capture. The single outages carry 99.2 percent of the probability and 90.2 percent of the risk; adding the pairs brings the risk to 99.6 percent; the triples close it. The risk curve lags the probability curve because the consequence grows with the size of the set, so the rare large sets carry more risk per unit of probability than the common small ones. On this network the lag is a few percent; on a network where pairs cause cascades it can be much larger, and the risk curve is the one to watch.

The worst sets. The worst single outage is line 1–3, at 3.5. The worst pair is line 1–3 with line 4–5, at 4.3: the overloads of the first plus the overload the second causes in region B. The pair of lines 1–2 and 1–3 is close behind at 4.1, and it is a different kind of event: with both lines from the reference to the triangle gone, bus 2 and bus 3 are served only through line 2–3 from each other, and nothing reaches them from bus 1, so the whole network is islanded except the reference; every load is shed. On seven lines these maxima are found by looking. On seven hundred they are the interdiction problem of §15.7.

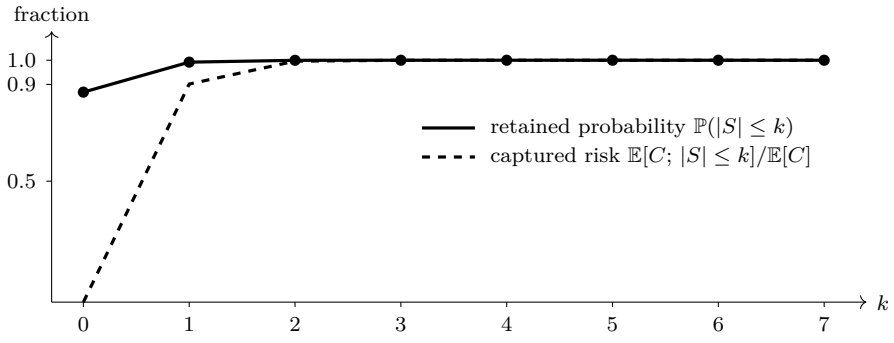


Figure 15.2: Coverage against enumeration depth on the six-bus network: the probability carried by the failure sets of size at most k (solid), and the fraction of the total risk they capture (dashed). The risk lags the probability because larger sets have larger consequences.

Uncertain loads. So far the loads were their nominal values and each branch's consequence was a number. Give each load the ± 0.2 prior of Chapter 8, and each branch's consequence becomes a distribution: the flows are linear in the loads, so within a branch the overload is a convex function of a Gaussian vector, and the script samples it. The probability of any consequence rises from 0.114 to 0.225, and the risk from 0.165 to 0.195. The doubling of the probability is the base case: with nominal loads it has no overload, but with the loads uncertain it has a 0.13 probability of one, since line 1–3 carries 2.33 against a rating of 2.8 and the load uncertainty pushes it over about one time in eight. Load uncertainty adds risk where the margins are thin, and it adds it to the branch that has all the probability.

15.5 The posterior over failure sets

Everything so far was prior: the risk before any reading. In operation the meters are reading, and the question changes from "which failures might occur" to "which failures have occurred, and what is the risk now". The hybrid model answers it the way Chapter 7 answered the one-line version: each failure set is a branch, each branch has an evidence given the readings, and the posterior over sets is the prior times the evidence, normalized. A meter on a failed line reads zero; a meter on a surviving line reads that branch's flow; and the posterior weighs every set by how well its flows explain what the meters say.

Let the meters on lines 1–2 and 3–4 read 4.10 and 2.10, each to ± 0.05; these are the flows the network carries with line 1–3 out and nothing else wrong. Figure 15.3 gives the posterior over the 128 sets. The outage of line 1–3 alone takes 94 percent: its prior was 1.8 percent, and two readings raised it by a factor of fifty, because no other single set puts 4.1 on line 1–2 while keeping 2.1 on the tie. The remaining six percent is divided equally among three sets, line 1–3 together with each of the three lines inside region B. The meters cannot distinguish these from the single outage, and the reason is Chapter 8's: a failure inside region B that islands no bus changes nothing that region B tells the rest of the network, so the tie flow stays at 2.10 and the meters are blind to it. Their posterior weight is their prior weight relative to the single outage, one extra failure at two percent each, and only a meter inside region B would separate them. The posterior says which meter to buy next, as Chapter 11 said it would.

The risk conditional on the readings is $\mathbb{E}[C \mid y] = 3.52$ per unit against a prior risk of 0.165:

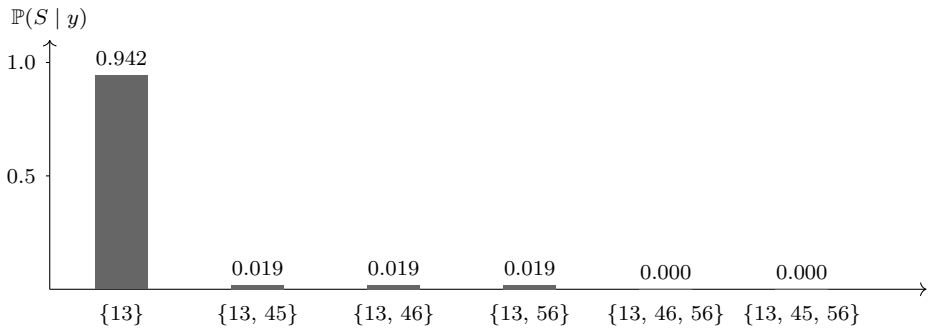


Figure 15.3: The posterior over failure sets on the six-bus network after two meters read $F_{12} = 4.10$ and $F_{34} = 2.10$, each to ± 0.05: the six most probable sets. The outage of line 1–3 is identified at 94 percent; the residual weight is on that outage together with a failure inside region B that the two meters cannot see.

not because the world has become more dangerous but because the readings have revealed that the largest single consequence has already occurred, and the posterior is nearly certain of it. The probability of a consequence given the readings is one. This is the operational form of the $N - k$ question: not "what could fail" but "what has failed, how sure are we, and what would the next failure cost". The last is a branch over the surviving lines with the posterior as its prior, and on this network the expected additional consequence of one more failure in the next horizon is close to zero, since the worst has happened and several further failures would island load that is already overloading the lines that serve it. That is a fact about this small network, not a rule; on a larger one the next failure after a first is often the one that cascades, and the conditional risk is the number to watch.

Principle 15.1. *Given readings, the failure sets have a posterior: prior times evidence per branch. It identifies what has failed, says what the meters cannot tell apart and why, and turns the prior risk into the conditional risk that operation needs.*

15.6 Enumeration by regions

The six-bus network has 128 failure sets and each was solved on the whole network. It did not have to be. The network is two regions and a tie, and its failure sets are the product of the regions' failure sets: region A's three lines give eight, region B's three lines give eight, the tie gives two, and $8 \times 8 \times 2 = 128$. By Chapter 8, each region's contribution to any branch is its message on the ports of the tie, a two-by-two precision and a two-vector, computed from the region alone. So the 128 branches need eight region solves for A, eight for B, and 128 combinations of two small messages with the tie's factor: sixteen region solves in place of 128 network solves, and the combinations are two-by-two arithmetic.

The saving is a factor of eight on six buses and it grows with the regions. A network of m regions with L_r failing lines in region r has $\prod_r 2^{L_r}$ failure sets, but only $\sum_r 2^{L_r}$ region solves, and the sum is smaller than the product by a factor that is exponential in the number of regions. With pruning by prior weight inside each region, so that only the sets of size up to k_r are kept, the count is $\sum_r \sum_{j \le k_r} \binom{L_r}{j}$ region solves, which is what makes a hundred-line network with three

lines out affordable when the lines are spread over ten regions and not when they are treated as one.

The script checks the one thing that makes this exact: that a failure inside region B changes region B's message only through what crosses the tie. With no failure in B, or with line 5–6 out, the tie carries 2.100 and the message is the same. With lines 4–5 and 4–6 both out, buses 5 and 6 are islanded, 1.60 of load is shed, and the tie carries 0.500: the message has changed, and it has changed because region B's total draw has changed, which is what its message carries. Region A's eight solves are reused across all of region B's eight failure sets, and region B's across all of A's, exactly as §8.5 arranged; and the posterior of §15.5 is computed the same way, one evidence per combination of region messages, with the readings inside each region entering that region's message and nowhere else.

15.7 The worst k-set as a query

The third question, $\max_{|S|=k} C(S)$, is an optimization over $\binom{L}{k}$ sets, and it is the one question here that the prior does not help with: the worst set is the worst whatever its probability. On small systems it is found by enumeration, as above. On large ones it is the *interdiction* problem, in which an adversary chooses the k components to disable so as to maximize the damage, and it is hard in general.

Three things the framework offers bear on it. The consequence per set is cheap when it is a rank-one update per component removed, so enumeration reaches further than a re-solve per set would. The convexity of §14.7 gives bounds, and the two directions of bound do different jobs. When C is the value of a linear program in the surviving capacities, the surrogate of Chapter 14 bounds it from below on any set. A lower bound certifies a set already found: its consequence is at least that much. It cannot prune. In a branch-and-bound that maximizes over sets, a branch is a partial set, and it may be discarded only when an upper bound on the consequence of every completion of it falls below the best value found so far. A score that can undercut the true completion value discards branches that may hold the optimum, and a search pruned by it returns a best-found set, not a proved one. And the region structure of Chapter 8 separates the search: a failure inside a region changes that region's message to its ports and nothing else, so the worst set can be sought region by region and then across the ports, at a cost that scales with the regions rather than with the whole.

For the smallest k the global optimum is enumerable, and there the answer is proved. Chapter 23 works a published stochastic benchmark on a 73-bus network with 216 candidate components and 200 background outage scenarios. It proves the $k = 1$ optimum by evaluating all 216 candidates. It proves the $k = 2$ optimum by evaluating all 15,602 candidates that remain once interchangeable components are merged, at 3,040,242 linear programs. For $k = 3$ to 10 a best-first search reproduces the published optima. Its pruning score is not a valid upper bound, so its certificate is conditional on that score. Every one of those runs stopped at its budget of 500,000 evaluated outage states, and no bound on the optimality gap is reported. The same study finds that ranking the components one at a time and taking the top k recovers the published objective at every k, at about 42,600 outage states whatever k is. The network is capacity-adequate, so the damage a set does is governed by how much reserve it removes, and the problem behaves like a knapsack rather than a search over interacting failures. Beyond the enumerable k, the honest form of the answer is a best-found set together with what certifies it; when nothing certifies it, the answer says so.

Principle 15.2. *A failure set is a branch, its consequence is an intervention's readout, and its probability is a product of availabilities. Risk is the mixture, ranking is the product, coverage is the binomial tail, and the worst case is a search that the prior does not shorten but the physics does.*

15.8 Common cause and correlated failure

Independence was the assumption of equation (15.1), and it is the assumption most often wrong. Components share a weather, a supply, a maintenance crew, a design flaw; when the shared thing fails, they fail together, at a rate that the product of their individual rates does not approach. Two models capture this, and both fit the factor graph as a variable with a prior.

15.8.1 A shared shock

Let a common cause $c \in \{0, 1\}$ occur with probability r, and let every component fail when it does, in addition to failing independently with probability q when it does not. This is the simplest Marshall–Olkin shock model: independent shocks, one hitting every component and L hitting one each. The probability that a given pair both fail is

$$\mathbb{P}(a_i = 0,\ a_j = 0) \ = \ r + (1 - r)\, q^2, \tag{15.3}$$

against q^2 under independence, and the probability that all L fail is $r + (1 - r)q^L$ against q^L. For r comparable to q^2 the pairs are twice as likely as independence says; for r comparable to q the shared cause dominates every joint failure, and no amount of redundancy helps, because redundancy protects against independent failures and the shared cause fails everything at once. In the factor graph, c is one discrete variable with a prior $\mathbb{P}(c = 1) = r$, and every availability's prior becomes conditional on it: $\mathbb{P}(a_i = 0 \mid c) = 1$ if $c = 1$ and q otherwise. The branch count doubles, from 2^L to 2^{L+1}, and the shared-cause branch is one of them.

15.8.2 A mixture over the rate

Let the failure rate itself be uncertain, taking the value q_w with probability π_w over a set of conditions w: a hot day, a storm, a normal day. Conditional on the condition the failures are independent; unconditionally they are not. The prior over a set is the mixture

$$\mathbb{P}(S) \ = \ \sum_{w} \pi_w \, q_w^{|S|} (1 - q_w)^{L - |S|}, \tag{15.4}$$

which is exchangeable but not independent. The induced correlation between two components' failures is

$$\rho \ = \ \frac{\mathbb{E}_w[q_w^2] - \bar{q}^{\,2}}{\bar{q}(1 - \bar{q})}, \qquad \bar{q} = \mathbb{E}_w[q_w], \tag{15.5}$$

which is the variance of the rate across conditions divided by the Bernoulli variance at the mean rate.

Proof. Two failures are indicators X_i, X_j with $\mathbb{E}[X_i] = \mathbb{E}_w[\mathbb{E}[X_i \mid w]] = \bar{q}$ and, by conditional independence, $\mathbb{E}[X_i X_j] = \mathbb{E}_w[q_w^2]$. Then $\mathrm{Cov}(X_i, X_j) = \mathbb{E}_w[q_w^2] - \bar{q}^2 = \mathrm{Var}_w[q_w]$, and each indicator's variance is $\bar{q}(1 - \bar{q})$. □

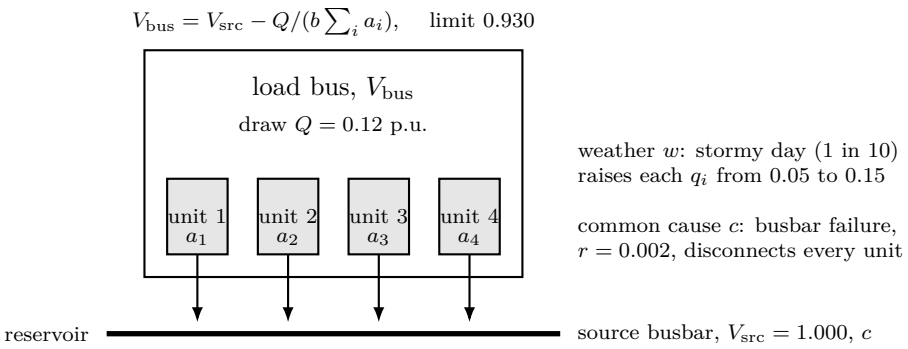


Figure 15.4: A load bus fed by four identical transformers sharing one source busbar. Each transformer has an availability a_i; the busbar's failure c is a common cause; the weather w raises every transformer's failure rate together.

The correlation is small when the rate varies little, and it is not the point. The point is the tail: a condition with a high rate makes large failure sets far more probable than the mean rate suggests, since $q_w^{|S|}$ grows with $|S|$ faster for the larger q_w. A ten-percent chance of a day on which the rate triples multiplies the probability of a triple failure by a factor that the correlation of a few hundredths does not hint at. In the factor graph, w is one discrete variable with prior π_w, and the availabilities' priors are conditional on it, exactly as for the shock; the difference is that the shock fails everything and the condition raises everything's rate.

15.9 Worked example: a transformer bank

A load bus draws 0.12 per unit of reactive power, and $n = 4$ identical transformers each connect it to a source busbar at 1.000 per unit, with susceptance 1.5 per unit each. The bus voltage must stay above 0.930. Each transformer fails over the horizon with probability 0.05; the source busbar, which all four share, fails with probability 0.002, and when it does no transformer carries anything. On a stormy day, which is one day in ten, the transformers' failure rate is 0.15. Figure 15.4 draws the system.

The consequence. With m transformers in service the bus is a single node with draw Q and a susceptance $m\,b$ to the busbar, and its steady voltage is the closed form

$$V_{\text{bus}}(m) \;=\; V_{\text{src}} - \frac{Q}{m\,b}, \tag{15.6}$$

which is the export feeder of Chapter 1 with one node. The voltages are 0.980, 0.973, 0.960 and 0.920 per unit for four, three, two and one transformers, and the bus is dark with none. Two transformers keep the bus above its limit; the bank is $N - 2$ tolerant, which is more redundancy than most.

Independent failures. The limit is breached when three or more transformers fail, which under independence at $q = 0.05$ has probability $\binom{4}{3}q^3(1-q) + q^4 = 4.8 \times 10^{-4}$: once in two thousand horizons.

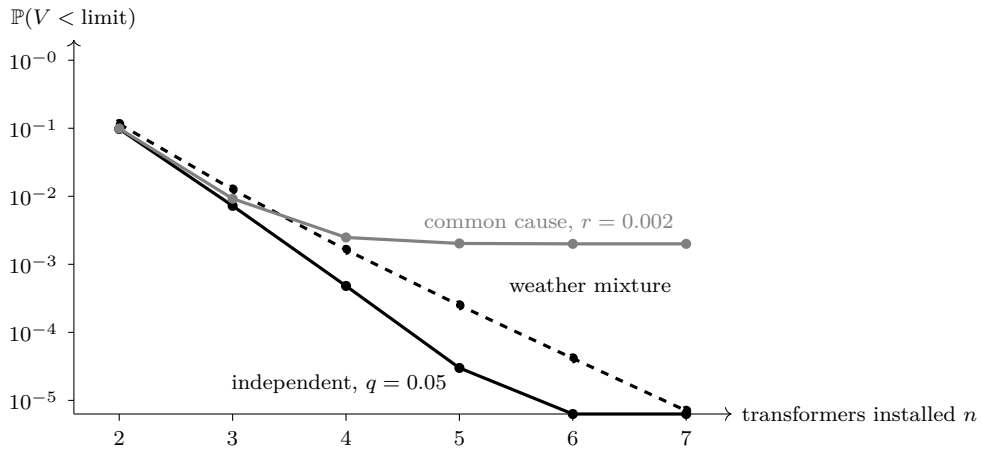


Figure 15.5: Probability of breaching the bus's limit against the number of transformers installed, on a logarithmic scale, under three priors: independent failures at $q = 0.05$; the weather mixture; and the shared busbar failure at $r = 0.002$. Adding transformers drives the independent risk down by a factor of twenty per unit and leaves the common-cause risk at its floor.

With the common cause. The busbar fails with probability 0.002, and when it does every transformer is out whatever its own state. The probability of breaching the limit is $r + (1 - r) \times 4.8 \times 10^{-4} = 2.5 \times 10^{-3}$, five times the independent figure, and four fifths of it is the busbar. The four transformers' redundancy protected against the independent failures to one part in two thousand; against the shared cause it protected against nothing, and the shared cause, at one part in five hundred, is now the risk.

With the weather. On a stormy day the exceedance probability is $\binom{4}{3}(0.15)^3(0.85) + (0.15)^4 = 1.2 \times 10^{-2}$, twenty-five times the calm-day figure; averaged over the one-in-ten stormy days, the unconditional probability is 1.6×10^{-3}, three and a half times the independent figure. The correlation the mixture induces between two transformers' failures, by equation (15.5), is 0.016, a number that would pass unnoticed in any table of correlations. The tail is where it shows: the probability that three named transformers fail together is $q^3 = 1.25 \times 10^{-4}$ under independence and $0.1 \times 0.15^3 + 0.9 \times 0.05^3 = 4.5 \times 10^{-4}$ under the mixture, nearly four times as much, and it is the triple failures that breach the limit.

Figure 15.5 draws the exceedance probability against the number of transformers installed, from two to seven, under the three priors. Under independence each added unit divides the risk by about twenty, and by seven transformers it is below one in a million. Under the weather mixture the curve falls too, but more slowly, since the stormy-day rate is three times the calm one and each added transformer divides the hot-day risk by only about six. Under the common cause the curve flattens at 0.002 and stays there: the seventh unit is as useless as the fifth, because the failure that matters takes all of them. An operator who reads only the independent curve buys transformers; one who reads the common-cause curve buys a second loop.

Principle 15.3. *Redundancy protects against independent failures and against nothing else. A shared cause puts a floor under the risk that no number of components lowers, and a shared condition fattens the tail far beyond what its correlation suggests. Both are one discrete variable in the graph, and the branches say what the redundancy is worth.*

15.10 In practice

Contingency lists. Operators do not enumerate; they maintain a list of the contingencies worth checking, built from experience and from screening. The ranking of §15.4 is the principled form of that list: every single outage costs one rank-one update, the pairs among the top singles cost two, and the list is the sets with the largest $\mathbb{P}(S)C(S)$ to whatever depth the tail requires. What the probabilistic version adds to the traditional list is the weight, so that a likely small event and an unlikely large one are compared on one scale, and the coverage figure, so that the list's completeness is a number rather than a judgment.

Screening by distribution factors. For a large network, the LODF matrix of Proposition 15.2, L columns of one solve each, screens every single outage for every line's post-outage flow at no further cost, and the pairs can be screened by a second application before any is solved exactly. The screen is exact in the DC approximation and a guide in the AC one, where the flows after an outage must be re-solved for the sets the screen flags.

Where to stop. Report the enumeration depth and the tail it leaves. A risk quoted without its unexamined probability is a risk quoted without an error bar. For a hundred components at one percent the depth is three for one-percent accuracy; for five hundred at half a percent no depth is affordable and the honest report is a sampled risk with a sampling error, or a risk over regions with the regions' messages combined exactly.

The common mistakes. Three recur. The first is to treat the $N-1$ verdict as the risk: a system can be $N-1$ secure and carry most of its risk in pairs, or insecure to a single failure that almost never happens. The second is to model failures as independent because the data on their dependence is poor; the shared loop of §15.9 raised the risk fivefold at a probability of one in five hundred, and a modeler who does not know r should carry it as a variable with a wide prior rather than set it to zero. The third is to report a worst case without a bound: on any system too large to enumerate, the worst set found is a lower bound on the worst set that exists, and it should be reported as one.

What to report. For each question, the number and its qualifier: the risk with its unexamined tail; the ranked list with its depth; the worst set with its bound; and, always, the do-nothing baseline of Chapter 13, since every mitigation proposed against the ranked list must be compared with leaving the list as it is.

15.11 What is missing

Every failure in this chapter was instantaneous and every consequence was a steady state. Real failures unfold: a line trips, flows redistribute, another line overloads and trips seconds later, and the cascade's consequence depends on the order and the timing. That needs the dynamics of Chapter 16. And every enumeration here was over a whole system; Chapter 17 organizes the enumeration by the regions of Chapter 8, where a region's failure sets are enumerated inside the region and combined at its ports, and Chapter 18 asks how large a region can be before that stops working. Part IV ends here; the operator's questions have each been asked, and each has been a readout of one posterior, a branch of it, or a search over its branches.

Notes and further reading

Power-system reliability evaluation, with independent-failure priors, contingency enumeration, and the loss-of-load indices that are this chapter's risk and exceedance probability, is treated at book length by Billinton and Allan [8]. The $N-1$ criterion and its contingency lists are standard operating practice; Wood, Wollenberg and Sheblé [98] derive the line outage distribution factors of Proposition 15.2 and their use in contingency screening, and Guler, Gross and Liu [28] generalize them to multiple simultaneous outages. The reading of the LODF as a Sherman–Morrison update, and of a failure set as a branch of the hybrid model whose consequence is an intervention's readout, is the book's.

The $N-k$ interdiction problem, finding the k components whose loss maximizes the consequence, is studied in a probabilistic form by Sundar, Coffrin, Nagarajan and Bent [88]. Sundar, Mastin, Garcia, Bent and Watson [89] pose its stochastic form on the RTS-GMLC network: the consequence is the minimum load shed of a DC dispatch, averaged over 200 background outage scenarios of weight $1/200$ each, for $k = 1$ to 10. Chapter 23 replicates that benchmark, with the findings summarized in §15.7. The same chapter also conditions the scenario distribution on evidence of fuel and storm regimes. Under fuel evidence the worst single outage moves from the generator at bus 121 to the one at bus 118, and no further linear program is solved, because only the scenario weights change. The shock model of §15.8 is Marshall and Olkin's [60]; the mixture over a shared rate is the standard exchangeable model of common-cause failure in reliability engineering, and the correlation formula (15.5) is elementary. That a shared condition shows in the tail and not in the correlation is the book's emphasis, and Figure 15.5 is its evidence.

Summary

- Four questions: risk, ranking, worst case, coverage. The $N-1$ check is the $k=1$ slice of the third.
- Under independent failures the prior decays geometrically in set size; coverage by depth k is the binomial tail. Seven lines are covered by 29 sets; a hundred need 161,700 for 98 percent; five hundred cannot be enumerated.
- A failure is an intervention, its consequence a readout, and in a DC network a rank-one update: the LODF, derived and checked to 10^{-15}.
- Six buses, seven lines, 128 sets: risk 0.165, singles carry 90 percent of it, pairs bring it to 99.6; the worst single is the heaviest line, the worst pair adds a region's overload. Load uncertainty doubles the probability of any consequence by threatening the base case.
- The worst k-set is a search the prior does not shorten; the rank-one consequence, the convex bound and the region structure do.
- A shared cause puts a floor under the risk that redundancy cannot lower: four transformers protect the bus to one in two thousand against independent failures and to one in five hundred, five times worse, against a loop failure at one in five hundred. A shared condition fattens the tail fourfold at a correlation of 0.016.

Exercises

Exercise 15.1. Derive equation (15.2) directly from the Sherman–Morrison identity applied to $\mathbf{B}'' = \mathbf{B}' - B_o \mathbf{a}_o \mathbf{a}_o^{\mathsf{T}}$, and show that the transfer argument in the proof of Proposition 15.2 and

the algebraic derivation give the same factor. Identify $\mathrm{PTDF}_{o,o}$ in terms of $\mathbf{a}_o^{\mathsf{T}}\mathbf{B}'^{-1}\mathbf{a}_o$.

Exercise 15.2. For L components at equal rate q, bound the binomial tail $\mathbb{P}(|S| > k)$ from above by a geometric series and find the smallest k for which the bound is below 10^{-6} when $L = 100$, $q = 0.01$. Compare with the exact tail in Table 15.1.

Exercise 15.3. On the six-bus network, add a second tie line from bus 6 to bus 1 with rating 1.5 and recompute the 256 failure sets. Which single outage is now the worst, does the tie's loss still island, and how does the risk change? Explain in terms of the region message of Chapter 8.

Exercise 15.4. Give the load bus a second source busbar serving two transformers each, with each loop failing at 0.002. Compute the exceedance probability under the two-loop common-cause model and compare with the single loop and with independence. At what per-loop failure rate does a second loop stop being worth more than a fifth unit?

Exercise 15.5. For the weather mixture with conditions w and rates q_w, show that the probability of a set of size k under the mixture, relative to the same set under the mean rate $\bar{q}$, is at least one, with equality only when the rate does not vary, and find the ratio for $k = 3$ in the transformer-bank example. (Hint: Jensen's inequality applied to $q \mapsto q^k$.)

Solutions to selected exercises

Exercise 15.1. Sherman–Morrison gives $(\mathbf{B}' - B_o\mathbf{a}_o\mathbf{a}_o^{\mathsf{T}})^{-1} = \mathbf{B}'^{-1} + \dfrac{B_o\,\mathbf{B}'^{-1}\mathbf{a}_o\mathbf{a}_o^{\mathsf{T}}\mathbf{B}'^{-1}}{1 - B_o\,\mathbf{a}_o^{\mathsf{T}}\mathbf{B}'^{-1}\mathbf{a}_o}$. The post-outage angles are $\boldsymbol{\theta}' = \mathbf{B}''^{-1}\mathbf{P} = \boldsymbol{\theta} + \dfrac{B_o\,\mathbf{B}'^{-1}\mathbf{a}_o\,(\mathbf{a}_o^{\mathsf{T}}\boldsymbol{\theta})}{1 - B_o\mathbf{a}_o^{\mathsf{T}}\mathbf{B}'^{-1}\mathbf{a}_o}$. Now $\mathbf{a}_o^{\mathsf{T}}\boldsymbol{\theta}$ is the pre-outage angle difference across line o, so $B_o\mathbf{a}_o^{\mathsf{T}}\boldsymbol{\theta} = F_o$; and $\mathbf{B}'^{-1}\mathbf{a}_o$ is the angle response to a unit transfer along o, so $B_e\mathbf{a}_e^{\mathsf{T}}\mathbf{B}'^{-1}\mathbf{a}_o = \mathrm{PTDF}_{e,o}$ for every line e, including $e = o$. The flow on line e after the outage is $B_e\mathbf{a}_e^{\mathsf{T}}\boldsymbol{\theta}' = F_e + \dfrac{\mathrm{PTDF}_{e,o}\,F_o}{1 - \mathrm{PTDF}_{o,o}}$, which is equation (15.2); the transfer argument's t is $F_o/(1 - \mathrm{PTDF}_{o,o})$, the same scalar. In particular $\mathrm{PTDF}_{o,o} = B_o\mathbf{a}_o^{\mathsf{T}}\mathbf{B}'^{-1}\mathbf{a}_o$, the fraction of a transfer along o's ends that flows on o itself, which is one exactly when o is the only path.

Exercise 15.2. Each binomial term satisfies $\binom{L}{j+1}q^{j+1}(1-q)^{L-j-1} = \binom{L}{j}q^j(1-q)^{L-j} \cdot \dfrac{(L-j)\,q}{(j+1)(1-q)}$, and for $j \geq k$ the ratio is at most $\rho_k = \dfrac{(L-k)\,q}{(k+1)(1-q)}$, which is below one once $k + 1 > (L-k)q/(1-q)$. Then the tail is bounded by the geometric series $\mathbb{P}(|S| > k) \leq \mathbb{P}(|S| = k)\,\dfrac{\rho_k}{1-\rho_k}$. For $L = 100$, $q = 0.01$: at $k = 5$, $\mathbb{P}(|S| = 5) = 2.9 \times 10^{-3}$ and $\rho_5 = 95 \times 0.01/(6 \times 0.99) = 0.160$, giving a bound of 5.5×10^{-4} against the exact 5.35×10^{-4}; at $k = 8$, $\mathbb{P}(|S| = 8) = 7.4 \times 10^{-6}$, $\rho_8 = 0.103$, bound 8.5×10^{-7}, below 10^{-6}. So $k = 8$ suffices, and the bound is within ten percent of the exact tail at every depth because the ratio is small. The $\binom{100}{8} = 1.9 \times 10^{11}$ sets of size eight are the reason enumeration to that depth is not done; the bound says how much is lost by stopping earlier.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Consequence is not capacity *consequence as a network property rather than a unit property; a screen that sorts by the wrong key; the largest component being irrelevant*
`foundation_power_flow/stochastic_nk_interdiction/problems/consequence_is_not_capacity`

The worst pair is not the two worst singles *interaction between simultaneous failures; greedy selection on marginal scores; superadditive consequence*
`foundation_power_flow/stochastic_nk_interdiction/problems/the_worst_pair`

When a screen stops paying *a screen that does not correlate with what it screens for; partial credit at a truncation depth; the cost of a screen that must run to the bottom*
`foundation_power_flow/contingency_risk_at_scale/problems/when_a_screen_stops_paying`

Part V

Time, hierarchy, and scale

Chapter 16

Dynamics and temporal factors

Every balance in the book so far had its accumulation term set to zero. The transformer's oil did not store heat; the bus did not store charge; the state was whatever the flows made it, instantly. Real systems store, and what they store changes the questions. A step in the losses does not move the oil's temperature at once but over hours; a line that is overloaded does not fail at once but when its conductor has heated; a reading taken now is evidence about the state a minute ago and a prediction about the state a minute hence. This chapter makes the accumulation term live. It shows that integrating the resulting equations in time is a sequence of the steady solves the book already has; that the graph of a system over many time steps is the graph of one step, copied and chained, so that everything Part III said about separators applies with the stored state as the separator between past and future; that filtering, smoothing and prediction are eliminations on that chain, and the Kalman filter is the rank-one update of Chapter 11 preceded by one Schur complement; and that a cascade, the failure that Chapter 15 could not model, is a sequence of events located on a trajectory. Two worked examples carry it: a transformer's winding and oil, filtered and smoothed through drifting losses, and the six-bus network after a line outage, whose conductors heat until one of them trips.

16.1 The accumulation term made live

Return to the balance law of Chapter 1,

$$\frac{\mathrm{d}X_{n,d}}{\mathrm{d}t} \;=\; \sum_{e \in \mathcal{E}(n)} A_{n,e}\, P_{d,e} + G_{n,d} - L_{n,d}, \tag{16.1}$$

and keep the left side. A node that can accumulate has an extensive state X, and a *storage relation* that ties it to the node's intensive state: $E = C\,T$ for a thermal mass of capacity C, $M = \rho V$ for a tank, $q = C_{\mathrm{el}} V$ for a capacitor. The unknowns now include the storage states, the rows now include the storage relations, and the balance rows at storing nodes are differential equations while every closure and every balance at a zero-storage node is algebraic. The whole is a *differential-algebraic* system,

$$\mathbf{F}\big(\mathbf{z},\ \dot{\mathbf{z}},\ t\big) = \mathbf{0}, \tag{16.2}$$

of index one: the algebraic rows can be solved for the algebraic variables given the differential ones, which is the well-posedness of §1.5 applied at each instant. Nothing about the graph has changed. The storage relation is one more closure, reading X and T; the balance row reads $\dot{X}$ as

well as the flows; the factor graph of Chapter 2 gains one variable and one square per storing node.

A transformer's winding and oil. Give a transformer's winding a heat capacity $C_1 = 1500$ kilojoules per kelvin and its oil $C_2 = 25{,}000$. The losses G enter the winding, the winding passes heat to the oil through $k = 5$ kilowatts per kelvin, and the oil sheds it to the air through $hA = 2$. The two balances become

$$C_1 \dot{T}_1 = G - k\,(T_1 - T_2), \qquad C_2 \dot{T}_2 = k\,(T_1 - T_2) - hA\,(T_2 - T_a), \tag{16.3}$$

with the closures substituted in. This is the export feeder's chain of Chapter 1 with its two nodes given mass, and at 100 kilowatts of losses it settles where that chapter's algebra put it: the oil at 70°C and the winding hot spot at 90°C, the numbers a transformer's rating is written against. It is a linear system $\dot{\mathbf{T}} = \mathbf{A}\mathbf{T} + \mathbf{b}$, and its eigenvalues say how it moves: from the script `ch16_dynamics.py` in the book's `scripts` directory, which supplies every number in the chapter, they are -3.5×10^{-3} and -8.0×10^{-5} per second, time constants of 283 seconds and 13,300 seconds. The winding responds in minutes; the oil, heavy and weakly coupled to the air, takes hours. The ratio of the two, about 47, is the system's *stiffness*, and it is the number that decides how the system may be integrated.

16.2 Integration as a sequence of steady solves

To advance the state from time t_n to $t_{n+1} = t_n + \Delta t$, replace the derivative in the differential rows by a difference. The simplest replacement that works for stiff systems is the backward difference,

$$\text{BDF1:} \qquad X^{n+1} - X^n - \Delta t\, \mathcal{B}(\mathbf{z}^{n+1}) = 0, \tag{16.4}$$

where $\mathcal{B}$ is the right side of equation (16.1) evaluated at the new time. The second-order version uses two past values,

$$\text{BDF2:} \qquad 3X^{n+1} - 4X^n + X^{n-1} - 2\Delta t\, \mathcal{B}(\mathbf{z}^{n+1}) = 0. \tag{16.5}$$

Both are *implicit*: the unknown state $\mathbf{z}^{n+1}$ appears inside $\mathcal{B}$, and each step is a nonlinear system in $\mathbf{z}^{n+1}$.

Proposition 16.1 (A time step is a steady solve)**.** *With the storage rows rewritten as equation* (16.4) *or* (16.5) *and every algebraic row unchanged, the system to be solved at each step has the same variables, the same sparsity, and the same Jacobian pattern as the steady-state system of Chapter 1, differing only in the storage rows' entries. Simulation is a loop over the steady solver.*

Proof. The steady system has the rows: balances with $\dot{X} = 0$, closures, storage relations, boundary conditions. Equation (16.4) is the balance row with $\dot{X}$ replaced by $(X^{n+1} - X^n)/\Delta t$, which adds the known X^n to the row's constant and the coefficient $1/\Delta t$ to the row's entry on X^{n+1}; it reads the same variables. Every other row is unchanged. So the Jacobian has the steady Jacobian's pattern with the storage-state columns of the balance rows carrying an extra $1/\Delta t$ (or $3/(2\Delta t)$ for BDF2), and Newton's method with the same assembly and factorization applies. □

Table 16.1: Integrating the transformer's step response over six hours: maximum error against the exact solution. Explicit Euler is accurate below its stability limit of 565 s and diverges above it; BDF1 is stable at every step with error proportional to Δt; BDF2's error is proportional to Δt^2 until the winding's fast transient limits it.

method	Δt [s]	max error in T_1 [K]	max error in T_2 [K]
explicit Euler	30	0.073	0.0043
	120	0.34	0.020
	300	1.45	0.086
	1200	5.7×10^9	3.5×10^8
BDF1	30	0.067	0.0043
	300	0.50	0.042
	1200	0.66	0.16
BDF2	30	0.020	0.0012
	300	0.50	0.028
	1200	0.66	0.049

The proposition is worth more than its proof suggests. It means there is one model, one assembly and one derivative path for the steady solve, the transient, and, by Chapter 6's Proposition 6.2, the probabilistic estimate at each time. It also means that everything said about the steady Jacobian's sparsity and factorization carries to each step, and, as §16.3 shows, to all steps at once.

Why implicit. The alternative, the forward difference $X^{n+1} = X^n + \Delta t\, \mathcal{B}(\mathbf{z}^n)$, is explicit: the new state is computed from the old one without solving anything. It is cheaper per step and it is unusable on a stiff system. For a linear system $\dot{\mathbf{T}} = \mathbf{AT}$ the explicit step multiplies each eigencomponent by $1 + \lambda\Delta t$, whose magnitude exceeds one, so that the component grows without bound, whenever $\Delta t > 2/|\lambda|$; the fastest eigenvalue sets the limit, and on the transformer it is $\Delta t < 565$ seconds, dictated by a winding whose transient is over in minutes and, for a rating question, of no interest. BDF1 multiplies by $1/(1 - \lambda\Delta t)$, whose magnitude is below one for every negative λ and every Δt: it is stable at any step, and the step can be chosen for the slow dynamics one cares about. Table 16.1 shows all three on the transformer's response to a step in the losses from 100 to 120 kilowatts, against the exact solution by matrix exponential.

At a twenty-minute step, four times the winding's time constant, BDF1 is within two thirds of a degree over the whole transient and BDF2 within five hundredths on the oil. Past the stability limit the explicit method does not merely lose accuracy, it diverges, and at a step chosen for the oil it would produce nothing usable. The stiff component that the implicit methods damp is the same one the explicit method amplifies, and the book's systems, with light and heavy masses coupled, are stiff as a rule.

Choosing the step. A fixed step is rarely right for a whole transient: fine when the state moves fast, coarse when it settles. The standard control accepts a step when a weighted measure of the change it produced is below one and chooses the next step from the ratio, with a safety factor and a cap on growth. The details are in the notes; what matters here is that the control acts on the same step-change measure for every variable of the graph, and that it shrinks the step

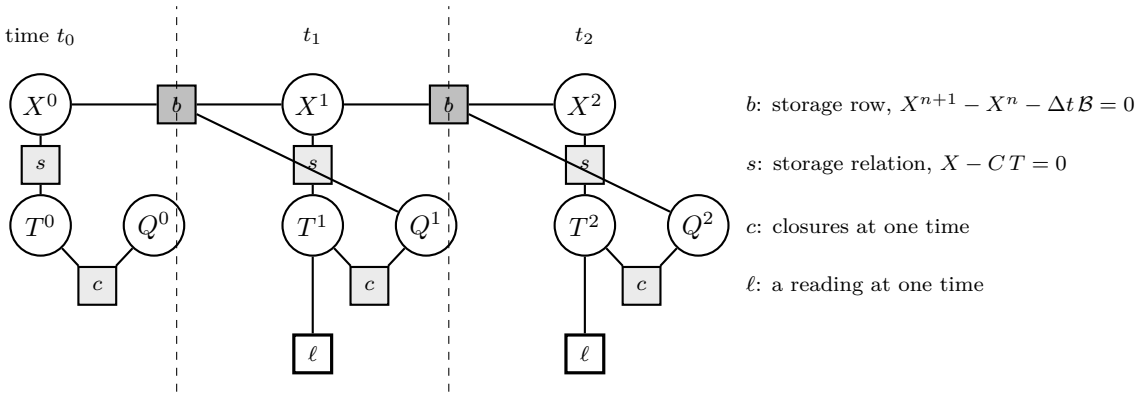


Figure 16.1: The unrolled factor graph of a storing node over three time steps, schematically. Within a step, the storage relation and the closures read variables of that step only. Only the storage rows (dark) read two times, and they read the storage state at both. The dashed cuts pass through the storage states, which are the separators between consecutive steps.

wherever a fast trajectory appears, which on a stiff system is exactly where an explicit method would have failed.

16.3 The unrolled graph

Take the factor graph of one step and copy it once per step. The copies are joined by the storage rows, equation (16.4), each of which reads X at two times; every other row reads one time only. The result is the *unrolled* factor graph of the transient, Figure 16.1, and it has a structure that decides everything that follows.

Proposition 16.2 (The stored state separates past from future)**.** *In the unrolled factor graph, the storage states X^n at one time form a separator between all variables at earlier times and all variables at later times. Hence, given X^n, the past and the future are conditionally independent.*

Proof. Every row of the model reads variables of one time step, except the storage rows (16.4), which read X^n, X^{n+1} and the flows and sources at t_{n+1} that enter $\mathcal{B}(\mathbf{z}^{n+1})$. So in the Markov graph, every edge joins two variables of the same step or joins X^n to a variable of step $n+1$. A path from step $m < n$ to step $p > n$ must pass through each intermediate step, and it can enter step n only through X^{n-1}'s edges to step n's variables and leave it only through X^n's edges to step $n+1$. Removing X^n removes every edge out of step n into the future, so no path remains, which is separation, and Proposition 5.1 gives the independence. For BDF2 the storage row reads X^{n-1} as well, and the separator is the pair (X^{n-1}, X^n). □

This is the Markov property of a dynamical system, which filtering theory takes as an axiom; here it is a consequence of the row structure, and it says which variables are the "state" in the sense that matters: the storage states, and nothing else. The temperatures, flows and sources at time n are algebraic functions of X^n and the inputs and need not be carried.

The precision is block-tridiagonal. With every row Gaussian, the unrolled model's precision has the block structure the proposition implies: a diagonal block per step, and off-diagonal

blocks only between consecutive steps, from the storage rows. Ordered by time it is banded, with bandwidth equal to the number of variables in two steps. On the transformer's chain, a three-variable state over 144 steps, the precision is 435×435 with bandwidth 5 and 1.8 percent of its entries nonzero, and its factorization costs what a chain's does: linear in the number of steps.

16.4 Process noise

The storage row (16.4) is a physics row and, like every physics row in Part II, it is given a width. Its width has a name in the filtering literature, *process noise*, and it means what the closure width of Chapter 4 meant: the amount by which the model's account of how the state moves may be wrong over one step. Unmodeled disturbances, neglected dynamics, a discretization error, an input that drifts: each is a reason the state at t_{n+1} is not exactly what the row predicts from t_n, and the width is their combined size.

Process noise has a second use, and it is the one the worked example needs. An input that is not constant, a source that drifts, a load that ramps, a parameter that ages, can be modeled as a variable at every time step joined to itself at the next by a storage row of its own, $u^{n+1} - u^n = 0$, with a width: a *random walk*. The width says how far the input may move in one step; the prior on u^0 says where it starts; and the readings, through the physics, then track it. The model does not know the input's trajectory. It knows that the trajectory is continuous at a stated rate, and that is often all that is known.

16.5 Filtering, smoothing, and prediction as eliminations

On the unrolled graph, the three questions of estimation in time are three elimination orders.

Filtering asks for the state at time n given readings up to time n. Eliminate the past forward: fold step 0 into step 1, then the result into step 2, and so on, conditioning on each step's readings as it is reached. By Proposition 16.2 what is passed from step to step is a message on the storage state alone, and for a Gaussian model the message is a mean and a covariance.

Smoothing asks for the state at time n given readings up to a later time N. Eliminate forward to N as in filtering, then back-substitute: the same two sweeps as Chapter 8's tree messages, on a tree that is a chain. Equivalently, solve the whole unrolled precision at once; its banded factorization is the forward sweep and its solve is the backward one.

Prediction asks for the state at a time beyond the last reading. It is filtering with no likelihood factors after the last one: the forward messages continue, growing wider by the process noise at each step, and no reading narrows them.

Proposition 16.3 (The Kalman filter is a Schur complement and a rank-one update)**.** *Let the state at time n have posterior $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ given readings to time n, let the storage rows be linear, $\mathbf{x}^{n+1} = \mathbf{F}\mathbf{x}^n + \mathbf{c} + \mathbf{w}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, and let a reading $y = \mathbf{h}^\mathsf{T}\mathbf{x}^{n+1} + \nu$ arrive. Then the prediction is*

$$\boldsymbol{\mu}_{n+1|n} = \mathbf{F}\boldsymbol{\mu}_n + \mathbf{c}, \qquad \boldsymbol{\Sigma}_{n+1|n} = \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^\mathsf{T} + \mathbf{Q}, \tag{16.6}$$

and the update is the rank-one conditioning of Proposition 11.1 applied to $(\boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n})$.

Proof. The joint of $\mathbf{x}^n$ and $\mathbf{x}^{n+1}$ before the reading is the product of the step-n posterior and

the storage factor $\exp(-\frac{1}{2}(\mathbf{x}^{n+1} - \mathbf{F}\mathbf{x}^n - \mathbf{c})^\mathsf{T}\mathbf{Q}^{-1}(\cdot))$. Its precision, in the blocks $(\mathbf{x}^n, \mathbf{x}^{n+1})$, is

$$\begin{pmatrix} \boldsymbol{\Sigma}_n^{-1} + \mathbf{F}^\mathsf{T}\mathbf{Q}^{-1}\mathbf{F} & -\mathbf{F}^\mathsf{T}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{F} & \mathbf{Q}^{-1} \end{pmatrix}. \tag{16.7}$$

Eliminating $\mathbf{x}^n$ is the Schur complement of the first block, equation (6.8): the precision on $\mathbf{x}^{n+1}$ becomes $\mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{F}(\boldsymbol{\Sigma}_n^{-1} + \mathbf{F}^\mathsf{T}\mathbf{Q}^{-1}\mathbf{F})^{-1}\mathbf{F}^\mathsf{T}\mathbf{Q}^{-1}$, which by the Woodbury identity is $(\mathbf{Q} + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^\mathsf{T})^{-1}$; inverting gives $\boldsymbol{\Sigma}_{n+1|n}$, and the information vector gives the mean. The reading is then one linear row on $\mathbf{x}^{n+1}$, and Proposition 11.1 conditions on it. □

The filter is thus nothing the book had not already built: a region message from the past onto the state, then a reading folded in. The smoother's backward pass is Chapter 8's equation (8.6), region by region along the chain. The literature has names for each, Kalman for the forward pass and Rauch, Tung and Striebel for the backward, and they are the sum-product messages of Chapter 8 on a chain of storage states.

Principle 16.1. *Time unrolls the graph into a chain whose separators are the stored states. Filtering is forward elimination, smoothing is forward then backward, prediction is forward without readings, and the Kalman filter is one Schur complement and one rank-one update per step.*

16.6 Worked example: tracking drifting losses

Take the transformer with its masses, losses that ramp from 100 to 130 kilowatts between one hour and three hours and then hold, and a thermometer in the top oil reading every five minutes with accuracy 0.3 kelvin. The model does not know the ramp. It carries the losses as a random walk with a width of one kilowatt per step and starts them at 100 ± 20. The state is (T_1, T_2, G); the transition is the BDF1 step with the losses held at their new value across the step; the horizon is twelve hours, 144 steps.

Filter against smoother. Figure 16.2 draws the losses' true trajectory against the filter's estimate and the smoother's. Table 16.2 gives the errors. The filter tracks the losses to 3.0 kilowatts root-mean-square and the winding hot spot to 0.69 kelvin, from a thermometer in the oil that never reads either. The smoother cuts both errors by a factor of four, to 0.72 kilowatts and 0.17 kelvin, and halves the reported spread of the losses mid-window, from 2.69 to 1.38 kilowatts: it has the future readings, which the filter did not, and the future of slowly moving losses is evidence about their present. The filter lags the ramp, as a random-walk prior must when the truth moves in one direction; its standardized innovations, which should average zero, average $+0.95$ during the ramp and $+0.09$ after it. That drift in the innovations is the test of Chapter 12 running in time, and it is how a filter says that its model of the input's motion is too slow.

The batch check. The script also solves the whole unrolled precision, 435 variables, banded, in one factorization, and compares the result with the smoother's backward pass. They agree to 10^{-9}. The smoother and the batch posterior are the same object, computed by the same elimination in a different bookkeeping, which is what Proposition 16.3 and Chapter 8 said.

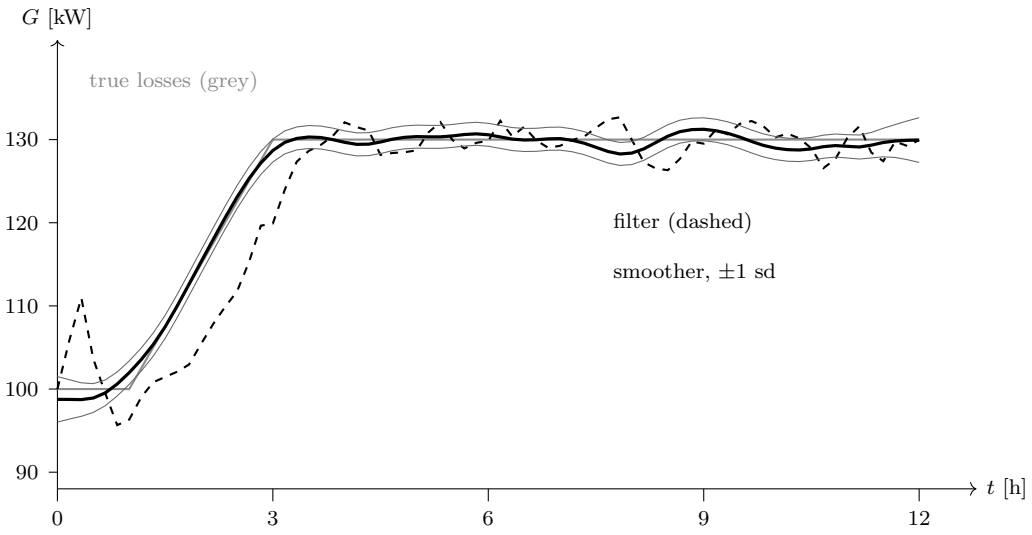


Figure 16.2: The losses: true trajectory (grey), the filter's estimate (dashed), and the smoother's estimate with its one-standard-deviation band (black). The filter lags the ramp and catches up when it ends; the smoother, seeing the later readings, follows the ramp.

Table 16.2: Tracking the drifting losses: errors of the filter and the smoother over the settled window, and the reported spread of the losses at mid-window. The batch solve of the unrolled precision agrees with the smoother to 10^{-9}.

	filter	smoother
rms error of G [kW]	3.03	0.72
rms error of T_1 [K]	0.69	0.17
reported sd of G at six hours [kW]	2.69	1.38

16.7 Detecting a change in time

The innovation of Chapter 11 was a single number per reading. In time it is a sequence, and the sequence is the most useful diagnostic a running model has. Under an adequate model the standardized innovations are independent standard Gaussians: mean zero, spread one, no memory. Anything the model does not know shows up as a departure from that, and the departure has a shape that says what kind of thing it was. A ramp the model is too slow to follow gives a sustained bias, as §16.6 showed; a bad reading gives one outlier; a sudden change in the system gives a jump that persists until the model has caught up.

Run the transformer's filter again with the losses held at 100 kilowatts and their random-walk width lowered to 0.3 kilowatts per step, and at six hours let the losses drop abruptly to 75: a load transferred away. Figure 16.3 draws the standardized innovations. Before the fault they have mean 0.01, spread 0.77, and never exceed 1.9 in magnitude. The alarm fires fifteen minutes after the drop, on the reading at 6.25 hours, at 4.5 spreads below prediction. The filter, whose prior on the losses' motion is slow, takes an hour and a quarter to bring its estimate within three kilowatts of the new level, and its estimate is 88.3 kilowatts at seven hours and 78.6 at eight; the innovations return to white as it arrives. The lag is the price of the narrow random walk, and it

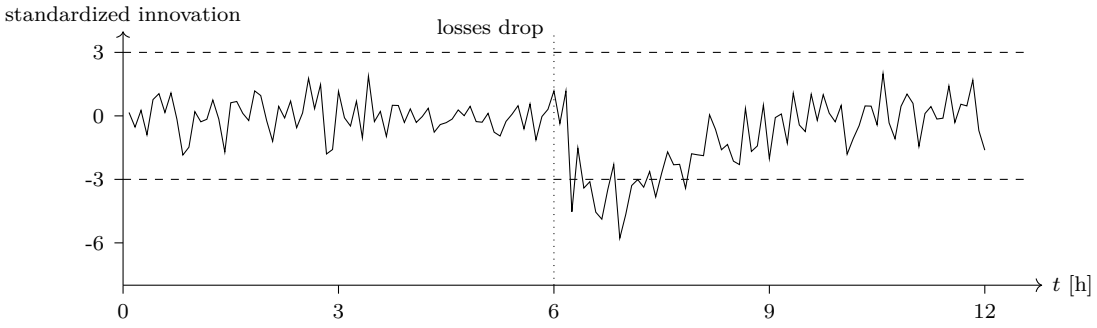


Figure 16.3: Standardized innovations of the filter on the transformer when the losses drop from 100 to 75 kilowatts at six hours and the model carries the losses as a slow random walk. Before the fault the sequence is white within ±3; a quarter of an hour after it the readings fall several spreads below their predictions, and the sequence stays there until the filter has moved the losses.

is the trade the operator makes: a wide walk tracks changes quickly and calls noise a change, a narrow one filters noise and calls a change an anomaly for as long as it takes to believe it. The alarm is right either way; what the width sets is whether the estimate follows the alarm at once or after a decent interval.

Two things distinguish this from a threshold on the thermometer. The threshold would fire when the oil had cooled by some fixed amount, hours after the drop, since the oil's time constant is nearly four hours; the innovation fires within a quarter of an hour, because the filter predicted where the oil should be and the oil was not there. And the threshold says the oil is cool; the innovation, through the model, says the losses have fallen, which is the thing that changed.

Principle 16.2. *A running model's innovations are its diagnostic in time: white when the model is adequate, biased when it is too slow, spiked when something changed. The alarm fires when the prediction fails, not when the state has drifted far enough to notice.*

16.8 Two hypotheses the steady state cannot separate

The alarm says something changed. Chapter 12 said what to do next: name the candidates, give each a latent variable with a zero-centered prior, and let the evidence choose. Two candidates explain a sudden fall in the oil's readings. The losses may have dropped, by an amount a; or the thermometer's bias may have jumped, by an amount b. Each is the running model with one more state, born at the step of the alarm with a prior of thirty units, kilowatts or kelvins, on its amplitude: under the first, the jump enters the storage row of the losses exactly as a process-noise step does; under the second, it enters the likelihood row of the thermometer as an offset.

In steady state the two are the same hypothesis. A loss drop of 25 kilowatts lowers the oil's steady reading by 10.0 kelvins, and a thermometer bias of -10.0 kelvins lowers it by the same, and no number of steady readings tells them apart. Chapter 12 met this in Table 12.4, where an unmodeled draw and a sensor bias were separated only by their priors. In time they are different hypotheses. A bias moves the reading at once and holds it; a loss drop moves the oil over its time constant of 13,300 seconds, nearly four hours. The shapes differ, and the evidence weighs shapes.

Table 16.3: Log Bayes factor for the true hypothesis against the other, accumulated over the k readings after the alarm, for a loss drop of 25 kilowatts and for a thermometer bias of the same steady effect, -10.0 kelvins. Amplitude priors of thirty units on both.

k readings after the alarm	1	2	3	5	10	20	30	60
truth: loss drop, log (losses : bias)	-3.6	-2.2	-2.2	0.1	18.7	60.6	78.2	83.5
truth: bias jump, log (bias : losses)	343	382	416	606	999	1520	1734	1802

The evidence of a model in time factors along the chain. Write the joint density of the readings as a product of conditionals, $p(y_1, \dots, y_N) = \prod_n p(y_n \mid y_1, \dots, y_{n-1})$; each factor is the reading's predictive density under the filter, which for the linear-Gaussian model is $\mathcal{N}(e_n;\, 0,\, s_n)$ with e_n the innovation and s_n its variance. Hence

$$\log Z \;=\; \sum_{n=1}^{N} \log \mathcal{N}(e_n;\, 0,\, s_n) \;=\; -\tfrac{1}{2} \sum_{n} \Big(\log 2\pi s_n + \frac{e_n^2}{s_n} \Big), \tag{16.8}$$

which is equation (12.1) computed one reading at a time. The Occam factor is there too: a hypothesis whose extra state is wide inflates s_n at the reading where the state is born, and pays $\frac{1}{2} \log s_n$ for the room it gave itself. The log Bayes factor between two hypotheses is the difference of two such sums, and it can be watched growing reading by reading.

Table 16.3 gives the result for both truths. When the losses have dropped, the first reading points the wrong way: a fall of one reading is a fall, and the bias explains it more cheaply, being able to produce all of it at once where the losses can produce only the first sliver. The loss hypothesis is behind by three and a half nats. Over the next readings the oil keeps falling, as a mass cooling through its time constant does and as a stuck bias does not, and the evidence turns and accumulates at about two and a half nats a reading: level at five readings, nineteen after ten, sixty after twenty. Then it slows. By the sixtieth reading the oil is most of the way to its new steady state, both hypotheses predict nearly the same reading, and the readings from the thirty-first to the sixtieth add under two tenths of a nat each. Ninety percent of the evidence arrives in the first twenty-eight readings, two hours and twenty minutes. The loss hypothesis ends with a jump estimate of -23.6 ± 1.6 kilowatts and losses of 74.8; the bias hypothesis, to fit the falling readings, has had to move its random-walk losses as well, and ends with a bias of -1.25 kelvins and losses of 77.1, a compromise that fits neither the steady state nor the transient.

When the truth is a bias, the asymmetry shows. A bias of ten kelvins moves the reading by thirty spreads at once, and a loss drop cannot do that: through the winding's and the oil's storage, a step of the losses moves the oil by almost nothing in the first five minutes. The first reading alone is three hundred and forty nats against the losses, and the transient it fails to produce adds the rest.

Two lessons. The distinction between a fault in the physics and a fault in the sensing, which Chapter 12 had to buy with a prior, is bought here with time: the storage rows give each hypothesis a shape, and the shapes are not the same. And the window in which they differ is the system's own time constant. An operator who waits for the new steady state before asking which hypothesis holds has waited past the only readings that could answer.

Principle 16.3. *Hypotheses that the steady state cannot separate are separated by the transient, because storage gives each a shape in time. The evidence factors along the chain, one innovation density per reading, and it accumulates only while the shapes differ: for as long as the time constant, and no longer.*

16.9 Events and cascades

A transient rarely runs to its end undisturbed. A conductor crosses a temperature limit and a breaker opens; a generator reaches its reactive limit and a bus loses its voltage support; a tap changer runs out of travel and a regime changes. Each is an *event*: a scalar indicator $\psi(\mathbf{z}, t)$ crosses zero, and at the crossing the model changes, by an intervention in the sense of Chapter 13, a closure replaced or a component removed. Locating the crossing is part of the integration: after each accepted step the indicator is checked for a sign change, the crossing is found by bisection on the interpolated state, and the handler applies the change and the integration continues on the new model.

A *cascade* is a sequence of events each caused by the last. A line trips; the flows redistribute; another line, now overloaded, heats; it trips in its turn. Chapter 15 treated failure sets as instantaneous and their consequences as steady states, and could not say which set a cascade would produce or how long it would take. With dynamics it can. The failure set grows one event at a time along the trajectory, and the operator's question changes from "what could fail" to "how long do I have".

16.10 Worked example: the conductors of the six-bus network

The conductor of an overhead line heats by its current and cools to the air, and its temperature, not its current, is what limits it. Let each line's conductor obey

$$C\,\frac{\mathrm{d}T_\ell}{\mathrm{d}t} \;=\; R\,F_\ell^2 - h\,(T_\ell - T_{\mathrm{amb}}), \tag{16.9}$$

with F_ℓ the line's flow, $T_{\mathrm{amb}} = 30$ degrees, a limit $T_{\max} = 90$ degrees, and a time constant $\tau = C/h$ of 600 seconds. Choose R so that a line carrying exactly its rating settles at the limit; then the steady conductor temperature at any flow is

$$T_\infty(F_\ell) \;=\; T_{\mathrm{amb}} + (T_{\max} - T_{\mathrm{amb}})\Big(\frac{F_\ell}{F_\ell^{\mathrm{rating}}}\Big)^2, \tag{16.10}$$

and the rating of Chapter 15 is the current at which the steady temperature is the limit. A line above its rating does not fail; it heats toward a steady temperature above the limit, and fails when it gets there.

Proposition 16.4 (Time to the limit)**.** *If a line's flow steps to a constant F_ℓ at $t = 0$ from a conductor temperature $T_0 < T_{\max}$, and $T_\infty(F_\ell) > T_{\max}$, the conductor reaches the limit at*

$$t_{trip} \;=\; \tau\,\ln\frac{T_\infty - T_0}{T_\infty - T_{\max}}. \tag{16.11}$$

Proof. Equation (16.9) with constant F_ℓ is linear with solution $T_\ell(t) = T_\infty - (T_\infty - T_0)\,e^{-t/\tau}$; set $T_\ell = T_{\max}$ and solve for t. The argument of the logarithm exceeds one exactly when $T_\infty > T_{\max} > T_0$. □

The cascade. At $t = 0$ line 1–3 of the six-bus network trips. Chapter 15 found the immediate consequence: line 1–2 carries 4.1 against a rating of 2.5, and line 2–3 carries 2.9 against 1.0. Both conductors begin to heat. Their steady temperatures under the new flows are 191 and 535

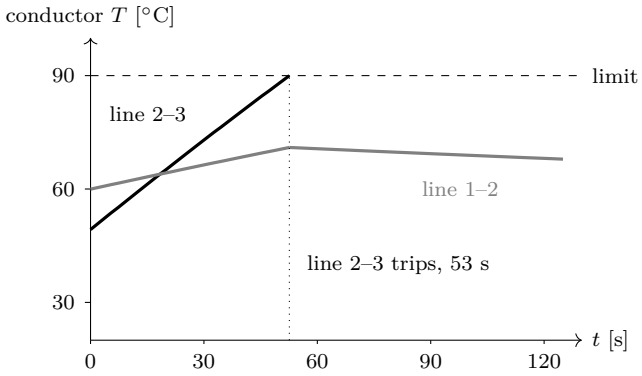


Figure 16.4: Conductor temperatures of lines 2–3 (black) and 1–2 (grey) after the outage of line 1–3 at $t = 0$. Line 2–3 reaches its limit at 52.6 seconds and trips, islanding four buses; line 1–2, relieved, cools toward a new steady value. The operator's window is the 52.6 seconds.

degrees; line 2–3, three times over its rating, is heading for the limit far faster than line 1–2, and by equation (16.11) it reaches 90 degrees at $t = 52.6$ seconds, while line 1–2 would have needed 156. At 52.6 seconds line 2–3 trips. The script re-solves the flows: with lines 1–3 and 2–3 both gone, bus 3 is reached from bus 1 by no path at all, and with it buses 4, 5 and 6; four buses are islanded and 2.9 per unit of load is shed, a third of the network. Line 1–2, now serving bus 2 alone, drops to 1.2 and cools. No further conductor heads for its limit. The cascade is two events long and it has ended worse than either alone: the $N - 1$ consequence of the first outage was an overload of 3.5, and the cascade it started shed 2.9 of load and left the overload gone.

Figure 16.4 draws the two conductor temperatures. The operator's window is the 52.6 seconds between the first trip and the second: curtail bus 2 or bus 3 within it, by the branch machinery of Chapter 13, and line 2–3 never reaches its limit. A steady-state analysis says line 2–3 is overloaded; the dynamic one says by how long.

How long, with error bars. The cooling coefficient h depends on wind and on the conductor's condition, and it is not known to better than thirty percent. Draw it from a log-normal with that spread, rerun the cascade for each draw, and the time to the first trip has a distribution: a median of 52 seconds, a fifth percentile of 32, a ninety-fifth of 86, and an 18 percent chance that the window is shorter than 40 seconds. That is the honest form of "how long do I have": a time with a tail, and the tail is what the operator plans against.

> **Principle 16.4.** *A cascade is a sequence of events located on a trajectory, each an intervention caused by the last. The steady state says what is overloaded; the dynamics say how long until it fails, and the uncertainty in the dynamics says how much of that time can be counted on.*

The window as a decision. The window is worth something only if an action inside it changes the outcome, and the branch machinery of Chapter 13 says which action does. Suppose the operator acts at 30 seconds, when line 2–3's conductor is at 73 degrees and still 17 short of its limit. The obvious action, curtailing bus 2, whose line 1–2 is also overloaded, does nothing for line 2–3: with line 1–3 gone, line 2–3 carries the load of bus 3 and of region B beyond it, 2.9

Table 16.4: Curtailing the load downstream of line 2–3 at 30 seconds after the outage of line 1–3: the flow on line 2–3, its conductor's steady temperature, and the resulting trip time. Curtailing bus 2, which is upstream, changes nothing.

curtailment [p.u.]	F_{23}	steady T_{23} [°C]	trip
0.8 at bus 2	2.90	535	at 53 s, unchanged
0.5 downstream	2.40	376	at 65 s (+12)
1.0 downstream	1.90	247	at 92 s (+39)
1.5 downstream	1.40	148	at 186 s (+133)
1.9 downstream	1.00	90	none; holds at the limit
2.0 downstream	0.90	79	none; cools to 79

per unit in all, and bus 2's load flows in on line 1–2 and stops there. Curtailing 0.8 at bus 2 leaves line 2–3 at 2.9. The graph says where the flow comes from, and the action must be taken downstream of the line it is meant to relieve.

Curtail the downstream load instead, bus 3 and region B pro rata, and Table 16.4 gives the result. Half a unit buys twelve seconds; a full unit buys thirty-nine; one and a half buys more than two minutes; and 1.9, which brings line 2–3 to exactly its rating, stops the conductor at the limit and holds it there. Below 1.9 the trip is delayed, not prevented, and every delay is time in which a further action, a redispatch or a repair, can be taken. The cascade that would have shed all 2.9 of the downstream load is prevented by curtailing 1.9 of it, and the operator has 52 seconds, or with the cooling coefficient uncertain some 32, to do so.

16.11 In practice

Integrate implicitly, step for the slow dynamics. The systems of this book couple light components to heavy ones and are stiff by construction. An implicit method with a step chosen for the dynamics of interest is the default; an explicit method is right only when the fastest time constant is the one of interest, which is rare. Let the step control shrink the step at events and transients and grow it when the state settles.

Carry the storage states; recover the rest. The filter's state is the set of storage states and the random-walk inputs, and nothing else, by Proposition 16.2. Temperatures at zero-storage nodes and every flow are algebraic functions of the state and are recovered by a solve when asked for. Carrying them in the state is a common mistake and it makes the covariance larger and singular.

Set the process noise from the physics, then check it. The width of a storage row is the model's error over one step; the width of a random walk is how far the input may move in one step. Both are physical statements and should be set as such, then checked by the innovations: a filter whose standardized innovations drift, as in §16.6 during the ramp, has too little process noise on the input that is moving; one whose innovations are too small has too much and is tracking noise.

Smooth when you can, filter when you must. Operation in real time filters; every analysis after the fact should smooth, since the smoother has the same cost and less error. A common mistake is to report filtered estimates of a past state when the later readings were available.

Locate events; do not step over them. An integrator that steps past a limit crossing and applies the change at the end of the step has mis-timed the event by up to a step, and in a cascade the timing is the answer. Locate the crossing by bisection, apply the change there, and restart the step.

Report a window, not a time. The time to a trip depends on quantities, wind and ambient among them, that are not known precisely. Report the window as a distribution, and plan against its short tail.

16.12 What is missing

Every model in this chapter was one system. The unrolled graph is a chain of copies of it, and the chapter organized computation along the chain. Chapter 17 organizes it across the other direction, the regions of Chapter 8 arranged in a tree of subsystems, and asks what it means for a region to be summarized to its parent when both have dynamics; and Chapter 18 asks how to cut a large system into such regions, with the measurements that Chapter 5 and Chapter 9 promised.

Notes and further reading

Differential-algebraic systems, their index, and the backward differentiation formulas for stiff problems are treated by Brenan, Campbell and Petzold [12] and by Hairer and Wanner [32], who also give the stability analysis behind Table 16.1; the step-change controller mentioned in §16.2 follows the PI design of Gustafsson, Lundh and Söderlind [29]. That a BDF step is the steady system with its storage rows rewritten, so that simulation is a loop over one solver, is the book's framing of a standard fact.

The Kalman filter is Kalman's [39] and the fixed-interval smoother is Rauch, Tung and Striebel's [75]; Särkkä and Svensson [81] present both as Gaussian inference on a chain, which is the view of §16.5, and treat process noise, random-walk models and innovation diagnostics. The derivation of the predict step as a Schur complement and of the update as the rank-one conditioning of Chapter 11 is the book's, and it is the reason the filter needed no new machinery.

The conductor model of §16.10 is a linearized lumped form of the IEEE standard for the current-temperature relationship of overhead conductors [35]. In the standard, convective cooling, the larger of a wind-driven and a natural term, grows as a power of the temperature rise, radiative cooling as the fourth power of absolute temperature, and solar heating enters as a source; equation (16.9) keeps the storage and replaces the losses by one linear term. Dynamic line rating is the practice of rating a line from the weather its conductors see rather than by a fixed seasonal current. A rating computed that way inherits the uncertainty of the weather along the line, and because the line is limited by its hottest span, the rating is a quantile of a minimum over correlated spans; a wind sensor on the line then has a value in the sense of Chapter 11, computable before the sensor is installed. Cascading failure in power networks has a large literature; the event-driven view of §16.9, in which each trip is an intervention on the

model that the next integration starts from, is how the reference implementation of Appendix C handles it.

Summary

- With storage, the model is an index-one differential-algebraic system; the graph gains one variable and one square per storing node.
- A BDF step is the steady system with its storage rows rewritten: same variables, same sparsity, same solver. Implicit methods are stable at any step; the explicit method fails above $2/|\lambda_{\max}|$, 565 seconds on the transformer, and is wrong by 5×10^9 degrees at twenty minutes.
- The unrolled graph is a chain; the storage states separate past from future; the precision is block-tridiagonal and factors in linear time.
- Process noise is the storage row's width; a random walk on an input tracks its drift.
- Filtering, smoothing and prediction are elimination orders on the chain. The Kalman predict step is a Schur complement and the update is a rank-one conditioning. On the transformer the smoother cuts the filter's errors by four and the batch solve agrees with it to 10^{-9}.
- The innovations in time are the model's diagnostic: white when it is adequate, biased when it lags, spiked when the system changed. A drop of a quarter in the losses is four and a half spreads on the reading a quarter of an hour later.
- A loss drop and a sensor bias with the same steady effect are one hypothesis in steady state and two in time; the evidence, one innovation density per reading, separates them by 78 nats within the oil's time constant and by five more over all the hours after it.
- A cascade is events on a trajectory. After the loss of line 1–3, line 2–3's conductor reaches its limit in 52.6 seconds and its trip islands four buses; with the cooling coefficient uncertain by thirty percent the window is 52 seconds with a fifth percentile of 32.
- The window is a decision when an action inside it changes the outcome. Curtailing upstream of the overloaded line does nothing; curtailing 1.9 of the 2.9 per unit downstream holds the conductor at its limit and prevents the cascade that would shed all 2.9.

Exercises

Exercise 16.1. Derive the BDF2 formula (16.5) by fitting a quadratic through X^{n-1}, X^n and X^{n+1} at equal spacing and requiring its derivative at t_{n+1} to equal $\mathcal{B}(\mathbf{z}^{n+1})$. Show that on $\dot{X} = \lambda X$ the method's amplification factor has magnitude below one for every $\lambda < 0$ and every Δt.

Exercise 16.2. Complete the proof of Proposition 16.3: show that the Schur complement of the joint precision onto $\mathbf{x}^{n+1}$ equals $(\mathbf{Q} + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^{\mathsf{T}})^{-1}$ by the Woodbury identity, and derive the predicted mean $\mathbf{F}\boldsymbol{\mu}_n + \mathbf{c}$ from the information vector.

Exercise 16.3. Rerun the tracking example with the random-walk width raised to 3 kilowatts per step and lowered to 0.3. Report the filter's error and the mean standardized innovation during the ramp for each, and explain the trade-off between lag and noise that the width controls.

Exercise 16.4. In the cascade, suppose the operator, instead of curtailing, opens the tie line 3–4 deliberately at $t = 30$ seconds, islanding region B on purpose. Recompute the flow on line

2–3, its conductor's trajectory from its value at 30 seconds, and whether it still trips. Compare the load shed by this action with the 1.9 per unit of Table 16.4 and with the 2.9 the cascade sheds, and say which the operator should prefer.

Exercise 16.5. Give the conductor of line 2–3 a thermometer reading every 10 seconds with accuracy 2 degrees, and build the filter for its temperature with h as a random-walk parameter. Show that after three readings the filter's predicted time to the limit has a spread narrower than the 32-to-86-second window of §16.10, and say why a direct temperature reading is worth more here than a flow meter.

Solutions to selected exercises

Exercise 16.1. Put $t_{n+1} = 0$ and let the quadratic through $(-2\Delta t, X^{n-1})$, $(-\Delta t, X^{n})$, $(0, X^{n+1})$ be $p(s) = X^{n+1} + as + bs^2$. Then $X^n = X^{n+1} - a\Delta t + b\Delta t^2$ and $X^{n-1} = X^{n+1} - 2a\Delta t + 4b\Delta t^2$. Subtracting twice the first from the second gives $X^{n-1} - 2X^n = -X^{n+1} + 2b\Delta t^2$, so $b = (X^{n+1} - 2X^n + X^{n-1})/(2\Delta t^2)$, and then $a = (X^{n+1} - X^n)/\Delta t + b\Delta t = (3X^{n+1} - 4X^n + X^{n-1})/(2\Delta t)$. The derivative at $s = 0$ is a, and setting $a = \mathcal{B}(\mathbf{z}^{n+1})$ gives $3X^{n+1} - 4X^n + X^{n-1} = 2\Delta t\,\mathcal{B}(\mathbf{z}^{n+1})$, which is equation (16.5). On $\dot X = \lambda X$ with $z = \lambda\Delta t$ the recurrence is $(3-2z)X^{n+1} = 4X^n - X^{n-1}$, whose characteristic roots satisfy $(3-2z)r^2 - 4r + 1 = 0$; for $z < 0$ the product of the roots is $1/(3-2z) < 1/3$ and their sum is $4/(3-2z) < 4/3$, and one checks that both roots lie inside the unit disk for every $z < 0$, with the larger root $\to 1$ as $z \to 0^-$ and both roots $\to 0$ as $z \to -\infty$: the method damps every stable mode at every step, and damps the stiffest modes most.

Exercise 16.2. Write $\mathbf{S} = \mathbf{\Sigma}_n^{-1} + \mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}\mathbf{F}$. The Schur complement is $\mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{F}\mathbf{S}^{-1}\mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}$. The Woodbury identity states $(\mathbf{Q} + \mathbf{F}\mathbf{\Sigma}_n\mathbf{F}^{\mathsf T})^{-1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{F}(\mathbf{\Sigma}_n^{-1} + \mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}\mathbf{F})^{-1}\mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}$, which is the same expression, so the predicted precision is $(\mathbf{Q} + \mathbf{F}\mathbf{\Sigma}_n\mathbf{F}^{\mathsf T})^{-1}$ and the predicted covariance is $\mathbf{F}\mathbf{\Sigma}_n\mathbf{F}^{\mathsf T} + \mathbf{Q}$. For the mean, the joint information vector has blocks $\mathbf{\Sigma}_n^{-1}\boldsymbol{\mu}_n - \mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}\mathbf{c}$ on $\mathbf{x}^n$ and $\mathbf{Q}^{-1}\mathbf{c}$ on $\mathbf{x}^{n+1}$; the Schur-complement information vector is $\mathbf{Q}^{-1}\mathbf{c} + \mathbf{Q}^{-1}\mathbf{F}\mathbf{S}^{-1}(\mathbf{\Sigma}_n^{-1}\boldsymbol{\mu}_n - \mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}\mathbf{c})$. Multiplying by the predicted covariance and simplifying with $\mathbf{S}^{-1}\mathbf{\Sigma}_n^{-1} = \mathbf{I} - \mathbf{S}^{-1}\mathbf{F}^{\mathsf T}\mathbf{Q}^{-1}\mathbf{F}$ gives $\mathbf{F}\boldsymbol{\mu}_n + \mathbf{c}$; the reader who does not want the algebra can verify it on the scalar case, where every matrix is a number and the identity is one line.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

The slow ledger *a time constant read off a capacity and a conductance; subsystems whose horizons differ by orders of magnitude; algebraic against differential parts of one model*
`power_grid/battery_electrothermal_cosim/problems/the_slow_ledger_sets_the_step`

A line has a memory *a thermal time constant between a limit and a violation; why an instantaneous overload is not an emergency; temperature as the real constraint and current as its proxy*
`transmission/dlr_n11_value/problems/a_line_has_a_memory`

Chapter 17

Hierarchy as physical decomposition

Every engineered system of any size is described at several levels at once. A collector substation is one injection to the transmission network, a set of feeders to the station's own operator, a set of generating units to each feeder, and a set of converters and windings to each unit; a distribution feeder is one load to the transmission network and a tree of loads to itself; a chemical plant is units to the site and vessels to each unit. The levels are not a convenience of drawing. Each is a boundary-exchanging system in the sense of Chapter 1, with its own balance rows and closures and its own ports, and the levels nest because the cuts nest: the boundary of a unit lies inside the boundary of its feeder, which lies inside the boundary of the station.

Chapter 8 showed what one cut buys. A region's factors, eliminated onto its ports, are the region as its neighbors see it, exactly. This chapter applies that statement to nested cuts, and finds that it needs nothing new. The same conservation law that makes the flows across a cut exact makes a region's balance the sum of its parts' balances, so that a parent's model of a child is the child's message, and a grandparent's model of the child is the child's message passed through the parent's. Schur complements compose, so the hierarchy of regions is a junction tree whose messages are computed level by level and reused across levels. And the cut that organizes the model organizes control and scale as well: a setpoint is imposed at a port, a child meets it inside, and the cost of a local change is the cost of one path from leaf to root.

17.1 One conservation law used twice

Take a region S of a physical graph: a set of nodes, the edges among them, and the sources at them. Chapter 1's Proposition 1.1 said that in steady state the net flow across the boundary of S equals the net source inside S. Write it as an equation among the residual rows. The balance at node n is

$$r_n \;=\; \sum_{e \ni n} \pm F_e \;-\; q_n \;=\; 0, \tag{17.1}$$

with the sign by the edge's orientation at n. Sum the rows over $n \in S$. Every edge with both ends in S appears twice with opposite signs and cancels; every edge with one end in S appears once. Hence

$$\sum_{n \in S} r_n \;=\; \sum_{e \in \partial S} \pm F_e \;-\; \sum_{n \in S} q_n, \tag{17.2}$$

which is the balance row of S regarded as a single node with the cut edges as its edges and the total source as its source. This is the whole content of the proposition, and it has a consequence

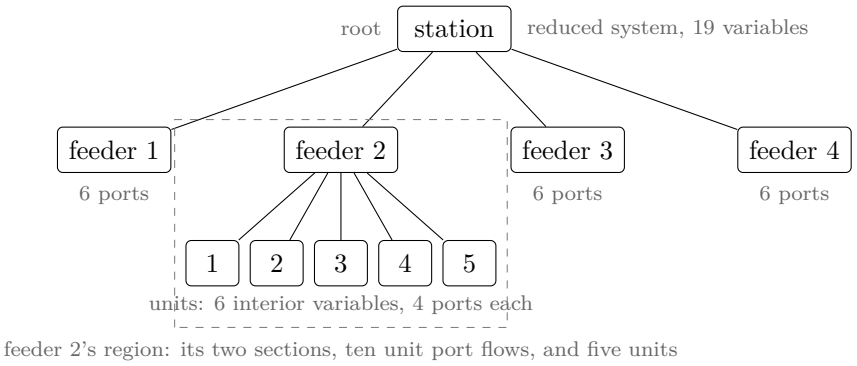


Figure 17.1: The collector substation's hierarchy of regions. Each unit eliminates six interior variables onto four ports; each feeder eliminates its own twelve (two section voltages and the ten unit port flows) onto six; the station solves the reduced system of nineteen. Every edge of the tree is a cut of the physical graph, and the ports of a cut are the message that crosses it.

for modeling that is easy to pass over.

Proposition 17.1 (Aggregation)**.** *Let a region S be replaced, in the parent's model, by one node whose edges are the cut edges of S and whose source is $\sum_{n\in S} q_n$. The parent's balance row at that node is the sum of the region's balance rows. It is exact, and it reads only the cut flows and the total source.*

The proposition is the license for every meter that reads a total. A feeder's meter reads the feeder's total injection; the transmission operator's meter at a substation reads the feeder's total load; a building's utility meter reads the sum of everything inside. None of these instruments knows the inside, and none needs to: equation (17.2) says that the total source is a function of the cut flows alone, so a factor on the total is a factor on the ports. A meter on a feeder's export is, to the station, a factor on the feeder's port flows; the station can use it without a model of the feeder. Where the meter is placed in the hierarchy is therefore a choice, and Section 17.4 shows the two placements agree to the physics width.

What the proposition does not license is as important, and Chapter 1 said it: the intensive states do not aggregate. The region-as-node has one balance row and no voltage. If the parent's model gives it a voltage, that voltage is a modeling assumption, not a consequence of conservation, and Section 17.3 shows what the assumption costs. The exact object that replaces the region is not a node with a potential but the region's message, which carries as many potentials as the region has ports.

17.2 Nested messages

Proposition 8.2 eliminated one region's interior onto its ports. In a hierarchy the interior of a feeder contains the interiors of its units, and there are two ways to eliminate it: all at once, or unit by unit and then the rest. The two agree, because the Schur complement has a composition property that is the algebraic form of nested cuts.

Proposition 17.2 (Schur complements compose)**.** *Let $\mathbf{\Lambda}$ be a precision on variables partitioned as $I_1 \cup I_2 \cup S$, with I_1 and I_2 disjoint. Eliminating I_1 first and then I_2 from the result gives the*

same precision on S as eliminating $I_1 \cup I_2$ at once:

$$(\mathbf{\Lambda}/\mathbf{\Lambda}_{I_1 I_1})/(\mathbf{\Lambda}/\mathbf{\Lambda}_{I_1 I_1})_{I_2 I_2} \;=\; \mathbf{\Lambda}/\mathbf{\Lambda}_{II}, \qquad I = I_1 \cup I_2, \tag{17.3}$$

where $\mathbf{\Lambda}/\mathbf{\Lambda}_{AA}$ denotes the Schur complement of the block A. The same holds for the information vector.

Proof. Both sides are the precision of the marginal on S of the Gaussian with precision $\mathbf{\Lambda}$. The left integrates out I_1 and then I_2; the right integrates out both together. Marginalization is one integral whatever the order in which its variables are grouped, so the two marginals are the same Gaussian and have the same precision and information. In matrix terms the identity is the quotient formula for Schur complements, and it can be verified by block elimination: eliminating I_1 leaves a bordered system on $I_2 \cup S$ whose I_2 block is $\mathbf{\Lambda}_{I_2 I_2} - \mathbf{\Lambda}_{I_2 I_1}\mathbf{\Lambda}_{I_1 I_1}^{-1}\mathbf{\Lambda}_{I_1 I_2}$, and eliminating that block reproduces the 2×2 block inverse of $\mathbf{\Lambda}_{II}$ term by term. □

The proposition is why a hierarchy needs no new machinery. Take the station. Each unit's factors read its interior and its four ports; its message is a 4×4 precision and a 4-vector on those ports, computed by equation (8.5). The feeder's factors read the feeder's two sections, the unit port flows, and the feeder's own ports. Add the five unit messages to the feeder's precision, and the feeder's interior at this level is twelve variables, its sections and the ten unit port flows, with the unit interiors already gone. Eliminate the twelve onto the feeder's six ports. By Proposition 17.2 the result is the message the feeder would have sent had its forty-two variables been eliminated at once, and it was computed with five 6×6 solves and one 12×12 solve instead of one 42×42. Add the four feeder messages to the station's factors and solve the reduced system on the nineteen variables the station reads. Then back-substitute downward: Proposition 8.3 gives each feeder's interior from its ports, and, applied again with the feeder's interior as the unit's ports, each unit's interior from the feeder's. Figure 17.1 draws the tree.

The pattern generalizes. A hierarchy of regions is a junction tree in which each region's cluster is its own interior together with its ports, and the separators are the ports. Messages go up the tree by Schur complement and down by back-substitution, and the running intersection property of Chapter 8 holds because a variable shared by two regions is a port of the lower one and lies on the path between them. The cost model of Section 8.5 applies level by level: a region with n_r variables eliminated at its level and k_r ports costs one factorization of an $n_r \times n_r$ block and k_r solves against it, and the root costs a solve of the size of what it reads. What the hierarchy adds to the flat partition of Chapter 8 is that n_r at each level is the region's own variables, not the whole subtree's: the unit interiors are eliminated once, at the unit level, and every level above sees them only through four numbers per unit.

Principle 17.1. *Nested cuts give nested messages. A region's message to its parent is computed from its children's messages and its own factors, and it is the same message the region would send if it were eliminated whole. Each level pays for its own variables only.*

17.3 What the parent needs to know

A level of description is a decision about what to carry. The hierarchy makes the decision precise: the parent needs, of each child, exactly the child's message, a precision and an information vector on the child's ports. Nothing less is exact and nothing more is needed. Two readings of that message are useful.

The message is an equivalent network. On the feeder's six ports the message is a 6×6 precision $\mathbf{S}$ and a 6-vector $\mathbf{h}$. Order the ports as the four flows $\mathbf{f}$ and the two potentials $\boldsymbol{\phi}$, and partition. The Gaussian with precision $\mathbf{S}$ has, conditional on the potentials, a mean and covariance for the flows of

$$\mathbb{E}[\mathbf{f} \mid \boldsymbol{\phi}] \;=\; \mathbf{S}_{ff}^{-1}\mathbf{h}_f \;-\; \mathbf{S}_{ff}^{-1}\mathbf{S}_{f\phi}\,\boldsymbol{\phi}, \qquad \mathrm{Cov}(\mathbf{f} \mid \boldsymbol{\phi}) \;=\; \mathbf{S}_{ff}^{-1}, \tag{17.4}$$

which is a network: a matrix of equivalent susceptances $-\mathbf{S}_{ff}^{-1}\mathbf{S}_{f\phi}$ from the port potentials to the port flows, an equivalent injection $\mathbf{S}_{ff}^{-1}\mathbf{h}_f$, and an error bar $\mathbf{S}_{ff}^{-1}$ on the flows. This is the Kron reduction of Chapter 8 with uncertainty, and it is what a modeler who knows only the ports would have to build by hand. For feeder 1 of the collector substation the equivalent susceptance from the collector bus's voltage to the feeder's flow into it is -0.58 megavars per millivolt, where the physical edge alone has 5.0: the message knows that the feeder's tail section has a voltage sensor, so that raising the collector bus mostly raises the tail with it rather than reversing the flow. The flows' spreads given the voltages are 0.04, 0.01, 0.33 and 0.03 megavars. A transmission model that treats a collector substation as "an injection of 128 megavars at 1.006 per unit" has built a one-node equivalent; the message is the many-port equivalent, with the region's readings folded in, and Section 17.4 measures the difference.

The message is the parent's whole knowledge of the child. Whatever the parent asks about the child's interior, it asks by back-substitution, and back-substitution needs the child's factorization, which the child keeps. The parent does not hold the child's variables. This is the level-of-description discipline enforced by the algebra: the station's reduced system has nineteen variables, not one hundred eighty-seven, and the station answers questions about units by passing them down.

Lumping is the message with an assumption added. The traditional alternative to the message is to *lump*: replace the child by a node with one potential and the total source, as Proposition 17.1 allows for the balance and forbids for the potential. Lumping is exact for the flow through the cut and wrong for everything that depends on which port the flow used. In the feeder, the reactive power leaves through the tail section at 1.0125 per unit and a little returns to the grid busbar from the head section at 1.0025; a node at one voltage cannot do both, and the parent's estimate of which way the power went is wrong by the difference. The structural literature knows this as the difference between Guyan reduction, which is the Schur complement onto the retained degrees of freedom and is exact for the static response [30], and a lumped mass, which is not. The message needs no judgment about which potential to keep; it keeps them all, and there are only as many as the cut has edges.

> **Principle 17.2.** *The parent's model of a child is the child's message: exact on the ports, silent on the interior, and answered for by back-substitution when the interior is asked about. A lumped node is the message with one potential where the cut has several, and it is exact for the total flow and wrong for its distribution.*

17.4 Worked example: a collector substation in three levels

Figure 17.2 draws the station. The transmission busbar sits at 1.000 per unit and the model treats it as a reservoir. Each of four collector feeders is a ring with a head section C and a

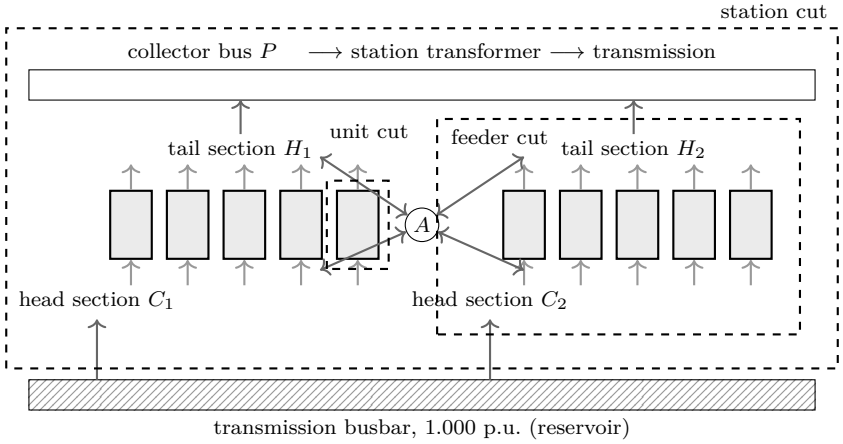


Figure 17.2: The collector substation, two of its four feeders drawn. Reactive power injected by the units (tap node F, terminal T with its injection, export node E) leaves by the tail sections and returns to transmission through the collector bus P and the station transformer, with a little going directly out of the head sections; the tie bus A exchanges with every section. The dashed rectangles are three nested cuts. A unit's ports are its tap and export flows with the section voltages on the far side; a feeder's are its four exchange flows with P and A; the station's is the exchange with the transmission busbar.

tail section H; five generating units sit on the ring between them. The tail section returns to the collector bus P, which reaches transmission through the station transformer, and both sections also exchange with a tie bus A that joins the feeders to one another. A unit has a tap node on the head section, a terminal bus with its reactive injection, and an export node on the tail side; the ring is operated open at the tap, so almost all of the injection leaves by the export path and almost none flows back to the head. Every exchange is a susceptance in the decoupled linearization $Q_{ij} = b_{ij}(V_i - V_j)$ of Chapter 1, and the model is the linear-Gaussian one of Chapter 6: one balance row per bus, one closure per branch, both with a physics width of ten kilovars; a prior of 8 ± 1.5 megavars on each unit's injection; voltage sensors on every section, on P and on A with accuracy half a millivolt, which is 5×10^{-4} per unit; terminal voltage sensors on two units per feeder with accuracy one millivolt; and a megavar meter on each feeder with accuracy 0.4. The model has 187 variables, seventy voltages, ninety-seven flows and twenty injections, and 209 factors. The truth draws the twenty injections from their prior, and the readings are the truth's values plus noise. Every number is from `ch17_hierarchy.py`.

Three solves agree. The monolithic solve inverts the 187×187 precision, whose condition number is 1.4×10^8 in these units. The hierarchical solve computes twenty unit messages, four feeder messages, one 19-variable station solve, and back-substitutes; Table 17.1 gives the sizes. The two agree to 1×10^{-9} in every mean and 6×10^{-11} in every standard deviation, which is Proposition 17.2 in floating point. A third solve writes each feeder's meter on the feeder's five injections instead of on its four port flows, the two placements Section 17.1 said are equivalent, and agrees with the first to 1.6 kilovars, the physics width.

Table 17.1: The levels of the collector substation: how many regions at each, how many variables each eliminates at its own level, and how many ports it keeps.

level	regions	eliminated	ports	what the level reads
unit	20	6	4	tap, terminal, export node, injection, two flows
feeder	4	12	6	two sections and ten unit port flows
station	1	19	–	P, A, transformer flow, sixteen port flows

Table 17.2: Posterior spreads in the collector substation as the readings of each level are admitted, from the root down. Injections and flows in megavars.

readings admitted	injection, sensed	injection, unsensed	feeder 2 total	transformer
none (priors)	1.50	1.50	3.35	5.58
station voltage sensors (P, A)	1.48	1.48	3.17	4.20
+ feeder voltage sensors and meters	1.34	1.34	0.39	0.64
+ unit terminal voltage sensors	0.58	1.26	0.39	0.63

What the posteriors say. The truth has head sections at 1.0025 to 1.0027 per unit, tail sections at 1.0125 to 1.0135, the collector bus at 1.0064, and terminal voltages from 1.0244 to 1.0410. The posterior of the collector bus is 1.00639 ± 0.00003 and of the reactive power the station transformer carries 127.8 ± 0.6 megavars against a truth of 128.8. Feeder 1's export through its cut, the combination $f_p + f_{ao} - f_s - f_{ai}$ of its four port flows, is 36.78 ± 0.39 megavars, and the sum of its five injection posteriors is 36.78, the same number, as equation (17.2) requires. Inside the units the picture is what Chapter 12 would predict: a unit with a terminal voltage sensor has its terminal at 1.03293 ± 0.00157 and its injection at 7.94 ± 0.58 megavars (truth 1.03336 and 8.03); a unit without one has 1.03176 ± 0.00340 and 7.50 ± 1.26 (truth 1.03240 and 7.67), the feeder meter and the section voltage sensors having narrowed its injection from the prior's 1.5 but the terminal remaining a virtual sensor two levels below any reading.

Information by level. Table 17.2 asks which readings narrow what, by solving the station with the readings of each level admitted in turn. The station's two voltage sensors, on the collector bus and the tie bus, narrow the transformer's flow from 5.6 to 4.2 megavars and the unit injections hardly at all: two readings at the root cannot say much about twenty injections three levels down. The feeder's voltage sensors and meters narrow the feeder total from 3.2 to 0.39 megavars, which is the meter's accuracy, and the transformer's flow to 0.64, but leave each injection at 1.34: the meter pins the sum of five injections and says nothing about their split, so the five posteriors are negatively correlated and each stays wide. Only the unit's own terminal sensor narrows its injection, to 0.58, and it does nothing for the unit next to it. Information enters at a level and flows through the ports; what a port carries is what a level can learn from above, and a port that carries a total carries a total.

Uncertainty through two levels. Equation (8.6) splits a region's interior covariance into two terms, the region's own conditional uncertainty given its ports and the port uncertainty carried in by the sensitivity $\mathbf{X}$; in a hierarchy the second term is itself the sum of the feeder's own conditional term and what the feeder's ports carried down from the station, by the same equation applied one level up. For the unsensed unit the split is stark. Given its four ports, the tap and export flows and the two section voltages, its terminal is known to 2×10^{-5} per unit

Table 17.3: The exact hierarchical posterior against the lumped station, in which each feeder is one node. Flows in megavars, voltages in per unit.

quantity	exact	lumped	true
collector bus P	1.00639 ± 0.00003	1.00386 ± 0.00002	1.00644
flow through the station transformer	127.8 ± 0.6	77.1 ± 0.4	128.8
feeder 1 flow to the collector bus f_p	30.6 ± 0.3	18.2 ± 0.2	30.4
feeder 1 total injection	36.8 ± 0.4	36.8 ± 0.4	36.6
feeder 1 voltage(s)	1.0025 / 1.0125	1.0075	1.0025 / 1.0125
misfit on the readings	23.5 / 22	749 / 14	–

and its injection to 0.02 megavars: the unit's own factors are physics rows of width ten kilovars, and once what crosses its boundary is fixed, conservation fixes the inside. All of its 0.0034 per unit and 1.26 megavars are carried in through the ports, almost entirely through the export flow, whose posterior spread is 1.16 megavars. Principle 1.3 said the boundary is where a model's ignorance concentrates; the hierarchy makes the statement quantitative at every level, because each level's uncertainty about a child is a term in the child's covariance, and the terms can be read off separately.

The lumped station. Now replace each feeder by one node L_i with the feeder's total injection, the feeder's four exchange branches, and the feeder's two voltage sensors both reading L_i. Table 17.3 compares. The lumped model gets the feeder totals right, to the meter's accuracy, because Proposition 17.1 guarantees it. It gets the collector bus wrong by 0.0025 per unit and the transformer's flow wrong by 50 megavars, because its one node at 1.00749 pushes 19 megavars per feeder straight back out of the head section, where the true head at 1.00254 pushes six. The reported spreads are 2×10^{-5} per unit and 0.4 megavars: the lumped model is confidently wrong, in the sense of Chapter 4, and its misfit says so, 749 on fourteen readings against the exact model's 23.5 on twenty-two. Two voltage sensors ten millivolts apart on one node cannot both be fitted. A modeler could lump more carefully, with two nodes per feeder, and would do better; the message does not require the judgment, because it keeps every port.

A local change. Add a terminal voltage sensor to unit $(2, 4)$, which had none. Its terminal posterior moves from 1.03176 ± 0.00340 to 1.03223 ± 0.00155 and its injection from 7.50 ± 1.26 to 7.68 ± 0.58, on a truth of 7.67. The largest change of any posterior mean is 0.47 inside that unit, 0.19 in the other units of its feeder, whose injections share the feeder meter with it and are explained away, 0.002 in its feeder's sections, 0.001 in the units of the other feeders, and 0.0005 in the station's buses. The hierarchy recomputes exactly what moved: the unit's message, a 6×6 elimination; its feeder's message, a 12×12; and the station solve of nineteen. Nineteen unit messages and three feeder messages are reused. The monolithic route refactorizes the 187×187 precision.

Figure 17.3 prices the two routes as the station grows. At 187 variables the monolithic sparse solver is faster, three tenths of a millisecond against six tenths, because the hierarchical route's bookkeeping costs more than the algebra it saves. At 2 659 variables the hierarchical route is already seven times quicker, and at 26 000 the hierarchical update takes three milliseconds against two hundred and sixteen: a factor of sixty, and growing, because the update's cost is fixed by the path from the changed unit to the root while the refactorization's grows with the

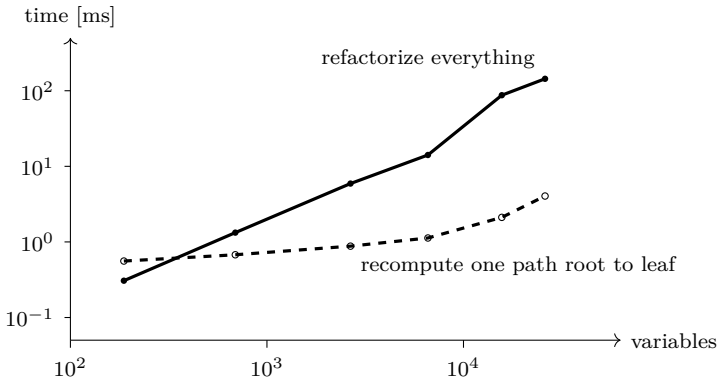


Figure 17.3: Wall-clock cost of incorporating one new voltage sensor on one unit, as the station grows from four feeders of five units to sixty-four feeders of fifty. Filled: refactorize the whole precision with a sparse direct solver. Open: recompute the changed unit's message, its feeder's message and the station solve, with every other message held. Both axes logarithmic; each point is the best of three runs.

station. Both routes give the same answer, to 8×10^{-10} at the size where the check was run. The sparse solver's factorization is itself a nested elimination, chosen by its ordering heuristic, so the hierarchy does not beat it at solving; what the hierarchy adds is the reuse, which a solver that does not know the physics cannot offer.

17.5 The cut, three times

The hierarchy of cuts has served so far as a way to organize inference. It is the same hierarchy that organizes two other things engineers do with large systems, and the coincidence is not one.

Modeling. A level of description is a choice of cut. The station's operator models feeders, the feeder's designer models units, the unit's designer models windings; each draws the boundary that the questions of that level need and treats what is inside as a message and what is outside as a reservoir or a port. Principle 1.3 said a system is defined by what crosses its boundary; the hierarchy says a system is defined at every level by that level's boundary, and the levels agree because the messages compose.

Control. A controller acts at a port. The station transformer's tap sets the voltage at the collector bus, which is the station's boundary condition; a transmission operator asks a feeder to hold its substation flow; a system operator gives a collector station a reactive schedule. Each is an intervention in Chapter 13's sense, a named quantity imposed at a cut, and each is met by the child, which back-substitutes the imposed port value into its interior and finds what its own inputs must do. The parent needs the child's message to know how the port responds to what it imposes, and nothing else. This is the structure of hierarchical control as Mesarović, Macko and Takahara set it out [61]: coordination at the interfaces, local decisions inside, the interfaces being exactly the ports. Section 17.6 works one.

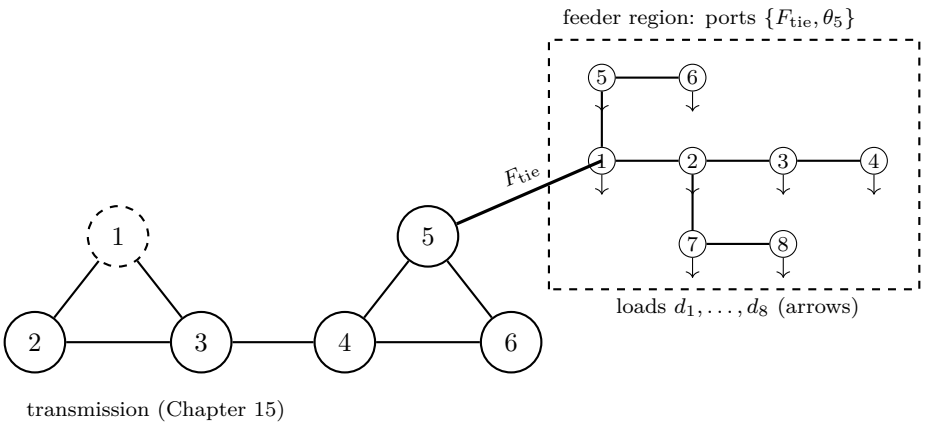


Figure 17.4: A radial feeder under bus 5 of the six-bus network. To transmission the feeder is a region behind the tie; its message is a Gaussian on the tie flow and the substation angle.

Scale. A large system is solved as a tree of regions, and the cost is set by the ports, not by the interiors. The cuts that give a good hierarchy for inference are those with few ports relative to their interiors, and the physical hierarchy, which was drawn for modeling and control, tends to have that property, because engineers build subsystems with narrow interfaces on purpose. Chapter 18 takes this up and measures it.

> **Principle 17.3.** *One cut serves three uses. As a level of description it fixes what is modeled and what is summarized; as a control interface it is where a setpoint is imposed and met; as a partition for inference it fixes the cost. The physical hierarchy is a good hierarchy for all three because its interfaces are narrow by design.*

17.6 Worked example: a feeder under a transmission bus

The six-bus network of Chapter 15 had a load of one per unit at bus 5. Now bus 5 is a substation, and the load is a radial distribution feeder of eight nodes with uncertain loads whose prior means sum to one and whose spreads are a quarter of their means; Figure 17.4 draws it. The feeder is a region with twenty-four interior variables, eight angles, eight branch flows and eight loads, and two ports, the tie flow F_{tie} from bus 5 into the feeder and the angle θ_5 on the transmission side. Transmission has its own factors, the closures and balances of Chapter 15 with two flow meters, and reads the feeder only through the tie.

The message is an injection. Eliminating the feeder's interior onto its two ports gives a precision of 110.65 on the tie flow and 1.9×10^{-9} on the angle. The feeder says nothing about θ_5: with only loads inside and no source, its net draw is the sum of its loads whatever the angle at which it is supplied, by Proposition 17.1. Its message is the one-dimensional statement $F_{\text{tie}} \sim \mathcal{N}(1.000,\, 0.095^2)$, an equivalent injection with error bars, the spread being the loads' prior spreads in quadrature, 0.090, widened slightly by the physics rows. This is what the transmission planner's "net load at bus 5" has always been, made exact and given a variance. Transmission's posterior, with its two meters, has the tie at 1.005 ± 0.090 and the line from bus 4 at 0.877 ± 0.066.

Table 17.4: The feeder's eight single-branch outages as messages to transmission, and transmission's flows on lines 4–5 and 5–6 with each message in place. Line 5–6 is rated 0.6.

branch out	load shed	message F_{tie}	F_{45}	F_{56}
0–1 (the tie's own branch)	1.00	0.000 ± 0.010	0.375	0.373
1–2	0.60	0.400 ± 0.062	0.589	0.160
1–5	0.25	0.750 ± 0.083	0.762	−0.012
2–7	0.30	0.700 ± 0.078	0.737	0.012
2–3	0.20	0.800 ± 0.087	0.786	−0.036
7–8	0.14	0.860 ± 0.088	0.813	−0.064
5–6	0.11	0.890 ± 0.090	0.827	−0.078
3–4	0.08	0.920 ± 0.092	0.841	−0.091

A meter at the port narrows the interior. Put a meter on the tie, reading 1.12 ± 0.02. The tie's posterior is 1.115 ± 0.020, and the correction passes down: every feeder load moves up from its prior, by 0.25 to 0.51 of its spread, in proportion to its prior variance, which is the virtual-sensor arithmetic of Chapter 12 applied through a port. Each load's spread shrinks only to between 0.91 and 0.98 of its prior, because one number cannot narrow eight; what the meter has narrowed is their sum, from 0.090 to 0.020, and their posterior is correlated to say so.

Failure sets enumerated inside, combined at the port. Chapter 15 enumerated failure sets over a whole system and promised that a hierarchy would enumerate them by region. Here is the mechanism. Each of the feeder's eight branches is a candidate outage; in a radial feeder an outage islands everything downstream, whose load is shed. Table 17.4 lists the eight, each as a feeder message: an injection reduced by the shed load, with a variance reduced by the shed loads' priors. Transmission's consequence of each is one solve of its reduced system of seventeen variables with the feeder's message swapped. Combined with transmission's own seven line outages, the fifty-six pairs cost eight feeder messages and fifty-six reduced solves, not fifty-six solves of the whole forty-one-variable model. With more feeders the saving compounds: a system of m feeders with b branches each has mb feeder messages and one reduced system, and the pairs are combinations of held messages.

The port as a setpoint. Transmission, finding line 5–6 loaded, asks the feeder to hold the tie at 0.85. In the model that is one factor on the port, a setpoint with a negligible width, which Chapter 13 would call imposing an input. The feeder's posterior distributes the required reduction of 0.15 over its eight loads in proportion to their prior variances, shares of 0.16, 0.07, 0.10, 0.04, 0.14, 0.08, 0.18 and 0.14 against prior-variance shares of 0.17, 0.08, 0.11, 0.05, 0.15, 0.09, 0.20 and 0.15, the small differences again the physics rows. That is the feeder's own decision to make, on its own model, and with its own priors replaced by curtailment costs it would be an optimization inside the feeder; transmission sees only the port, where the flows on lines 4–5 and 5–6 become 0.800 and −0.050. The parent imposed, the child met, and each used only its own factors and the message between them.

17.7 Hierarchy in time

Chapter 16 unrolled the graph in time and found the storage states to be separators between past and future. A hierarchy in time therefore has two families of separators, the ports of each cut at each step and the storage states of the whole system between steps, and the question is whether they can be used together. They can, in one order, and not in another that is tempting.

Proposition 17.3 (Nested elimination in space and time)**.** *In the unrolled graph of a hierarchy of regions, eliminating at each step the regions' interiors onto their ports and the storage states, and then eliminating the step onto its storage states, is exact. The message carried from step to step is a Gaussian on all the storage states of the system jointly.*

Proof. Within a step, the regions' factors read their interiors, their ports and their own storage states at the previous step; the ports separate the interiors as in Proposition 5.3, so eliminating each interior onto its ports and storage states is a region message, and Proposition 17.2 lets it be done level by level. Across steps, Proposition 16.2 says the storage states separate; eliminating everything at the step except the states gives a Gaussian on the states, which is the filter's carried message. Every elimination is a Schur complement and every one is exact. □

The tempting order is to let each region carry its own storage states and nothing else: a filter per unit, a filter for the station, each updating from its own readings and its neighbors' ports. The proposition says what that loses. The carried message is a Gaussian on the states *jointly*, and after one step of conditioning on shared readings, a feeder meter or a collector-bus voltage sensor, the states of different regions are correlated. A feeder meter that reads five injections at once makes their errors negatively correlated, because a reading that says the total is high says that if one injection is higher than estimated another is lower. A per-region filter keeps each region's block of the covariance and drops the rest, which is to say it treats the regions as independent at the start of every step, and then conditions on the same shared reading again as if it were new information about each.

The mechanism is visible in two injections. Let q_1 and q_2 have independent priors of variance v, and let a meter read their sum, $y = q_1 + q_2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, once per step, with the sources not moving. Exactly: the sum $s = q_1 + q_2$ has prior variance $2v$ and after n readings has variance $(1/2v + n/\sigma^2)^{-1}$, while the difference $d = q_1 - q_2$ is never read and keeps its variance $2v$; since $q_1 = (s + d)/2$,

$$\mathrm{Var}(q_1 \mid y_1, \dots, y_n) \;=\; \tfrac{1}{4}\Big[\big(\tfrac{1}{2v} + \tfrac{n}{\sigma^2}\big)^{-1} + 2v\Big] \;\longrightarrow\; \tfrac{v}{2} \quad (n \to \infty). \tag{17.5}$$

Half the prior variance is all a sum meter can ever remove from one injection. A filter that carries q_1 and q_2 in separate blocks conditions each reading on the current variances v_n of both as if independent, $v_{n+1} = v_n - v_n^2/(2v_n + \sigma^2) = v_n\,(v_n + \sigma^2)/(2v_n + \sigma^2)$, which decreases at every step and tends to zero. It comes to believe each injection is known exactly, when only their sum is. Every shared reading is treated as fresh information about each part because the correlation that records "this was already used" has been thrown away.

Table 17.5 measures the loss on the station with storage at the unit terminals and the station buses, the injections random-walking by 0.15 megavars per step, and the same readings every step as before. The exact filter's errors are what it reports: root-mean-square 1.06 against a reported 1.04, and three percent of errors beyond two spreads, as a Gaussian should have. The regional filters have the same errors and report a third of them. Forty-two percent of their errors

Table 17.5: Filtering the station's twenty injections over eighty steps of thirty seconds, with every state carried jointly (exact) and with each unit carrying only its own terminal and injection and the station only its two buses (regional). Errors over the last sixty steps.

	exact	regional
rms error of an injection [MVAr]	1.06	1.07
reported spread of an injection [MVAr]	1.04	0.34
ratio, error to reported	1.02	3.12
fraction of errors beyond two spreads	0.033	0.425
reported spread of the twenty injections' sum [MVAr]	0.95	1.53

lie beyond two reported spreads. And the error is not one-sided: the exact posterior correlation between two injections on the same feeder is -0.18, which makes their sum better known than the injections individually, 0.95 against $\sqrt{20} \times 1.04$; the regional filters, having dropped the correlation, report the sum at 1.53, worse than the truth, while reporting each injection better than the truth. They are overconfident about the parts and underconfident about the whole, which is the signature of dropped negative correlations. The remedy in the filtering literature is either to carry the joint, as the proposition prescribes, or to fuse with a rule that is consistent for any correlation at the price of conservatism, which is covariance intersection [37]; what is not a remedy is to pretend the regions are independent.

Principle 17.4. *In time, ports separate regions within a step and storage states separate steps, and both eliminations are exact in that order. What is carried between steps is the joint Gaussian on all storage states. A region that carries only its own states forgets the correlations that shared readings created, and reports its parts too well and its whole too badly.*

17.8 In practice

Cut where the engineers cut. The physical hierarchy of a system, its stations, feeders and units or its regions and tie lines, was drawn by people who wanted narrow interfaces. Use it. A cut that follows a drawing boundary has ports that already have names and usually have meters.

Put the meter on the ports. A meter that reads a region's total is a factor on the region's port flows by Proposition 17.1. Written that way it belongs to the parent, and the parent can use it before the child's message is available, or with a child that is not modeled at all.

Hold the messages. The value of the hierarchy is reuse, and reuse requires that every region's message and factorization be kept between questions. A change recomputes one path from the changed region to the root; a question about an interior back-substitutes down one path. Nothing else is touched.

Do not lump what you can reduce. A region summarized to one node with one potential is exact for its total flow and wrong for its distribution among the ports. The message costs one elimination and keeps every port. If a lumped model must be used, run its misfit test; the lumped station's misfit was fifty times its degrees of freedom.

Impose at the port, meet inside. A setpoint is a factor on a port. The parent imposes it on the child's message; the child back-substitutes it into its interior and decides, with its own model and costs, how to meet it. Neither needs the other's variables.

In time, carry the joint. A hierarchy of filters that each carry their own states is not a filter; it is a set of filters that condition on shared readings twice. Carry the joint Gaussian on the storage states, or fuse conservatively, and check the reported spreads against the innovations.

17.9 What is missing

The hierarchy of this chapter was given, drawn from the physical layout, and every region's ports were few by construction. Two questions remain. When the layout does not give a hierarchy, or gives one whose regions have many ports, how should a large graph be cut, and what does the cut cost? Chapter 18 answers with the boundary dimension of a partition and measures it on test networks. And when the messages are too many or the regions too large for exact elimination, what approximations respect the hierarchy? Chapter 18 also treats co-simulation, in which regions exchange port values by iteration rather than by message, and shows when it diverges.

Notes and further reading

Aggregation of flows across a cut, Proposition 17.1, is the summation of balance rows and appears wherever conservation laws are discretized; the chapter's contribution is only to say that it places total-reading meters on the ports. The composition of Schur complements, Proposition 17.2, is the quotient formula for Schur complements, and its use to eliminate nested substructures is the substructuring of structural analysis [30] and the nested dissection of sparse direct methods [25]; its extension to dynamics as component mode synthesis is Craig and Bampton's [15]. Kron reduction as the electrical form of the same elimination is [18, 47], and the equivalent injection of Section 17.6 is the Ward equivalent [95] with a variance. The view of the hierarchy as a junction tree is Chapter 8's, following [46, 51]. Model reduction in the control-theoretic sense, which approximates where the message is exact, is surveyed by Antoulas [2]. Hierarchical control as coordination at interfaces is Mesarović, Macko and Takahara's [61]. The failure of per-region filters and the covariance-intersection remedy are Julier and Uhlmann's [37]. The reading of the physical hierarchy as simultaneously a level of description, a control interface and an inference partition is the book's.

Summary

- Summing a region's balance rows gives the balance of the region as a node, exactly; a meter on a region's total is a factor on its ports. Intensive states do not aggregate.
- Schur complements compose, so a hierarchy of regions is a junction tree whose messages are computed level by level. Each level pays for its own variables: the unit six, the feeder twelve, the station nineteen, for a model of 187.
- Information enters at a level and flows through the ports. A feeder meter pins the feeder's total and leaves each injection wide; only a unit's own voltage sensor narrows its injection.

- The parent's model of a child is the child's message, an equivalent network with error bars. A lumped node is exact for the total flow and wrong for its distribution: the lumped station misplaced 50 megavars and reported a spread of 0.4.
- A local change recomputes one path from leaf to root; at 26 000 variables that is sixty times cheaper than refactorizing, and the ratio grows.
- The same cut is a level of description, a control interface and an inference partition. A setpoint at a port is imposed by the parent and met inside the child.
- A feeder's message to transmission is an injection with a variance; its failure sets are enumerated inside and combined at the port; a substation meter narrows its loads through the port.
- In time, carry the joint Gaussian on all storage states. Per-region filters report each injection three times too well and the sum too badly.

Exercises

Exercise 17.1. Prove equation (17.3) by block elimination. Write $\mathbf{\Lambda}$ in the blocks (I_1, I_2, S), eliminate I_1 to obtain a 2×2 block matrix on (I_2, S), eliminate its I_2 block, and show that the result equals $\mathbf{\Lambda}_{SS} - \mathbf{\Lambda}_{SI}\mathbf{\Lambda}_{II}^{-1}\mathbf{\Lambda}_{IS}$ with $I = I_1 \cup I_2$, using the block inverse of $\mathbf{\Lambda}_{II}$.

Exercise 17.2. Feeder 1 of the collector substation has seventeen balance rows, two sections and three per unit, each with physics width σ_p. Show that the identity $f_p + f_{ao} - f_s - f_{ai} = \sum_j q_{1j}$ holds in the model with a width of $\sqrt{17}\,\sigma_p$, and hence that placing the feeder meter on the port flows or on the injections gives posteriors that differ by at most that order. Compare with the 1.6 kilovars the script found for $\sigma_p = 10$ kilovars and a meter width of 400.

Exercise 17.3. Lump each feeder of the collector substation to two nodes instead of one, a head node carrying the grid-busbar and tie-in branches and a tail node carrying the collector-bus and tie-out branches, with the feeder's injection at the tail node. Recompute Table 17.3. Which entries does the two-node lump fix, which does it still get wrong, and what would the lump need in order to be exact on all six ports?

Exercise 17.4. In the feeder example, give each load a curtailment cost c_k per unit shed, and replace the Gaussian prior on the loads by the constraint that the shed amounts are nonnegative and the objective $\sum_k c_k s_k$. With the tie held at 0.85, which loads are curtailed? Show that transmission's view of the feeder is unchanged by the replacement: the message on the port is still a point at 0.85, and the flows on lines 4–5 and 5–6 are the same as in Section 17.6.

Solutions to selected exercises

Exercise 17.1. Write

$$\mathbf{\Lambda} = \begin{pmatrix} A & B & C \\ B^{\mathsf{T}} & D & E \\ C^{\mathsf{T}} & E^{\mathsf{T}} & G \end{pmatrix}$$

on (I_1, I_2, S). Eliminating I_1 gives, on (I_2, S),

$$\begin{pmatrix} D - B^{\mathsf{T}}A^{-1}B & E - B^{\mathsf{T}}A^{-1}C \\ E^{\mathsf{T}} - C^{\mathsf{T}}A^{-1}B & G - C^{\mathsf{T}}A^{-1}C \end{pmatrix} =: \begin{pmatrix} D' & E' \\ E'^{\mathsf{T}} & G' \end{pmatrix},$$

and eliminating I_2 from that gives $G' - E'^{\mathsf{T}} D'^{-1} E'$. On the other side, $\mathbf{\Lambda}_{II} = \left(\begin{smallmatrix} A & B \\ B^{\mathsf{T}} & D \end{smallmatrix}\right)$ has the block inverse, with $D' = D - B^{\mathsf{T}} A^{-1} B$,

$$\mathbf{\Lambda}_{II}^{-1} = \begin{pmatrix} A^{-1} + A^{-1} B D'^{-1} B^{\mathsf{T}} A^{-1} & -A^{-1} B D'^{-1} \\ -D'^{-1} B^{\mathsf{T}} A^{-1} & D'^{-1} \end{pmatrix},$$

and $\mathbf{\Lambda}_{SI} \mathbf{\Lambda}_{II}^{-1} \mathbf{\Lambda}_{IS}$ with $\mathbf{\Lambda}_{IS} = (C;\, E)$ expands to

$$C^{\mathsf{T}} A^{-1} C + C^{\mathsf{T}} A^{-1} B D'^{-1} B^{\mathsf{T}} A^{-1} C - C^{\mathsf{T}} A^{-1} B D'^{-1} E - E^{\mathsf{T}} D'^{-1} B^{\mathsf{T}} A^{-1} C + E^{\mathsf{T}} D'^{-1} E$$
$$= C^{\mathsf{T}} A^{-1} C + E'^{\mathsf{T}} D'^{-1} E',$$

since $E' = E - B^{\mathsf{T}} A^{-1} C$. Hence $G - \mathbf{\Lambda}_{SI} \mathbf{\Lambda}_{II}^{-1} \mathbf{\Lambda}_{IS} = G' - E'^{\mathsf{T}} D'^{-1} E'$, which is the two-stage result. The information vector follows by the same substitution with $\boldsymbol{\eta}$ in place of the last block column.

Exercise 17.2. Each balance row of the feeder's region is $r_n = \sum_{e \ni n} \pm F_e - q_n$ with $r_n \sim \mathcal{N}(0, \sigma_p^2)$ independently. Summing the seventeen rows, every internal edge cancels and the cut edges remain: $\sum_n r_n = (f_p + f_{ao} - f_s - f_{ai}) - \sum_j q_{1j}$, and the sum of seventeen independent widths σ_p has width $\sqrt{17}\,\sigma_p$. So the model holds the identity to $\sqrt{17} \times 10 = 41$ kilovars. A meter of width 400 kilovars placed on either side of the identity is the same factor up to a width of 41 kilovars added in quadrature, $\sqrt{400^2 + 41^2} - 400 = 2$ kilovars, and the posteriors differ by a quantity of that order in the meter's direction, scaled by how much the meter moves the estimate; the script's 1.6 kilovars is consistent.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Evidence that crosses a domain *one variable shared between two physical domains; a measurement in one domain informing an unknown in another; a negative control that isolates the mechanism*
`power_grid/cross_domain_inference_interdiction/problems/evidence_crosses_the_domain`

Chapter 18

Partitioning at scale

Chapters 8 and 17 took the cut as given. The collector substation came with feeders and units, the network with a feeder behind a substation, and the ports were few because the engineers who drew the system had made them few. This chapter asks what happens when the cut must be chosen, and what a bad choice costs. The answer has a single currency. Every route through a partitioned model, exact elimination, messages between regions, iteration between regional solvers, enumeration of failure sets region by region, pays in the number of ports: the *boundary dimension* of the partition. A cut with few ports relative to its interiors is cheap on every route; a cut with many ports is expensive on every route and, for the iterative routes, fails.

The chapter measures rather than argues. Three transmission networks of the standard kind, with 73, 118 and 1 354 buses, and the collector substation of Chapter 17 at four sizes, supply the graphs. On each the chapter measures the elimination width that Chapter 5 promised, the boundary dimension of partitions chosen three ways, the convergence of region-level message passing that Chapter 9 promised, and the convergence of co-simulation. It then reads, from the case studies of Part VI, what the same partitions do when the regions are not Gaussian but sets of failure states, which is the question Chapters 10 and 15 left open. The chapter's own computations, of width, boundary dimension, propagation and co-simulation, are from `ch18_partitioning.py`. The separator compression, the region trees and the data-hall chain are read from the committed results files of the case studies that measured them.

18.1 The cost of a cut

Let a partition divide the variables into region interiors $I_1, \dots, I_R$ and a port set S, as in equation (8.4). Region r has n_r interior variables and k_r ports, and $|S| \le \sum_r k_r$, with equality when no port is shared. The exact route of Chapter 8 costs one factorization of each $\mathbf{\Lambda}_{II}^{(r)}$, k_r solves against it to form the message, and one dense factorization of the $|S| \times |S|$ port system. Write w_r for the elimination width of region r's interior on its own, the largest front its factorization creates.

Proposition 18.1 (Width of the partition ordering)**.** *Eliminating every region's interior before any port, each interior in an ordering of width w_r, produces fronts no larger than* $\max_r(w_r + k_r)$ *during the interior phase and no larger than $|S|$ during the port phase. The elimination width of the whole is at most* $\max\big(\max_r(w_r + k_r),\ |S| - 1\big)$.

Proof. While region r's interior is being eliminated, no other region's interior has been touched

Table 18.1: Elimination width of the nodal graphs, by minimum-degree and minimum-fill orderings, against $\sqrt{n}$. The largest dense block a sparse factorization forms is one more than the width.

graph	nodes	edges	width (min. degree)	width (min. fill)	$\sqrt{n}$
RTS-GMLC	73	108	5	5	8.5
IEEE 118	118	179	4	4	10.9
PEGASE 1354	1 354	1 710	14	12	36.8
station, 4×5	70	92	2	2	8.4
station, 8×10	258	344	2	2	16.1
station, 16×20	994	1 328	2	2	31.5
station, 32×25	2 466	3 296	2	2	49.7

and none is adjacent, by the bordered structure; the only variables outside I_r that an interior variable of r can be joined to are r's own ports. So a front during that phase is a front of r's own elimination, of size at most w_r, together with at most k_r ports. Once every interior is gone the remaining graph is on S, and any ordering of it has fronts of size at most $|S|$. □

In operations the same accounting reads

$$\text{cost} \;\approx\; \sum_r \left(n_r w_r^2 + k_r\, n_r w_r\right) \;+\; \tfrac{1}{3}|S|^3, \tag{18.1}$$

the first term for factorizing each interior with fronts of width w_r, the second for the k_r solves that form the message, the third for the dense port system; back-substitution adds a solve per region, which is lower order. The two region terms are linear in the region sizes, so a partition can be as fine as one likes without penalty as long as the ports do not grow; the port term is cubic, and it is the one a bad cut inflates.

The proposition says that a partition's cost is governed by two numbers: the width of the worst region plus its ports, and the size of the port set. The first is small when regions are small or themselves well structured; the second is the boundary dimension. Nested dissection, the ordering that sparse direct solvers use, is this proposition applied recursively [25]: cut the graph by a small separator, order the two sides first and the separator last, and recurse into the sides. Planar graphs have separators of size $O(\sqrt{n})$ [54], which bounds the elimination width of nested dissection on them by $O(\sqrt{n})$, and physical networks are nearly planar. What the bound leaves open is the constant, and the constant decides whether a network of a thousand buses has a port system of thirty or of three hundred. The next section measures it.

18.2 Elimination width of physical networks

Table 18.1 gives, for each graph, the elimination width found by the two standard greedy orderings, minimum degree and minimum fill, which are upper bounds on the treewidth of Definition 5.2. The three transmission networks are the IEEE Reliability Test System of 73 buses in three areas, the IEEE 118-bus system, and the 1 354-bus PEGASE model of part of the European grid, all taken as their nodal graphs with parallel branches merged. The collector substations are the graph of Chapter 17's station, collector bus and tie bus and feeders of units, at four sizes.

Two things stand out in Figure 18.1. The transmission networks have widths of 4, 5 and 12, a third of $\sqrt{n}$ and less, so that the largest dense block a direct factorization ever forms on the

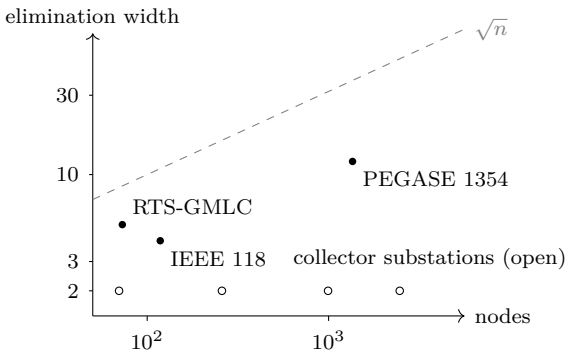


Figure 18.1: Elimination width against the number of nodes, both axes logarithmic. Filled: the three transmission networks. Open: the collector substation at four sizes. The dashed line is $\sqrt{n}$, the planar-separator bound's scaling. The transmission networks sit well below it and the substations do not grow at all.

1 354-bus network has thirteen rows. Exact inference on these graphs is not merely affordable; it is cheap, and the reason is the one Chapter 5 gave, that a transmission network is a sparse, nearly planar graph whose cycles are local. And the collector substation's width is 2 at every size. Its graph is a set of chains (tap, terminal, export node) hanging between the two section buses of each feeder, with every feeder hanging on the same two station buses; eliminate the chains, then the sections, and the front never exceeds three. This is the hierarchy of Chapter 17 seen by the elimination algorithm: a system built from subsystems with narrow interfaces has an elimination width set by the interfaces, not by the size.

Principle 18.1. *Physical networks have elimination widths far below the planar bound: four to twelve on standard transmission systems of* 73 *to* 1 354 *buses, two on a collector substation at any size. Exact Gaussian inference on them is cheap, and a partition is chosen for reuse and for what it enables, not to make the solve possible.*

18.3 Choosing the cut

Three ways to cut a graph into regions are compared. The *physical* cut follows labels the system carries: the three areas of the Reliability Test System, which are its three synchronous regions. The *spectral* cut bisects the graph by the sign of the Fiedler vector, the eigenvector of the susceptance-weighted Laplacian belonging to its second-smallest eigenvalue [24], splitting at the median so that the halves are equal, and recurses to give $2, 4, 8, \ldots$ regions [74]; a Kernighan–Lin pass [45] could refine it and, on these graphs, changes little. The *random* cut assigns buses to equal-sized regions at random, and stands for a partition chosen without regard to the physics: by owner, by alphabetical order, by whatever data happened to be at hand.

Table 18.2 and Figure 18.2 give the result. Cut in two, the 1 354-bus network has thirty boundary buses, two percent of the total; cut in two at random it has 950, seventy percent. The ratio is thirty, and it is the ratio of the port systems' sizes and, cubed, of their factorization costs. The same holds at every R: the random cut's boundary is essentially every bus once R exceeds a handful, because a random region of size n/R in a graph of average degree three has almost no interior. The spectral cut's boundary grows with R, as it must, since more regions mean more

Table 18.2: Boundary dimension of partitions of the three networks and of two collector substations. Ports are boundary nodes, counted once each; $k_{\max}$ is the largest region's port count. The substations' physical cut is by feeders, with the collector bus and tie bus as a root region.

network	cut	R	ports $\lvert S\rvert$	share of buses	$k_{\max}$	cut branches
RTS-GMLC	areas	3	10	14%	4	5
	spectral	2	11	15%	6	7
	spectral	4	26	36%	8	19
	spectral	8	42	58%	9	32
	random	2	60	82%	30	56
	random	8	71	97%	10	106
IEEE 118	spectral	2	16	14%	8	12
	spectral	4	38	32%	12	26
	spectral	8	54	46%	10	39
	random	2	96	81%	50	89
	random	8	118	100%	15	162
PEGASE 1354	spectral	2	30	2%	17	24
	spectral	4	87	6%	29	82
	spectral	8	235	17%	67	209
	spectral	16	327	24%	38	299
	spectral	32	508	38%	27	496
	random	2	950	70%	477	965
	random	32	1 334	99%	43	1 922
station 8×10	feeders	9	18	7%	2	24
	spectral	8	59	23%	22	43
	random	8	256	99%	33	308
station 32×25	feeders	33	66	3%	2	96
	spectral	32	2 174	88%	77	2 203
	random	32	2 464	100%	78	3 194

cuts, but from a base two orders of magnitude lower. On the Reliability Test System the physical cut, three areas with ten boundary buses, is as good as the best spectral cut and better than the spectral cut into four, which is the observation of Chapter 17 made quantitative: the areas were drawn by engineers who wanted few ties, and on a transmission network the eigenvector finds cuts of the same quality.

The substation, cut three ways. The collector substation makes the comparison on a hierarchy built by design, and adds a caution. Cut by feeders, with the collector bus and tie bus as a root region of their own, the station of eight feeders and ten units has eighteen boundary nodes among 258, seven percent, and no region has more than two ports; at thirty-two feeders of twenty-five units the shares are 66 of 2 466, under three percent, and still two ports per region. Cut spectrally into the same number of regions the smaller station has 59 boundary nodes, twenty-three percent, and a worst region with twenty-two ports where the physical cut has two; the larger has 2 174 of 2 466, eighty-eight percent. Cut at random, 256 and 2 464, all but two. The spectral cut fails here for a reason worth knowing: recursive bisection at the median insists on equal halves, and a graph in which every feeder hangs on the same two hub buses has no balanced cut that respects the feeders, so the Fiedler vector slices across them. The physical

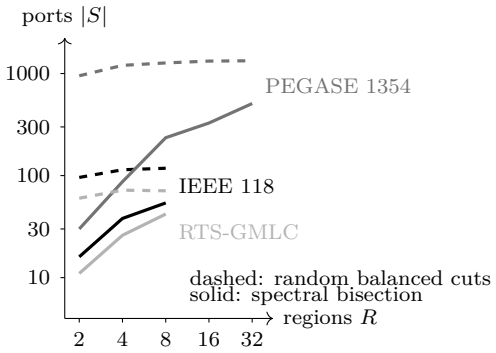


Figure 18.2: Boundary dimension against the number of regions for the spectral cuts (solid) and the random cuts (dashed) on the three networks, both axes logarithmic. A random cut puts most buses on the boundary at any R; a spectral cut's boundary grows roughly as $R^{0.8}$ from a small base.

cut is unbalanced, a root region of two nodes and thirty-two regions of seventy-seven, and that is exactly what makes it good. Balance is a constraint of parallel computing, not of inference; the cost model of Section 18.1 charges for ports and region widths, and an unbalanced partition with narrow interfaces beats a balanced one with wide ones.

The largest region port count $k_{\max}$ tells the same story from the region's side. A spectral region of the PEGASE network has 17 to 67 ports; a random region of the same size has hundreds. Since a region's message is a dense $k_r \times k_r$ block, and since the discrete enumeration of Section 18.5 is exponential in what the ports can express, $k_{\max}$ is the number that decides whether a region can be summarized at all.

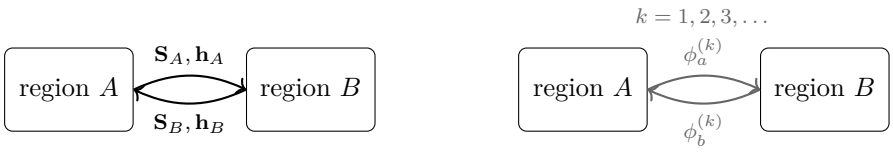
Principle 18.2. *Cut where the flows are thin. A cut chosen by the physics, by the system's own areas or by the Fiedler vector of its weighted Laplacian, has a boundary of a few percent of the variables; a cut chosen without the physics has a boundary of most of them, and every route through the partition pays the difference.*

18.4 Coupling the regions: messages, propagation, or iteration

Once the regions are cut, their interiors are eliminated onto the ports and what remains is the port system, a precision on S built from the region messages. There are three ways to solve it; Figure 18.3 draws the first and the third for two regions.

The first is exact: factor the port system as one dense matrix. It costs $|S|^3/3$ and never fails. When the region graph is a tree, as for two regions, the factorization is one message each way (Chapter 8).

The second is to pass messages between regions and iterate: Gaussian belief propagation on the region graph, with each region's ports as one vector-valued variable. Chapter 9 found that propagation on the physical graph itself never converges, and asked whether it converges on the region graph, whose factors are messages that have absorbed the tight ties. The walk-summability criterion of Chapter 9 for scalar variables is that the spectral radius of the absolute partial-correlation matrix, $\rho(|\mathbf{R}|)$, is below one; for vector variables the natural analogue replaces each scalar partial correlation by the spectral norm of the normalized block, $\|\mathbf{D}_r^{-1/2}\mathbf{\Lambda}_{rs}\mathbf{D}_s^{-1/2}\|_2$,



messages: one pass, exact on a tree co-simulation: exchange potentials, re-solve, repeat

Figure 18.3: Two ways to couple two regions. Left: each region sends the other its Schur complement onto the ports, a precision and an information vector; the port posterior is exact after one exchange. Right: each region re-solves with the other's last boundary potential and sends the new one; the sequence converges to the same answer at a rate set by the tie strength.

with $\mathbf{D}_r$ the region's own port block, and asks whether the spectral radius ρ_B of the matrix of those norms is below one. Both are computed here, and the propagation is also simply run, to 10^{-8} or to failure.

The third is co-simulation: each region solves its own physics with its neighbors' boundary potentials held at the last exchanged values, and the regions exchange again. Exchanging simultaneously is block Jacobi on the nodal system; exchanging in sequence is block Gauss–Seidel. Both converge on these systems, since the nodal matrix with a slack is an M-matrix and block splittings of M-matrices are regular [92], but at a rate set by the partition.

The convergence factor of an exchange. The rate has a closed form for two regions joined by one tie. Eliminate each region's interior onto its boundary bus; region A becomes an equivalent conductance g_a from its boundary potential ϕ_a to the reference together with an equivalent injection, region B likewise with g_b, and the tie has conductance g_t. The port system is

$$\begin{pmatrix} g_a + g_t & -g_t \\ -g_t & g_b + g_t \end{pmatrix} \begin{pmatrix} \phi_a \\ \phi_b \end{pmatrix} = \begin{pmatrix} p_a \\ p_b \end{pmatrix}. \tag{18.2}$$

Simultaneous exchange solves each row with the other's potential from the previous round, $\phi_a^{(k+1)} = (p_a + g_t \phi_b^{(k)})/(g_a + g_t)$ and likewise for b, whose iteration matrix has spectral radius

$$\rho_J \;=\; \frac{g_t}{\sqrt{(g_a + g_t)(g_b + g_t)}}, \qquad \rho_{GS} = \rho_J^2 \tag{18.3}$$

for the sequential variant. The factor is near zero when the tie is weak against the regions' own stiffness and near one when it is strong; a tie as strong as the regions it joins gives $\rho_J = 1/2$ and twenty exchanges for six digits, and a tie ten times stronger gives 0.91 and 150. The message route computes the fixed point of this iteration in one step, because the message *is* g_a, g_b and the injections, and the port solve inverts the 2×2 matrix. With more ports the factor is the spectral radius of the corresponding block iteration, and the closed form becomes a measurement.

Table 18.3 gives the measurements; the model is the linear-Gaussian one of Chapter 6 in the book's own variables, angles, flows and injections, with a physics width of 10^{-3}, injection priors of spread 0.1 and flow meters on a third of the branches. Four things are worth reading off.

First, the scalar walk-summability radius of the port system is above one for every partition, at 1.03 to 3.00, as it was 3.2, 5.1 and 8.1 for the three full graphs. Eliminating the interiors does not make the port system walk-summable in the scalar sense; the region messages are dense blocks, and dense blocks of partial correlations of order one half among a region's own ports are

Table 18.3: Coupling the regions of each partition. $\rho(|\mathbf{R}|)$ is the scalar walk-summability radius of the port system; ρ_B is its block analogue with regions as vector variables; the propagation column says whether region-level belief propagation reached 10^{-8} and in how many sweeps; the last two columns are the block Jacobi and block Gauss–Seidel convergence factors of co-simulation on the physics and the sequential exchanges needed for 10^{-6}.

network	cut	ports	$\rho(\lvert\mathbf{R}\rvert)$	ρ_B	propagation	ρ_J	ρ_{GS}	exchanges
RTS-GMLC	areas 3	10	1.05	1.48	converged, 20	0.734	0.541	22
	spectral 2	12	1.03	0.90	converged, 1	0.906	0.821	70
	spectral 4	32	3.00	1.99	converged, 19	0.975	0.950	270
	spectral 8	53	3.00	2.75	failed	0.979	0.958	322
	random 4	139	2.69	3.00	failed	0.995	0.990	1 364
IEEE 118	spectral 2	19	2.17	1.00	converged, 1	0.872	0.761	51
	spectral 4	43	2.97	1.86	failed	0.968	0.937	213
	spectral 8	64	3.00	2.76	failed	0.975	0.951	273
	random 4	192	2.93	3.00	failed	0.996	0.991	1 560
PEGASE 1354	spectral 2	37	2.99	1.00	converged, 1	0.961	0.924	175
	spectral 4	117	3.00	2.84	failed	0.991	0.983	791
	spectral 8	328	3.00	4.63	failed	0.995	0.989	1 307
	spectral 32	768	3.00	6.99	failed	0.999	0.999	9 740
	random 4	2 211	3.00	3.00	failed	1.000	1.000	55 369

exactly what Chapter 9 found fatal. Scalar propagation among the ports is no better than scalar propagation among the buses.

Second, block propagation, with regions as vector variables, behaves as Chapter 9 predicted. On two regions the region graph is a tree and one sweep is exact. On the Reliability Test System's three areas, whose region graph is a triangle, it converges in twenty sweeps, and on its spectral cut into four, in nineteen; on every partition into eight or more, and on every random partition, it fails, the beliefs' precisions turning indefinite within a sweep. The block radius tracks this: below one on the two-region cuts, 1.5 and 2.0 where propagation converged on a cycle, 2.8 and above where it failed. The condition $\rho_B < 1$ is sufficient, and the measurements say that a small excess over one is survivable on a region graph with one short cycle and a large excess is not. Region-level propagation is therefore a tool for a few large regions, and not a substitute for the port solve when the regions are many.

Third, co-simulation always converges on the physics, and the rate is what equation (18.3) predicts. On the three areas, with five tie lines against three regions of two dozen buses, the Jacobi factor is 0.73, the Gauss–Seidel factor is 0.54, close to the square 0.54 that the two-region formula gives, and six digits take twenty-two exchanges. On two spectral regions of the PEGASE network the factor is 0.92 and six digits take 175 exchanges; on thirty-two regions, 9 740; on a random cut into four, 55 000, the factor being one to four decimals. Figure 18.4 plots the exchange counts against the number of regions for both kinds of cut on the three networks. Each exchange is a full regional solve. The message route does the same regional solves once, plus the port factorization, and is finished.

Fourth, the port count orders every column. Read down any network's rows: as $|S|$ grows the block radius grows, propagation goes from converging to failing, and the exchange count grows by an order of magnitude and more. The boundary dimension is the one number that predicts

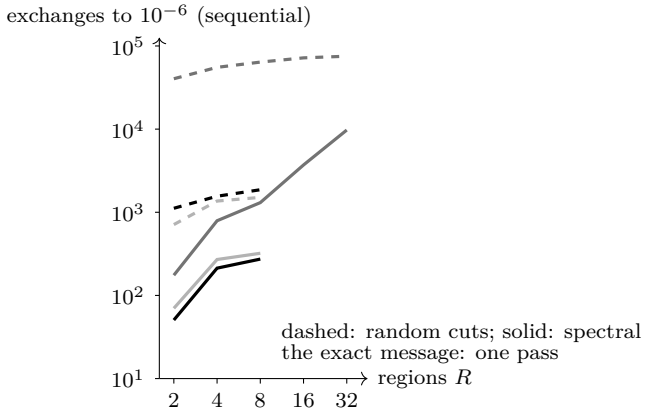


Figure 18.4: Sequential exchanges of co-simulation needed for six digits, against the number of regions, for spectral (solid) and random (dashed) cuts of the three networks. The exact message route needs one. Both axes logarithmic.

the cost of every route.

The port count as a cost predictor. The exact route was also timed: every region's message by sparse factorization of its interior, then the dense port solve, best of three runs. Figure 18.5 plots the time against the port count for every partition of the three networks. The points from three networks and two kinds of cut fall on one curve, flat where the port solve is negligible and rising with the third power of $|S|$ once it dominates: on the 1 354-bus network the spectral cut in two takes eighteen milliseconds and the random cut in two six tenths of a second, the spectral cut into thirty-two also six tenths and the random cut into thirty-two three seconds. A monolithic sparse solve of the same model takes eleven milliseconds, less than any partition, which is Chapter 17's point again: the partition is not for the one solve but for the reuse, and its cost is a function of the ports. Proposition 18.1 said as much; the figure says the constant is one.

> **Principle 18.3.** *Between regions, pass the exact message and solve the port system. Scalar propagation on the ports fails as it does on the buses; block propagation converges only among a few large regions; co-simulation converges at a rate set by the ties' strength against the regions' and needs tens to tens of thousands of exchanges where the message needs one. Every route's cost rises with the boundary dimension.*

18.5 Enumeration by region

The Gaussian measurements above bound what a partition costs when the regions' messages are precisions. Chapters 10 and 15 asked a harder question: when a region's state is a set of failed components, its message is not a Gaussian but a distribution over what the ports can take, one value per distinct separator value the region's failure states produce, and the question is how many such values there are. If many failure states of a region produce the same port values, the region compresses: its $|E_r|$ states become $|S_r|$ separator values, and the regions combine on the ports at the cost of $|S_r| \times |S_s|$ pairs rather than $|E_r| \times |E_s|$. If they do not, the partition buys nothing on this route.

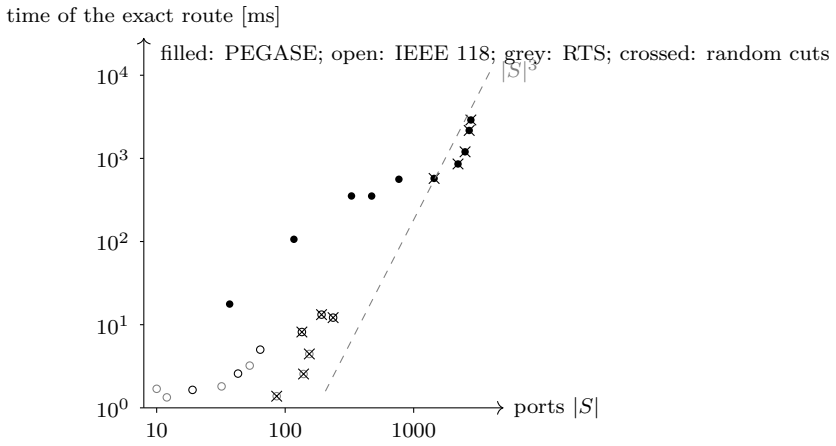


Figure 18.5: Time of the exact route, all region messages and the port solve, against the port count, for every partition of the three networks, both axes logarithmic. The dashed line has slope three. The points of three networks and two kinds of cut lie on one curve.

The case studies of Chapter 24 measured the compression ratio $C_r = |E_r|/|S_r|$ on the same three networks, with separator values distinguished at the precision the physics gives. On the Reliability Test System and the IEEE 118-bus system it took physical, spectral and random cuts and enumerated each region's failure states to a residual mass of 10^{-2}, with a cap of 100 000 states per region. On the PEGASE system it took spectral cuts only, enumerated to a residual target of 10^{-1} with a cap of 5 000 states, and the cap binds: the enumerated states leave a residual mass of 0.97 to 1.00. Where a region stops at its cap, its ratio is censored, measured on a shallower enumeration than the target. Table 18.4 summarizes what the committed results contain, and its notes mark the censored rows.

The finding is sobering and instructive. Compression is large only where the cut is a single tie and the region has interchangeable parts: a replica of the Reliability Test System's Area 1 behind one tie compresses sixteenfold, because its many twin generating units produce the same tie flow whichever twin fails. It falls to three at two ties and to between one and six on the real networks. On the IEEE 118-bus system, whose generating units are all different, it is 1.02 on one region, where every failure state is its own separator value, and 1.63 on the other, measured at the state cap; the partition saves almost nothing on the enumeration route. The study's own gate, a median compression of ten at three ports or fewer on two grids, was not met, and its verdict is recorded as unfavorable. What the partition does predict, even there, is which cut is cheapest: on three of the four grids the partition with the fewest ports was also the one that measured cheapest, and on the fourth the two differed by a refinement pass.

This is the honest answer to Chapter 15's question of how large a region can be before enumeration by region stops working. It works when the region's failure states collapse onto few port values, which is a property of the physics inside the region, twin units, radial feeders, symmetric layouts, and not of the cut alone. Where the regions are heterogeneous, the port values are as many as the states, and the route through the ports is a reorganization of the enumeration rather than a reduction of it. The Gaussian route has no such condition, because a Gaussian message is a fixed-size object whatever the region contains, and this is the strongest practical argument for the Laplace approximation of Chapter 7 within regions wherever it is valid.

Table 18.4: Separator compression measured by a case study of Part VI: enumerated region failure states per distinct separator value, for the physical, spectral or refined spectral (spectral+KL) cut of each network, with the boundary dimension of the region measured. Ranges run over the regions of the cut. The last column is the partition the study found cheapest by measurement against the one the port count alone predicted.

network	cut	k_r	C_r	best: measured / predicted
RTS Area 1, two replicas, one tie	replica	1	15.9	–
RTS Area 1, two replicas, two ties	replica	2	3.4	–
RTS-GMLC, three areas	areas	2–4	3.7–5.7	areas / areas
RTS Area 1 (24 buses)	spectral	3–4	1.8–2.5	spectral / spectral
RTS Area 1 (24 buses)	spectral+KL	3–4	1.6–3.3	–
IEEE 118	spectral	8	1.02, 1.63[b]	spectral+KL / spectral
PEGASE 1354	spectral	13–17	1.5–2.0[a]	spectral / spectral

[a]Enumerated to a residual target of 10^{-1} with a cap of 5 000 states per region; the cap binds, with residual mass 0.97 and 1.00, so both ratios are censored.
[b]The second region stopped at the cap of 100 000 states with residual mass 0.032, so its ratio is censored. The spectral cuts of the full RTS-GMLC, not tabulated, stop at the same cap with residuals 0.023 and 0.026.

What the discrete message costs. When the regions are discrete the message is a histogram: one entry per distinct separator value, carrying the total probability of the region states that produce it and, for each consequence of interest, their contribution. Combining two regions is then a sum over pairs of entries, $|S_r| \times |S_s|$ of them, and the physics is evaluated once per region state, not once per pair, because a region state's flows are fixed by its own separator value and the other region's. The two-replica system with one tie, dispatched at peak load, shows the arithmetic at a residual of 10^{-2} per region. Each region enumerates 11 336 states, which fall on 712 separator values, so one table over pairs of separator values has $712^2 = 506\,944$ entries; that is the interface count of Table 18.4. The two-region pass builds one such table for each region, 1 013 888 interface pairs in all, with 21 916 physics evaluations. Its bracket on the probability of any overload, $[0.664, 0.684]$, has the joint residual mass 0.020 as its width and contains the Monte Carlo reference 0.679. The global enumeration of the same system, with 100 000 states, reached $[0.58, 0.69]$. It needs five million states to reach a residual of 0.019, about the two-region residual, which the regional route reaches with 22 672 enumerated states, some 220 times fewer. The difference is ordering: the global enumeration must order 124 components' failure sets by probability, while the regional one orders sixty-two of them twice and takes the product. Where the ports compress, the pairs are far fewer than the states; where they do not, the pairs are the states, and only the ordering advantage remains.

Where the condition holds, the same studies show what it buys. Table 18.5 reads their results for chains of Area 1 replicas joined by single ties, whose region graph is a path and whose messages therefore combine exactly on a one-dimensional lattice of tie flows. The joint state count grows from 10^8 for two regions to 10^{24} for six, beyond any enumeration; the region-tree pass brackets the probability of any overload on the enumerated mass to a width of two to three thousandths in minutes. With the residual mass added to its upper end, the bracket contains the Monte Carlo reference at every length; without it, the reference lies above the bracket for three regions or more. These chains are dispatched pro rata at sixty percent load, so their two-region member is not the peak-dispatch system of the previous paragraph: its probability of

Table 18.5: Region trees for chains of R single-tie regions, read from a case study of Part VI: the joint state count, the certified bracket on the probability of any overload before the residual mass is added, and the reference. At $R = 2$ the reference is the exact value on the same enumerated mass. Wall-clock is that study's last refinement pass.

R	components	joint states	bracket	reference	pass time
2	124	$10^{8.1}$	$[0.1207, 0.1227]$	0.1227 (exact)	7 s
3	186	$10^{12.2}$	$[0.1000, 0.1023]$	0.113 (MC)	63 s
4	248	$10^{16.2}$	$[0.0765, 0.0793]$	0.088 (MC)	230 s
6	372	$10^{24.3}$	$[0.0452, 0.0476]$	0.056 (MC)	511 s

any overload on the enumerated mass is 0.1227, where the peak system's is near 0.68.

Two remarks on that table. The Monte Carlo references at $R \geq 3$ lie above the bracket, by about one percentage point, 0.011, 0.009 and 0.008; the bracket is on the enumerated mass, and the un-enumerated residual, 10^{-2} per region and hence up to six percent for six regions, is what the reference contains and the bracket does not. The study reports the bracket with the residual added, 0.132, 0.119 and 0.106, which does contain the reference in each case. And the wall-clock grows with R although the per-region work does not, because the lattice on which the messages combine widens with the number of regions; the price of combining exactly is paid on the separator, as everywhere in this chapter.

18.6 The second domain

The same partition machinery, separators, exact two-region messages, certified brackets, was run in Chapter 24 on a chain of data halls modeled as a steady thermal network: racks with heat loads, cooling units with capacity, plenum and wall paths with conductance and a heat-flow limit. The domain entered through a forty-line adapter, conductance for susceptance, heat for power, a cooling unit for a generator, and the probability layer never saw a room. Table 18.6 summarizes the two systems. Two halls joined by one wall path, twelve failing components and 4 096 joint states, gave the probabilities, of any overload 0.15773 and of islanding, equal to brute force to within 4×10^{-16}, and the expected severity, 1 754 W, to within 8.9×10^{-12} W, in 13 milliseconds against 211 for the brute force. Three halls with two wall paths per interface, twenty-eight components and $10^{4.7}$ joint states, gave a bracket of $[0.011, 0.025]$ on the probability of any overload. Its lower end is the largest single hall's overload probability; its upper end is the union bound over hall and wall events with the joint residual mass, 0.003, added. It contains the Monte Carlo reference of 0.01275, 255 overloads in 20 000 draws, and was computed in a fifth of a second against two thirds of a second for twenty thousand draws, and, what a Monte Carlo cannot, a certified per-path bracket that includes the walls themselves, which on this system do overload when a cooling unit is lost and heat is pushed to the neighbors.

That the collector substations of Table 18.1 have elimination width two, and that the data halls compress and combine exactly on their walls, are the same fact: systems assembled from repeated units, in either domain, are hierarchies with narrow interfaces, and every route through a partition of them is cheap. The transmission networks are harder on every count, wider, with more ports per region and with heterogeneous regions that do not compress, and they are still far from the bounds. Between the two lie most physical systems.

Table 18.6: The data-hall chain, read from a case study of Part VI: the two-hall system solved by two-region messages against brute force, and the three-hall system bracketed by messages against Monte Carlo. The three-hall bracket includes the residual mass: it runs from the largest single hall's overload probability to the union bound over hall and wall events plus the joint residual.

	two halls, one wall	three halls, two walls per interface
components / joint states	12 / 4 096	28 / $10^{4.7}$
ports per hall	1	2
probability of any overload	0.15773 (messages)	$[0.011, 0.025]$ (bracket)
reference	0.15773 (brute force)	0.01275 (Monte Carlo, 20 000 draws)
agreement, probabilities	4×10^{-16}	reference inside the bracket
agreement, expected severity	8.9×10^{-12} W	–
time	13 ms against 211 ms	0.2 s against 0.66 s

18.7 In practice

Measure the width before partitioning. A minimum-degree ordering costs nothing and gives the elimination width. If it is small, and on physical networks it is, exact inference on the whole is cheap and a partition is for reuse, for hierarchy and for enumeration, not for tractability.

Take the physical cut when there is one, the spectral cut when there is not. Areas, feeders, buildings and rows are cuts with few ports by design. Without labels, the Fiedler vector of the weighted Laplacian finds cuts of the same quality on a meshed network; on a hub-and-spoke system it does not, because it insists on balance, and the physical cut, unbalanced, is the right one. Never partition by anything that ignores the graph, and never require balance that the physics does not have.

Count the ports and believe them. The boundary dimension predicts the port solve, the message sizes, the convergence of propagation, the exchanges of co-simulation and, where regions compress, the enumeration. A candidate partition with more ports is worse on every route.

Solve the port system; do not iterate on it. Scalar propagation on the ports fails. Block propagation is for a handful of regions. Co-simulation converges but pays a regional solve per exchange, and the number of exchanges is set by the ties against the regions. The message and the port factorization do the work once.

Expect compression only where the region is uniform. A region behind one tie with interchangeable parts compresses its failure states by an order of magnitude; a heterogeneous region does not. Check the ratio on the region before building the enumeration around it, and use a Gaussian message within the region wherever the Laplace approximation holds.

18.8 What is missing

Part V has treated time, hierarchy and scale as the three directions along which a model grows, and found in each the same structure: a separator, a message, a cost set by the separator's size. What the part has not done is run the whole machine on a whole problem from the operator's

question to the answer, with every choice of Chapters 1 to 18 made and defended against a naive baseline. That is Part VI, one chapter per study, each read cold, each with its numbers rendered from the committed results that the two tables of this chapter have already drawn on.

Notes and further reading

Elimination width, treewidth and the greedy orderings are Chapter 5's, following [3, 78, 99]; nested dissection is George's [25] and the planar separator theorem Lipton and Tarjan's [54]. Spectral bisection is Fiedler's vector [24] as Pothen, Simon and Liou used it for sparse-matrix orderings [74]; Kernighan–Lin refinement is [45], and multilevel partitioners such as METIS [41] combine the two at scale. Block splittings and their convergence for M-matrices are treated in the domain-decomposition literature [92]; co-simulation as the exchange of interface values between subsystem solvers, and the conditions under which it converges or diverges, is surveyed by Gomes and co-authors [26]. Walk-summability is Malioutov, Johnson and Willsky's [59]; the block analogue ρ_B is the chapter's own measurement device and is offered as a sufficient condition, not as a theorem. The measurements of elimination width, boundary dimension, region-level propagation and co-simulation on the three networks are the book's; the separator compression, the region trees and the thermal chain are read from the case studies of Part VI, which state their own assumptions, in particular that reliability data for the IEEE 118 and PEGASE systems are class rates borrowed from the Reliability Test System.

Two water-network estimators partition for the reasons of §18.1. Han, Eguchi, Mehrotra and Venkatasubramanian [33] cut a large damaged network at its articulation points so that components can be estimated independently under disaster conditions, accepting the coupling through the bridge flows that Chapter 5's Notes describe. Irofti, Romero-Ben, Stoican and Puig [36] cut in time rather than in space: a leak-free estimation over a window, then a localization graph on its residuals, each solved once, which bounds the cost per scenario on their 268-junction benchmark at the price of passing point estimates rather than posteriors between the two stages. The incremental re-elimination of Dellaert and Kaess [17], which re-solves only the part of the Bayes tree that new factors touch, is the robotics form of §18.3's question of where the cut should fall when the graph grows.

Summary

- The elimination width of a partition ordering is at most the worst region's width plus its ports, or the port count; every route through a partition pays in the boundary dimension.
- Measured widths: 5, 4 and 12 on transmission networks of 73, 118 and 1 354 buses, 2 on a collector substation at any size. Exact inference on physical networks is cheap.
- A physical or spectral cut of the 1 354-bus network in two has 30 boundary buses; a random cut has 950. Random regions have almost no interior.
- Scalar propagation on the ports is not walk-summable on any partition. Block propagation converges among two to four large regions and fails beyond. Co-simulation converges at the rate $g_t/\sqrt{(g_a+g_t)(g_b+g_t)}$ and needs 22 to 55 000 exchanges where the message needs one.
- Enumeration by region compresses only where the region is uniform: sixteenfold behind one tie on a system of twin units, hardly at all on the IEEE 118-bus system. Where it compresses, chains of six regions with 10^{24} joint states are bracketed to a few thousandths on the enumerated mass.

- The same machinery on a chain of data halls reproduces brute force to 4×10^{-16} in probability and 8.9×10^{-12} W in expected severity, sixteen times faster.

Exercises

Exercise 18.1. Derive equation (18.3): write the block Jacobi iteration matrix of the two-port system, find its eigenvalues, and show that the block Gauss–Seidel matrix has the square of the Jacobi radius. Then evaluate both for a tie of conductance 10 joining regions of equivalent conductance 2 and 5.

Exercise 18.2. Prove that the elimination width of a graph is at least the size of its largest clique minus one, and use Proposition 18.1 to show that a partition of the collector substation into feeders, with the station buses as ports, has an elimination width of at most three whatever the number of feeders and units. Compare with Table 18.1.

Exercise 18.3. Take the IEEE 118-bus network's spectral cut into two and compute the block radius ρ_B of the two-region port system with the regions' own port blocks as normalizers. It is 1.00 in Table 18.3, yet propagation converges in one sweep. Explain why a two-region system converges regardless of ρ_B, and what ρ_B measures there.

Exercise 18.4. A region behind a single tie contains m identical generating units, each failed independently with probability q, and the tie flow depends only on how many are failed. Compute the number of enumerated states with probability above ϵ and the number of distinct separator values, and hence the compression ratio C as a function of m, q and ϵ. Then replace the units by m units of distinct sizes and show that $C = 1$.

Solutions to selected exercises

Exercise 18.1. With D the block diagonal $\operatorname{diag}(g_a+g_t,\ g_b+g_t)$ and $E = A - D$ the off-diagonal part, block Jacobi iterates $\phi^{(k+1)} = D^{-1}(p - E\phi^{(k)})$, whose iteration matrix is

$$M_J = -D^{-1}E = \begin{pmatrix} 0 & \dfrac{g_t}{g_a+g_t} \\ \dfrac{g_t}{g_b+g_t} & 0 \end{pmatrix}, \qquad \lambda^2 = \frac{g_t^2}{(g_a+g_t)(g_b+g_t)},$$

so $\rho_J = g_t/\sqrt{(g_a+g_t)(g_b+g_t)}$. Gauss–Seidel updates a then b with the new ϕ_a: $M_{GS} = -(D+L)^{-1}U$ with L the strictly lower and U the strictly upper part, which for a 2×2 system gives $M_{GS} = \left(\begin{smallmatrix} 0 & \rho_{ab} \\ 0 & \rho_{ab}\rho_{ba} \end{smallmatrix}\right)$ with $\rho_{ab} = g_t/(g_a+g_t)$ and $\rho_{ba} = g_t/(g_b+g_t)$; its nonzero eigenvalue is $\rho_{ab}\rho_{ba} = \rho_J^2$. For $g_t = 10$, $g_a = 2$, $g_b = 5$: $\rho_J = 10/\sqrt{12 \times 15} = 0.745$ and $\rho_{GS} = 0.556$, so six digits take $\log 10^{-6} / \log 0.745 = 47$ simultaneous or 24 sequential exchanges.

Exercise 18.2. In any elimination ordering, consider the first vertex of a largest clique Q to be eliminated. At that moment every other vertex of Q is still present and adjacent to it, since edges are never removed, so its front has at least $|Q| - 1$ vertices, and the width is at least $|Q| - 1$. For the station, take each feeder as a region with interior its two sections and its units' nodes, and the two station buses P and A as the port set, $|S| = 2$. Within a feeder, eliminate each unit's chain tap-to-export first: the tap node has neighbors C and T, the terminal then has C

and E, the export node then has C and H, so w_r counts at most two interior neighbors, and each feeder node is adjacent to at most the two ports, so $w_r + k_r \leq 2 + 2 = 4$ is the proposition's bound; a sharper count notes that a unit node's front never contains both ports, giving 3. Then the two section buses each see $\{P, A\}$ and each other's eliminated fill, a front of three. The port phase is on two nodes. The width is at most three at any size, and Table 18.1 measures two, one better, because the minimum-degree ordering eliminates the feeders' sections before the tie bus and never forms the three-clique.

Problems from the studies

Each of these is stated, solved and committed beside the study it came from, with a script that regenerates its answer. The path is relative to `docs/terra_graph_engine/case_studies/`.

Pick the cut *the cost of a decomposition living on its boundary; a spectral cut against a random one; local refinement that finds nothing left to improve*
`foundation_power_flow/separator_compressibility/problems/pick_the_cut`

Part VI

Case studies

Chapter 19

Case studies

The chapters before this one taught the methods on small systems chosen so that every number could be checked by hand. The chapters of this part apply them to systems where no number can be checked by hand, and report what happened. Each is a complete application of the book to one question on one system, and each was run as a study in its own right, with its code and its results committed to the repository that accompanies the book.

19.1 What a case study is here

A case study in this part is a question, a system, a model and a run. The question is one an engineer or an operator would ask. The system is a standard test network or a constructed plant, described fully enough to rebuild. The model is a factor graph of the kinds in Part I, with the uncertainty placed where Part II puts it. The run is a script that produces a results file, and every number in a chapter of this part is read from that file by a script in the book's `scripts` directory. Nothing is retyped.

Every study was run with TERRA, the author's implementation of the constructions of this book, and its plugins. The engine compiles a conservation graph into its residual rows, solves them by Newton's method, and lifts the same rows into the MAP and Laplace inference of Part III. The plugins carry what is specific to a domain or a method: power-flow and dispatch models, reliability enumeration, the census and its surrogate. Appendix C describes both and maps each construction of the book to the module that implements it. In the chapters of this part, "the engine" means TERRA. The studies count cost in *engine solves*, one evaluation of the physics at one setting of the uncertain inputs. In the three-line, AC and bad-data studies that is a Newton or MAP solve of a compiled model. In the census and $N - k$ studies it is a DC minimum-load-shed linear program solved by the power-flow plugin. In the partitioning studies it is a region-local linear network flow under a fixed dispatch rule. Neither of those passes through the Newton solver.

The chapters share one order. Each states the question, describes the system, draws the model as a factor graph, lists the method with a pointer to the chapter that justifies each step, gives the results, says what surprised, and ends with how to reproduce the run. The section on what surprised is not decoration. Several of these studies found that a method did less than expected, or that a bound was not a bound, and those findings are reported with the same care as the ones that went well.

19.2 The studies

- Chapter 20, *Three routes on three lines*: enumeration, Monte Carlo and factorized inference costed in physics solves on a problem with a closed form, the study behind Chapter 10.
- Chapter 21, *Nine questions from one census*: a census of solves with gradients on a 73-bus network with fourteen uncertain ratings answers nine planning questions, and the cost of each against the standard sampling method.
- Chapter 22, *What survives on AC*: the same nine questions on an AC model, where the convexity the certificates rest on is no longer available, and which of them keep their guarantees.
- Chapter 23, *The worst k outages*: a published interdiction benchmark replicated, its optimum proved by enumeration for one and two outages, and the outage conditioned on evidence about the weather and the fuel supply.
- Chapter 24, *Partitioning in practice*: measured compression at the ports of real networks, region trees along chains of regions, and the same machinery on a chain of data halls.
- Chapter 25, *Bad data on fourteen buses*: the diagnostics of Chapter 12 on an AC network whose loads are known only by forecast, including what a meter that no other meter checks lets through.

The studies are in the power domain, and the partitioning chapter also runs a thermal one. The first five follow the parts of the book they draw on most: Part III first, then Part IV, then Part V. The last returns to diagnosis, on a system large enough for its failures to show.

19.3 Reading a case study

Each chapter can be read on its own after the chapters it points to. A reader who wants to rerun a study needs the repository, a Python environment with the engine of Appendix C, and the command in the chapter's last section. Most studies run in minutes on a workstation; the chapters say where one takes longer. A rerun reproduces the committed results file, and the book's script for the chapter then reproduces every number in the chapter from it.

Chapter 20

Three routes on three lines

This study asks what it costs, in solves of the physics, to answer two questions about a small network whose line ratings are uncertain: how much load is curtailed on average, and how much of the variation in curtailment is owed to each line. It answers both three ways, with one engine solve as the only primitive: by a product grid over the ratings, by Monte Carlo sampling, and by enumeration on a cut that the compiled model exposes. Chapter 10 used the study's results to make the book's comparison of these routes and derived the error laws behind them. This chapter presents the study as a study: the model as TERRA holds it, the design of the runs, the checks on the engine and on the cut, and the results that Chapter 10 did not need, including several that do not flatter the grid. It belongs in the book because it is the smallest system on which the elements of Parts II and III appear in one computation: balance rows, closures, a boundary, priors on parameters, a separator, and a question that needs conditional expectations.

Every number in this chapter is read from the study's committed results by the script `ch20_three_line.py`, which prints each one and writes the generated tables and figures.

20.1 The question

A program that rates lines dynamically has to decide which conductors to instrument. Two quantities bear on that decision. The first is the expected curtailment $\mathbb{E}[G]$ when the ratings fall short of the demand. The second is the share of $\mathrm{Var}[G]$ attributable to each rating, which orders the lines by how much a measurement of each would tell. Both are functionals of the distribution of the network's response to uncertain inputs.

The study poses three questions about that computation.

1. How many engine solves does each route need to reach a given error in $\mathbb{E}[G]$?
2. How many further solves does the attribution cost, once the mean is in hand?
3. Does the compiled model expose a cut of its own structure without being told, and does enumeration on that cut reproduce the grid exactly?

The currency is the solve, for the reason given in §10.1: on a network of operational size, the solve is what costs. The study makes no claim about the speed of any one solve. It counts them.

20.2 The system

Two lines, A and C, run in parallel from a source to a middle bus. A third line, B, runs from the middle bus to a load of 160 megawatts. The ratings are independent and uniform: A and

Table 20.1: The rows of the compiled three-line model. Each unknown is listed with the row that determines it and the variables that row reads, as recorded in the dependency graph of the study's results. The last column says whether the unknown lies upstream of the curtailment G.

engine name	symbol	row	relation	reaches G
`cap_pair`	κ	closure	$\kappa = A + C$	yes
`req_mid`	ρ	closure	$\rho = \min(D, B)$	yes
`f_B`	f_B	closure on line B	$f_B = \min(\rho, \kappa)$	yes
`served`	δ	balance at the load bus	$f_B - \delta = 0$	yes
`G`	G	closure	$G = D - \delta$	target
`f_A`	f_A	closure on line A	$f_A = f_B\, A/(A + C)$	no
`f_C`	f_C	balance at the middle bus	$f_A + f_C - f_B = 0$	no

C on $[70, 120]$ megawatts, B on $[140, 180]$. The curtailment is $G = (160 - \min(A + C, B))^+$, equation (10.1). Its moments have closed forms (Exercise 10.1): $\mathbb{E}[G] = 16/3$ megawatts, $\mathrm{Var}[G] = 41.956$ square megawatts and $\mathbb{P}(G = 0) = 0.46$. The Shapley shares of the variance have no short closed form. The study's reference values, 93.5976 percent for B and 3.2012 percent each for A and C, come from the partial-sum computation on a grid of 241 points per rating, evaluated with the closed-form G in place of the engine. Every route is scored against these values.

The network was chosen for two properties. It is small enough to have the closed form that scoring needs. And it is the smallest network on which the attribution is not trivial: A and C reach the load only through their sum, so neither matters alone, and an attribution has to decide how to split what they do together.

20.2.1 The model in TERRA

The study builds the network as a conservation graph in the sense of Chapter 1. It has four nodes. The source and the sink are reservoirs (§1.4); the middle bus and the load bus are zero-storage nodes balanced in the power domain. It has four edges, each with one power channel: line A and line C from the source to the middle bus, line B from the middle bus to the load, and a demand edge from the load to the sink. The four channels carry the flows f_A, f_C, f_B and the served power, written δ here.

Table 20.1 lists the seven rows that determine the seven unknowns. Five are closures and two are balances. The results file records the compiled model as 7 rows in 7 unknowns and well posed. The demand and the three ratings are substituted constants, the form that Table 2.2 gives a reservoir or a parameter before probability is attached.

Each closure has a plain physical reading. The requirement ρ at the middle bus is the demand $D = 160$, capped by what line B may carry. The flow on line B is that requirement, capped by what the parallel pair may deliver, so $f_B = \min(D, B, A + C)$. The load bus passes f_B on as served power, and the curtailment is the demand less what is served. Nothing in the model writes a positive part: the minimum with D inside ρ keeps δ at or below the demand, so G is never negative.

The two parallel lines share their flow in proportion to their ratings. Line A carries $f_A = f_B\, A/(A + C)$, and line C has no closure at all: its flow is whatever the balance at the middle bus leaves, $f_C = f_B - f_A$. The proportional split has a consequence the model never states. Since $f_B \le A + C$, each line carries at most its own rating, $f_A \le A$ and $f_C \le C$, without a limit

Table 20.2: The engine against the closed-form curtailment at the five spot-check triples of the study, ratings in megawatts. The error is the engine's value less the exact one, in units of 10^{-9} megawatts. Generated from the results file.

A	B	C	exact G	engine G	error [10^{-9} MW]
100	160	100	0	−0.000000004	−3.98
70	150	80	10	9.999999997	−3.32
120	145	110	15	14.999999997	−2.99
70	180	70	20	19.999999999	−1.01
90	141	95	19	18.999999997	−2.72

written on either.

20.2.2 One solve

Every estimate in the study goes through one function, shown here with its type conversions removed:

```
def curtailment(self, a, b, c):
    self.n_solves += 1
    cm = self.model.with_substitutions(
        {"rate_A": a, "rate_B": b, "rate_C": c})
    return tge.solve(cm).value("G")
```

The call rebinds the three substituted rating values and leaves the topology, the rows, their order and the registry of unknowns unchanged. It then solves the residual system $\mathbf{F}(\mathbf{z}) = 0$ by Newton's method from the same initial guess each time, and reads G from the solution. The counter is incremented once per call, and its final value is written to the results file. Every route therefore pays in the same unit, and no route can reuse a solve without the counter seeing it.

Before any estimate, the study checks the engine against the closed form at five rating triples. Table 20.2 gives the result. The largest error is 4.0×10^{-9} megawatts, nine orders of magnitude below any error the study measures. All five errors have the same sign: the engine's G sits a few parts in 10^9 below the exact value. That sign matters later (§20.6).

20.3 The model as a factor graph

Figure 20.1 draws the model as a factor graph in the sense of §2.3. Circles are the unknowns and the three ratings. Squares are the seven rows of Table 20.1, with the two balances shaded darker than the closures. Each rating carries a unary prior factor, its uniform distribution, which is how Chapter 3 turns a substituted parameter into an uncertain one. The demand is a constant inside the two factors that read it, drawn dashed as a reservoir is.

The graph has 20 nodes and 22 edges, so its cycle rank is 3. Every cycle passes through the split closure on line A. One runs A–sum–C–split–A, one runs through κ, the minimum on line B and f_B back to the split, and one closes through the balance at the middle bus.

The grey branch contributes nothing to G, and the hard rows say why. Its two unknowns, f_A and f_C, appear in no factor but the split closure and the middle balance. Given f_B, A and C, those two rows fix f_A and f_C uniquely, and the map from (f_A, f_C) to the two residuals is

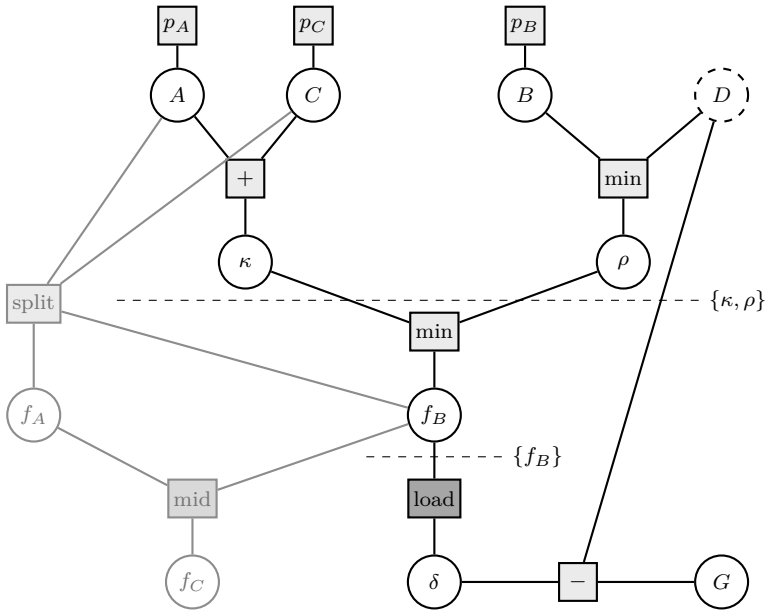


Figure 20.1: The three-line model as a factor graph. Circles: the three ratings and the seven unknowns (κ the capacity of the parallel pair, ρ the requirement at the middle bus, δ the served power); dashed circle: the demand D, a constant. Light squares: closures and the three priors; dark squares: the balances at the load bus and the middle bus. Grey: the flow split between the parallel lines, which never reaches G. The dashed lines mark two cuts of the ladder; the grey edge from the split closure to f_B crosses the upper one.

triangular with unit diagonal. Integrating the two hard rows over f_A and f_C, in the sense of §4.1, therefore gives one for every value of f_B, A and C. The branch is a constant factor and drops out. What remains has 16 nodes and 15 edges, cycle rank 0: a tree, running from the three priors through κ and ρ, the minimum, f_B, the load balance and δ to G.

On that tree the cuts of Table 10.3 are separators in the sense of Chapter 5. The pair $\{\kappa, \rho\}$ blocks every path from the ratings to G, and so does $\{f_B\}$. Before the branch is summed out, the pair is not a separator of the drawn graph: the path A–split–f_B runs around it, which is the grey edge crossing the upper dashed line. The study's dependency graph records the fact directly. It orients each row toward the unknown it determines, as in Proposition 2.1, and on that orientation neither f_A nor f_C is an ancestor of G. The orientation has no cycle, so every block of Proposition 2.1 has one row and the whole model is an order of substitutions. The engine solves it simultaneously anyway, as it would a model with loops.

20.4 Method

The study runs in two scripts. The first computes every estimate and the second reads the model's structure. The steps, each with the part of the book that justifies it, are these.

1. *Build and check the model.* The conservation graph of §20.2 is compiled once, following §1.2 and §1.3, and checked at five rating triples against the closed form (Table 20.2).

2. *Attach the priors.* The three ratings receive independent uniform priors (Chapter 3). The

demand stays a constant.

3. *Product grid.* For $n \in \{9, 17, 25, 33, 41\}$, each rating is discretized on n evenly spaced nodes across its range, with trapezoid weights: $h/(b-a)$ at interior nodes and half that at the two ends. The weight of a node triple is the product of its three weights. The grid solves the model once per triple, n^3 solves, and forms $\mathbb{E}[G]$, $\mathrm{Var}[G]$ and $\mathbb{P}(G = 0)$ as weighted sums. The last counts the nodes where the engine returns $G \le 10^{-9}$. The six conditional variances $v(u) = \mathrm{Var}(\mathbb{E}[G \mid X_u])$ are partial sums over the same table (Proposition 10.3), combined into Shapley shares with the weights 1/3, 1/6 and 1/3 for subsets of size 0, 1 and 2 (§14.3). The error of the rule is bounded by Proposition 10.1. Every n is odd, so $B = 160$ is always a node.

4. *Monte Carlo.* For $N \in \{300, 1{,}000, 3{,}000, 10{,}000\}$ and eight seeds, 20260906 to 20260913, the study draws N rating triples from the priors, solves each, and forms the same three sample moments. It records the root-mean-square error over the eight seeds of the mean and the variance against the closed form. The mean, variance and $\mathbb{P}(G = 0)$ stored beside those errors are the first seed's values, not averages. The error law is Proposition 10.2.

5. *Pick-freeze attribution.* For $N \in \{300, 1{,}000, 3{,}000\}$ and five seeds, 20260906 to 20260910, the study draws two independent sets of N triples and solves the first. For each of the six proper nonempty subsets u it builds a hybrid set, taking the columns in u from the first set and the rest from the second, and solves it. Then $v(u)$ is estimated as the mean of the product of paired curtailments less the squared mean (§10.5). This is $7N$ solves per seed. The study records the root-mean-square error of B's share over the five seeds, and the shares stored beside it are the first seed's.

6. *Read the structure.* A separate script with its own engine reads the dependency graph from the compiled model's declarations. A closure declares its inputs and its output, and a balanced node with exactly one closure-free incident flow determines that flow. The script marks as *explicit* the unknowns that closures alone compute from the ratings and the demand. Starting from $\{G\}$, it repeatedly replaces one variable by the variables that determine it, and keeps every set it reaches whose members are explicit, uncertain or constant. These sets are the *frontiers*, the cuts of Proposition 10.4. For each frontier and each grid it counts the distinct frontier values, rounded to nine decimals, which needs no solve. It then recomputes the $n = 17$ grid through one frontier, solving once per distinct value and reading the rest from a cache. The search is a routine of the study. The engine supplies the declarations it reads; it does not offer the search itself.

Table 20.3 gives the design as a budget. The main run made 391,030 engine solves: 5 spot checks, 126,125 on the five grids, 114,400 for Monte Carlo and 150,500 for pick-freeze. The structure script made 561 more. A solve of this seven-unknown model took about a millisecond of wall-clock time on every route, and the estimators of the main run took 7.8 minutes in all.

20.5 Results

Chapter 10 gives the errors of the mean and of B's share for both routes (Tables 10.1 and 10.2) and plots them against solves (Figure 10.2). This section shows what those tables do not: every grid quantity side by side, the probability of no curtailment, the cost ratio at equal accuracy, and the check of the cut.

Table 20.3: The experimental design as a solve budget. Solves are totals over all sizes and seeds. The last column is wall-clock time per solve, the range over the sizes of each route. The cut check ran in a separate script and is not in the main run's total. Generated from the results file.

route	sizes	seeds	solves	ms per solve
spot checks	five rating triples	—	5	—
product grid	$n = 9, 17, 25, 33, 41$, n^3 each	—	126,125	0.85–1.08
Monte Carlo	$N = 300, 1{,}000, 3{,}000, 10{,}000$	8	114,400	0.85–1.19
pick-freeze	$N = 300, 1{,}000, 3{,}000$, $7N$ each	5	150,500	0.96–1.47
total, main run			391,030	
cut check, own run	$n = 17$ through $(A{+}C, B)$	—	561	—

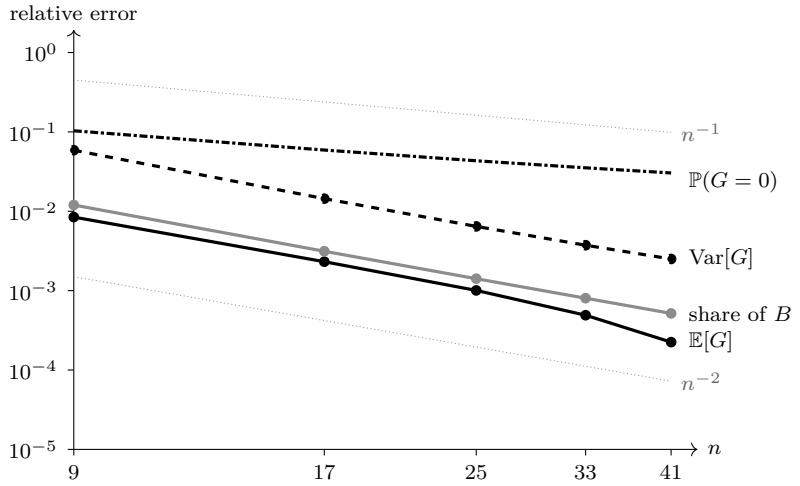


Figure 20.2: Relative error of four product-grid quantities against the number of nodes per rating, n: the mean, the variance and line B's Shapley share, each against its reference, and the probability of no curtailment. Dotted: guides of slope -2 and -1. Share and variance follow n^{-2}; the mean follows it and then steepens at the last grid; the probability falls more slowly than n^{-1}. Generated from the results file.

20.5.1 The grid, quantity by quantity

Figure 20.2 plots the relative error of four grid quantities against the grid size n, on logarithmic axes. Each error is divided by its reference value so the four fit one scale. The two dotted guides have slopes -2 and -1.

The share of B follows n^{-2} at every step. Measured against the step h, its local slopes between consecutive grids are 1.94, 1.96, 1.97 and 1.98. The variance follows the same law, with local slopes 2.03, 1.98, 1.90 and 1.78. Both are what Proposition 10.1 predicts for a piecewise-smooth integrand. The mean follows h^2 over the first two steps, with slopes 1.87 and 2.05, then steepens to 2.50 and 3.50. A single fit over the five grids gives a slope of -0.78 in solves, and §20.6 explains why that figure overstates the method. The probability of no curtailment converges slowest. Its local slopes are 0.81, 0.76, 0.71 and 0.67, and every grid value lies above the exact 0.46: from 0.5076 at $n = 9$ to 0.4740 at $n = 41$.

The grid also keeps a symmetry that the problem has. Lines A and C are exchangeable, and on every grid their Shapley shares agree to within 7.1×10^{-14} percentage points.

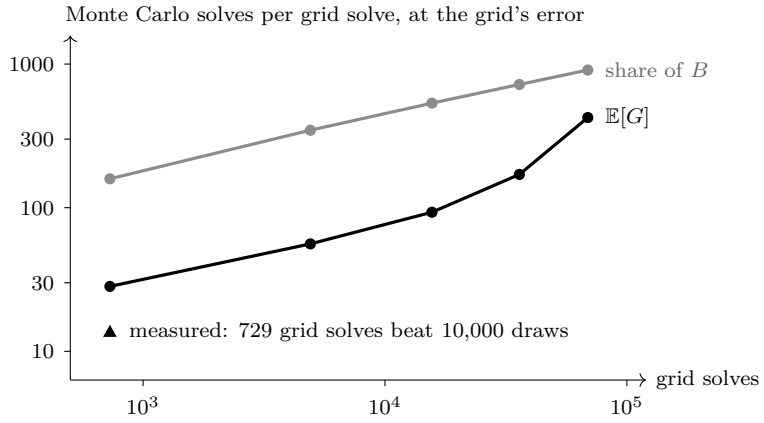


Figure 20.3: Monte Carlo solves needed to match each product grid's error, per grid solve, against the grid's solves. Black: the mean, by the $\sigma/\sqrt{N}$ law. Grey: line B's Shapley share, by pick-freeze at $7N$ solves, with the constant taken from the $N = 3{,}000$ runs. Triangle: the measured factor at $n = 9$, where 729 grid solves beat the 10,000-draw run. Generated from the results file.

20.5.2 Sampling

The Monte Carlo errors of the mean follow $\sigma/\sqrt{N}$ as closely as eight seeds allow (§10.4). The quantity that Chapter 10 did not score is $\mathbb{P}(G = 0)$. A proportion estimated from N independent draws has standard error $\sqrt{p(1-p)/N}$. At $N = 10{,}000$ and $p = 0.46$ that is 0.0050, and the first seed's value, 0.4578, is 0.0022 from the truth. The grid at $n = 41$, with 68,921 solves, is 0.0140 from it. By the same binomial law, about 1,271 draws would match the grid's error at $n = 41$, and about 110 would match it at $n = 9$. On this quantity the grid needs 54 times the solves of sampling at $n = 41$, and the gap widens as n grows.

Pick-freeze does not keep the symmetry between A and C. Its first seed at $N = 3{,}000$, 21,000 solves, gives A a share of 5.56 percent and C a share of 6.52, against 3.20 each. Every conditional expectation comes from its own hybrid sample, and the hybrid samples for $\{A\}$ and $\{C\}$ carry independent noise.

20.5.3 Equal accuracy

Figure 20.3 asks how many solves Monte Carlo needs to match each grid, and divides by the grid's own solves. For the mean the count comes from the $\sigma/\sqrt{N}$ law with the closed-form variance. For B's share there is no such law with a known constant, so the constant is taken from the largest pick-freeze run, where the root-mean-square error times the square root of the solves is 382. The smaller runs give 323 and 260. A constant lower by up to a third would lower this curve by up to a half, since the solves needed scale with its square. The triangle is the one measured comparison: the 729 solves of the $n = 9$ grid beat 10,000 draws, a factor of 13.7.

For the mean the ratio grows from 28 at $n = 9$ to 424 at $n = 41$. For the share it grows from 159 to 908. Both grow because the grid's error falls faster in solves than sampling's does, and the more accuracy is asked for, the more that difference is worth. The measured 13.7 sits below the law's 28 for a plain reason: the 10,000-draw run is more accurate than the 729-solve grid, not equal to it, so the measured factor is a lower bound on the equal-accuracy ratio. Chapter 10

explains why none of this is the book's own advantage: it is quadrature in three dimensions.

20.5.4 The cut ladder

The structure script found five frontiers, listed with their distinct solves in Table 10.3. At $n = 41$ they reduce the 68,921 grid solves by factors of 2,089 for $\{f_B\}$, 40.5 for $\{\kappa, \rho\}$, 20.8 for $\{\kappa, B\}$ and 2.0 for $\{A, C, \rho\}$. The fifth frontier is the ratings themselves.

The verification recomputed the $n = 17$ grid through the frontier $\{\kappa, B\}$, the capacity of the pair and the rating of line B. It solved 561 distinct frontier values in place of 4,913 grid triples, and the two means agree to 9.7×10^{-14} megawatts. Both read 5.345687862758 to twelve decimals. This is Proposition 10.4 run on the engine: the grid, its weights and its answer are unchanged, and only the number of solves differs.

The one-quantity frontier needs fewer solves than there are nodes per rating, and the reason can be read off the model. The flow f_B is $\min(160, B, A + C)$, so its values on the grid are the grid values of B at or below 160, together with those of $A + C$ at or below 160. At $n = 9$ there are 5 of the first and 4 of the second, with 1 shared, which makes 8. At $n = 41$ there are 21 and 17 with 5 shared, which makes 33. The script confirms this union count against the recorded ladder at all five grid sizes.

20.6 What surprised

The mean's last grid is lucky. The mean steepens at $n = 41$ because of where that grid puts its nodes. The value $B = 160$ is a node on every grid, since every n is odd. The kink at $A + C = 160$ falls on a node only when $0.4(n - 1)$ is an integer, which among the five grids happens only at $n = 41$. In the proof of Proposition 10.1 a kink at offset s in its cell costs $\frac{1}{2} j s (h - s)$, which vanishes at $s = 0$. At $n = 41$, two of the mean integrand's three kinks therefore cost nothing, and the error drops further than h^2 explains. The variance gets no such gain. Its integrand G^2 is piecewise quadratic, so the curvature term $(b - a)Mh^2/12$ remains wherever the kinks fall. The fitted exponent of 0.78 in solves is thus a property of this sequence of grids, not of the method. The rate the theory guarantees, and the rate the variance and the share show, is $2/d = 2/3$ in solves. A grid whose nodes happen to meet the kinks looks better than the method is.

On a probability, the grid loses. The probability of no curtailment is the expectation of an indicator, and an indicator jumps rather than bends. On a cell containing a jump, the trapezoid rule errs by an amount of order h, not h^2, and the measured slopes fall below even that. There is also a bias. A node on the edge of the zero region counts with its full weight toward the zero set, so every grid value lies above 0.46. At 68,921 solves the grid is matched by about 1,271 draws. This is the three-line counterpart of the tail probabilities of §10.7, and the remedy is the same: integrate one input in closed form so the jump becomes a kink.

The zero test is tighter than the engine. The grid counts a node as uncurtailed when the engine returns $G \le 10^{-9}$, but the spot checks show engine errors up to 4.0×10^{-9}. The count works because all five spot-check errors are negative: at a triple with no curtailment, the engine returns a slightly negative G, which passes the test. A solve that erred in the other direction

would move a node out of the zero set. Five spot checks cannot certify the sign everywhere, so the threshold sits below the accuracy the study has demonstrated.

The verified cut is not the cheapest. The search orders frontiers by size and then by name, and the verification takes the second entry. That entry is $\{\kappa, B\}$, at 561 solves for $n = 17$. The cheaper two-quantity frontier $\{\kappa, \rho\}$ needs 297, and $\{f_B\}$ needs 14. Both were counted, but neither was run. Their exactness rests on Proposition 10.4, not on a computation in the study.

The smallest frontier is set by the search's rule. The served power δ equals f_B and sits one row from G. But a balance determines it, and the search evaluates without a solve only what closures compute, so $\{\delta\}$ is not a frontier and $\{f_B\}$ is. The 2,089-fold reduction of $\{f_B\}$ says more about this model than about the method: every quantity upstream of the load bus is explicit. In a model whose intermediate quantities are fixed by balances, a grid point's frontier values are unknown until it is solved, and the ladder shortens to the frontiers that closures alone can evaluate.

Two cuts are separators only after a sum. The two-quantity frontiers do not separate the ratings from G in the factor graph as drawn (Figure 20.1). They separate them only after the flow-split branch is summed out. Here that sum is free and exact, because the branch is a pair of hard rows with a unit Jacobian. The dependency graph sees the cut at once because it records which row determines which unknown, and the undirected factor graph does not.

Each route pays its full cost. Under a given seed, the base set of the pick-freeze run is the same draw set as the Monte Carlo run of the same size. Both are generated by the same stream in the same order. The study solves it again rather than share it, so the 150,500 pick-freeze solves include sets already solved for the mean. This charges each route what it would cost on its own.

20.7 Reproduction

From the repository root, with the engine on the path, the study runs in three commands:

```
<study> = docs/terra_graph_engine/case_studies/
          foundation_power_flow/terra_vs_monte_carlo_three_line
PYTHONPATH=src python <study>/run.py
PYTHONPATH=src python <study>/run_structure.py
PYTHONPATH=src python <study>/make_report.py --pdf
```

The first command runs the spot checks, the five grids, the Monte Carlo and the pick-freeze estimators from seed 20260906, and writes `results.json`. It holds the configuration, the closed form, the reference shares, the compiled model's row and unknown counts, the spot checks, one record per grid, per Monte Carlo size and per pick-freeze size, and the total engine solves. The second command reads that file, reads the structure of a freshly compiled model, and appends a `structure` block with the dependency graph, the frontier ladder and the $n = 17$ verification. The third renders the study's own report from the same file.

The book's script, `scripts/ch20_three_line.py`, run from the book's directory, reads `results.json` and one number from the study's report: the 241 points per rating of the reference

grid. It prints every number in this chapter, checks that the solve budget sums to the recorded total and that the one-quantity cut's union count matches the recorded ladder, and writes Tables 20.2 and 20.3 and Figures 20.2 and 20.3. It runs no engine solve.

Chapter 21

Nine questions from one census

A planner who has estimated the expected load shed of a stressed network has done the cheap part of the work. The decisions come afterwards: which line to uprate, which ratings to measure, which to ignore, what changes when the forecast changes, how heavy the tail is and how bad the worst case can be. Sampling answers each of these with a sample designed for it, and each sample costs thousands of network solves. This chapter runs the construction of Chapter 14 on a real test network and asks all of these questions of one census of solves that return their gradients. It counts what every answer costs by both routes, scores every answer against a large independent reference, and reports where the census route is better, where it is worse, and why. Every number in the chapter is read from the committed results of two case studies by the script `ch21_census.py` in the book's `scripts` directory, which prints each one; no number was typed by hand.

21.1 The question

Transmission line ratings depend on the weather and are uncertain at planning time. When they fall short of what the dispatch needs, load is shed. The shed G is therefore a random variable, and a planner wants more than its mean. Nine questions are asked of it here, each with the decision it informs:

Q1. the shadow price $\partial\mathbb{E}[G]/\partial r_i$ of each line's rating: which line to uprate first;
Q2. the total Sobol index $S_{T,i}$ of each rating: which ratings drive the uncertainty;
Q3. a guaranteed ceiling on $S_{T,i}$: which ratings can be ignored;
Q4. the variance $\mathrm{Var}(\mathbb{E}[G \mid r_i])$ removed by learning one rating exactly: the value of a measurement;
Q5. $\mathbb{E}[G]$ under a hotter season's belief: re-planning when the forecast moves;
Q6. $\mathbb{E}[G]$ when ratings share their weather: correlated inputs;
Q7. the curve $\mathbb{E}[G \mid r_i = r]$ across one line's range: what an uprate buys;
Q8. $\mathbb{P}(G > g)$, value-at-risk and conditional value-at-risk: the tail;
Q9. $\max G$ over the box of possible ratings: the worst case.

The question of the study is not the answers themselves but their price. For each question there is a standard sampling method a practitioner would use. The census route pays once for a census and answers all nine from it. The comparison is in engine solves, one solution of the network model for one vector of ratings, because that is the count that transfers between machines, as §10.10 argued. And since a cheap wrong answer is worth nothing, every answer of both routes is

scored against a reference computed independently and at a larger budget.

21.2 The system

The network is the RTS-GMLC test system, the one Chapter 15 used for combinations of outages. It has 73 buses, 120 branches (lines and transformers), 96 generators with 9,076 MW of capacity and 51 loads totalling 8,550 MW, on a 100 MVA base. Every branch rating is scaled to 0.5 of its nominal value, so that the network is stressed: at the scaled ratings the best dispatch must already shed 154.06 MW.

One solve at the scaled ratings finds the branches whose flow bound is active. There are $K = 14$ of them, branches 11, 25, 30, 48, 53, 58, 64, 86, 89, 91, 96, 100, 102 and 107, with scaled ratings of 87.5, 200 or 250 MW. These are the uncertain inputs; the other 106 ratings are held at their scaled values. Under the base belief, belief 1, each uncertain rating is uniform on $[0.85, 1.15]$ times its scaled rating, independently of the others, so the box widths are 26.25, 60 or 75 MW. Two further beliefs enter through Q5 and Q6. Belief 2 shifts every box to $[0.80, 1.10]$, a hotter season. The shared-weather belief keeps belief 1's marginals and couples them by a one-factor Gaussian copula with pairwise latent correlation 0.6.

The consequence G is the minimum load shed, in megawatts, over all re-dispatches of the network under the DC power-flow approximation: the value of a linear program in which generation lies within its limits, each load may be curtailed, power balances at every bus, flows follow the DC angle equations, and each branch flow is bounded by its rating. The program is solved with the HiGHS solver. Each solve returns the shed and, from the dual variables of the fourteen uncertain flow bounds, the shed's derivative with respect to each uncertain rating: $1 + K = 15$ numbers per solve.

21.3 The model as a factor graph

Figure 21.1 draws the model in the book's notation. Each uncertain rating r_i is a variable with a prior factor, the uniform box of belief 1. A rating touches the network through exactly one factor, the bound β_i that holds the flow p_i on its branch within $\pm r_i$. The flows meet at the bus balance factors, which also read the generators' outputs and the served fractions of the loads. The dashed box is the linear program: given the ratings, it chooses every variable inside to minimize the shed. Its value is G, and the duals of the β_i are the subgradient $\mathbf{s} = \nabla G$ with respect to the ratings, by the sensitivity theorem of linear programming that §14.7 invoked.

Two properties of this graph carry the whole study. The first is that the ratings enter only through the β_i, as right-hand sides of linear constraints. The value of a linear program is convex in its right-hand sides, so G is convex in $\mathbf{r}$, and Proposition 14.2 applies. The second is monotonicity: raising a rating loosens one constraint and enlarges the feasible set, so the minimum shed cannot rise. Every component of $\mathbf{s}$ is therefore zero or negative. The first property makes the surrogate a floor; the second, as §21.6 shows, makes one of the nine questions easier than its standard method assumes.

The lower row is the computation. A census of solves at draws $\mathbf{r}_j$ from belief 1 yields triples $(\mathbf{r}_j, G_j, \mathbf{s}_j)$, and each triple is a supporting plane $\pi_j(\mathbf{r}) = G_j + \mathbf{s}_j^{\mathsf{T}}(\mathbf{r} - \mathbf{r}_j)$. The surrogate of equation (14.3) is their maximum, floored at zero. The model is not solved again except where a question needs a small residual sample.

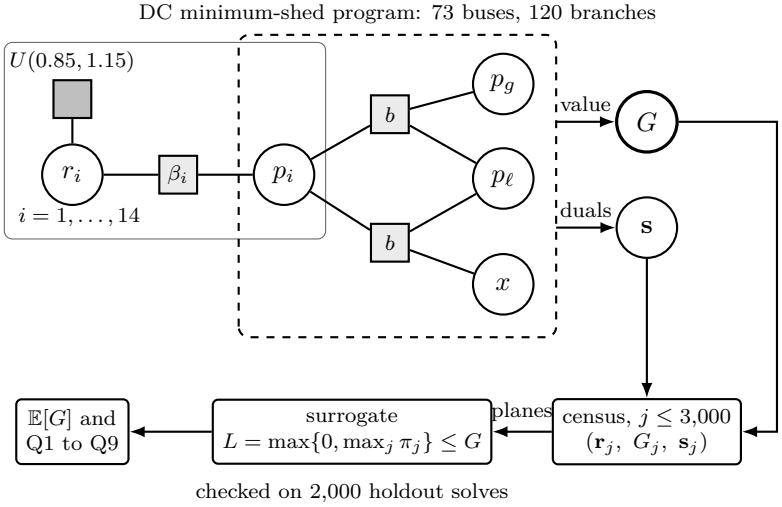


Figure 21.1: The model and the census route. Circles are variables and squares are factors. Top: the fourteen uncertain ratings r_i (plate), each with its prior factor (grey) and its bound factor β_i on the branch flow p_i; bus balance factors b join flows, generation p_g, the other branches' flows p_ℓ and served load x. The dashed box is the linear program; its value is the shed G and the duals of the β_i are its subgradient $\mathbf{s}$. Bottom: each solve adds a plane π_j to the surrogate L, and the questions are asked of L.

21.4 Method

The method has seven steps. Each is a construction the earlier chapters justified; the step names which.

Step 1: identify the uncertain inputs. One solve at the scaled ratings marks the 14 active flow bounds. Only those ratings are made uncertain. This is the factor-graph reading of Chapter 2 applied to sensitivity: an input whose factor is inactive at the operating point has a zero column in $\mathbf{s}$ there. The step is a modelling choice with a limit, stated in §21.6: a bound inactive at the centre of the box could activate in a corner of it.

Step 2: take gradients from the solve. The duals of the β_i are $\partial G/\partial r_i$, returned with the optimum at no extra cost; §14.2 gives the smooth analogue and §14.7 the linear-programming case. Where the program is degenerate the dual is a subgradient rather than a gradient, which is all Proposition 14.2 asks. No difference is taken, so the trap of §14.5 does not arise.

Step 3: harvest the census. Draw 3,000 rating vectors from belief 1, solve each, and keep the triples. The census is a matrix with $2K + 1 = 29$ columns: the ratings drawn, the shed, and the fourteen duals. This is the construction of §14.4.

Step 4: certify. Solve 2,000 further draws, independent of the census, and check $L \le G$ on every one. The inequality is a theorem given convexity; the check guards against a wrong sign in a dual or a declared convexity that does not hold (Principle 14.2).

Step 5: estimate the mean by a control variate. Write $\mathbb{E}[G] = \mathbb{E}[L] + \mathbb{E}[G - L]$. The first term is an average of L over scrambled Sobol points, as many as needed, at no solves. The second is an average over 512 solves at fresh draws, and it is small because L hugs G. The two terms' sampling errors combine into one 95 percent interval, and $\mathbb{E}[L]$'s own one-sided limit is a lower bound on $\mathbb{E}[G]$, since $G - L \geq 0$. The sampling theory is Chapter 9's and Proposition 10.2's; what the surrogate adds is that the expensive term has a small variance.

Step 6: answer each question from the surrogate. Each question becomes a functional of L, of the census matrix, or of both:

- Q1 is the column mean of the census duals, since differentiation and expectation commute.
- Q2 and Q4 are the pick-freeze estimators of Chapter 10 applied to L instead of G, on large samples of cheap evaluations; the first-order estimator comes from the same matrices as the total one.
- Q3 is the derivative-based bound of Sobol and Kucherenko: for r_i uniform on an interval of width w_i, $S_{T,i} \leq w_i^2 \, \mathbb{E}[(\partial G/\partial r_i)^2]/(\pi^2 \operatorname{Var}[G])$. The second moment is the mean squared census dual. The bound needs G to be absolutely continuous along each input with the duals as its derivatives, which a linear program's value is; it does not need convexity. A line whose bound falls below 1 percent is excluded.
- Q5 and Q6 repeat step 5 under the new belief. The planes are valid everywhere, so $\mathbb{E}[L]$ under the new belief is a floor with no new solve; the residual needs 512 solves drawn from the new belief.
- Q7 fixes one rating at each point of a grid, draws the other thirteen from belief 1, and averages L: a floor on $\mathbb{E}[G \mid r_i = r]$ at every point.
- Q8 evaluates L on a large draw from belief 1 and reads off the exceedance fractions, the 0.95 quantile and the mean above it. Since $L \leq G$ pointwise, each is a floor on the same quantity of G.
- Q9 maximizes L over the box. Each plane is separable in the ratings, so its maximum over the box is at the vertex that takes each rating's upper end where the plane's slope is positive and its lower end otherwise; the maximum over the planes is the maximum of L, found in one pass over the 3,000 planes without enumerating the 2^{14} vertices.

Only Q5 and Q6 solve again. The variance in the denominators of Q3 and Q4 was taken from the reference sample; the census's own values estimate it too, at no further solves.

Step 7: score both routes. The sampling alternative for each question is the standard method at a budget of thousands of solves: central differences of the mean with common random numbers, 2,000 draws per shift and a step of 0.02 per unit (Q1); Saltelli's design with $N = 1{,}000$ (Q2, and Q4 from the same run); Morris screening with 20 trajectories (Q3); a fresh scrambled-Sobol study of 4,096 solves (Q5, Q6); conditional Monte Carlo with 1,000 solves per grid point (Q7); the empirical tail of 10,000 solves (Q8); and the solve of every vertex of the box (Q9), which is exact for a convex G. The references are the mean of the duals over 100,000 draws (Q1, Q8), Saltelli with $N = 4{,}000$ (Q2 to Q4), fresh studies of 20,000 solves (Q5, Q6), conditional Monte Carlo with 4,000 draws per point (Q7), and the vertex enumeration itself (Q9).

Table 21.1: The nine questions: solves and error against the reference for the census route and the sampling alternative, and the kind of statement the census route makes. A floor or ceiling is one-sided by theorem; the alternatives give intervals without a direction. Q4's alternative reads the Saltelli run of Q2 and costs no solve of its own; Q9's alternative is exact by construction and is the reference.

		solves		error		census
	question and error measure	census	sampling	census	sampling	claim
Q1	shadow prices, max, MW/MW	0	56,000	0.018	0.019	estimate
Q2	total indices S_T, max	0	16,000	0.009	0.020	estimate
Q3	screening bound holds	0	300	14 of 14	no bound	ceiling
Q4	first-order S_1, max	0	0	0.029	0.044	estimate
Q5	mean, new belief, MW	512	4,096	0.091	0.036	floor
Q6	mean, correlated, MW	512	4,096	0.084	0.116	floor
Q7	what-if curve, max, MW	0	7,000	2.27	0.49	floor
Q8	exceedance, max	0	10,000	0.0050	0.0098	floor
Q9	worst case, MW	0	16,384	0.31	0 (exact)	floor
	census, holdout and the mean	5,512	—			
	total	6,536	113,876			

21.5 Results

21.5.1 The certificate and the cost

On the 2,000 holdout draws the surrogate violated $L \le G$ on none; the largest excess was 4.8×10^{-10} MW, the solver's tolerance. On the 100,000 reference draws it violated it on none. The residual $G-L$ on the reference has mean 0.609 MW and standard deviation 1.181 MW, against a standard deviation of G of 50.25 MW. The surrogate captures all but a small, nonnegative remainder.

Table 21.1 lists the nine questions with each route's solves and error against the reference, and the kind of claim the census route makes. Figure 21.2 draws the solve counts on a logarithmic scale. The census route spends 6,536 solves: the census and the holdout, 5,000; the mean, 512; and the residual samples of Q5 and Q6, 512 each. The alternatives spend 113,876, seventeen times as many. Seven of the nine questions cost the census route no solve at all beyond the census. With the references the study made 352,413 solves.

21.5.2 The mean

The control variate gives $\mathbb{E}[G] = 214.185 \pm 0.092$ MW (95 percent) from 512 residual solves, with a certified lower bound of 213.525 MW. The study's own reference, 100,000 plain draws, gives 214.279 ± 0.159 MW (one standard error). That reference is less precise than the answer it is meant to check, so the mean is checked against the method study described at the end of this section, which ran the same network, lines and box with a different seed and a 300,000-draw reference tightened by the same control variate: 214.114 ± 0.002 MW. The census answer lies 0.071 MW above it, inside its half-width of 0.092. The study's plain reference lies 0.165 MW above it, 1.04 of its own standard error.

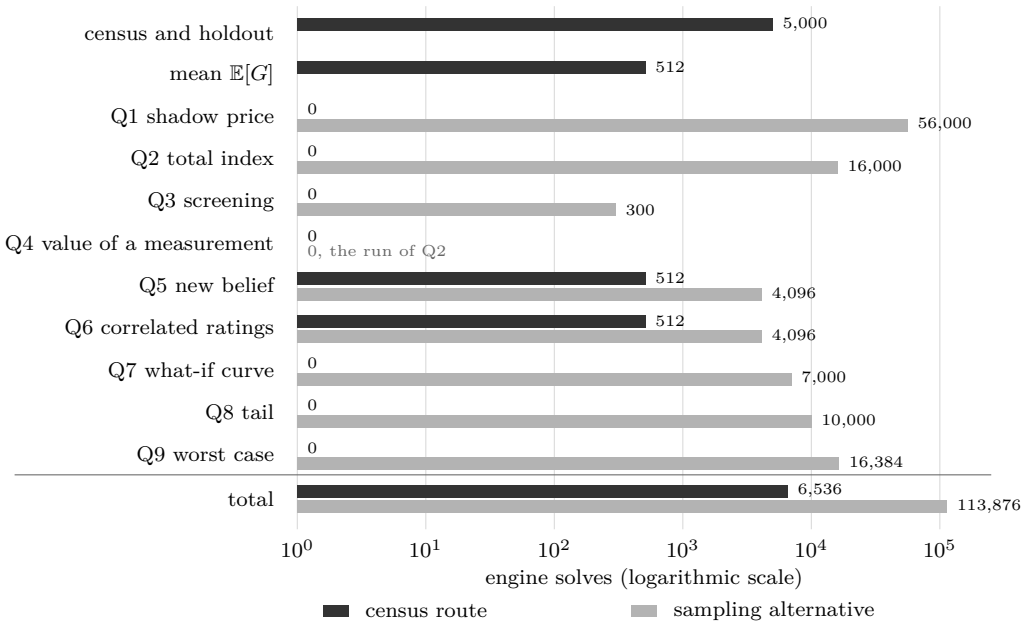


Figure 21.2: Engine solves per question on a logarithmic scale. Dark: the census route; light: the standard sampling method for the same question. The top two rows are paid once; a label 0 marks a question answered with no further solve.

21.5.3 Prices and attribution: Q1 to Q4

Table 21.2 gives, for each line, the census answers to Q1 to Q3 beside the alternatives and references. For Q1 the largest error of the census route over the fourteen lines is 0.0175 MW per MW; central differences at 56,000 solves reach 0.0192. Both name the same three lines as the most valuable to uprate, branches 102, 58 and 11. For branch 102 the census says one more megawatt of rating removes 1.070 MW of expected shed, with a standard error of 0.007; the reference says 1.066. The reference is the same estimator as the census answer at 33 times the sample, which is why the census route can afford to be the more accurate: it has no step to choose and no difference to take.

For Q2 the surrogate's total indices are within 0.0086 of the reference on every line, and Saltelli on the program at 16,000 solves is within 0.0200. All three routes rank branches 102, 25 and 58 first. Branch 102 alone carries a total index of 0.242 by the surrogate and 0.237 by the reference. For Q4 the first-order indices are within 0.0295 (surrogate) and 0.0444 (Saltelli) of the reference. The reference first-order indices sum to 0.847, so 15 percent of the variance is interaction among the ratings, which no single measurement removes: learning branch 102's rating exactly would remove 595 MW^2 of a variance of 2,525 by the reference, 570 by the surrogate.

For Q3 the bound lies above the reference total index on all fourteen lines, as the theorem requires, and it excludes four below 1 percent: branches 30, 89, 91 and 107. The largest of their bounds is 0.0057 and the smallest bound among the other ten is 0.0296, so the sampling error in the estimated second moments and variance does not move the decision. Morris screening at 300 solves puts the same four lines at the bottom, in a different order, with no statement of how much any of them could matter. The two outputs are of different kinds: a ranking, and an inequality one can sign.

Table 21.2: The fourteen uncertain lines, ordered by the reference total index. Rating: the scaled rating, the centre of the line's box. Shadow price: the mean census dual against the mean dual over 100,000 draws. Total index: from the surrogate, from Saltelli on the program (16,000 solves) and the reference (64,000 solves). DGSM bound: the certified ceiling on S_T from the census duals; * marks the four lines it excludes below 1 percent.

		$\partial\mathbb{E}[G]/\partial r_i$ [MW/MW]		total index S_T			DGSM
branch	rating [MW]	census	reference	census	Saltelli	reference	bound
102	250.0	−1.070	−1.066	0.2420	0.2213	0.2367	0.2934
25	250.0	−0.853	−0.846	0.1713	0.1826	0.1627	0.2449
58	200.0	−1.000	−1.016	0.1612	0.1647	0.1588	0.2728
100	250.0	−0.721	−0.704	0.1284	0.1291	0.1284	0.2753
48	200.0	−0.808	−0.796	0.1031	0.1057	0.0994	0.2215
86	200.0	−0.773	−0.773	0.0931	0.0891	0.0882	0.1353
64	250.0	−0.578	−0.582	0.0849	0.0876	0.0847	0.1650
96	200.0	−0.515	−0.519	0.0536	0.0577	0.0557	0.0993
53	87.5	−0.880	−0.873	0.0235	0.0248	0.0236	0.0296
11	87.5	−0.924	−0.909	0.0233	0.0235	0.0218	0.0338
91	87.5	−0.321	−0.328	0.0045	0.0047	0.0046	0.0057*
30	250.0	−0.047	−0.048	0.0018	0.0021	0.0015	0.0043*
89	87.5	−0.102	−0.107	0.0009	0.0007	0.0008	0.0010*
107	250.0	−0.029	−0.029	0.0005	0.0005	0.0006	0.0018*

21.5.4 New beliefs: Q5 and Q6

Under belief 2 the expected shed rises to 302.1 MW by the reference. The census, harvested under belief 1, answers 302.180 ± 0.095 MW from 512 residual solves, 0.091 MW from the reference, which is 1.88 of its own standard error and inside its interval. The fresh quasi-Monte Carlo study at 4,096 solves is 0.036 MW off. On this question the alternative is the more accurate, and the reference's standard error, 0.018 MW, is small enough to resolve it. The census route's gain here is the eight-fold saving in solves and the lower bound, 301.403 MW, which holds; it is not accuracy.

Under the shared-weather belief the expected shed falls to 202.2 MW. The census answers 202.278 ± 0.061 MW, 0.084 MW off, and the fresh study 0.116 MW off. Here the reference's own standard error, 0.052 MW, is comparable to both errors, so the order of the two is not resolved; the census answer's interval misses the reference value, which lies 2.69 of the census route's standard errors away. The lower bound, 201.870 MW, holds, and the residual samples of both questions satisfied $L \le G$ on every draw.

21.5.5 The what-if curve: Q7

The most influential line, branch 102, is swept across its box from 212.5 to 287.5 MW. Figure 21.3 plots each route's curve minus the reference. The census curve lies below the reference at every point, as a floor must, by 0.52 MW at the low end and by 2.27 MW at the high end. Conditional Monte Carlo at 7,000 solves deviates by at most 0.49 MW. On this question the census route is the less accurate, and the error is not uniform: it is more than four times larger at the top of the range than at the bottom. Two facts explain the shape. The reference curve bends most at the top, where its drop per 12.5 MW step shrinks from 15.96 to 7.29 MW as branch 102 stops binding in more draws; a maximum of planes is loosest where the function curves most. And at

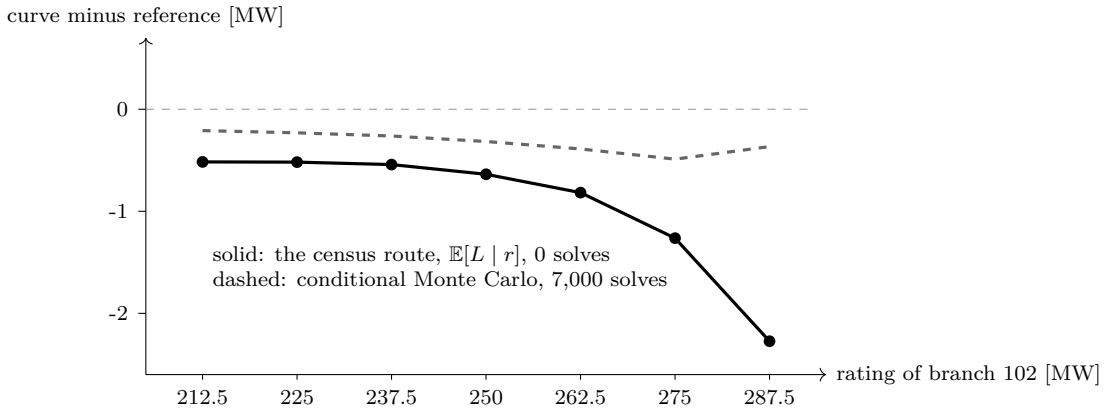


Figure 21.3: Q7: the error of each route's curve $\mathbb{E}[G \mid r_{102} = r]$ against the reference (4,000 draws per point). The census route (solid) is a floor and lies below the reference everywhere; its gap grows to 2.27 MW where the curve bends at the top of the range. Conditional Monte Carlo (dashed) is within 0.49 MW; its seven points share one draw of the other thirteen ratings, which is why its errors share a sign.

the edge of the box every census point lies on one side of the conditioning value, so no plane is anchored there.

The reference is not exact either. Its sampling error is not recorded, but at 4,000 draws per point on a consequence whose standard deviation is 50.25 MW its scale is $50.25/\sqrt{4{,}000} = 0.79$ MW, somewhat less since fixing one rating removes part of the variance. The alternative's 0.49 MW and the census route's gap at the low end sit inside that scale; the 2.27 MW gap at the top does not.

21.5.6 The tail and the worst case: Q8 and Q9

At the reference quantiles 0.5, 0.75, 0.9, 0.95 and 0.99, the levels 213.8, 248.5, 279.5, 298.1 and 331.5 MW, the census floors on $\mathbb{P}(G > g)$ are 0.49499, 0.24644, 0.09855, 0.04894 and 0.00978, below the reference at every level. Their largest error is 0.0050; the empirical tail of 10,000 solves has 0.0098, with no direction. At $\alpha = 0.95$ the floors are VaR ≥ 297.6 and CVaR ≥ 318.0 MW, against the reference 298.1 and 318.4; the empirical tail estimates 298.3 and 319.1. On value-at-risk the empirical estimate is the closer, on conditional value-at-risk the floor, and only the floor comes with a direction.

Over the whole box, the maximum of L is 471.69 MW, at the same vertex as the exact worst case, 472.00 MW, found by solving all 16,384 vertices. The gap is 0.31 MW.

21.5.7 How large a census

The method study ran the same network, lines and box with seed 20260906, where the queries study used 20260908, and asked how the certificate and the mean depend on the census size. Its residual samples were 2,048 solves rather than 512, and its reference 300,000 draws rather than 100,000. Figure 21.4 draws what it found at censuses of 100, 300, 1,000 and 3,000 solves. At every size the surrogate violated $L \leq G$ on none of the 2,000 holdout draws, and at 3,000 on none of the 300,000 reference draws. The mean gap $\mathbb{E}[G - L]$ fell from 6.137 to 0.559 MW.

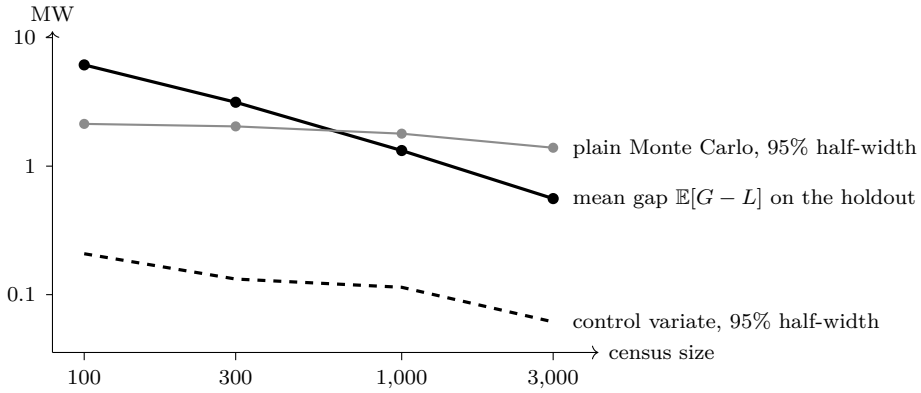


Figure 21.4: The method study: against census size, the mean gap between the program and the surrogate on the holdout (solid), and the 95 percent half-width of the mean by the control variate (dashed) and by plain Monte Carlo (grey) at the same total budget of the census plus 2,048 solves. Logarithmic scales on both axes. No holdout draw violated $L \leq G$ at any size.

The control variate's 95 percent half-width fell with it, from 0.208 to 0.061 MW, while plain Monte Carlo at the same total budget, census included, fell only from 2.132 to 1.391. At a census of 3,000 the control variate's interval is 22.7 times narrower than plain sampling's at an equal budget of 5,048 solves. Over five replicates at a budget of 7,096, its root-mean-square error is 0.0068 MW, against 0.040 for quasi-Monte Carlo over the solve and 0.762 for plain Monte Carlo. The method study's screening bound excludes the same four lines, 30, 89, 91 and 107.

21.6 What surprised

One solve at the expected ratings understates the shed by a quarter. At the scaled ratings, the centre of the box, the shed is 154.1 MW. The expected shed is 214.1 MW. The gap, 60.1 MW or 28 percent of the expectation, is Jensen's inequality for a convex consequence: the average of G exceeds G at the average input. A planner who solves once at the forecast ratings does not get an approximate answer but a biased one, biased in the unsafe direction.

Correlating the ratings lowered the mean. Shared weather moved the expected shed from 214.3 to 202.2 MW, a fall of 12.1 MW. One might expect correlation to hurt, since low ratings then arrive together. Convexity does not fix the direction. The three-line problem of Chapter 10 shows both cases. Where two lines act in parallel, their sum binds, correlation widens the spread of the sum, and a shed that is convex in the sum has a larger mean when the sum spreads wider. Where a line acts in series, the tightest one binds, and the minimum of correlated ratings is larger on average than the minimum of independent ones, so the shed falls. On this network the second effect wins. The census route did not need to know that; it read the answer from 512 solves.

The worst case needed one solve. The worst vertex, found by both routes, is the corner with every rating at the low end of its box. Monotonicity predicts it: the shed cannot rise when a bound loosens, so it is largest where every bound is tightest. For this consequence Q9's exact answer costs one solve, not 16,384. The ledger charges the enumeration because it is the general

method for a convex consequence, and the census route found the same vertex without being told. Charging one solve instead, the alternatives total 97,493 and the ratio is 14.9 to one rather than seventeen. A method comparison is only as fair as the knowledge each side is allowed; the monotonicity was in the factor graph all along.

A floor at every point is not a floor on a difference. The census curve of Q7 is a floor at each rating, but its gap grows along the range, so the curve is too steep: its slope is -1.101 MW per MW against the reference's -1.081. Read as the benefit of uprating branch 102 from 250 to 287.5 MW, the census curve promises 34.09 MW of relief; the reference gives 32.45 and conditional Monte Carlo 32.50. The difference of two floors carries no guarantee, and here it overstates the value of the uprate by 5 percent. Anyone who reads an investment off the surrogate should read it from the shadow price of Q1, which is a mean of exact duals, or buy a few solves at the two ratings that matter.

Accuracy is not uniform. The census route was the more accurate on Q1, Q2, Q4 and Q8, and less accurate on Q5 and Q7; on Q6 the reference cannot tell. The losses have one cause. The surrogate is tightest where the census put its points, under belief 1 and in the interior of the box, and loosest elsewhere: belief 2 reaches down to 0.80 of the scaled rating, below the census's lower edge of 0.85, and Q7 conditions on the box's edges. Belief 2 still gave a certified floor, because the planes are valid everywhere; it gave a looser one. A census meant to serve other beliefs should be drawn from their union, or seeded at the vertices and edges, as §14.8 advised.

Slope is not share. Morris screening ranked branch 53 third of fourteen; the reference total index ranks it ninth. Its mean slope is steep, -0.873 MW per MW, but its box is only 26.25 MW wide, and a share of variance is slope times range. The screening bound of Q3 carries the factor w_i^2 and ranks it where it belongs. The question a planner asks, which lines can be ignored, is a question about shares.

The limits. Three conditions carry the study, and each is a boundary of its claims. The solve must return its sensitivities; a program that returned only a value would pay $2K + 1 = 29$ solves per plane by differences, and across a kink would not get a subgradient at all. The consequence must be convex; an AC flow is not, and there the same calls return estimates without floors. And the uncertain inputs were chosen at the centre of the box: the 106 ratings inactive there were held fixed, and a bound that activates only in a corner is outside the model. Every certified statement is one-sided. The surrogate never overestimates, so the tail and the worst case are bounded from below only; their upper side remains statistical.

21.7 Reproduction

The two studies, `rating_uncertainty_queries` and `rating_uncertainty_surrogate`, live in the directory `foundation_power_flow` of the repository's case studies. From the repository root, each is regenerated by

```
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
    foundation_power_flow/rating_uncertainty_queries/run.py
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
    foundation_power_flow/rating_uncertainty_surrogate/run.py
```

with the line break removed. The first uses seed 20260908 and the HiGHS backend and makes 352,413 solves including its references; on the machine that produced the committed results it ran in 505 seconds, of which the census took 6.6. The second uses seed 20260906. Each writes its `results.json` at six significant digits. The network is the RTS-GMLC case file vendored with the studies; the evaluator that turns the program into a map from ratings to shed and duals is `MinShedRatingEvaluator` in `plugins.power_flow`, and the census, certificate and nine queries are generic calls in `plugins.uncertainty`, which never sees the network.

The chapter's numbers, tables and plotted figures come from

```
cd docs/book
python scripts/ch21_census.py
```

which reads the two committed `results.json` files and the queries study's `timings.json`, makes no solve, prints every number quoted above, and writes `figures/ch21_solves.tex`, `ch21_census_size.tex`, `ch21_whatif.tex`, `ch21_nine.tex` and `ch21_lines.tex`. The factor graph of Figure 21.1 carries no data.

Chapter 22

What survives on AC

The census of Chapter 14 answers many questions from one set of solves, and §14.7 named the two conditions it rests on: the solve must return its gradient, and the consequence must be convex in the inputs. Chapter 21 put nine planning questions to a load-shedding linear program, where both conditions hold. This chapter puts the same nine questions to a model where the first holds and the second does not: an AC power flow, solved by Newton's method, whose line loadings are not convex in the loads. Its gradients come from the implicit function theorem, at the price of one factorization per solve. Its surrogate is no longer a bound by theorem. The question is which of the nine answers keep their guarantee, which keep only their accuracy, and what each costs. The answer in brief: the gradients survive, one guarantee survives with them, and the five answers that read a floor from the surrogate rest on an inequality that held at every draw checked in one regime and failed at about one draw in nine in the other. Every number in the chapter is read from the study's committed results by the script `ch22_ac.py` in the book's `scripts` directory, which prints them and solves nothing.

22.1 The question

A planner has a transmission network, a set of line ratings and a load forecast that is uncertain by fifteen percent either way at each of the eight largest loads. The consequence of interest is the total overload, the sum over lines of how far each line's apparent power exceeds its rating. The planner's questions are the nine of Chapter 21: the shadow price of each load (Q1), the share of the overload's variance each load causes (Q2), which loads can be dropped from further study with a guarantee (Q3), how much variance metering a load would remove (Q4), the expected overload under a shifted forecast (Q5) and under loads that share the weather (Q6), the expected overload as one load grows (Q7), the tail of the overload (Q8), and the worst case over the forecast box (Q9).

On the linear program each answer came with a label that said what it rested on. Here the labels are assigned by one rule, the same in both chapters. An answer is *certified* when it is a theorem under a hypothesis the model declares and the study tests. It is *empirical* when it is a one-sided claim, a floor, that rests on the surrogate inequality $L \le G$, and that inequality held on every fresh draw that was checked; nothing guarantees it elsewhere. It is an *estimate* when it is a number with a standard error and no direction. The chapter's question is how the nine labels fall on a model that is not convex, and whether the answers remain worth their cost.

22.2 The system

The network is the IEEE 14-bus test system, a small transmission network long used as a reference case for power-flow methods. It has 14 buses, 20 branches (lines and transformers), 5 generator buses and 11 loads totalling 259 megawatts. The eight largest loads are uncertain: 94.2 megawatts at bus b3, 47.8 at b4, 29.5 at b9, 21.7 at b2, 14.9 at b14, 13.5 at b13, 11.2 at b6 and 9.0 at b10, together 241.8 megawatts. The three smallest loads are fixed.

Each uncertain load is scaled by a multiplier d_i, uniform on $[0.85, 1.15]$ and independent of the others, which scales its active and reactive power together, at constant power factor. The model has 19 load inputs, the active and reactive loads of the buses that carry them, and the eight multipliers map onto them affinely. Two further beliefs appear in the questions: every interval shifted to $[0.80, 1.10]$, a milder season (Q5), and the original marginals coupled by a one-factor Gaussian copula with pairwise correlation 0.6, loads that share the weather (Q6).

The consequence is the total apparent-power overload,

$$G(\mathbf{d}) \;=\; 100 \sum_{\ell=1}^{20} \left(|S_\ell(\mathbf{d})| - r_\ell \right)^+ \text{ megawatts}, \tag{22.1}$$

where $|S_\ell|$ is the larger of the apparent powers at the two ends of branch ℓ in per unit on the 100 megavolt-ampere base, and r_ℓ is its rating. Two sets of ratings define two regimes. In the *congested* regime every rating is 0.90 of the branch's loading at the nominal loads, so every line is overloaded throughout the box and G is a sum of twenty smooth loadings less constants. Its mean is 72.41 ± 0.28 megawatts and its standard deviation 27.68. In the *intermittent* regime every rating is 1.05 of the nominal loading, so a line is overloaded only when the loads run high; G is zero on 9.98 percent of the draws, its mean is 4.52 ± 0.06 megawatts, its standard deviation 6.45, and the set of overloaded lines changes across the box.

A second system, from the thermal domain, tests that nothing in the construction is specific to power flow: a duct heater, in which air at mass flow $\dot{m}$ and inlet temperature T_{in} receives an electric input W and loses 0.5 kilowatts, with the outlet temperature $T_{\text{out}} = T_{\text{in}} + (W - 0.5)/(\dot{m}\, c_p)$ as the consequence. Its three inputs are uniform on $\dot{m} \in [2.4, 3.6]$ kilograms per second, $T_{\text{in}} \in [285.15, 295.15]$ kelvin and $W \in [10, 20]$ kilowatts. It is not convex either: its Hessian in $(\dot{m}, W)$ is indefinite, a saddle like the generation areas' derating of §14.6.

22.3 The model as a factor graph

Figure 22.1 draws a slice of the model around buses b2, b3 and b4, whose three lines form a triangle. Each multiplier d_b carries a uniform prior factor and enters the balance factor of its bus, which holds the active and reactive power balance. Each balance factor couples the state $x_b = (V_b, \theta_b)$ of its own bus with the states of its neighbours, through the AC flow closures of the lines between them. The solved system has 114 unknowns and as many equations: bus states, branch-end flows and the midpoint balances that the engine's compiled form carries. Below the states sit the overload factors, one per branch, each a positive part of a smooth loading, and below them the consequence, their sum.

The dotted line in the figure is the one that matters for the method. Above it the map from the multipliers to the states is smooth: at a converged solution with a nonsingular Jacobian it is differentiable, and equation (14.1) of Chapter 14 gives its derivative from the Jacobian that Newton's method has already assembled and factored, with the input columns kept. Below

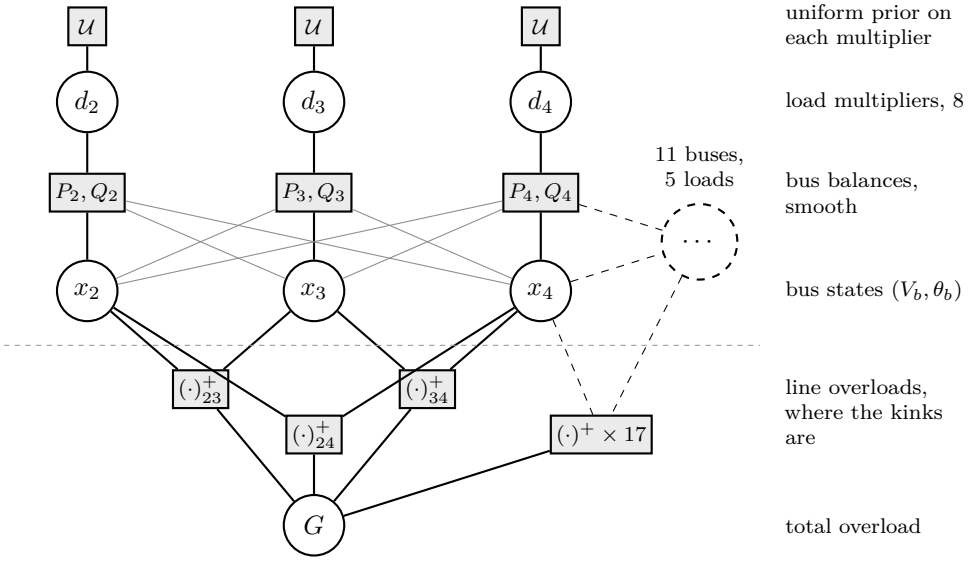


Figure 22.1: A slice of the AC model around buses b2, b3 and b4. Circles are variables, shaded squares factors. Each load multiplier d_b has a uniform prior and enters its bus's active and reactive balance; each balance couples its bus's state $x_b = (V_b, \theta_b)$ with its neighbours' through the lines' flow closures (grey edges). The dashed circle stands for the other 11 buses and 5 uncertain loads. Above the dotted line the map from loads to states is smooth and its derivative comes from one factorization of the Jacobian; below it each line's overload is a positive part, with a kink where the line reaches its rating.

it each overload factor has a kink at its rating. The gradient of G is therefore assembled as §14.5 prescribes: the derivative of the smooth inner map, the loadings, by the implicit function theorem, and the active piece of each positive part at the solved point. With $|S_\ell| = \sqrt{P_\ell^2 + Q_\ell^2}$ at the loaded end,

$$\frac{\partial G}{\partial \mathbf{d}} \;=\; 100 \sum_{\ell \text{ active}} \frac{P_\ell \, \partial P_\ell / \partial \mathbf{d} + Q_\ell \, \partial Q_\ell / \partial \mathbf{d}}{|S_\ell|}, \tag{22.2}$$

where a line is active when its loading exceeds its rating at the solved point. A census point costs one solve and one factorization. The alternative, central differences of G in each of the eight multipliers, costs $2K + 1 = 17$ solves per point and, where a step crosses a line's rating, returns the average of two slopes.

22.4 Method

The census. In each regime 300 multiplier vectors are drawn from the prior, and at each the model is solved and returns G and its gradient, equation (22.2). The surrogate is the maximum of the census planes and zero, equation (14.3). Proposition 14.2 makes $L \le G$ a theorem when G is convex. Here it is not. A line's apparent power is a nonlinear function of the loads, through voltages and angles that enter the flow closures in products and trigonometric functions, and nothing makes it convex. A plane tangent to a loading that bends downward lies above it nearby. The model therefore declares itself not convex, and the software never labels a floor certified on

it, whatever the checks find. The model also declares its gradients exact, because they come from the solve and not from differences.

The certificate, as an observation. The check of §14.4 is run as before and read differently. G is solved at 2,000 holdout draws and at 10,000 reference draws, and the excess $L - G$ is recorded at each, with a tolerance of 10^{-4} megawatts for the solver. On a convex model one violation would prove a wrong gradient. Here a violation is the non-convexity showing, and no violation is an observation about 12,000 draws, not a proof about the box. To see how the certificate behaves as the census grows, the census is also run at 100 and 1,000 points.

The routes to the nine answers. Each question has a route through the census and a standard sampling route that returns G only, at the budgets below, per regime.

- Q1: the mean of the census gradients, which is the gradient of the mean, at zero further solves; against central differences of the mean with 500 common draws per shift, 8,000 solves.
- Q2 and Q4: the total and first-order Sobol indices by Jansen's pick-freeze estimators on the surrogate, 20,000 rows per matrix, at zero solves; against the same design on the model with 1,000 rows, 10,000 solves, which serves both questions.
- Q3: the derivative bound of Proposition 22.1 below, from the census gradients, at zero solves, with a load excluded when its bound is below one percent; against Morris's elementary effects with 20 trajectories, 180 solves, which rank but do not bound.
- Q5 and Q6: the control variate $\mathbb{E}[G] = \mathbb{E}[L] + \mathbb{E}[G - L]$ under the new belief, the first term over quasi-random draws of L at no solves and the second from 256 solves; against a fresh quasi-Monte Carlo study of 2,048 solves.
- Q7: $\mathbb{E}[L \mid d_{\mathrm{b3}} = d]$ on a grid of seven values, the other loads drawn, at zero solves; against conditional Monte Carlo with 300 solves per grid point, 2,100 in all.
- Q8: $\mathbb{P}(L > g)$ at the reference quantiles of G, and the value at risk and conditional value at risk of L at 95 percent, at zero solves; against the empirical tail of 5,000 solves.
- Q9: the maximum of L over the box in closed form, then one solve of G at the maximizing vertex; against G at all $2^8 = 256$ vertices.

The closed form of Q9 uses the separability of each plane: the plane at census point j, with slope $\mathbf{c}_j$, is largest over the box at the vertex that takes each multiplier to the end its slope favours, so $\max L = \max\{0, \max_j [G_j + \sum_k \max(c_{jk}(a_k - d_{jk}),\, c_{jk}(b_k - d_{jk}))]\}$ over the census, for the box $\prod_k [a_k, b_k]$, which is $O(CK)$ arithmetic. On a convex G the vertex enumeration is the exact worst case. On this one it is not, because a non-convex function need not take its maximum over a box at a vertex, and both routes are estimates.

The one guarantee that needs no convexity. The screening bound of Q3 rests on the gradients alone.

Proposition 22.1 (A derivative bound on the total index)**.** *Let the inputs be independent, with d_i uniform on an interval of width w_i, and let G be absolutely continuous along each coordinate with square-integrable partial derivatives. Then*

$$S_{T,i} \operatorname{Var}[G] \;\le\; \frac{w_i^2}{\pi^2}\, \mathbb{E}\Big[\Big(\frac{\partial G}{\partial d_i}\Big)^2\Big]. \tag{22.3}$$

Proof. The total index satisfies $S_{T,i}\,\mathrm{Var}[G] = \mathbb{E}[\mathrm{Var}(G \mid \mathbf{d}_{-i})]$, the expected variance left when every input but d_i is known. With $\mathbf{d}_{-i}$ fixed, G is an absolutely continuous function of one variable on an interval of width w_i, uniformly distributed. The Poincaré inequality for the uniform law on an interval, Wirtinger's inequality, bounds the variance of such a function by $(w_i/\pi)^2$ times the mean square of its derivative. Take the expectation over $\mathbf{d}_{-i}$. □

The hypothesis is regularity, not convexity. The solution map is smooth on the box and each positive part is Lipschitz, so G qualifies, and the census gradients, being exact where G is differentiable, are its almost-everywhere derivatives. The inequality is a theorem on this model. Its two ingredients, the mean square gradient over 300 census points and the variance of G, are still Monte Carlo estimates, and the bound inherits their error.

Where the census helps a sampler. The control variate of Q5 and Q6 also estimates $\mathbb{E}[G]$ itself. It is unbiased whether or not $L \le G$ holds; the surrogate's accuracy decides only its variance.

22.5 Results

22.5.1 The certificate

In both regimes the census of 300 points cost exactly 300 solves and 300 factorizations, and no solve in the study needed a fallback start. In the congested regime $L \le G$ held at all 2,000 holdout draws and all 10,000 reference draws. The residual $G - L$ has a standard deviation of 0.0046 times that of G: the surrogate lies below the overload wherever it was checked, and close to it. In the intermittent regime L exceeds G at 205 of the 2,000 holdout draws and 1,149 of the 10,000 reference draws, 11 percent of the draws, by at most 4.65×10^{-3} megawatts, which is 7.2×10^{-4} of the standard deviation of G. The residual ratio is 0.043. The surrogate is an accurate interpolant of the intermittent overload but not a bound on it.

Figure 22.2 shows the certificate as the census grows. Panel (a) plots the spread of the residual against the census size in both regimes, with the number of holdout violations beside each point. The residual falls as the census grows in both regimes, roughly halving with each tripling. In the congested regime the count of violations stays at zero. In the intermittent regime it rises, 140, 205 and 296 of 2,000 at 100, 300 and 1,000 points. Panel (b) plots the largest violation, as a fraction of the standard deviation of G, in the intermittent regime. The filled circles are this study's census, whose gradients are exact: the largest violation grows only from 4.9×10^{-4} to 6.0×10^{-4} of the spread while the census grows tenfold. The other two series come from a separate run of the same model and regime, a census with its own seed that differenced each gradient at $2K + 1$ solves per point. The open circles difference the smooth loadings and apply the active set at the centre, as §14.5 prescribes, and sit with the exact gradients. The shaded squares difference G itself. Their violations grow with the census, from 0.14 to 1.09 megawatts, until at 1,000 points the largest is a sixth of the spread of G, whose estimate in that run is 6.38 megawatts, and 238 times the active-set value of 0.0046 megawatts.

The two panels separate two failures. The differenced gradient is wrong at the kinks, as Proposition 14.3 predicts, and a larger census puts more points across a rating. The exact gradient fails far more mildly, because the function bends the wrong way. Correcting the gradient removes more than two orders of magnitude of the violation and leaves the residue that belongs to the model.

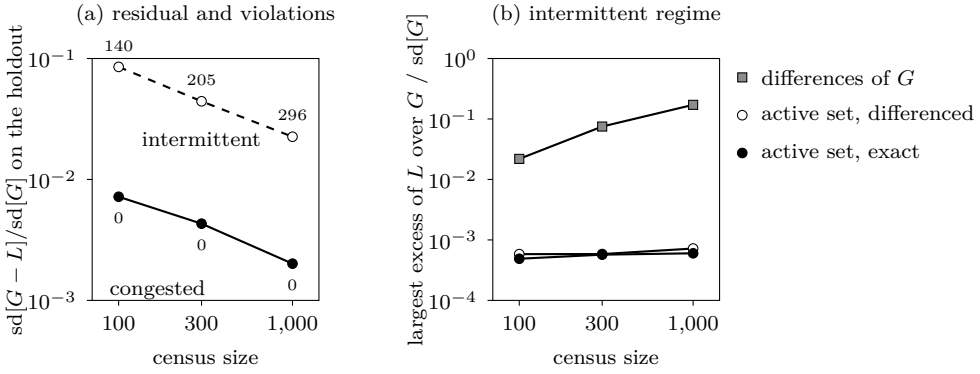


Figure 22.2: The certificate $L \leq G$ against the census size. (a) The standard deviation of $G - L$ on the 2,000-draw holdout, relative to that of G, in the congested (filled, solid) and intermittent (open, dashed) regimes; the numbers are the holdout draws at which $L > G$. (b) The largest excess of L over G on the holdout in the intermittent regime, relative to the standard deviation of G, for three gradients: central differences of G and differences of the smooth loadings with the active set, both from a separate census of the same model, and the exact gradients of this study. Both axes are logarithmic.

22.5.2 What survives

Table 22.1 gives the label of each answer in each regime, with what the answer rests on. The labels come from the rule of §22.1 and the declarations of the model; the software assigns them, not the analyst. Q1, Q2 and Q4 are estimates in both regimes, as they were on the linear program, because they never claimed a direction. Q3 is certified in both, by Proposition 22.1. Q5 to Q9 read a floor from the surrogate. On the linear program they were certified. Here they are empirical in the congested regime, where $L \leq G$ held on all 12,000 draws checked, and estimates in the intermittent one, where it failed on 1,354 of them.

The expected overload under the census belief comes first, since every later answer builds on the same census. The control variate gives 72.40 ± 0.01 megawatts in the congested regime and 4.45 ± 0.04 in the intermittent one, at 95 percent, against reference values of 72.41 ± 0.28 and 4.52 ± 0.06 from 10,000 solves. Per solve, the control variate's variance is 58,697 times smaller than plain Monte Carlo's in the congested regime and 613 times in the intermittent one. The difference of nearly a hundredfold between the regimes is consistent with the kinks: a surrogate of planes can follow twenty smooth loadings more closely than a sum of positive parts that switch on and off across the box.

22.5.3 Accuracy and cost

Table 22.2 compares each answer with its sampling alternative against the reference, in each regime, and gives the solves each route spent. The comparison is mixed. The census route is the more accurate in most entries and less accurate in three: the shadow prices in the congested regime, and the shifted forecast and the what-if curve in the intermittent regime. Its advantage is not accuracy. It is comparable accuracy at a small fraction of the solves.

Over the whole study, both regimes and the duct heater, the census route spent 8,200 solves on censuses and holdouts, including the 1,000-point censuses behind Figure 22.2, and 1,538

Table 22.1: The nine questions on the AC model: what each answer rests on, and its label in each regime. Certified: a theorem under a declared, tested hypothesis. Empirical: a floor resting on $L \le G$, which held on every one of 12,000 fresh draws. Estimate: a number with a standard error and no direction.

	question	rests on	congested	intermittent
Q1	shadow price of each load	unbiased gradients	estimate	estimate
Q2	total Sobol index S_T	variance decomposition of L	estimate	estimate
Q3	screening bound on S_T	exact gradients	**certified**	**certified**
Q4	value of metering, S_1	variance decomposition of L	estimate	estimate
Q5	$\mathbb{E}[G]$, shifted forecast	$L \le G$ under the new belief	empirical	estimate
Q6	$\mathbb{E}[G]$, correlated loads	$L \le G$ under the copula	empirical	estimate
Q7	what-if curve at b3	$L \le G$ along the grid	empirical	estimate
Q8	tail floors	$L \le G$ on the draws	empirical	estimate
Q9	worst case over the box	$L \le G$ at the vertex	empirical	estimate

Table 22.2: Each answer against its sampling alternative, as the error against the reference, in each regime, with the solves per regime. Q1: largest error over the eight loads, in megawatts per unit multiplier, on scales of 240 and 62. Q2, Q4: largest error in an index. Q5, Q6: error in megawatts. Q7: largest deviation over the grid, megawatts. Q8: error in the conditional value at risk at 95 percent, megawatts. Q9: the value found, megawatts, surrogate against the vertex maximum.

	congested		intermittent		solves
	census	sampling	census	sampling	census / sampling
Q1 shadow price	0.94	0.71	1.64	1.78	0 / 8,000
Q2 S_T	0.021	0.043	0.027	0.060	0 / 10,000
Q4 S_1	0.023	0.033	0.028	0.065	0 / same run
Q5 shifted forecast	0.003	0.011	0.006	0.001	256 / 2,048
Q6 correlated loads	0.007	0.030	0.016	0.031	256 / 2,048
Q7 what-if curve	0.28	0.48	0.38	0.14	0 / 2,100
Q8 CVaR_{95}	0.03	0.32	0.27	0.42	0 / 5,000
Q9 worst case	180.6	180.8	74.1	74.3	1 / 256

beyond them: the residual samples of the mean, Q5 and Q6, and the one solve at the worst vertex. The sampling alternatives spent 71,264, seven times as many.

The gradient cost almost nothing in time. In the congested regime the 1,000-point census, a solve and a factorization at each point, took 8.2 milliseconds per point, against 8.6 milliseconds per solve for the 10,000 reference solves; in the intermittent regime 14.8 against 9.5. A differenced census would have paid 17 solves a point.

22.5.4 The answers, question by question

Q1, shadow prices. The census gradients rank the loads b3, b9, b4 by their shadow price in both regimes, as the reference does. In the congested regime b3's is 241.3 megawatts of expected overload per unit multiplier against a reference of 240.4: growing the load at b3 by one percent adds about 2.4 megawatts of expected overload. In the intermittent regime it is 62.3 against 62.5.

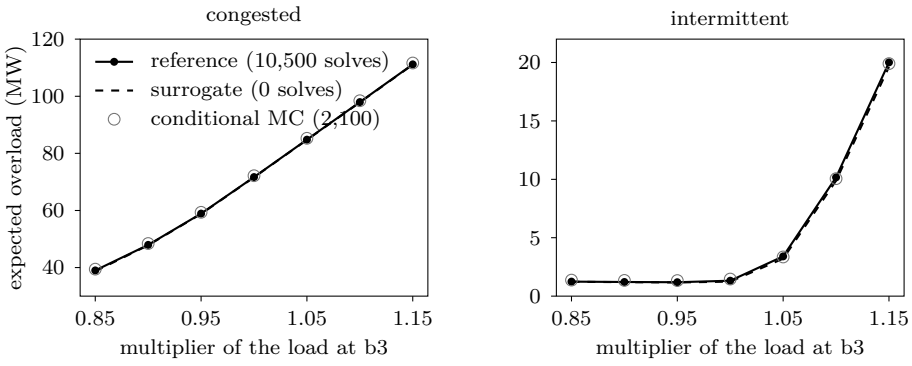


Figure 22.3: The expected overload as the load at b3 varies over the forecast box, the other seven loads averaged over their prior (Q7). Solid line and dots: the reference, 1,500 solves per grid point. Dashed line: the surrogate, at zero solves. Circles: conditional Monte Carlo, 300 solves per point. Left, the congested regime; right, the intermittent one, whose curve is flat below the nominal load and steep above it.

Q2 to Q4, attribution, screening and metering. The total indices from the surrogate are within 0.021 of the reference on every load in the congested regime and within 0.027 in the intermittent one, against 0.043 and 0.060 for pick-freeze on the model. The load at b3 dominates: its reference total index is 0.627 and 0.832. The derivative bound of Proposition 22.1 lies above the reference total index on all eight loads in both regimes, as the theorem requires. It excludes one load below one percent in the congested regime, b2, whose reference total index is 0.0046, and none in the intermittent one. Morris's screening at 180 solves names the same three least influential loads, in the congested regime in the same order, without saying how much any of them could matter. The first-order indices tell the regimes apart. In the congested regime they sum to 0.979 by the reference, so the overload is nearly additive in the loads. In the intermittent regime they sum to 0.633, so 37 percent of the variance is interaction: a line overloads when several loads run high together. Metering the load at b3 would remove 462 of the 766 square megawatts of variance in the congested regime by the reference, and the surrogate says 472.

Q5 and Q6, a new belief. Under the milder season the expected overload falls to 39.90 megawatts in the congested regime and 1.11 in the intermittent one. The census, drawn under the original belief, recovers both from 256 residual solves, within 0.003 and 0.006 megawatts, where a fresh quasi-Monte Carlo study of 2,048 solves is within 0.011 and 0.001. Under shared weather the expected overload rises to 73.60 and 9.36; the census route is within 0.007 and 0.016, the fresh study within 0.030 and 0.031. The surrogate mean alone, 39.45 and 73.40 in the congested regime, lies below the reference under both beliefs in both regimes.

Q7, the what-if curve. Figure 22.3 plots the expected overload against the multiplier of the load at b3, by the three routes. In the congested regime the curve is nearly straight, with a slope of 245.0 megawatts per unit multiplier from the surrogate against 244.5 from the reference, and the surrogate lies within 0.28 megawatts of the reference everywhere, below it at every grid point; conditional Monte Carlo at 2,100 solves strays by up to 0.48. In the intermittent regime the curve has two regimes of its own. Below the nominal load it is flat, near 1.2 megawatts, and even dips slightly; above it lines cross their ratings and the curve climbs to 20.0 megawatts at

the top of the box. The surrogate follows the shape and stays below the reference at every point, by up to 0.38 megawatts, at the top of the box, where the curve is steepest. Conditional Monte Carlo is within 0.14.

Q8, the tail. At the reference quantiles of G the surrogate's exceedance probabilities lie at or below the reference at every level, 0.0497 against 0.05 at the 95th percentile in the congested regime and 0.0486 in the intermittent one. The conditional value at risk at 95 percent is 127.19 megawatts from the surrogate, 127.22 by the reference and 127.54 from the empirical tail of 5,000 solves, in the congested regime; in the intermittent regime 25.06, 25.34 and 25.76. The empirical value at risk of the intermittent regime, 18.67 megawatts, matches the reference to the second decimal, where the surrogate's is 18.41: on this one number the sampler wins.

Q9, the worst case. The surrogate's maximum over the box is 180.6 megawatts in the congested regime, at the vertex where every load is at 1.15 of nominal, and one solve there finds $G = 180.8$. Solving all 256 vertices finds the same vertex and the same value. In the intermittent regime the numbers are 74.1 and 74.3, again at the all-high vertex. Neither route is a certified worst case on this model. The only unconditional floors on hand are the largest reference draws, 157.2 and 50.6 megawatts, far below the vertex values: random draws rarely visit a corner of an eight-dimensional box.

The duct heater. The same calls, with no change, were run on the duct heater with a census of 200 points. The gradients from the solve agree with the closed form to 1.1×10^{-13} kelvin. The shadow prices are within 1.2 percent of their closed form over the box, where finite differences of the mean at 12,000 solves are within 0.5 percent. The total indices from the surrogate are within 0.013 of the reference, where pick-freeze on the model at 5,000 solves is within 0.009. The derivative bound is a theorem here too and holds on all three inputs. And $L \le G$ fails at 1,997 of 2,000 holdout draws, by up to 0.60 kelvin against a spread of 3.11, and at 19,971 of 20,000 reference draws: a tangent plane of a saddle lies above the surface in one direction and below it in the other. The surrogate is an interpolant here and nothing more, and the labels say so: Q3 certified, Q1 and Q2 estimates, nothing empirical.

22.6 What surprised

The congested regime did not violate at all. Every line loading in the congested regime is a non-convex function of the loads, and yet on 12,000 fresh draws not one lay below the surrogate. The data are consistent with an explanation but do not prove it. With every line active, G is the sum of the twenty loadings less constants, and its curvature is the sum of theirs; a sum of loadings that bend both ways can be convex in aggregate over the box even when no single loading is. The intermittent regime removes that protection. A line that is not overloaded at a census point contributes nothing to that point's plane, and since its positive part is never negative, it can only leave the plane below G elsewhere. A line that is overloaded there contributes its linearization, and where its loading bends downward the linearization rises above it. A plane in the intermittent regime therefore stays below G only if the partial sum of the loadings active at its census point is convex, a sum over fewer lines, which averages less. The violations in the intermittent regime are the model's non-convexity showing through.

More census points, more violations. On a convex model a larger census is always better: every plane is below G and each one added lifts L closer. Here the residual shrinks with the census, as it should, but the count of violations grows, 140, 205, 296, because each plane added is one more chance to cross a loading that bends down. The size of the worst violation barely moves. A larger census buys a better interpolant and more places where it is not a floor.

The floors that were not floors still held. In the intermittent regime every floor the questions asked for lay below its reference: the means under both new beliefs, the what-if curve at every grid point, the exceedance probabilities at every level, the tail measures and the worst case. The label is still estimate, and rightly. The rule does not ask whether the floors happened to hold on the questions asked; it asks what they rest on, and here they rest on an inequality that failed 1,354 times.

The census does not always beat the sampler. On three entries of Table 22.2, on the value at risk of the intermittent regime and on both comparisons of the duct heater the sampling route is the more accurate. None is a large loss. The census route answers all nine questions for what the sampler spends on one or two of them, and that is its whole advantage on a model without convexity.

The guarantee that survives screens but does not rank. The derivative bound excluded b2 with a guarantee in the congested regime. On the load that matters it says little: in the intermittent regime b3's bound is 2.66, above one and therefore empty, because the gradient with respect to b3 is large exactly where the overload is positive, and the bound charges the full mean square gradient against the variance. The bound is sharp where gradients are small and uniform, and it is built for saying what can be dropped, not how much the rest matter.

22.7 Reproduction

The study lives in the repository directory

```
docs/terra_graph_engine/case_studies/foundation_power_flow/
    load_uncertainty_queries_ac/
```

and from that directory

```
PYTHONPATH=src python run.py
```

rebuilds `results.json`, `timings.json` and the pinned arrays with seed 20260909, in about 41 minutes on a workstation: 200,570 AC solves including the references, 22,000 factorizations and 79,200 duct-heater solves, with no fallback start. The results used here are those of commit `5a9493c3` of 9 September 2026. The finite-difference series of Figure 22.2 come from the AC part of the sibling study `rating_uncertainty_surrogate`, seed 20260906. A replica test, `test_case_study_replica_ac.py`, recomputes the answers from the pinned arrays. From the book's directory,

```
python scripts/ch22_ac.py
```

prints every number in this chapter, checks that the solve ledger adds up to the committed totals, and writes the two data figures.

Chapter 23

The worst k outages

Chapter 15 asked which k failures would hurt most and left the question, beyond the smallest systems, as a search that the prior does not shorten. This chapter works that question on a published benchmark. The network is the RTS-GMLC test system. The consequence of losing a set of components is the expected load that an operator with full re-dispatch still cannot serve, averaged over two hundred published background-outage scenarios. The budgets run from one component to ten. The chapter does four things with the problem. It reproduces the published answers exactly. It proves the answers for one and two components by evaluating every alternative. It tests whether a bound-guided search finds the answers for larger budgets more cheaply than a sorted list does, and finds that it does not. And it conditions the scenario distribution on evidence of a fuel shortage or a storm, and shows the worst outage moving without a single further linear program. Every number in the chapter is read from the committed results of the two case studies by the script `ch23_nk.py` in the book's `scripts` directory, which re-runs no physics.

23.1 The question

A planner has a network in which some components are already out. The planner does not know which, but has a set of plausible background states s, each with a weight. The question is which k further components, if lost, would cost the most load shed on average over those states. Let A be a set of k candidate components, and let $Q(s, A)$ be the minimum load shed, in megawatts, when every component out in state s and every component in A is out of service. The objective and its maximizer are

$$J(A) \;=\; \sum_{s} w_s \, Q(s, A), \qquad A_k^{\star} \;=\; \operatorname*{arg\,max}_{|A|=k} J(A), \qquad k = 1, \ldots, 10. \tag{23.1}$$

This is the worst-case question of Chapter 15 in its stochastic form. The worst set is still the worst whatever its probability; what is new is that its damage is judged against a distribution of background states rather than against one intact network. The formulation is that of Sundar, Mastin, Garcia, Bent and Watson [89], who solve it with a mixed-integer master problem and lazily added cuts; an earlier probabilistic $N - k$ formulation is in [88].

Four questions follow, and the chapter answers them in order. Can the published optima be reproduced by an independent implementation? Are they global optima, or only the best found? Can they be found without an integer program, by searching the sets directly? And how do they move when the distribution of background states is conditioned on evidence?

23.2 The system

The RTS-GMLC network has 73 buses in three areas, 96 active generators and 120 active branches. Every generator and every branch is a candidate, so there are 216 candidate components. Installed capacity is 9,076 MW against a load of 8,550 MW, a reserve margin of 526 MW, or 6.2 percent of load. One unit, g74 at bus 121, is rated 400 MW; twelve more are rated 350 or 355 MW. The margin is smaller than any one of them. Losing a single large unit therefore spends the whole reserve, and that fact organizes most of what follows.

The background states. The two hundred states are a fixed published sample, each with weight $w_s = 1/200$. They are not per-component failure probabilities, and the study never resamples them. Each state removes between 2 and 6 components, with a median of 5. In 65.5 percent of the states nothing is shed before any interdiction, and the largest shed in any one state is 334 MW. Averaged over all two hundred, the background states alone shed $J(\emptyset) = 16.899$ MW. That baseline is not zero, and any computation of marginal damage that treats it as zero is biased by that amount.

The consequence. For each outage set the consequence $Q(s, A)$ is the optimal value of a direct-current minimum-load-shed linear program with full re-dispatch:

$$
\begin{aligned}
Q \;=\; & \min \textstyle\sum_i p_i^d\,(1 - x_i) \\
& \text{s.t. power balance at every bus, with load } p_i^d x_i \text{ served,} \\
& \quad x_\ell\, p_\ell = \theta_{f(\ell)} - \theta_{t(\ell)}, \quad |p_\ell| \le r_\ell \quad \text{for branches in service,} \\
& \quad p_g^{\min} \le p_g \le p_g^{\max} \quad \text{for generators in service,} \\
& \quad 0 \le x_i \le 1.
\end{aligned}
\tag{23.2}
$$

The variable x_i is the served fraction of load i, p_g a generator's output, p_ℓ a branch flow, θ the bus angles, x_ℓ a branch reactance and r_ℓ its rating. An outage acts on bounds only: a generator out has its output fixed at zero, and a branch out has its flow fixed at zero and its angle coupling released. The matrix of the program is the same for every outage set, which is what lets one factorization be warm-started across millions of solves. A set of outages that splits the network leaves each island to balance through its own rows; no load is shed by rule. Minimum outputs are enforced, so a unit that must run can make a small island infeasible, and it also makes Q non-monotone in A: removing a unit can reduce the shed. A set that is infeasible in any background state is excluded from every maximum and counted, never scored as infinite damage.

Interchangeable components. Two generators at the same bus with the same limits, knocked out by exactly the same background states, are interchangeable: exchanging them changes no $Q(s, A)$. Merging such components leaves 177 classes, 65 of generators and 112 of branches. A candidate set is then a multiset of classes, since both units of a twin pair can be chosen. The published five-unit optimum contains such a pair, the two units at bus 323.

23.3 The model as a factor graph

Figure 23.1 draws the model. Its variables are the availabilities $a_c \in \{0, 1\}$ of the 216 components. Three factors set them. The background factor sets $a_c = 0$ for every component out in state s,

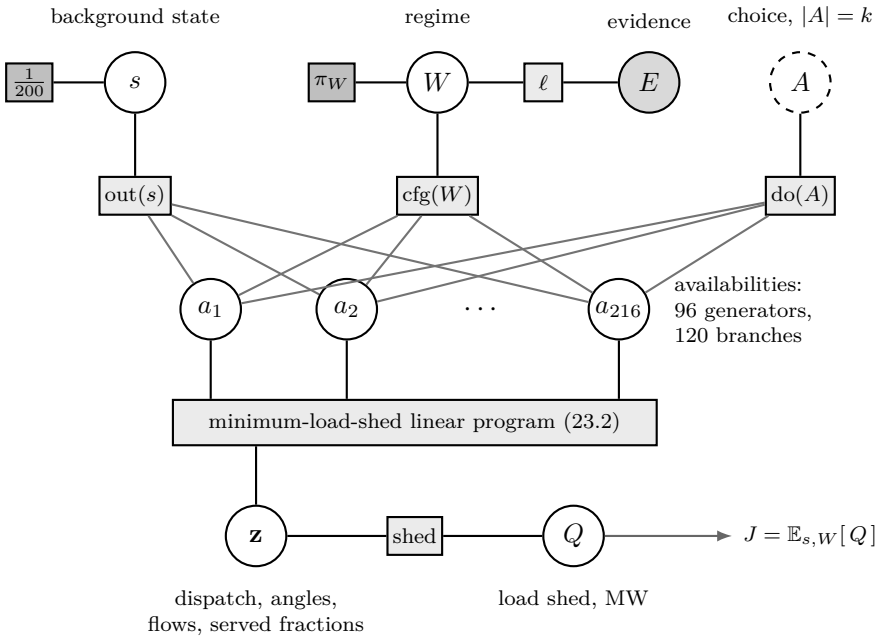


Figure 23.1: The model as a factor graph. Circles are variables and squares are factors; the shaded squares are priors, the shaded circle is the observed evidence, and the dashed circle is the choice, which enters through an intervention. The background state s and the regime W are discrete, and each value of them is a branch; within a branch the shed Q is the value of one linear program. The objective J is the average of Q over the branches, weighted by their probabilities.

and s has the uniform prior $1/200$. The regime factor sets further components out according to a regime variable W; it enters only in §23.4.4, and before that W is fixed at "calm", which adds nothing. The interdiction factor sets $a_c = 0$ for every $c \in A$. The set A is drawn dashed because it is not a random variable. It is a choice, and the factor it enters is an intervention in the sense of Chapter 13: it replaces whatever the availabilities would otherwise be. Below the availabilities, one factor carries the linear program (23.2), connecting them to the dispatch variables $\mathbf{z} = (p_g, \theta, p_\ell, x_i)$, and a readout gives the shed Q.

The graph is the hybrid model of Chapter 7 with one change. The discrete variables s and W split it into branches, one per value, exactly as in §7.2. Within a branch there is no Gaussian: the physics is a set of hard constraints and an optimization, and the branch's consequence is a number, the optimal value. The objective (23.1) is the mixture of those numbers by the branch weights. Two consequences of the structure matter below. First, the branch consequences do not depend on the branch weights, so a change of weights, which is what evidence does, costs arithmetic and no solves. Second, the linear program's value is convex in the capacities that bound it (§14.7), so first-order estimates of the damage built from its dual prices bound the damage from below and never from above.

23.4 Method

23.4.1 Replication

The study vendors the published inputs and results, 24 files, and checks each against its SHA-256 hash on every run, so an edited input cannot pass as a replication. It then evaluates J at each published optimal set and compares with the published objective. The tolerance is 0.01 MW. The same comparison is made for the deterministic variant of the benchmark, which has no background states. A replication of the objective at the published sets says nothing yet about optimality: it establishes that the two implementations compute the same function.

23.4.2 Proof by complete enumeration

A maximum over a finite set is proved by evaluating every member of it, provided each evaluation is exact. Each $Q(s, A)$ is the value of a linear program solved to optimality, so each $J(A)$ is exact to solver precision. At $k = 1$ the study evaluates all 216 components without merging any. At $k = 2$ it evaluates all 15,602 multisets of classes. Merging is exact, since members of a class give identical objectives, and the study checks this at $k = 1$: the largest spread of J within a class is 8.5×10^{-12} MW. Enumeration stops there because the merged candidate space grows to 913,182 sets at $k = 3$ and 6.9×10^{15} at $k = 10$, each needing up to two hundred linear programs.

23.4.3 Search, and what certifies it

For $k \geq 3$ the study searches the lattice of candidate sets best first. It extends partial sets one component at a time, orders them by a score, solves only the complete sets that survive, and discards a partial set whose score falls below the best complete set found so far. Section 15.7 stated the rule for such a search: a partial set may be discarded only when an *upper* bound on every completion of it falls below the incumbent. The study has two quantities to hand. The first is an island-capacity bound: within each island of the post-outage network, the shed is at least the island's load minus its surviving capacity. It is a valid lower bound on Q and costs no linear program. The second is the score the search prunes with, a first-order estimate of each component's damage read from the dual prices of the background states' linear programs with nothing interdicted. That is the coefficient the published method places in its cuts. By the convexity of the last section it is a floor for the capacity part of the damage, not a ceiling, and a small counterexample in the study's tests shows it undercutting the true damage of a completion. It is therefore not a valid pruning bound, and a search pruned by it returns a best-found set whose certificate is conditional on the score.

The search has a preset budget of $B_k = 500{,}000$ distinct outage states, where a state is one union of a background state and a candidate set, and so one linear program. Every arm of the study counts its work in the same ledger, so costs compare directly. The comparator is the cheapest method that could possibly work: evaluate $J(\{c\})$ for every single component, sort, and take the top k.

23.4.4 Conditioning on evidence

The second study makes the regime $W \in \{\text{calm}, \text{fuel}, \text{storm}\}$ a variable. Each regime is a distribution over extra outage configurations added to every background state. Calm adds nothing. Fuel takes g74 out with probability 0.9, the reading of a gas-supply restriction that

makes the largest unit likely but not certainly unavailable. Storm acts on the two tie lines that are the only connections between Area 3 and the rest of the network. It takes each out with marginal probability m, with correlation ρ between them, so that both are out with probability

$$\mathbb{P}(\text{both ties out}) \;=\; m^2 + \rho\, m\,(1-m), \tag{23.3}$$

which is 0.195 at the study's $m = 0.3$, $\rho = 0.5$. The correlation is the point: one weather system trips both ties together, and a product of independent outages, m^2, cannot represent that. The two ties are a separator in the sense of Chapter 8, and the storm acts on the separator.

For a single candidate c, each regime has its own objective, the background average taken over the regime's configurations,

$$J_w(c) \;=\; \sum_{\text{cfg}} \mathbb{P}(\text{cfg} \mid w)\, \frac{1}{200} \sum_{s} Q\big(s,\ \{c\} \cup \text{cfg}\big), \tag{23.4}$$

and evidence E enters through Bayes' rule, equation (7.4):

$$\mathbb{P}(w \mid E) \;\propto\; \mathbb{P}(w)\,\mathbb{P}(E \mid w), \qquad J_E(c) \;=\; \sum_{w} \mathbb{P}(w \mid E)\, J_w(c). \tag{23.5}$$

The evidence is modeled as a signal with likelihood p under the unusual regime and $1-p$ under calm, and p is swept from zero to one. The prior probabilities, recovered from the committed posteriors, are 0.2 for fuel against calm and 0.1 for storm against calm. The regime objectives are computed once, over a screened list of candidates, the largest generators and every generator in Area 3, at a cost of 39,314 linear programs. Every posterior after that is a reweighting of numbers already held.

A certified interval. The same study asks what can be certified when the background states come from a physical reliability model rather than a fixed sample. It takes Area 1 alone: 24 buses, 38 branches, 24 thermal generators, 2,850 MW of load and 3,018 MW of capacity. Each of the 62 components fails independently with its own probability, between 4.6×10^{-4} and 0.12, for an expected 1.237 outages. This variant lets a generator's output fall to zero, so that no deep-tail state is infeasible. The states are enumerated in decreasing probability until the enumerated mass reaches $1 - \varepsilon$. The consequence of any state is at most the area's load $q_{\max}$, so, as in the risk bound of Proposition 15.1, the unenumerated tail adds at most its probability times the largest consequence, $\varepsilon\, q_{\max}$, to J. Each candidate's objective is then an interval $[J^-, J^- + \varepsilon q_{\max}]$, and the incumbent is certified when its lower end clears every rival's upper end by a guard band of 2×10^{-6} MW, which exceeds the agreement floor of two independent solvers.

23.5 Results

Table 23.1 lists the ten budgets. For each it gives the objective, its distance from the published value, the number of distinct sets within 10^{-4} MW of it, the increase over the previous budget, how the objective is established, and the number of distinct states the bound-guided search evaluated.

Table 23.1: The worst k outages on RTS-GMLC. $J_k^\star$ is the expected shed at the optimal set; $|\Delta z|$ its distance from the published value; "optima" the number of distinct sets within 10^{-4} MW of it; "increment" is $J_k^\star - J_{k-1}^\star$, with $J_0^\star = J(\emptyset)$. "Proved" means established by complete enumeration; "conditional" means found by a search whose certificate rests on a score that is not a valid bound. †The search stopped at its budget of 500,000 states.

k	$J_k^\star$ (MW)	$\lvert\Delta z\rvert$ (MW)	optima	increment	status	search states
1	151.8247	3.5×10^{-5}	1	134.926	proved	12,062
2	481.3227	1.5×10^{-5}	1	329.498	proved	165,841
3	835.3350	2.8×10^{-12}	11	354.012	conditional	500,173†
4	1190.3350	3.9×10^{-12}	14	355.000	conditional	500,118†
5	1545.3350	4.1×10^{-12}	11	355.000	conditional	500,072†
6	1900.3350	1.4×10^{-12}	5	355.000	conditional	500,032†
7	2255.3350	1.4×10^{-12}	1	355.000	conditional	500,164†
8	2605.3350	4.5×10^{-13}	1	350.000	conditional	500,025†
9	2942.5850	1.4×10^{-12}	1	337.250	conditional	500,044†
10	3274.5100	9.1×10^{-13}	1	331.925	conditional	500,023†

Replication. All ten stochastic objectives agree with the published values to 3.5×10^{-5} MW, against a tolerance of 0.01 MW. The residual is the rounding of the published files, which carry four decimals. The deterministic variant agrees to 1.8×10^{-12} MW at every budget, from 15 MW at $k = 1$ to 3,069 MW at $k = 10$. Two implementations with different solvers and no shared code compute the same function.

Proof at one and two. At $k = 1$ the enumeration evaluates all 216 components with 42,426 linear programs. The maximum is g74, at 151.8247 MW, the published answer. No other component comes within 1 MW of it, so the optimum is isolated. One branch is excluded as infeasible. At $k = 2$ the enumeration evaluates all 15,602 merged pairs with 3,040,242 linear programs and excludes 180 infeasible pairs. The maximum is $\{\text{g18}, \text{g74}\}$, at 481.3227 MW, the published answer again. It wins narrowly. The runner-up, $\{\text{g71}, \text{g74}\}$, is 0.4754 MW behind, and the next three are g74 paired with another 355 MW unit, each within 0.52 MW. These two results are proofs. The published methods do not establish global optimality on their own: the exact formulation assumes bounds on dual variables that it does not compute, and the heuristic's cuts are of the same non-valid family as the search score above. Complete enumeration establishes it, and it confirms the published answers.

Many optima from three on. At $k = 3$ there are eleven distinct sets within 10^{-4} MW of the optimum; at $k = 4$, fourteen; at $k = 5$ and $k = 6$, eleven and five. The large units are nearly interchangeable in value, and several combinations of them tie. That is why the optimal sets are not nested, and why the published three-unit set, $\{\text{g57}, \text{g71}, \text{g74}\}$, does not contain the published two-unit set's g18. The search's best set at $k = 3$ is a different member of the same tie. Recovery is therefore judged by whether the published set's class is among the tied optima, and it is at all ten budgets.

The increments. From $k = 4$ to $k = 7$ each added unit raises the objective by exactly 355.000 MW, the capacity of the unit added. At $k = 8$ the increment is 350.000 MW, the

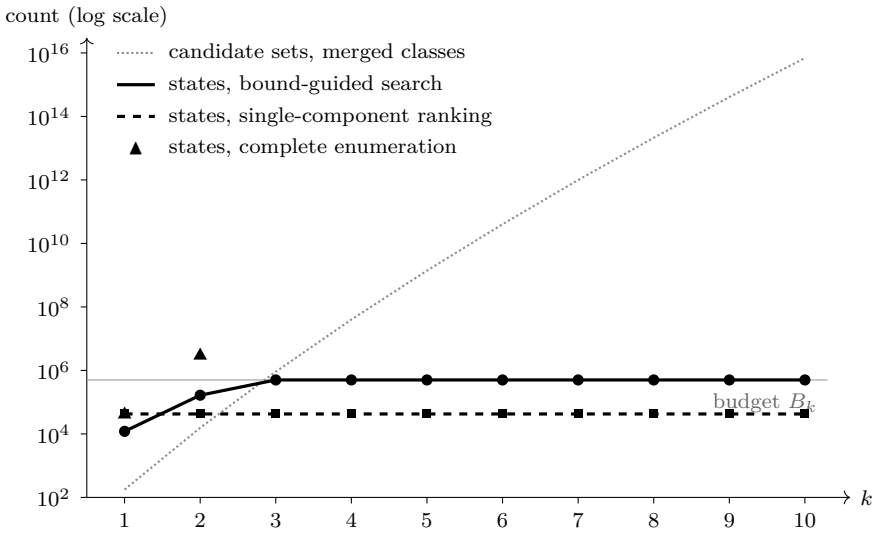


Figure 23.2: Cost of finding the worst k outages, on a logarithmic scale. Dotted: the number of candidate sets after merging interchangeable components. Solid: distinct outage states, each one linear program, evaluated by the bound-guided search. Dashed: the same count for ranking single components and taking the top k. Triangles: complete enumeration at $k = 1$ and $k = 2$; at $k = 1$ it coincides with the ranking, since both evaluate every single component. The horizontal line is the search's budget of 500,000 states.

capacity of g20. At $k = 9$ and $k = 10$ the increments, 337.250 and 331.925 MW, fall short of the added units' 355 MW, because those units are already out in some of the background states, where removing them again costs nothing. Past the second unit, every megawatt removed is a megawatt shed in nearly every state. The network hardly matters: the island bound, which ignores flows inside an island, is within 4.125 MW of the exact objective from $k = 3$ onward, and at those budgets the network, rather than the capacity, limits the shed in only 15 of the 200 states. At $k = 0$ the same bound is 32.9 percent low and at $k = 2$ 1.06 percent low. It is loose where the answer is small and tight where the answer is large.

The search against a sorted list. Figure 23.2 plots cost against the budget. The horizontal axis is k; the vertical axis is a count on a logarithmic scale, from 10^2 to 10^{16}. The dotted curve is the size of the merged candidate space, which rises by one to two decades per unit of k. The solid curve is the bound-guided search. At $k = 1$ it evaluates 12,062 states, fewer than the comparator. At $k = 2$ it evaluates 165,841, nearly four times the comparator. From $k = 3$ it runs into the budget line and stops there, having evaluated about 2,535 candidate sets each time. At $k = 3$ that is 2.8×10^{-3} of the merged space; at $k = 10$ it is 3.7×10^{-13}. The dashed curve is the comparator. It evaluates 42,426 states at $k = 1$ and 42,626 at every larger budget, the extra two hundred being the one evaluation of the chosen set. The top k single components reproduce the published objective at every budget from one to ten. At $k \geq 3$ the search spends 11.7 times the comparator's work and ends with a weaker statement: its best-found sets match the published objective, but no bound on the gap to the true optimum is available, and none is reported. The two triangles show what proof costs: 42,426 states at $k = 1$ and 3,040,237 at $k = 2$.

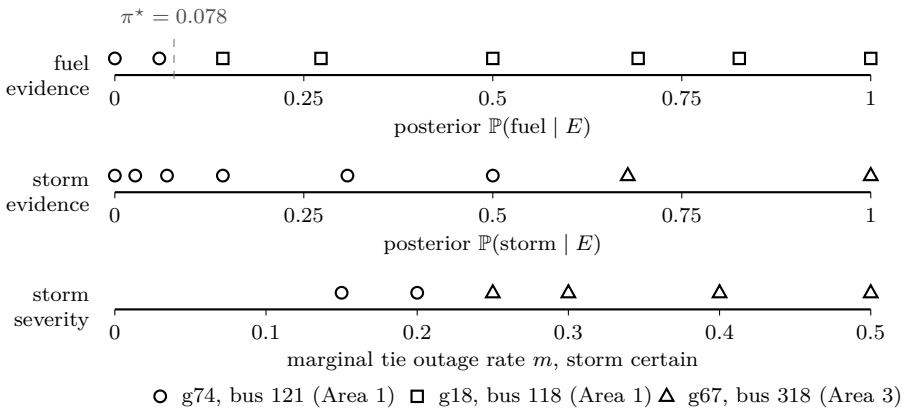


Figure 23.3: The worst single outage under evidence. Top: against the posterior probability of a fuel restriction on g74; the dashed tick $\pi^\star$ is where g18 overtakes g74 by the two-branch arithmetic of the text. Middle: against the posterior probability of a correlated storm on the two Area 3 tie lines, at $m = 0.3$, $\rho = 0.5$. Bottom: against the storm's marginal tie outage rate m, with the storm certain. No linear program is solved after the first evaluation of the three regimes.

Evidence moves the answer. Figure 23.3 shows where the worst single outage sits as evidence accumulates. It has three strips. The top two place the optimum against the posterior probability of the fuel regime and of the storm regime, at the eight evidence strengths of the sweep; the bottom strip places it against the storm's severity m, with the storm certain. The mark says which unit is the optimum: a circle for g74 at bus 121, a square for g18 at bus 118, a triangle for g67 at bus 318 in Area 3.

Fuel evidence moves the optimum early. At a posterior of 0.059 the optimum is still g74; at 0.143 it is g18, and it stays there. The move follows from the branch arithmetic of equation (23.5). Under the fuel regime g74 is probably out already, so removing it again adds nothing: $J_{\text{fuel}}(\text{g74}) = J_{\text{calm}}(\text{g74}) = 151.825$ MW. Removing g18 under the fuel regime removes the pair nine times in ten, so $J_{\text{fuel}}(\text{g18}) = 0.1 \times 126.844 + 0.9 \times 481.323 = 445.875$ MW. The mixture puts g18 above g74 once the posterior of fuel exceeds $\pi^\star = 0.078$, inside the bracket the sweep finds. The move costs no linear program. The pair value it uses was already computed.

Storm evidence moves the optimum late, and into a different area. At a posterior of 0.50 the optimum is still g74. At 0.68, the next point of the sweep, it is g67 in Area 3. The move needs a severe storm as well as a likely one. With the storm certain, the optimum stays on g74 for $m = 0.15$ and $m = 0.20$ and moves to g67 from $m = 0.25$, where both ties fail together with probability 0.156. Within the storm, the worst single outage sheds 170.81 MW against the calm optimum's 151.82 MW. The margin is thin, and that is why the posterior must be well above one half. With both ties forced out, the worst single outage sheds 350.31 MW, and five Area 3 units tie for it, two of which, g68 and g72, appear in none of the published optimal sets of four or fewer units. A storm that can trip only one tie does not move the optimum: the remaining tie carries what Area 3 then needs, and g74 stays worst. The reproduction check comes first, and it passes. With the regime pinned to calm, the conditioned computation returns all ten published objectives to 3.5×10^{-5} MW with intervals of zero width.

Table 23.2: Certified intervals on Area 1, with the states enumerated to 99 percent of the probability. Each objective lies in $[J^-, J^+]$ with $J^+ - J^- = 28.50$ MW. "Gap" is the incumbent's lower end minus the runner-up's; the incumbent is certified when the gap exceeds the interval width.

k	incumbent	$[J^-, J^+]$	runner-up	$[J^-, J^+]$	gap	certified
1	{g74}	[427.38, 455.88]	{g20}	[397.55, 426.05]	29.82	yes
2	{g20, g74}	[746.63, 775.13]	{g18, g74}	[737.17, 765.67]	9.46	no

What can be certified. In Area 1, enumeration to 99 percent of the probability takes 9,714 states; to 99.9 percent, 111,873; to 99.99 percent, 829,001. At the 99 percent level the truncation slack is $\varepsilon q_{\max} = 28.499$ MW. Table 23.2 gives the two leading intervals at $k = 1$ and $k = 2$. At $k = 1$ the incumbent, g74, has a lower end 29.824 MW above the runner-up's, more than the slack, so the intervals do not overlap and the optimum is certified, by 1.3 MW. At $k = 2$ the two leading pairs are 9.459 MW apart at their lower ends, far less than the slack, and the optimum is not certified. The gap would clear the 2.850 MW slack of the 99.9 percent enumeration, so the pair is certifiable in principle with 111,873 states for each of 276 candidate pairs. That run is not in the results. In this model the leading pair is $\{\text{g20}, \text{g74}\}$, not the benchmark's pair: a different network, a different outage distribution and a different treatment of minimum outputs give a different answer, and the method reports it with its interval.

23.6 What surprised

A sorted list beat the search. The search was built to exploit interactions between failures, and on this system the interactions carry no information about the ranking. Take the 13 units rated 350 MW or more, and for each of their 78 pairs compute the interaction $I(a, b) = J(\{a, b\}) - J(\{a\}) - J(\{b\}) + J(\emptyset)$. All 78 are positive. They range from 183.17 MW, for g9 with g43, to 219.55 MW, for g18 with g74, with a mean of 197.05 MW. The pairs are strongly super-additive, because the first unit removed spends the 526 MW margin and the second is then shed almost in full. But the interaction is nearly the same for every pair. A nearly constant interaction adds nearly the same amount to every pair's value and reorders nothing, so the best pair is the two best singles, and the best k-set is the k best singles. On a network with enough transmission, generation is fungible, and the damage a set does is set by one scalar, the reserve it removes. The problem is a knapsack. A search that looks for interacting failures finds none worth finding, and pays for looking.

The same fact explains why the obvious structural screen fails. Ranking components by the damage each does alone to the intact network, the deterministic $N - 1$ severity, finds none of the ten optimal sets and falls short by up to 2,892.6 MW. To contain the ten optimal units, that ranking must be read to its 80th entry of 216; ranking by the components' expected damage over the background states needs only the first 11. On the intact network a large unit's loss is absorbed by re-dispatch and does no damage; what is severe there is a branch whose loss islands a load. Under background stress the large unit is the worst thing to lose. The two measures answer different questions.

Uniform weights hide the interactions; evidence uses them. The super-additivity that the search could not exploit is what moves the optimum under fuel evidence. The interaction of

g18 with g74 is 219.55 MW, and once g74 is probably out, that interaction is charged to g18. It is enough to move the optimum at a posterior of 0.078. Under uniform weights over a fixed sample, every set is scored against the same background, and the interactions shift every value alike. Under a posterior that concentrates on one background, they stop shifting alike, and the ranking changes.

The storm's threshold has two dimensions. The storm moves the optimum into Area 3 only when it is both severe and well evidenced: a tie outage rate of at least 0.25 with the storm certain, and a posterior between 0.50 and 0.68 at the study's severity. A single degraded tie never moves it. A threshold on either axis alone would have said the wrong thing about the other.

The flat objective resists certification. The same flatness that makes ranking work makes certification hard. At $k = 1$ the leading unit is ahead by enough to clear a 99 percent truncation. At $k = 2$ the leading pairs differ by a third of the slack. A certificate is a margin between rivals, and a knapsack with many near-equal items has small margins.

23.7 Reproduction

The figures and tables of this chapter are regenerated, from the committed results alone, by

```
cd docs/book
python scripts/ch23_nk.py
```

which writes `figures/ch23_*.tex`, the two figures and two tables of the results section, and prints every number quoted here.

The results themselves come from the study drivers, run from each study's directory under `docs/terra_graph_engine/case_studies/foundation_power_flow/` with the repository's `src` directory on the Python path. In `stochastic_nk_interdiction`:

```
python snk_run.py --symmetry-classes    # symmetry_classes.json
python snk_run.py --level1              # level1_replication.json
python snk_run.py --k1-exhaustive       # k1_exhaustive.json
python snk_run.py --k2-exhaustive       # k2_exhaustive.json
python snk_bench.py --benchmark         # benchmark_k1_10.json
python snk_bench.py --bound-gap         # bound_gap.json
python snk_ablation.py --ablation --kmax 3 --exhaustive-kmax 1
python snk_lattice.py                   # lattice_ablation.json
python snk_structure.py --h4 --seed 20260825
```

and in `inference_conditioned_nk_interdiction`:

```
python icn_run.py       # reduction, evidence_sweep, decision_boundary
python icn_s6_gate.py   # enum_separability_gate.json
```

The solver is HiGHS, warm-started by changing bounds on one assembled program; one warm solve takes 1.78 ms and one island bound 0.245 ms on the workstation that produced the results. On that workstation the $k = 1$ enumeration took 70.5 s and the $k = 2$ enumeration 5,694 s; wall-clock times are one machine's, and the state counts are the portable measure of cost. Nothing

in either study samples: the background states are a fixed file, the candidate space is enumerated in a fixed order, and J is summed in a fixed scenario order, so a re-run reproduces the committed results to every digit they carry. Every driver verifies the 24 vendored reference files by hash before it computes anything.

Chapter 24

Partitioning in practice

Chapter 18 measured what a cut costs when the regions' messages are Gaussian, and closed by reading four numbers from studies in which the regions are sets of failed components. This chapter is those studies. They share one probability layer and one question: when a network of independent, failure-prone components is cut into regions, does the cut make the probability of an overload cheaper to compute, and by how much? The first study measured, before any algorithm was built around the answer, how much a region's failure states collapse at its ports on four real networks, and it judged the answer against a threshold fixed in advance. The second built the two-region route on the one system where the collapse is large and compared it, at equal certified accuracy, with a single enumeration of the whole. The third took that system to chains of up to six regions, whose joint state count no enumeration reaches. The fourth ran the same code on a chain of data halls. The threshold was not met, and that negative result is as much the finding of the chapter as the positive ones. Every number in the chapter is read from the committed results of the four studies by the script `ch24_partitioning.py` in the book's `scripts` directory, which re-runs no physics.

24.1 The question

An operator wants the probability that at least one branch of a network carries more than its rating, when every generator and branch fails independently with a known probability. Chapter 15 computed that probability by enumerating failure sets in order of decreasing probability and certifying the result with the mass not yet enumerated. The number of failure sets needed for a given certified width grows exponentially with the number of components.

A cut offers a way around the growth. By Chapter 5, the interiors of two regions are independent given their ports. So a region's failure state matters to the rest of the network only through what it does at the ports: its equivalent at the boundary buses and the power it adds to or takes from the shared balance. Call that the region's *separator value*. If many failure states produce the same separator value, the region *compresses*. Its $|E_r|$ enumerated states become $|S_r|$ distinct separator values, and two regions combine over $|S_r| \times |S_s|$ pairs instead of $|E_r| \times |E_s|$. The compression ratio is

$$C_r \;=\; \frac{|E_r|}{|S_r|}, \tag{24.1}$$

the number of enumerated region states per distinct separator value.

The four studies asked four questions.

Table 24.1: The systems of the four studies. Components are generators and branches that can fail; branches on a cut are held in service and not counted. k_r is the number of boundary buses of a region. ε_r is the residual probability mass at which each region's enumeration stops, and the cap is the largest number of states it may enumerate.

system	size	cuts, k_r	operating point	enumeration
RTS Area 1	24 buses, 62 comp.	spectral, refined, four random, one bad; 3–12	load 0.6, pro rata	$\varepsilon_r = 10^{-1}, 10^{-2}$; cap 10^5
RTS-GMLC	73 buses, 193 comp.	areas, 2–4; spectral, 5–6; random, 29–33	load 0.6, pro rata	same
IEEE 118	118 buses, 240 comp.	spectral, 8–9; random, 47–52	load 0.6, pro rata	same
PEGASE 1354	1,354 buses, 2,251 comp.	spectral, 13 and 17	load 0.6, pro rata	$\varepsilon_r = 10^{-1}$; cap 5,000
two replicas of Area 1	48 buses, 124 comp.	the replicas, 1 or 2	load 0.920, merit order	$\varepsilon_r = 10^{-1}$ to 10^{-3}
chains of R replicas	48–144 buses, 124–372 comp.	the replicas, 1	load 0.6, pro rata	$\varepsilon_r = 10^{-2}$
two data halls	12 comp.	the halls, 1	–	exhaustive
three data halls	28 comp.	the halls, 2	–	$\varepsilon_r = 10^{-3}$

1. Do the cuts of real networks compress? The criterion was fixed before measuring: the answer is favorable if the median C_r over regions with at most three ports is at least 10 on at least two realistic grids, and if the three-area cut of the RTS-GMLC system needs at most 10^6 interface pairs.
2. Where compression holds, is the two-region route exact, and what does it cost against one enumeration of the whole at the same certified width?
3. Does the route extend beyond two regions, to systems whose joint state count is out of reach of any enumeration?
4. Does the same code run unchanged on a system that is not a power network?

24.2 The systems

Table 24.1 lists the systems. Each row gives the size, the cuts tried with the number of ports k_r per region, the operating point, and how deeply each region was enumerated. The table is read row by row; the paragraphs below say why the rows differ.

The four grids of the first block are the Reliability Test System's Area 1 on its own, the full three-area RTS-GMLC system, the IEEE 118-bus system and the 1,354-bus PEGASE model. The two RTS systems carry published failure rates. The IEEE 118 and PEGASE systems have none, and the study borrows the RTS class rates by component type. All four are dispatched pro rata at sixty percent of peak load: every generator's output scales with its capacity. Each region is measured against its complement with the two-region evaluator described below.

The second block is built from Area 1. Two copies joined by one tie line, or by two, form a system of 124 components whose regions are identical, so every twin unit in one region has a twin in the other. The two-replica system is dispatched at its peak snapshot, with a load of

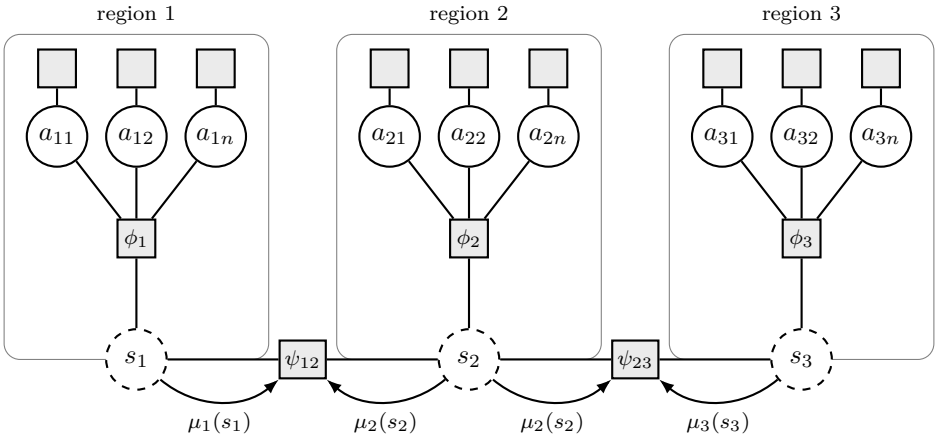


Figure 24.1: The failure model of a chain of three regions as a factor graph. Circles: component availabilities, each with a prior factor (small squares). ϕ_r: the region's physics, mapping its failure state to its separator value s_r (dashed), which sits on the region boundary. $\psi_{r,r+1}$: the tie between neighbors. Arrows: the messages, one histogram over separator values per region.

0.920 of peak served in merit order. The chains join R copies slack bus to slack bus, so each region has a single boundary bus even when it has two neighbors. They are dispatched pro rata at sixty percent, with ties rated 500 MW. The two systems are different. At peak dispatch the two-replica probability of any overload is near 0.68; the two-region member of the chains has 0.1227 on its enumerated mass. The chapter keeps them apart throughout.

The third block is a chain of data halls modeled as a steady thermal network: racks with heat loads, cooling units with capacity, and plenum, rack and wall paths with a conductance and a heat-flow limit. A cooling unit fails with probability 0.03 and a path with probability 0.002. Two halls are joined by one wall path; three halls by two wall paths at each interface.

The physics is the same in every power system. A failure state removes its components. The lost generation L is re-supplied by the surviving units in proportion to their headroom H, at a common participation $\alpha = \min(L, H)/H$, and any excess $D = \max(L - H, 0)$ is a deficit taken at the slack bus. Flows follow from the DC model. An overload is a flow above the branch's continuous rating, beyond a small dead-band. Islanded states count as islanding, not as overloads. The thermal system enters through an adapter of about forty lines that maps conductance to susceptance, heat to power, a cooling unit to a generator and a path's heat-flow limit to a rating; the probability layer never sees a room.

24.3 The model as a factor graph

Figure 24.1 draws the model for a chain of three regions. Each region holds the availabilities $a_{r1}, \ldots, a_{rn}$ of its components, binary variables, each with a prior factor. One factor per region, ϕ_r, is the region's physics: it joins every availability of the region to the region's separator value s_r, drawn dashed because it is a summary and not a component. The separator value sits on the region's boundary. Tie factors $\psi_{r,r+1}$ join neighboring separator values; they carry the tie flow and whether it is within its rating. Messages μ_r run from each separator value into the tie factors.

The figure shows two things. The region interiors meet only at the separator values, so the graph is a chain of regions and the messages are exact. And the message is the only object that crosses a boundary, so its size, the number of distinct separator values, is the cost.

In formulas, write X_r for region r's failure state and $m_r(X_r) = s_r$ for its separator value. In the power studies s_r is the region's Ward equivalent at its boundary buses together with its lost generation L_r and headroom H_r, rounded to 10^{-6} MW. The equivalent is exact for DC physics, because injections enter linearly. The region's message is the probability of each separator value over the enumerated set E_r,

$$\mu_r(s) \;=\; \sum_{X_r \in E_r,\; m_r(X_r)=s} \mathbb{P}(X_r). \tag{24.2}$$

Given both separator values, whether region A overloads depends on X_A and on s_B alone. So one table per region,

$$T_A(s_A, s_B) \;=\; \sum_{X_A :\, m_A(X_A)=s_A} \mathbb{P}(X_A)\, \mathbb{1}\big[\text{no overload in } A \mid X_A, s_B\big], \tag{24.3}$$

gives the probability of no overload on the enumerated mass as

$$\sum_{s_A} \sum_{s_B} T_A(s_A, s_B)\, T_B(s_B, s_A)\, \mathbb{1}\big[\text{tie within rating} \mid s_A, s_B\big]. \tag{24.4}$$

Each table has $|S_A| \times |S_B|$ entries. The physics is evaluated once per region state, against all of the neighbor's separator values at once, because the region's flows are affine in the boundary condition.

24.4 Method

The method is assembled from earlier chapters, and each piece keeps the guarantee it had there.

Exactness from separation. That the tables of equation (24.3) lose nothing is Proposition 5.3 of Chapter 5: the ports separate the interiors. The discrete message of equation (24.2) is the region message of Chapter 8 for discrete variables (§8.8), grouped by separator value, which is where it can shrink.

Truncation with a certificate. Each region is enumerated in order of decreasing probability until the unenumerated mass falls to ε_r, as in Chapter 15 (§15.6). The joint unenumerated mass is then exactly $\varepsilon = 1 - \prod_r (1 - \varepsilon_r)$. The certified bracket on the probability of any overload runs from the value on the enumerated mass to that value plus ε, which is Chapter 9's pruning with the pruned weight kept as a bound (§9.5).

Boundary dimension. The number of ports k_r is Chapter 18's cost variable. The first study also tested whether $\sum_r k_r^2$, computable before any physics, ranks candidate cuts the way their measured cost does.

Messages along a path. For chains of single-port regions under pro-rata dispatch, the regions couple only through the sums $\sum_r L_r$ and $\sum_r H_r$, which fix α and D. Each region's message is then a histogram on a common lattice, and the histograms combine by convolution along the chain: the nested messages of Chapter 17 (§17.2) with convolution in place of the Schur complement. A lattice cell holds states with slightly different sums, so a total cell whose states straddle a rating is only possibly overloaded. That ambiguity is bracketed by boxes of cells and narrowed by refining the boxes. The second lattice axis is sheared to $W = H + cL$, with c fitted to the dispatch. Under pro-rata dispatch the fit gives $c = 0.589$ and W is constant at 1,008 MW, so the lattice becomes one-dimensional in L.

Two ports per interface. When a region has two ports, as the three halls do, each region's per-path overload probabilities are computed against the binned distribution of its neighbors' separator values, and the probability of any overload is bracketed below by the largest single-region probability and above by the union bound over region and wall events.

What was run. The first study measured every region of every candidate cut at $\varepsilon_r = 10^{-1}$ and 10^{-2}, and repeated the 10^{-2} measurement with the separator values rounded coarsely, from 10^{-6} MW down to 10^{-1} MW, to see whether round-off was splitting values that should coincide. One cut at one truncation took from under a second on Area 1 to about fifty seconds for the spectral cuts of IEEE 118 at 10^{-2} and 332 and 443 seconds for its random cuts; the whole measurement took 31 minutes. The second study checked exactness on a sixteen-component model, eight per region, with 65,536 joint states, then ran the full 124 components at three truncations against the global enumeration at a budget of 100,000 states and against 20,000 Monte Carlo draws, and checked the per-state physics of 20,000 global states against the regional evaluator. It also re-ran the two-region route after evidence of a storm in one region. The third study checked the chain pass against brute force on truncated models of three and four regions, ran chains of 2, 3, 4 and 6 full regions with four refinement passes each, and repeated the three-region chain under the merit-order dispatch; the sheared chains took 36 minutes. The fourth study compared two halls exhaustively against brute force and bracketed three halls at three lattice resolutions against 20,000 Monte Carlo draws.

24.5 Results

24.5.1 Compression on real cuts

Figure 24.2 plots, for every region of every cut, the compression ratio against the number of ports, both axes logarithmic. Marker shape gives the grid. Filled markers are regions enumerated to the target residual; open markers stopped at the state cap. The shaded corner is the gate's target: at most three ports and a ratio of at least ten.

The target corner is empty except for one point: the two-replica system with one tie, at $C_r = 15.9$. That system was built to have twins and is not one of the realistic grids the gate asks about. On the realistic grids, the ratios lie between one and six and a half at every port count. The IEEE 118 and PEGASE systems have no region with three ports or fewer on any cut. RTS Area 1 has three such regions, with a median ratio of 1.6; the full RTS-GMLC system has one, area 3 with two ports, at 5.7. The three-area cut of RTS-GMLC needs 51,395,594 interface pairs, fifty times the gate's million. The verdict recorded by the study is unfavorable.

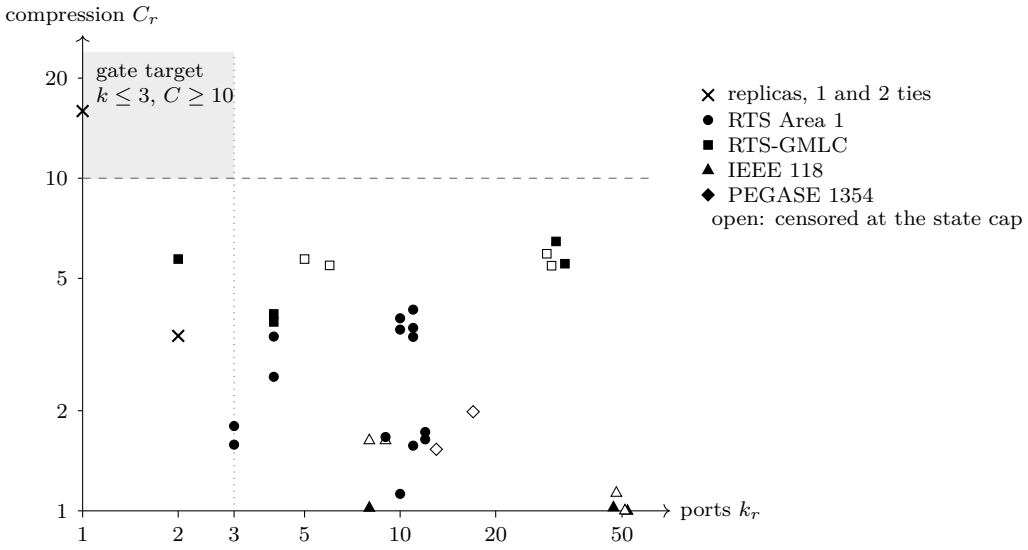


Figure 24.2: Compression ratio C_r against ports k_r for every region measured, at $\varepsilon_r = 10^{-2}$ (10^{-1} for PEGASE). Filled: enumerated to the target residual. Open: stopped at the state cap, so the ratio is censored. Dashed: the gate's $C = 10$. Dotted: $k = 3$. Only the one-tie replica, which is not a realistic grid, enters the shaded target.

Table 24.2 shows where the separator values come from. For selected regions it gives the enumerated states, the number of distinct sets of failed generators among them, the distinct separator values and the ratio.

Three readings follow from the table. First, compression comes from the generators. On RTS-GMLC and the replicas, whose many units come in identical sizes, the separator values number between 8 and 39 percent of the generator sets: a region that loses one of two twin units produces the same boundary equivalent and the same lost power whichever twin it was. On IEEE 118, whose units are all different, the separator values are as many as the states; in region A they outnumber the generator sets, because branch outages split them further.

Second, the ports multiply what the generators leave. The first two rows are the same region under the same dispatch, and its 11,336 states hold the same 8,564 generator sets. Behind one tie they fall on 712 separator values; behind two ties, on 3,373. The third row is the same enumeration inside RTS-GMLC, under the pro-rata dispatch and with four ports, and it falls on 2,900. The region's content sets how far the states can collapse, and each additional port reveals more of the difference between them.

Third, a random cut compresses as much as a good one. The random region of RTS-GMLC has a ratio of 5.5, the same as the spectral region, with 33 ports instead of 6. Compression is a property of what is inside the region; the port count is the property of the cut, and it is the port count that sets the size of every interface table. That random cut's interface tables have 176,410,244 pairs.

The coarse rounding changed almost nothing. Rounding the separator values to 10^{-1} MW instead of 10^{-6} MW took IEEE 118's region B from 61,172 values to 59,727, and area 3 of RTS-GMLC from a ratio of 5.72 to 6.04. Separator values differ by megawatts, not by round-off.

What the cut's geometry did predict was which cut is cheapest. The study ranked the candidate cuts of each grid by a measured cost, a weighted sum of $\sum_r k_r^2$, the enumerated states,

Table 24.2: Where separator values come from, at $\varepsilon_r = 10^{-2}$ (10^{-1} for PEGASE). $|E_r|$: enumerated states, with the residual mass reached. Generator sets: distinct sets of failed generators among them. $|S_r|$: distinct separator values. A star marks a region stopped at its state cap, whose ratio is censored.

network	region	k_r	$\lvert E_r\rvert$ (residual)	generator sets	$\lvert S_r\rvert$	C_r
replica, one tie	A	1	11,336 (0.010)	8,564	712	15.9
replica, two ties	A	2	11,336 (0.010)	8,564	3,373	3.4
RTS-GMLC, areas	area 1	4	11,336 (0.010)	8,564	2,900	3.9
RTS-GMLC, areas	area 3	2	13,736 (0.010)	9,993	2,402	5.7
RTS-GMLC, spectral	A	6	100,000 (0.023)*	76,714	18,275	5.5*
RTS-GMLC, random	A	33	57,881 (0.010)	47,582	10,462	5.5
IEEE 118, spectral	A	8	30,452 (0.010)	16,451	29,781	1.0
IEEE 118, spectral	B	8	100,000 (0.032)*	72,709	61,172	1.6*
IEEE 118, random	A	47	33,188 (0.010)	23,960	32,322	1.0
PEGASE, spectral	A	17	5,000 (0.971)*	5,000	2,516	2.0*
PEGASE, spectral	B	13	5,000 (1.000)*	5,000	3,263	1.5*

the interface pairs and the imbalance of region sizes, and compared that ranking with the ranking by $\sum_r k_r^2$ alone. The two agree on the best cut of RTS Area 1, RTS-GMLC and PEGASE, with rank correlations of 0.98 and 0.94 on the first two; PEGASE had only two candidates. On IEEE 118 the measured best was the weighted refinement of the spectral cut and the port count preferred the unrefined one, at measured costs of 2,068 and 2,080, a near tie between two cuts whose regions were both censored.

24.5.2 Two regions at peak load

The exactness check came first. On the sixteen-component model, with 65,536 joint states, the two-region route reproduced every metric of the global enumeration to 3.5×10^{-15} in the probability of any overload and 4.5×10^{-12} MW in expected severity, with one tie or with two. It needed 510 physics evaluations against 65,536, and 0.22 seconds against 21.2. Each region's 255 states fell on 16 separator values behind one tie, a ratio of 15.9, and on 104 behind two, a ratio of 2.45.

Table 24.3 gives the full 124-component system at three truncations, beside the global enumeration and the Monte Carlo reference. The columns are the states enumerated, the physics evaluations, the interface pairs, the certified bracket on the probability of any overload, its width, and the wall-clock. Chapter 18 quoted the middle row; the table adds the ladder around it.

Figure 24.3 plots the width against the states for both routes. The regional line lies two orders of magnitude to the left of the global one. At a width near 0.02 the global enumeration needs five million states and the regional one 22,672, about 221 times fewer. Below that the global route has no point at all: five million states was as far as it was run, and the regional route reaches 0.002 with 271,032. Two effects add up here. The regional route orders sixty-two components' failure sets twice instead of 124 components' once, and the product of two ordered lists covers the joint space far faster than one ordered list does; and each region's 11,336 states collapse onto 712 separator values, so the pair tables stay near a million entries.

The brackets agree with the references. At $\varepsilon_r = 10^{-1}$ and 10^{-2} they contain the Monte Carlo estimate. At 10^{-3} the bracket $[0.6806, 0.6826]$ lies above the estimate 0.6789 but inside its

Table 24.3: The two-replica system at peak load, one tie, 124 components. Brackets are certified: their width is the unenumerated mass. The Monte Carlo row gives the point estimate and its 95 % Wilson interval. The five-million-state global run reports only the mass it reached.

route	states	physics	pairs	$\mathbb{P}$(any overload)	width	time
global, budget 10^5	100,000	100,000	–	$[0.5765, 0.6879]$	0.1114	24 s
global	5,000,000	–	–	–	0.0191	–
two regions, $\varepsilon_r = 10^{-1}$	756	756	13,448	$[0.5103, 0.7002]$	0.1899	0.2 s
two regions, $\varepsilon_r = 10^{-2}$	22,672	21,916	1,013,888	$[0.6638, 0.6837]$	0.0199	19 s
two regions, $\varepsilon_r = 10^{-3}$	271,032	248,360	15,859,712	$[0.6806, 0.6826]$	0.0020	646 s
Monte Carlo	20,000	8,361	–	0.6789 $[0.6724, 0.6853]$	–	4 s

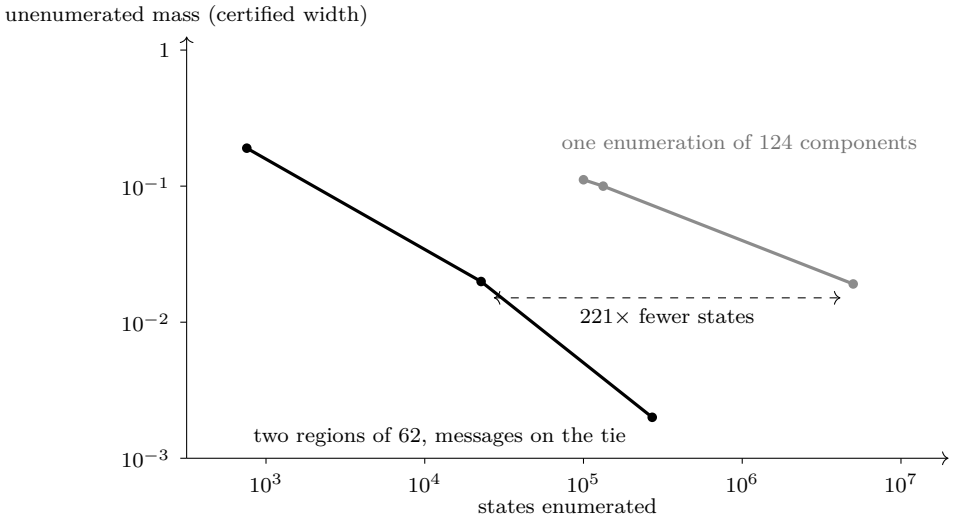


Figure 24.3: Certified width, the unenumerated mass, against the number of states enumerated, for the two-replica system at peak load, both axes logarithmic. Grey: one enumeration of all 124 components, at 100,000, 133,031 and 5,000,000 states. Black: two regions of 62 components combined by messages on the tie, at $\varepsilon_r = 10^{-1}$, 10^{-2} and 10^{-3}.

95 percent interval, which is all that 20,000 draws of a probability near 0.68 can resolve. The per-state physics of the regional evaluator matched the global one on all 20,000 checked states.

The two-region route also keeps evidence local. When a storm multiplies the outage rates of twelve lines in region A by ten, the update re-enumerates region A alone, 24,942 states with 13,606 new physics evaluations, and reuses region B's 11,336 states without a single new evaluation. Its bracket, $[0.6620, 0.6819]$, keeps the width 0.0199, in 32 seconds. The global route at its 100,000-state budget re-used its cached states, needed 11,335 new evaluations and 10 seconds, and its width grew to 0.1320. Under the storm, region A's states compress more, at a ratio of 24.7, because the storm adds line outages that the one tie does not see.

24.5.3 Chains beyond enumeration

The chain pass was first checked against brute force. On three regions of four components each, 4,096 joint states, its last bracket was $[0.001726, 0.001764]$ around the exact 0.001726. On four

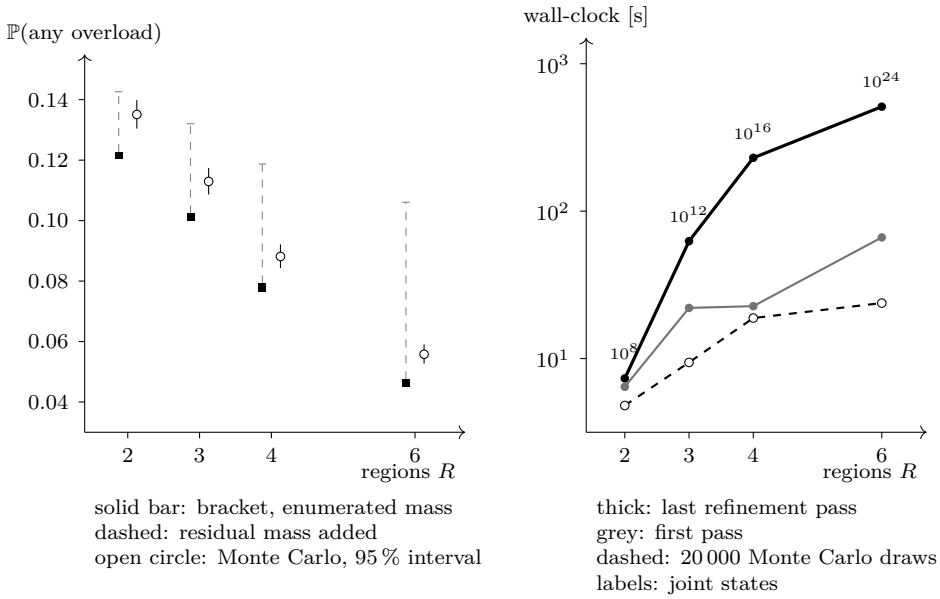


Figure 24.4: Chains of R single-tie regions under pro-rata dispatch at sixty percent load, $\varepsilon_r = 10^{-2}$ per region. Left: the certified bracket on the probability of any overload on the enumerated mass, its extension by the residual, and the Monte Carlo estimate. Right: wall-clock of the first and last refinement passes and of 20,000 Monte Carlo draws, logarithmic, with the joint state count of each chain.

regions, 65,536 joint states, it closed on the exact 0.006281. Islanding was exact in both.

Figure 24.4 shows the full chains. The left panel gives, for each chain length R, the certified bracket on the enumerated mass (solid bar), its extension by the joint residual mass (dashed), and the Monte Carlo estimate with its 95 percent interval (open circle). The right panel gives the wall-clock of the first and last refinement passes and of the Monte Carlo, with the joint state count of each chain above the last pass.

The bracket on the enumerated mass is two to three thousandths wide at every length, while the joint state count grows from 10^8 to 10^{24}. The last pass grows from 7 seconds to 511, a factor of seventy, because the lattice on which the messages combine widens with R. The residual grows too, and it dominates the certificate. Table 24.4 separates the two sources of width. Its bracket columns exclude the residual mass, which is listed on its own; adding it to the upper end gives the certified interval.

The shear is worth an order of magnitude. On the plain lattice the last pass leaves a width of two to three hundredths; on the sheared lattice, two to three thousandths, at a fraction of the plain lattice's pass time, 511 seconds against 1,529 at six regions. Refinement narrows the sheared bracket by a factor of two to five across four passes. The residual, meanwhile, is between nine and twenty-four times the last box width. At $\varepsilon_r = 10^{-2}$ it is the enumeration depth, not the lattice, that limits the certificate. At $R = 2$ the exact two-region value on the same enumerated mass, 0.12269, lies inside the sheared bracket $[0.12065, 0.12273]$. Under this dispatch the two-region separator is very small: each region's 11,336 states fall on 291 values, a ratio of 39.0, against 15.9 at peak dispatch.

Table 24.4: Sources of width in the chain brackets. Box widths are on the enumerated mass, before the residual: after the first and the last refinement pass on the sheared lattice, and after the last pass on the plain two-axis lattice of 2×20 MW cells. The residual is the joint unenumerated mass. The last column is the Monte Carlo estimate minus the upper end of the sheared bracket before the residual.

R	residual	first pass	last pass	plain lattice	MC above bracket
2	0.0199	0.0099	0.0021	0.0260	0.0124
3	0.0297	0.0064	0.0023	0.0246	0.0106
4	0.0394	0.0078	0.0028	0.0283	0.0089
6	0.0585	0.0054	0.0024	0.0198	0.0082

Table 24.5: Three data halls, two wall paths per interface, $\varepsilon_r = 10^{-3}$ per hall, against 20,000 Monte Carlo draws with 95 % Wilson intervals. The per-hall, wall, islanding and severity rows are on the enumerated mass and exclude the joint residual, 0.0030. The row for any overload includes it: it runs from the largest hall's value to the union bound plus the residual.

event	messages	Monte Carlo
overload in hall 0	0.00430	0.0048 $[0.0039, 0.0059]$
overload in hall 1	0.01111	0.0123 $[0.0109, 0.0139]$
overload in hall 2	0.00430	0.0046 $[0.0038, 0.0056]$
overload of a wall path	$[0.00053, 0.00212]$	0.00205 $[0.0015, 0.0028]$
islanding	0.00732	0.00695 $[0.0059, 0.0082]$
expected severity	$[291.4, 291.5]$ W	304.7 ± 21.3 W
any overload	$[0.01111, 0.02480]$	0.01275 $[0.0113, 0.0144]$

24.5.4 The second domain

On two halls joined by one wall path, the two-region messages matched brute force on every metric: the probability of any overload, 0.157727, of islanding and of deficit to within 4×10^{-16}, the expected severity, 1,754 W, to within 8.9×10^{-12} W, and every path's overload probability to 2.8×10^{-17}. Each hall's states fell on 8 separator values, and the two tables held 128 pairs. The messages took 12.9 milliseconds and brute force 210.8, a ratio of sixteen.

Table 24.5 gives the three-hall system, which has two wall paths at each interface, per hall and per kind of event. Chapter 18 gave its bracket on any overload; the table adds what that bracket is made of.

The per-hall values are points: the environment messages resolve each hall's own probability exactly on the enumerated mass. The width of the any-overload bracket is the union bound, 0.0107, plus the residual 0.0030. Each wall path overloads with probability 0.00053: when a cooling unit is lost, heat is pushed through the walls to the neighbors' units. Refining the cells from 5,000 W to 200 W changed no bracket. The whole ladder of three passes, with the enumeration, took 0.18 seconds, against 0.659 for the 20,000 draws.

24.6 What surprised

The gate came out unfavorable. The first study was designed to say whether realistic cuts compress enough to build on, and it said no. No realistic grid came near a ratio of ten at

three ports or fewer, and two of the four had no region that small. The reason is in Table 24.2: compression is supplied by interchangeable parts inside a region and consumed by the ports, and real grids have few of the first and more of the second than a single tie. The negative result sets the conditions under which partitioning a discrete failure model pays. It pays where a region with interchangeable units sits behind one or two ties, as in the two-replica system and the chains, and in the halls, whose cooling units are alike. It pays in coverage even where compression is modest, because two ordered lists cover the joint space faster than one; the factor of 221 in states was measured where compression was also large, so the two effects cannot be separated on this system. And it pays for evidence, which stays in its region. Where the region is heterogeneous, as on IEEE 118, the route through the ports reorganizes the enumeration without shortening it, and a Gaussian message inside the region, where the Laplace approximation holds, is the better tool.

Censored measurements. Thirteen of the 47 regions used for the partition-search fit stopped at the state cap. On PEGASE the residual after 5,000 states is 0.97 and 1.00: those ratios describe the most probable 5,000 states, nothing more. The direction of the censoring can be read from the uncensored regions. On the one-tie replica the ratio grows from 4.6 to 15.9 to 48.1 as ε_r falls from 10^{-1} to 10^{-3}; on IEEE 118's region A it grows from 1.00 to 1.02. A censored ratio is therefore likely an underestimate, but the IEEE 118 rows show that the growth can be negligible. A fitted law predicting separator counts from the port count, and state counts from the summed outage probability, missed by a factor of about ten at the median, $10^{1.05}$ and $10^{0.95}$; the censored rows bias it, and the study keeps the port count for ranking cuts and does not use the law for magnitudes.

The one-point gap. In Figure 24.4 the Monte Carlo estimate lies above the bracket on the enumerated mass at every length, by about one percentage point. At $R = 2$ the exact value on that mass lies inside the bracket, so the bracket is right and the gap is the unenumerated tail. The gap is below the residual at every length: 62, 36, 22 and 14 percent of it at $R = 2$, 3, 4 and 6. With the residual added the certified intervals are $[0.1207, 0.1426]$, $[0.1000, 0.1320]$, $[0.0765, 0.1187]$ and $[0.0452, 0.1061]$, and each contains the Monte Carlo estimate. The same pattern appears at a smaller scale in the halls, where each hall's Monte Carlo estimate exceeds its value on the enumerated mass by less than the residual 0.0030. The lesson is the one in the chapter's residual conventions: a bracket on the enumerated mass is a statement about that mass, the failure sets it leaves out are the deep ones, and deep failure sets overload more often than shallow ones. Report the bracket with the residual, or enumerate deeper.

The shear is a property of the dispatch. Under the merit-order dispatch of the peak snapshot the three-region chain did not converge. The fitted shear was zero: the headroom sits in the marginal units, so it is not a linear function of the lost power, and W ranged from 230 to 425 MW instead of collapsing to one value. The bracket stayed at $[0.7929, 0.9695]$ through three passes of 133, 190 and 287 seconds. It contained the Monte Carlo estimate, 0.8215, but it did not narrow. The chains' efficiency came from the pro-rata dispatch, which makes each region's contribution a function of one scalar.

24.7 Reproduction

The four studies live under `docs/terra_graph_engine/case_studies/`. Each is run from the repository root with the source tree on the path:

```
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
  foundation_power_flow/separator_compressibility/run.py
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
  foundation_power_flow/local_to_global_poc/run.py --deep
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
  foundation_power_flow/region_tree_chain/run.py
PYTHONPATH=src python docs/terra_graph_engine/case_studies/\
  foundation_hvac/thermal_local_to_global/run.py
```

Each writes a `results.json` and renders a `results.md` from it; `--render-only` re-renders without recomputing. The option `--deep` adds the $\varepsilon_r = 10^{-3}$ truncation to the two-replica study, about ten minutes on top of a run of about fifteen. A fifth directory, `local_to_global_scaling`, holds a consolidated table rebuilt from the committed results of the source studies, and its test requires the rebuilt file to be byte-identical to the committed one. The invariant tests of each study are under `tests/case_studies/` in the matching directory. Wall-clocks in this chapter are those of one machine running Python and NumPy.

The chapter's figures and every number in its text come from

```
cd docs/book
python scripts/ch24_partitioning.py
```

which reads the committed `results.json` files, prints the quantities quoted above, and writes the three figure files `ch24_compression.tex`, `ch24_coverage.tex` and `ch24_chains.tex` into the `figures` directory.

Chapter 25

Bad data on fourteen buses

Chapter 12 built the diagnostics of a posterior on a six-bus network with linear physics, where every number could be checked by hand. This chapter runs the same diagnostics on the AC IEEE 14-bus network, whose physics is nonlinear and whose loads are known only by forecast. The misfit test, redundancy, studentized residuals, sequential innovations, evidence over hypotheses, the value of a reading and identifiability are each applied to one meter set and one seeded truth. Three findings carry the chapter. The misfit test is right only when the degrees of freedom are counted right, and a broad regularizing prior is not a degree of freedom. A meter that no other meter checks can still look checkable, because the load forecasts check it. And a gross error that the tests miss moves the estimate by six standard deviations without widening its error bars. Every number in the chapter is read from the study's committed results file by the script `ch25_bad_data.py` in the book's `scripts` directory, which re-runs no physics.

25.1 The question

A transmission operator's state estimator reads a few dozen meters every few seconds. Sooner or later one of them reads wrong: a failed transducer, a wiring change nobody recorded, a scaling error after maintenance. The question is which tests see the error, which do not, and how far the estimated state moves when the error goes unseen. State estimation has answered the first half of the question for decades, with the chi-squared test on the misfit and the largest normalized residual. This study asks the question on a network where the loads are uncertain and carry forecasts, as they do in practice, and it asks what those forecasts do to the answer.

Four sub-questions follow, in the order of Chapter 12. Is the model adequate on clean data, and does the test's false-alarm rate match its level? How checkable is each meter, before any reading? When one meter reads 7.5 of its accuracies high, what do the misfit, the residuals and the evidence say, and where does the estimate go? And which meters are checkable only because the forecasts check them?

25.2 The system

The network is the IEEE 14-bus test system on a 100 MVA base: fourteen buses, seventeen lines and three transformers (Figure 25.1). Bus 1 is the slack. Buses 2, 3, 6 and 8 are voltage-controlled: a generator or synchronous condenser at each holds the bus voltage at its setpoint with whatever

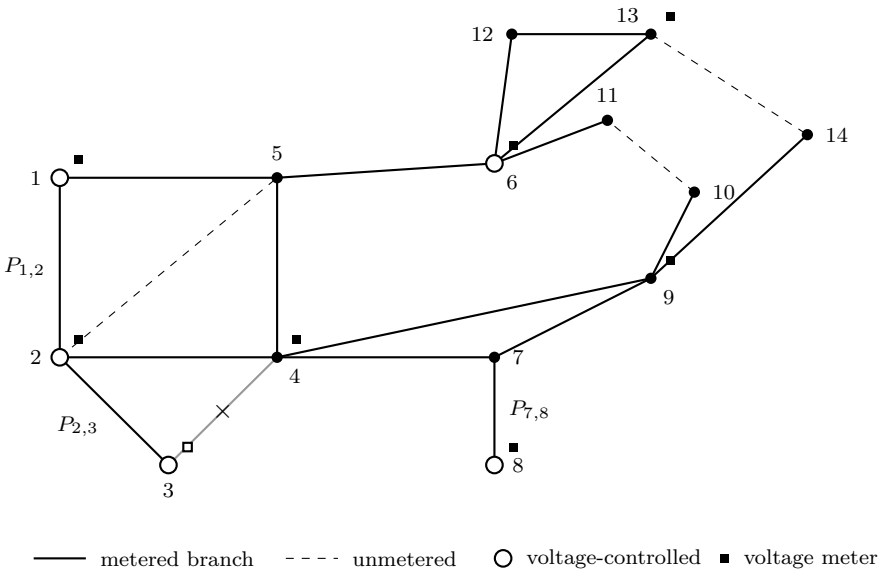


Figure 25.1: The IEEE 14-bus network and its meters. Solid branches carry an active and a reactive flow meter at the end of the lower-numbered bus; dashed branches are unmetered. Black squares are voltage meters. Open circles are the slack bus 1 and the voltage-controlled buses 2, 3, 6 and 8. The thinned meter set of §25.5.9 removes both meters on line 3–4 (the cross), the voltage meter at bus 3 (the open square) and the reactive meter on line 2–3, leaving the active meter $P_{2,3}$ as the only meter on bus 3's angle.

reactive power that takes. Eleven buses carry a load, every bus except 1, 7 and 8, and the load at bus 3 is the largest on the network.

Meters. Eight voltage meters, at buses 1, 2, 3, 4, 6, 8, 9 and 13, have accuracy 0.004 per unit. Seventeen of the twenty branches carry an active and a reactive flow meter at the end of the lower-numbered bus, 34 flow meters of accuracy 0.01 per unit, one megawatt or one megavar. The three unmetered branches are lines 2–5, 10–11 and 13–14. That is 42 meters. In what follows $P_{i,j}$ and $Q_{i,j}$ are the active and reactive flows on the branch between buses i and j, read at bus i, and V_i is the voltage magnitude at bus i.

Loads and their forecasts. Each of the eleven loads has a latent active deviation δP_i and a latent reactive deviation δQ_i from its nominal value, 22 latent loads in all. Each has a Gaussian prior with mean zero and a standard deviation of 20 percent of the nominal load, with a floor of 0.01 per unit. These priors are the pseudo-measurements a real estimator uses for loads it does not meter: forecasts, with the forecast's error as their width.

Truth. The truth is an exact AC power flow with each load moved by a deviation drawn from its prior. The meters read the truth plus Gaussian noise at their stated accuracies. The estimator therefore sees a model that is correctly specified in every factor that has a data-generating story, which is what makes the misfit law testable.

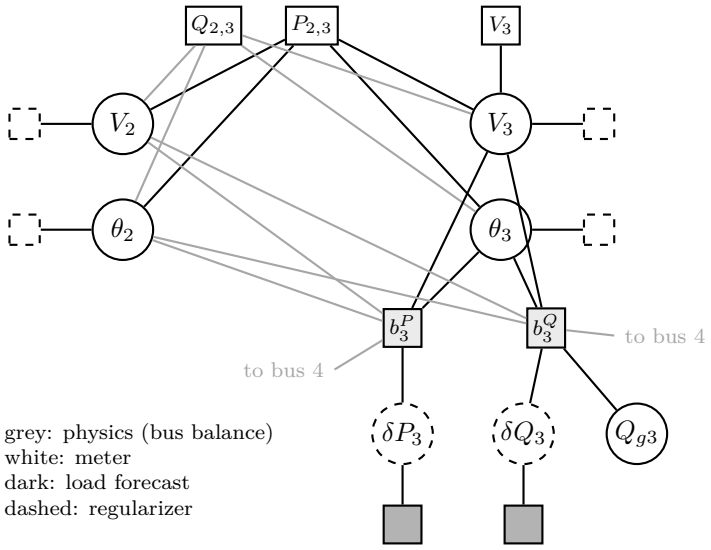


Figure 25.2: The factor graph around line 2–3. Circles are unknowns; dashed circles are the latent load deviations at bus 3. Grey factors are the active and reactive balances at bus 3, which also touch the voltages and angles of buses 2 and 4 through the branch flows (grey edges). White factors are meters, dark factors the load forecasts, and dashed factors the broad regularizing priors on every voltage and angle. The reactive load δQ_3 and the condenser's free output Q_{g3} enter only the reactive balance b_3^Q, and only through their difference.

25.3 The model as a factor graph

Figure 25.2 draws the part of the factor graph around line 2–3. The unknowns are the voltage magnitude and angle at every bus, the generators' free outputs, and the 22 latent loads. The physics contributes the AC active and reactive balance at every bus, each a row that sums the branch flows leaving the bus, the generation and the load, as in Chapter 1, held at a physics width of 10^{-3}. Every flow meter is a factor on the four voltages and angles at the two ends of its branch. Every voltage meter is a factor on one variable. Every latent load has its forecast as a prior factor.

One more family of factors is needed. Every bus voltage carries a prior of mean one and width 0.3 per unit, and every angle a prior of mean zero and width one radian: 28 priors, the dashed factors of Figure 25.2. They keep the posterior's precision matrix full-rank where meters are sparse, and they are broad enough that they bias nothing the meters determine. They are *regularizers*. The truth is not drawn from them, and that distinction is the subject of the first result.

The graph also shows a symmetry before any computation. At bus 3 the condenser's reactive output Q_{g3} is free, because it is whatever holds V_3 at its setpoint. The reactive load deviation δQ_3 enters the same balance row and nothing else. Only $Q_{g3} - \delta Q_3$ reaches any meter, so no reading can separate the two. The same holds at the other voltage-controlled buses that carry a load, 2 and 6. This is Principle 12.3 of Chapter 12 on a real network, and §25.5.8 confirms it to two decimals.

25.4 Method

The estimator is the MAP mode of the posterior with its Laplace Gaussian, computed by Newton's method on the AC rows (§6.4). Every diagnostic is a readout of that one posterior, as Table 12.1 listed.

1. *Adequacy.* The misfit, the sum of squared standardized residuals at the mode, against the chi-squared law of Proposition 12.1 with $m_g - n$ degrees of freedom. The law is checked over 200 seeded truths, each a fresh draw of loads and meter noise.
2. *Calibration.* The same 200 seeds give the error of every state variable and latent load in units of its posterior standard deviation, which should be standard Gaussian.
3. *Redundancy.* For each meter, $r_k = 1 - \mathrm{Var}[\mathbf{h}_k^{\mathsf{T}}\mathbf{z}]/\sigma_k^2$ under the posterior on clean data (§12.5). A gross error b moves the meter's studentized residual by $b\sqrt{r_k}/\sigma_k$, so the smallest error flagged at three is $3\sigma_k/\sqrt{r_k}$.
4. *Localization.* Studentized residuals, ranked (§12.4), after a gross error of $7.5\sigma_k$ is added to one meter's clean reading.
5. *Sequential innovations.* The 42 readings folded into the prior's Laplace Gaussian one at a time by rank-one updates, each standardized innovation checked against three and the running sum against the chi-squared quantile (Proposition 11.2). The updates are never re-linearized, and the end result is compared with the batch AC solve (§11.6).
6. *Evidence.* Named hypotheses scored by equation (12.1) at equal prior odds (§12.6). A *bad meter* hypothesis widens one meter's accuracy thirtyfold; a *load anomaly* hypothesis widens one load's forecast tenfold. With the null hypothesis, that is one plus 42 plus 22, 65 hypotheses.
7. *Value of a reading.* The variance a candidate meter would remove from a target, equation (11.3), chosen greedily in two steps (§11.7).
8. *Identifiability.* The ratio ρ_i of each latent load's posterior spread to its forecast's, from the parameter Schur complement (§12.7). The engine calls a load well identified below 0.25, weakly between 0.25 and 0.75, and not identified above.
9. *Critical meters.* Redundancy has two sources, the other meters and the forecasts. Widening every forecast a hundredfold leaves the meters alone; a meter whose redundancy then falls below 0.01 is critical in the classical sense. A thinned meter set then puts one critical meter on the largest load.

25.5 Results

25.5.1 Is the model adequate?

Count every prior as a pseudo-observation, which is what Proposition 12.1 does, and this model claims 70 degrees of freedom. Table 25.1 sets that claim against the 200 seeds. The mean misfit is 41.7, far below 70, the variance 92.9 against the law's 140, and the test rejects no seed at either level. A test that never rejects clean data looks reassuring, and it is not: it will not reject much else either.

The count is wrong, and the proof of Proposition 12.1 says where. The proof assumes that every factor's standardized error is a standard Gaussian draw: that the prior's mean minus the true value, divided by the prior's width, is as large as the width says. For the load forecasts that is true by construction, since the truth is drawn from them. For the regularizers it is false. A voltage prior of width 0.3 per unit centered on one, against a true voltage within about a tenth

Table 25.1: The misfit over 200 seeded truths under two counts of the degrees of freedom: every prior counted as a pseudo-observation, and the 28 regularizing voltage and angle priors left out of both the misfit and the count. The last two columns are the fractions of seeds rejected at the 1 and 5 percent levels.

count	dof	mean	variance	law's variance	$p < 0.01$	$p < 0.05$
every prior a pseudo-observation	70	41.7	92.9	140	0.0%	0.0%
regularizers excluded	42	40.6	92.7	84	0.5%	4.5%
the law, if right					1%	5%

of one, has a standardized error of a fraction, not of order one. The regularizer adds a row to the count and one to the law's mean, and adds almost nothing to the misfit.

The results measure it directly. Removing the 28 regularizers' terms lowers the mean misfit from 41.7 to 40.6: together they contribute 1.07, which is 0.038 per term against the 1 each that the count assumes, an rms departure of about a fifth of the prior's width. Removing their rows lowers the count from 70 to 42. The law then holds, as the second row of Table 25.1 shows: mean 40.6 on 42, variance 92.7 close to the law's 84, and rejection on 0.5 percent of seeds at the 1 percent level and 4.5 percent at the 5 percent level. This count is also easy to read. Each load forecast pays for its own latent load, and the physics rows pay for the voltages, angles and generator outputs. What is left is one degree of freedom per meter, 42. The engine lets a prior be declared a regularizer when it is attached. Its misfit test then leaves that prior out of both the sum and the count, and the grid's default voltage and angle priors are declared that way.

Figure 25.3 draws both laws. The seed means sit together at the center of the χ^2_{42} density and deep in the left half of the χ^2_{70}. The misfit did not change between the two counts; the law it was judged against did. The two dots mark the gross error of §25.5.3. The same misfit is a p-value of 0.0056 on the correct count and 0.49 with every prior counted, where it would pass as ordinary.

> **Principle 25.1.** *A prior whose width says nothing about how the truth was drawn is a regularizer, not an observation. Count it in neither the misfit nor its degrees of freedom, or the test will pass everything.*

Error bars. The same seeds calibrate the posterior's spreads. Over 9,800 standardized errors of the voltages, angles and latent loads, 2.5 percent exceed two, against 4.6 percent for a standard Gaussian, and 0.27 percent exceed three, against the Gaussian's 0.27. The rms standardized error is 0.874. The error bars are about fourteen percent wider than the errors they cover, with tails of the right size. They are safe to report as they are.

On the seeded reference truth used for every result below, the clean misfit is 32.02 on 42 degrees of freedom, a p-value of 0.868.

25.5.2 How checkable is each meter?

Figure 25.4 draws the redundancy of all 42 meters on the clean reference data, sorted. The voltage meters are all highly redundant, from 0.881 to 0.973: the physics and the flow meters determine the voltages to well within the voltage meters' accuracy. The flow meters spread from 0.113 to 0.971, and no meter falls below a tenth.

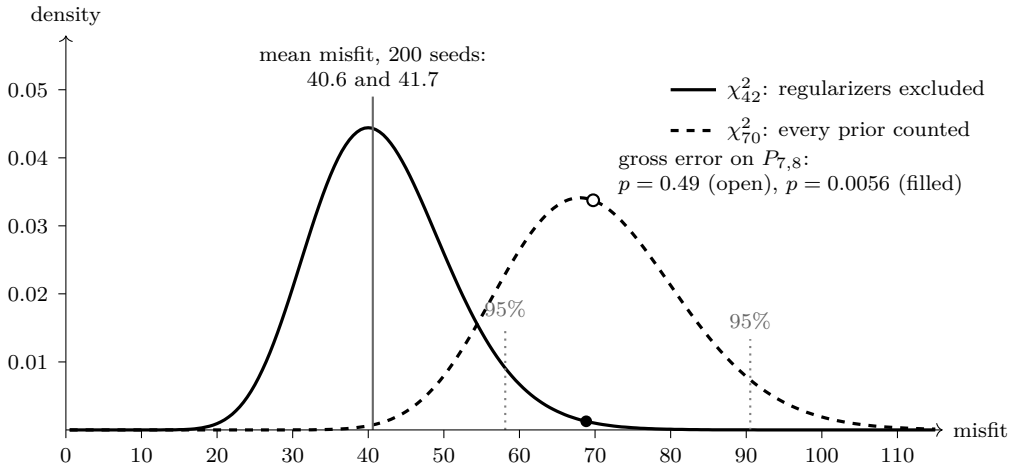


Figure 25.3: The misfit law under the two counts. Solid: χ^2_{42}, the correct law with the regularizers excluded. Dashed: χ^2_{70}, every prior counted. Dotted lines are the 95 percent points. The grey line is the mean misfit over 200 seeds, 40.6 on the correct count and 41.7 with every prior counted. The dots are the misfit with a 7.5σ error on the meter $P_{7,8}$: filled on the correct law, open on the law that counts every prior.

The two ends of the flow meters are the two cases of the next subsections. The active flow on line 7–8 has redundancy 0.971. Bus 8 carries a condenser and no load, so its active balance fixes the flow on the line whatever the loads are, and the meter reads a quantity the physics already knows. The smallest error it flags at three spreads is 0.030 per unit, 3.04 of its accuracies. The active flow on line 1–2 has redundancy 0.113. It is the heaviest line on the network, it leaves the slack bus, whose output is free, and it arrives at bus 2, whose balance includes the unmetered line 2–5 and a load known only to its forecast. The smallest error it flags is 0.089 per unit, 8.91 of its accuracies.

25.5.3 A gross error where it can be seen

Add 0.075 per unit, 7.5 accuracies, to the clean reading of $P_{7,8}$. Table 25.2 gives the diagnostics in its first column. The studentized residual of the faulty meter is 6.19, first in the ranking by a wide margin; the next is V_6 at -2.43, the same ordinary residual it has on clean data. The redundancy predicts a shift of $7.5\sqrt{r_k} = 7.39$ for the error alone; the observed value is that shift added to the meter's residual on clean data. The misfit with the regularizers excluded is 68.83 on 42, a p-value of 0.0056, rejected at the 1 percent level.

The evidence agrees. Among the 65 hypotheses the bad meter $P_{7,8}$ has probability 1.000, ahead of the runner-up, a load anomaly at bus 14, by 14.5 nats. The error does little harm: the largest shift of any state variable or latent load is 0.21 of its standard deviation, at θ_8. A meter that the physics checks this well cannot move the state much, because the rest of the model outvotes it.

25.5.4 A gross error where it cannot

Add the same 0.075 per unit to $P_{1,2}$, the second column of Table 25.2. The redundancy predicts a shift of 2.53, and the observed studentized residual is 2.40, second in the ranking behind the

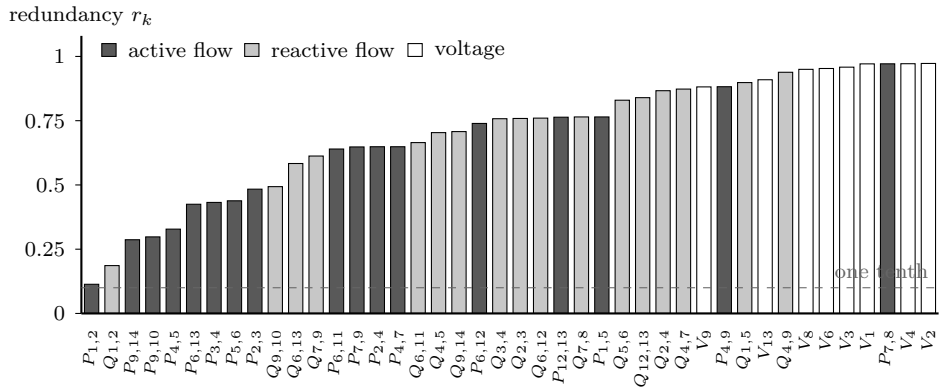


Figure 25.4: Redundancy r_k of the 42 meters on the clean reference data, sorted. The dashed line is at one tenth, below which Chapter 12 called a meter uncheckable. The least redundant meter is the active flow $P_{1,2}$ at 0.113; the most redundant flow meter is $P_{7,8}$ at 0.971.

Table 25.2: A gross error of 7.5 accuracies, 0.075 per unit, on three meters: the most redundant flow meter, the least redundant one, and the only meter on bus 3's angle in the thinned set of §25.5.9. The thinned column has 38 meters, 38 degrees of freedom and 61 hypotheses. State shifts are in units of the clean posterior's standard deviation.

	$P_{7,8}$	$P_{1,2}$	$P_{2,3}$, thinned
redundancy r_k	0.971	0.113	0.086
smallest error flagged, $3\sigma_k/\sqrt{r_k}$ [pu]	0.030	0.089	0.102
gross error [pu]	0.075	0.075	0.075
shift the error alone predicts, $7.5\sqrt{r_k}$	7.39	2.53	2.20
studentized residual	6.19	2.40	1.91
its rank among the meters	1	2	2
misfit p, every prior counted	0.486	0.999	–
misfit p, regularizers excluded	0.0056	0.658	0.727
largest state shift [sd]	0.21 (θ_8)	3.89 (δP_2)	6.71 (δP_3)
evidence: rank of the truth	1 of 65	2 of 65	7 of 61
evidence: its probability	1.000	0.101	0.044

ordinary V_6 at -2.43. No threshold at three sees it. The misfit is 37.75 on 42, a p-value of 0.658: the model looks better than on an average clean day.

The state has moved. The estimate of the load at bus 2 has risen by 3.89 of its standard deviations, and the angles at buses 2, 4, 3 and 5 have fallen by 3.16, 2.03, 2.03 and 2.01. The model has explained the extra flow on line 1–2 as extra load at the bus it flows into, which is exactly what the graph allows: bus 2's forecast is wide enough to absorb it, and nothing else in the model reads the difference. This is the tie-line case of §12.5 on a meshed network. The line 1–2 is not a bridge, yet a load forecast of twenty percent next to it makes it behave like one.

The evidence does better than the residuals, but only just. It ranks the bad meter $P_{1,2}$ second among 65, at probability 0.101, behind a load anomaly at bus 14 at 0.201, a gap of 0.69 nats, odds of two to one. An operator would be told that something is probably wrong near line 1–2 or in bus 14's forecast and that the answer is not certain, which is the truth.

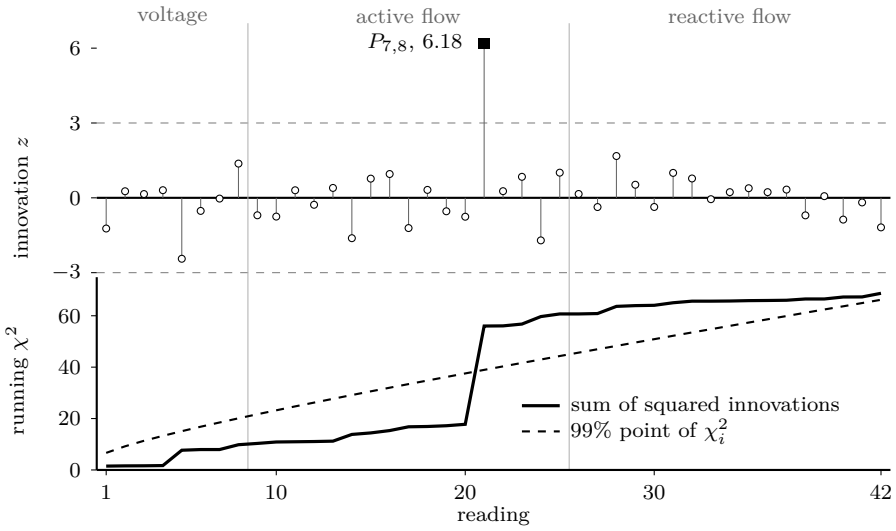


Figure 25.5: The 42 readings of the gross-error case on $P_{7,8}$, folded in one at a time. Top: standardized innovations; the black square is the faulty meter, and the dashed lines are at ± 3. Bottom: the running sum of squared innovations against the 99 percent point of the chi-squared law with as many degrees of freedom as readings.

25.5.5 One reading at a time

Fold the readings of the first gross-error case into the prior one at a time, in the order voltages, active flows, reactive flows. The prior's Laplace Gaussian is linearized at the nominal loads and is never re-linearized. Figure 25.5 plots the standardized innovations and their running sum of squares.

The faulty reading, the twenty-first, lands 6.18 spreads from its prediction, and no clean reading exceeds 2.44, the voltage at bus 6. The running sum sits at 17.77 before the faulty reading, well under the quantile of 37.57, jumps to 55.99 against 38.93, and stays above the quantile for every reading after, ending at 68.74 against 66.21. The margin narrows as clean readings accumulate, which is Chapter 11's observation that a cumulative test dilutes a sudden fault and a per-reading test does not.

Two further numbers settle the method. The sequential result, 42 rank-one updates of a Gaussian linearized at the nominal loads, ends within 0.068 posterior standard deviations of the batch AC solve on every variable, at worst at V_5, with an rms difference of 0.020. At this operating point and with loads within their forecasts, the AC nonlinearity does not matter for sequential updating, which is the check of §11.6 passing on a real network. And withholding the flagged reading changes the largest state error from 1.904 to 1.905 standard deviations: on a meter this redundant, detection matters for the record and not for the state.

25.5.6 Which explanation?

Table 25.2 gave the rank and probability of the true hypothesis in each case. Two regularities in the full rankings are worth stating. First, a load anomaly at bus 14 is the runner-up on the data of the first gross-error case and first on the second and on the thinned data below, so its rank reflects the clean readings rather than any gross error. Second, the load-anomaly hypotheses for

Table 25.3: Identifiability of the 22 latent loads: the ratio ρ of posterior to forecast spread on the clean reference data. Bold: well identified, $\rho < 0.25$. Above 0.75 the engine calls a load not identified. The reactive loads at the voltage-controlled buses 2, 3 and 6 have $\rho = 1.00$.

bus	2	3	4	5	6	9	10	11	12	13	14
ρ, active	0.42	**0.07**	**0.19**	0.78	0.62	0.27	0.62	0.88	0.73	0.54	0.51
ρ, reactive	1.00	1.00	0.88	0.89	1.00	0.41	0.76	0.86	0.79	0.79	0.84

the reactive loads at buses 3 and 6 have the same log-evidence to every figure in every data set: 48.326 on the first, 63.863 on the second. That is the symmetry of Figure 25.2 again. Widening the forecast of a quantity that no reading can see changes nothing a reading can judge, so every such hypothesis scores the same.

25.5.7 Which meter next?

Bus 14 has no voltage meter, and its load is weakly identified. Take each as a target and choose two new meters greedily by equation (11.3), among voltage meters at the unmetered buses, flow meters on the unmetered branches, and flow meters at the other end of the metered ones.

For the voltage at bus 14 the best candidate is a voltage meter there, and it removes only 20.9 percent of the variance. The reason is that the voltage is already known to 0.00205 per unit, half the voltage meter's accuracy of 0.004: the physics and the neighboring meters make the virtual sensor twice as good as the real one would be, and equation (11.3) gives $v/(\sigma^2 + v) = 0.2086$. A flow meter on line 13–14 then removes a further 8.4 percent, and the spread falls to 0.00175.

For the load at bus 14 the best candidate is the active flow on line 9–14 read at bus 14, $P_{14,9}$, which removes 34.5 percent; the active flow $P_{13,14}$ is a close second at 33.1. After the first, the second removes 28.5 percent of what remains, and the spread falls from 0.0152 to 0.0104 per unit, 32 percent narrower. Bus 14 has only those two branches, so its load is the sum of the two flows into it, and once one is read the other is the load. That is the mechanism of Figure 11.4, on a bus with a nonlinear balance.

25.5.8 Which loads can be recovered?

Table 25.3 gives ρ_i for the 22 latent loads. Two are well identified, the active loads at buses 3 and 4, at 0.07 and 0.19. Eight are weakly identified, and twelve not at all. The active loads fare better than the reactive ones, because the flow meters' active readings are sharper relative to the forecasts. Of the eleven reactive loads, only the one at bus 9 is identified at all, and only weakly.

Three of the twelve are not a question of meters. The reactive loads at buses 2, 3 and 6 have $\rho = 1.00$: the posterior is the forecast. These are the three voltage-controlled buses that carry a load, and the reason is the symmetry of §25.3: the load and the generator's free reactive output enter one balance row and only through their difference. The flattest direction of the load block is the reactive load at bus 3 alone, with no other load in it. No meter on the network can resolve it; a meter on the generator's reactive output or on the load itself would.

25.5.9 Critical meters and a thinned set

Redundancy has two sources, and the results so far have mixed them. Widen every load forecast a hundredfold and the meters stand alone. Figure 25.6 draws the result. Seven meters of the

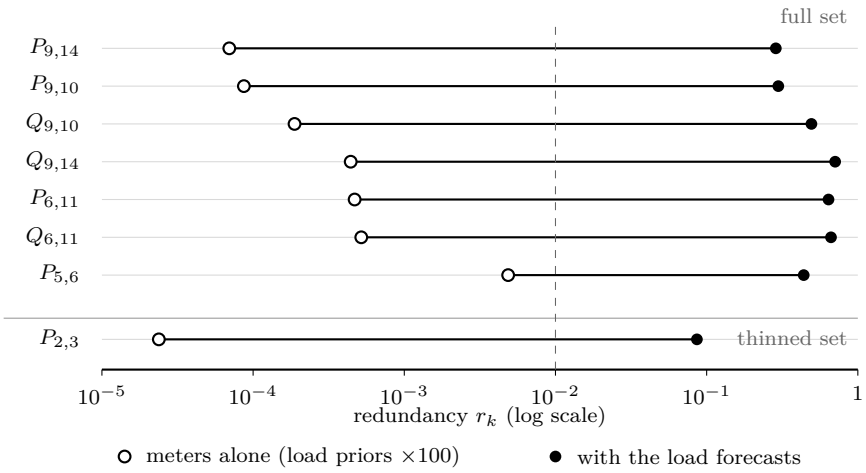


Figure 25.6: Redundancy of the meters that are critical among meters, on a log scale. Open circles: every load forecast widened a hundredfold, so that only the other meters can check the meter. Filled circles: with the forecasts at their stated width. The dashed line is at 0.01. Top seven: the full meter set. Bottom: the active meter $P_{2,3}$ in the thinned set, the only meter on bus 3's angle.

full set fall below 0.01, and so are critical in the classical sense: $P_{9,14}$, $P_{9,10}$, $Q_{9,10}$, $Q_{9,14}$, $P_{6,11}$, $Q_{6,11}$ and the transformer flow $P_{5,6}$, from 7.0×10^{-5} to 4.9×10^{-3}. They are the flows into the lower network, where every bus has a load and two of the branches are unmetered; each of these meters is the only reading of some combination of loads. With the forecasts in place the same seven have redundancy from 0.287 to 0.708, comfortably checkable. All of that checking comes from the forecasts. The least redundant meter of the full set, $P_{1,2}$, is not among the seven: low redundancy and criticality are different things.

Now thin the set. Remove the voltage meter at bus 3, both meters on line 3–4 and the reactive meter on line 2–3, leaving 38 meters. The active flow $P_{2,3}$ is then the only meter on bus 3's angle, and bus 3 carries the largest load. The clean misfit is 28.83 on 38, a p-value of 0.858. The lowest redundancies are $P_{2,3}$ at 0.086, $P_{1,2}$ at 0.108 and $Q_{1,2}$ at 0.175. With the forecasts widened, the redundancy of $P_{2,3}$ is 2.4×10^{-5}: the other meters say nothing about the flow it reads, and only the forecast of bus 3's load, 20 percent of the largest load on the network, checks it. The smallest error it would flag at three spreads is 0.102 per unit, 10.2 of its accuracies.

Add 0.075 per unit to it. The third column of Table 25.2 gives the diagnostics. The studentized residual is 1.91, second behind V_6. The misfit is 32.37 on 38, a p-value of 0.727. Nothing is flagged. The estimate, meanwhile, has moved further than in any other case. The estimated load at bus 3 rises by 6.71 of its standard deviations and θ_3 falls by 6.04, while the loads at buses 2 and 4 fall by 3.38 and 3.23. Against the truth, the estimated load at bus 3 goes from 0.80 standard deviations below the true load to 5.94 above it, θ_3 from 1.10 above the truth to 4.94 below it, the load at bus 2 from 0.03 to 3.41 below, and the load at bus 4 from 0.49 above to 2.74 below. The model has moved load from buses 2 and 4 to bus 3 to explain a flow into bus 3 that did not happen.

The error bars do not widen to cover the move. The posterior standard deviation of each of the five most-moved variables changes by less than half a percent between the clean and the

faulty data: by a factor of 0.995 for the load at bus 3, 1.000 for θ_3. A reader of the estimate sees a load five standard deviations from where it is, and a spread that says it cannot be.

The evidence ranks the true hypothesis seventh of 61, at probability 0.044, behind the load anomaly at bus 14 at 0.213 and behind five hypotheses tied at 0.065, among them that nothing is wrong at all, a gap of 1.19 nats. It keeps the truth in view, which the residuals did not, and it does not pick it.

The fix is a meter, and which one matters. Restoring each removed meter in turn, the redundancy of $P_{2,3}$ rises to 0.455 with the active meter on line 3–4 restored, to 0.162 with the reactive meter on line 3–4, to 0.091 with the reactive meter on line 2–3, and stays at 0.086 with the voltage meter at bus 3. Only the second active flow at bus 3 makes the first checkable. On a transmission network active power moves with angles and reactive power with voltage magnitudes, and a meter that reads the wrong half of that split adds almost nothing.

25.6 What surprised

The degrees of freedom were wrong, and wrong safely. The misfit test is the first line of defence, and on this model its default count made it blind: it never rejected a clean seed, and it passed a 7.5σ error on the best-covered meter at a p-value of 0.49. The miscount is not an arithmetic slip. It is the difference between a prior with a data-generating story, whose standardized error is a standard Gaussian draw, and a prior with none, whose term is near zero but whose row is counted all the same. The model knows which of its priors are which only if the modeler says so, and the count must be made with that knowledge: here, one degree of freedom per meter.

Critical meters were hidden by the forecasts. Seven meters of the full set are critical among meters, and every one of them looked checkable, with redundancy from 0.29 to 0.71. The check they receive is a load forecast. That is a real check, and it is only as good as the forecast: a meter checked by a forecast cannot tell its own error from a wrong forecast, and the evidence between the two is close, as the thinned case showed at 0.044 against 0.213. A redundancy report that does not separate the two sources overstates how well the meters check one another.

The error bars did not widen. An absorbed gross error moved the bus-3 load estimate from 0.80 standard deviations off the truth to 5.94, with its spread unchanged to half a percent. This follows from Chapter 11: the Laplace covariance depends on the linearization point and not on the readings, and on a nearly linear model a gross error moves the mean and leaves the curvature. A posterior can therefore be confidently wrong, and nothing in its own error bars says so. Only the misfit and the residuals can, and on a meter with redundancy 0.086 they cannot.

Some loads are invisible to every meter. The reactive loads at the voltage-controlled buses have $\rho = 1.00$, and their load-anomaly hypotheses score identically on every data set. The cause is not meter placement but a symmetry of the physics: a voltage-controlled bus's reactive generation is free, and the load trades against it one for one. Principle 12.3 predicted it from the graph, and a sensor-placement study that did not look for it would spend meters on a direction that no meter on the network can resolve.

25.7 Reproduction

The study is one script, run from the repository root with the repository's `src` directory importable:

```
python docs/terra_graph_engine/case_studies/power_grid/bad_data_ieee14/run.py
```

It runs in under a minute on a workstation, every draw is seeded, and it writes `results.json` next to itself with every float rounded to six significant figures, so that a rerun reproduces the committed file byte for byte. The findings are enforced on the committed file by the artifact tests in `tests/case_studies/power_grid/bad_data_ieee14/`: the misfit law holds only with the regularizers excluded, the well-covered error is found and the least-redundant one is not, the sequential pass flags only the bad reading and matches the batch solve, the critical meters are checked only by the forecasts, the thinned error is unseen and corrupts the state, and the reactive loads at the voltage-controlled buses are unidentifiable.

The figures and tables of this chapter are regenerated, from the committed results alone, by

```
cd docs/book
python scripts/ch25_bad_data.py
```

which prints every number quoted here and writes the chapter's four generated figures and three generated tables, the files `ch25_*.tex` in the book's `figures` directory.

Chapter 26

An independent implementation agrees

Numbers in this chapter are produced by `scripts/t7_annual_ra.py` from the study's committed results files. Nothing is retyped.

26.1 The question

Every check in this part so far has been a check the method performed on itself. A closed form on a system small enough to have one. An invariant that must close. A bound that must hold. A posterior width tested against the spread of its own estimates. All of those are worth doing and none of them answers the question a planner actually has.

A resource-adequacy number decides whether a region builds a power plant. Its owner does not ask whether the arithmetic is self-consistent. They ask: would a different team, using a different tool, in a different language, get the same number?

That question has exactly one test, and the test is expensive, because it requires a second implementation. This chapter reports one. An annual resource-adequacy workload — every hour of a year, fifty outage draws per hour, a linear program per scenario — is run twice. Once through the construction of this book. Once through an independently written toolchain of commercial lineage, in a different language, with a different optimizer and no shared code. The inputs are pinned by a committed file that both read.

Two questions follow, and the second is the one worth the chapter:

- **Do the headline numbers agree, and to what precision?**
- **Where do they *not* agree, and what does that reveal about what was being compared?**

26.2 The workload

The system is Area 1 of a standard transmission test case. The workload is a full-year enumeration: 8,784 hours, fifty independent generator-outage draws at each, 439,200 scenarios in all. Each scenario is a direct-current minimum-load-shed linear program — given this hour's load, this draw's available generation, and the network's line ratings, dispatch to serve as much load as possible and report what could not be served, at what cost.

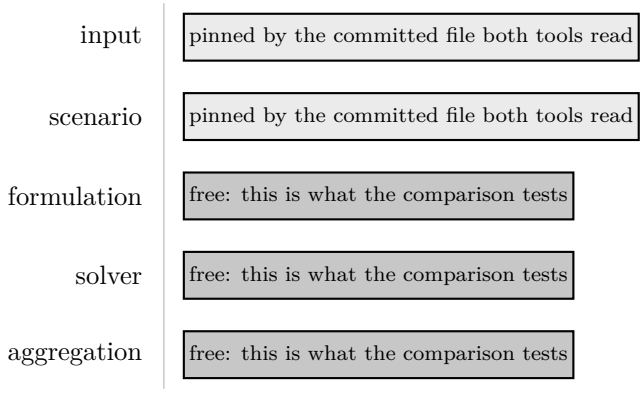


Figure 26.1: When two implementations disagree, the disagreement has one of five sources. Pinning the inputs and the scenario draws in a file that both tools read removes the first two by construction. Whatever is left — a difference in how the model was written down, in how the linear program was solved, or in how per-scenario results were summed into a year — is what the comparison is actually testing. A comparison that leaves the inputs free cannot tell which of the five moved; a comparison that changes only the inputs, holding the formulation fixed, isolates the other direction.

Three numbers come out of the year. *Loss of load expectation*, the expected number of hours in which some load goes unserved. *Expected unserved energy*, the expected megawatt-hours not delivered. And the annual dispatch cost.

A smaller design — one hundred hours, the same fifty draws, 5,000 scenarios — was run first and reported alongside, so that a disagreement found at full scale could be reproduced at a size a person can inspect.

26.3 What “independent” has to mean

Figure 26.1 states the protocol. The fifty outage draws at each of the 8,784 hours are not generated twice; they are written once to a file, committed, and read by both tools. So are the loads, the ratings and the generator data. Neither tool can be right by having drawn a luckier sample.

That leaves three places a difference could come from: the two teams wrote down different models, or the two optimizers found different answers to the same model, or the two summaries aggregated the same per-scenario results differently. Sections 26.5.1 through 26.5.3 separate them.

26.4 Method

Both tools solve every scenario and record, per scenario, the total unserved power, the unserved power at each bus, the total dispatch, the total cost, and a flag for whether the scenario is a loss-of-load event at all. The comparison then asks, in order:

1. Are the same scenarios infeasible under both?
2. Are the same scenarios flagged as events?

Table 26.1: The three annual figures, computed twice over 439,200 scenarios by two independently written toolchains reading one committed scenario file. The relative differences are a few tens of times the machine epsilon of a double-precision number.

metric	unit	this construction	the other tool	relative difference
loss of load expectation	h/yr	13.68	13.68	5.8×10^{-15}
expected unserved energy	MWh/yr	1735.7684	1735.7684	2.8×10^{-15}
annual dispatch cost	USD	15621915.614	15621915.614	2.7×10^{-15}

Table 26.2: The per-scenario comparison. The tolerances are the study's, set before the run. The last row is the one that does not belong with the others, and Section 26.5.3 is about it.

compared over all 439,200 scenarios	largest difference	tolerance	within it
total unserved power	7.5×10^{-12} MW	0.5 MW	439,200
total dispatch	1.8×10^{-11} MW	0.5 MW	439,200
total cost	0.00053 USD	10 USD	439,200
unserved power *at a bus*	300.7 MW	—	—

3. Do the per-scenario totals agree within tolerance — half a megawatt on power, ten dollars on cost?
4. Do the three annual headline figures agree to three significant figures?

Separately, a sample of scenarios is compared at the level of the *state* rather than the summary: every branch flow, every bus angle, every locational price, and the dispatch of every generator.

Two references stand outside the comparison. A bootstrap over the outage draws gives a confidence interval for the annual figures, which says how much of the answer is sampling rather than physics. And a copper-plate reliability model — one that treats the region as a single bus with no network at all — gives the number a planner would have had without a transmission model.

26.5 Results

26.5.1 The headline

Table 26.1 gives it. Loss of load expectation: 13.68 hours a year, both. Expected unserved energy: 1735.7684 MWh a year, agreeing to 2.8×10^{-15} relative. Annual dispatch cost: $15,621,915.61, agreeing to 2.7×10^{-15}.

The largest of the three relative differences is 5.8×10^{-15}, about twenty-six times the smallest difference a double-precision number can represent. Two independently written programs, in two languages, with two optimizers, summing 439,200 linear programs, produce the same three numbers to fourteen digits.

26.5.2 Every scenario, not just the total

An annual total can agree by cancellation. Table 26.2 rules that out. No scenario is infeasible under either tool; the two never disagree about which scenarios are infeasible, and never disagree about which are loss-of-load events. Across all 439,200, the largest difference in total unserved

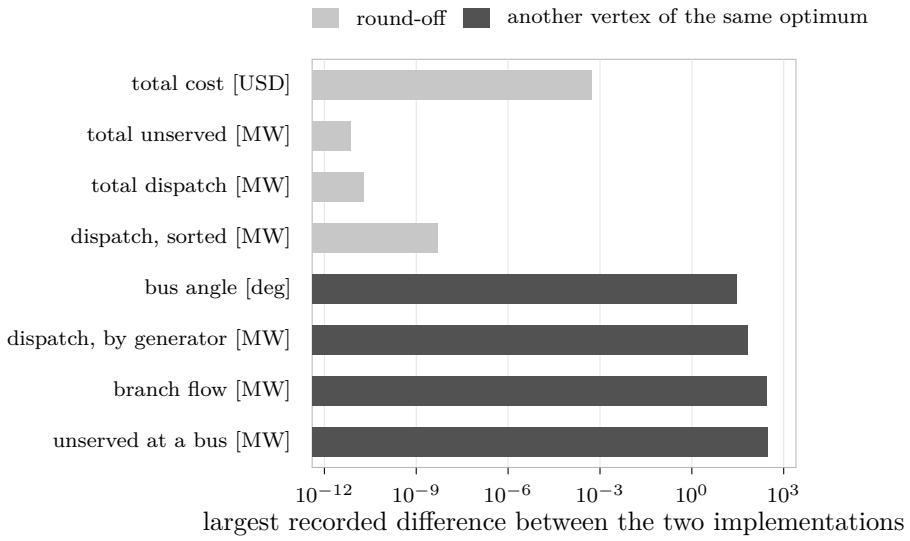


Figure 26.2: Every disagreement the study recorded, on one logarithmic axis. The pale bars are the quantities the two tools agree on at round-off; the dark bars are the ones they do not. The split is not between important and unimportant quantities. It is between quantities the linear program's optimum determines and quantities it does not.

power is 7.5×10^{-12} MW, in total dispatch 1.8×10^{-11} MW, and in total cost five hundredths of a cent. Every scenario is inside every tolerance.

So the agreement is not an accident of summation. It holds scenario by scenario, at round-off.

26.5.3 Where they do not agree

And now the last row of Table 26.2. The largest difference in unserved power *at a bus* is 300.7 MW.

Figure 26.2 puts every recorded disagreement on one axis, and the picture is not a gradient. It is two clusters twelve orders of magnitude apart. Totals, costs and sorted dispatch sit at 10^{-12}. Bus angles, branch flows, per-generator dispatch and per-bus unserved power sit between 30 and 300.

The state-level comparison says what is happening. Over 2,875 states, the median difference in branch flow is 4.9×10^{-9} MW and the largest is 278 MW. The median difference in bus angle is 7.8×10^{-11} degrees and the largest is 30. Almost every state agrees exactly; a few do not agree at all.

The diagnostic that settles it is the dispatch vector, compared two ways. Compared generator by generator, the ninety-ninth percentile difference is 66.9 MW. Compared after sorting — so that the comparison is between the two *multisets* of output levels, ignoring which machine produced which — the ninety-ninth percentile falls to 4.9×10^{-9} MW. A factor of 1.4×10^{10}.

The same set of output levels is being produced by different generators.

This is degeneracy, and it is a property of the problem rather than of either implementation. When two generators have the same marginal cost and the network does not distinguish between them, the linear program has a continuum of optima with identical objective value. Both optimizers find an optimum. They find different ones. The objective agrees to round-off because

the objective is unique; the solution differs by hundreds of megawatts because the solution is not.

Principle 26.1. *Cross-validating two implementations validates the quantities their shared problem determines, and nothing else. Where the optimum is degenerate, two correct programs will disagree by any amount, and the disagreement is evidence about the problem rather than about the programs.*

The practical consequence is a rule for what a comparison of this kind may conclude. This one licenses the annual figures, the per-scenario totals, the event flags and the prices, which agree at round-off. It does not license a statement about which generator ran or which bus was shed — and neither implementation should be quoted on those without first checking whether the optimum that produced them was unique.

26.5.4 What the agreement is not

Fourteen digits is a seductive number and it is worth saying plainly what it does not mean.

A bootstrap over the outage draws, 2,000 resamples, puts a 95 percent interval on loss of load expectation of 12.6 to 14.8 hours a year, and on expected unserved energy of 1557 to 1909 MWh. The answer is 13.68 ± 1.1 hours — uncertain at the eight-percent level, from the fifty draws per hour and no other cause.

So the two implementations agree on a quantity to fourteen digits while the quantity itself is known to one and a half. The agreement is roughly 10^{13} times tighter than the uncertainty.

Principle 26.2. *Cross-validation tests implementation, not truth. Two programs agreeing to machine precision establishes that they solve the same problem; it says nothing whatever about whether that problem answers the question, or about how well the answer is determined by the data.*

Sampling instead of enumerating makes the same point from the other side. Runs at ten, twenty-five, fifty and a hundred draws per hour — each drawing its own scenarios rather than reading the committed file — give loss-of-load expectations between 13.20 and 14.60 hours, against 13.68 from the full enumeration. An eight-percent spread, from the sampling alone, at budgets that cost between nineteen seconds and three and a half minutes.

26.5.5 What a network-blind model would have said

A copper-plate reliability model of the same region, 10,000 samples in seventy seconds, reports a loss of load expectation of 8.3 hours a year and expected unserved energy of 1050 MWh. With the network, the figures are 13.68 and 1735.8.

The copper plate is low by 39.3 percent on one and 39.5 percent on the other. That gap is the *deliverability* component: hours in which the region has enough generation and cannot get it to the load.

Figure 26.4 shows where the gap lives, and it is a smaller place than the size of the effect suggests. Of the 684 loss-of-load events, 579 — eighty-five percent — have branch A11 sitting exactly at its 175 MW rating. In the remaining 105 no branch is at its rating at all: those are genuine generation shortfalls, which the copper plate would also have found. A11 reaches its rating in 39,745 of the 439,200 scenarios. The second most loaded branch in the system peaks at 96.5 percent of its rating and never reaches it.

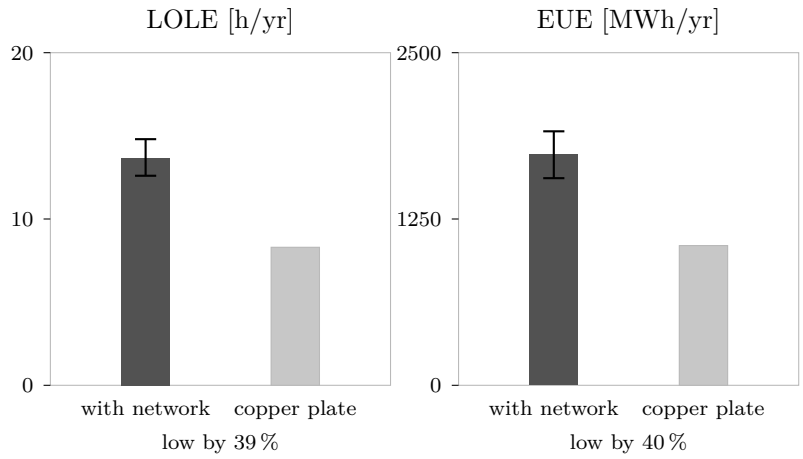


Figure 26.3: The two annual figures with the transmission network modelled, and from a copper-plate model of the same region — one that treats it as a single bus and asks only whether enough generation is available. The bars with the network carry the bootstrap interval of Section 26.5.4; the copper-plate figures are outside it by a wide margin.

A forty-percent error in a planning number, traceable to one 175 MW line. That is the case for carrying the network, and it is also a warning about how concentrated such a result can be: a study that reported only the aggregate gap would have obscured the fact that a single reconductoring decision is what the number is about.

26.5.6 Cost

Over 5,000 scenarios the linear programs take 1.36 seconds warm-started from the previous solve, 7.86 seconds solved cold, and 12.97 seconds if the model is rebuilt each time — 0.27, 1.57 and 2.60 milliseconds per scenario. Warm-starting is 5.8 times faster than solving cold and 9.6 times faster than rebuilding, which is what makes a full-year enumeration a reasonable thing to do rather than an ambitious one.

26.6 What surprised

Fourteen digits on the total, three hundred megawatts on the allocation. The expectation going in was a single verdict: either the two tools agree or they do not. They do both. Every quantity the optimum determines agrees at round-off; every quantity it does not determine disagrees by hundreds of megawatts. What made this legible rather than alarming was one cheap diagnostic — comparing the dispatch vector sorted as well as by generator. Sorted, the ninety-ninth percentile difference falls by ten orders of magnitude, which identifies the disagreement as a permutation rather than a discrepancy. Any cross-validation of an optimization model should carry that check, because without it the same evidence reads as a serious defect.

The agreement is thirteen orders of magnitude tighter than the uncertainty. Two implementations matching to 5.8×10^{-15} on a number whose bootstrap interval is ± 8 percent is an arresting combination, and it is the most useful thing in the chapter for calibrating what a

Figure 26.4: The eight most heavily loaded branches, by the highest utilization each reaches anywhere in the year. One branch reaches its rating. The next one peaks at 96 percent and never binds. The entire forty-percent deliverability effect of Figure 26.3 is that one line.

cross-validation is worth. It is a very strong statement about the software and no statement at all about the answer. The fifty draws per hour, not the arithmetic, are what limits this study, and the remedy for that is more draws, not a better implementation.

One line carries the entire network effect. A copper-plate model misses forty percent of the risk, which reads as a broad argument for network-aware modelling. The census says it is not broad. Eighty-five percent of the events have one specific branch at its rating, and the next most loaded branch in the system never binds at all. The aggregate number and the actionable number are different: forty percent is the case for the method, and A11 is the case for the capital project.

26.7 Reproduction

The full-year run is long; the hundred-hour design reproduces the structure in minutes:

```
python docs/terra_graph_engine/case_studies/foundation_power_flow/\
       annual_ra_rtsgmlc/run.py
```

It writes `crossval_metrics.json`, `crossval_metrics_full.json` and `hardening_demos.json` beside itself. The chapter's numbers and figures are then reproduced from those three files alone:

```
cd docs/books/transmission && python scripts/t7_annual_ra.py
```

which prints every value quoted above — including the sorted-versus-named dispatch ratio of Section 26.5.3 and the copper-plate gaps of Section 26.5.5, both derived from the recorded values — and rewrites the five fragments in `figures/`. A representative line of its output:

```
  the diagnostic: sorting the dispatch vector before comparing takes
  the 99th percentile from 66.92 MW to 4.913e-09 MW
  -- a factor of 1.36e+10.
```

Reproducing the other tool's half needs its own toolchain and its own language; the committed scenario file and the recorded comparison are what this chapter reads.

Notes

The test system, its generator data and its reliability parameters are standard and public. The second toolchain is an independently developed open-source power-systems stack of commercial lineage, and the copper-plate reference is a separate published reliability tool; both were run by their own maintainers' conventions rather than reimplemented here. The minimum-load-shed formulation, the direct-current approximation, loss of load expectation and expected unserved energy are standard resource-adequacy practice. Degeneracy in linear programs, and the non-uniqueness of the optimal vertex when it occurs, are textbook. The same non-uniqueness appears wherever a model has symmetries the data cannot break, and reporting it rather than resolving it arbitrarily is the general rule.

What is this chapter's own is the protocol of Section 26.3, which pins the inputs and the scenario draws in a file both implementations read so that only formulation, solver and aggregation remain free; and the diagnostic of Section 26.5.3, comparing the dispatch vector both sorted and by generator, which separates a permutation of a degenerate optimum from a genuine disagreement and turns a three-hundred-megawatt difference from a defect into a finding. The concentration result of Section 26.5.5 — that eighty-five percent of the events bind on one branch while the second most loaded branch never binds — is recorded here because the aggregate deliverability gap is the number usually quoted and it is not the number that tells a planner what to do.

Chapter 27

Where enumeration stops paying

Numbers in this chapter are produced by `scripts/t8_at_scale.py` from the study's committed results files. Nothing is retyped.

27.1 The question

Chapter 26 established that the implementation is right. This chapter asks whether the *method* is, and answers by finding the place where it is not.

The method is probability-ordered enumeration. Rare-event risk on a power system is conventionally estimated by sampling: draw outage states, evaluate them, report a confidence interval. Enumeration does the opposite. It walks the outage states in decreasing order of probability, evaluates each exactly once, and stops at a probability floor — which buys a *certified* answer, an interval with no sampling error in it, rather than a confidence statement. Chapters 23 and 21 both lean on that property.

The question a planner needs answered before adopting it is not whether it works. It is: on what systems, at what rarity, does it beat sampling — and where does it stop? A method whose advocates cannot say where it fails has not been characterized.

This chapter characterizes it. Four systems, three severity thresholds, three probability floors — thirty-seven cells, each with a complete certified reference — against fourteen sampling and screening baselines at matched evaluation budgets, over 328 recorded method runs. The result is a cost law, a regime where enumeration dominates by a factor of two, and a regime where it loses by a factor of ninety-five.

27.2 The systems, and what is being counted

Table 27.1 gives them. Replication is the trick that makes the experiment a controlled one: two copies of an area, joined, have twice the components and the same kind of topology, so component count can be varied without changing what the network is like. The real three-area system is then the control on the control — 193 components, fewer than the four-copy replica's 248, but a topology nobody designed by copying.

The quantity being measured is *watchlist coverage*. A watchlist is defined by two numbers: a probability floor $p_{\min}$, and a severity threshold. A state belongs to it if its probability is at least $p_{\min}$ *and* it causes an overload of at least the threshold. Coverage is the fraction of that watchlist a method has found when its evaluation budget runs out.

Table 27.1: The four systems. The first three are one, two and four copies of the same 24-bus area, which scales the component count while holding the topology's character fixed; the fourth is the real three-area test system. Note that the four-copy replica is the *larger* of the last two, which Section 27.4.3 makes use of. The budget is the number of distinct states any method is allowed to evaluate.

system		buses	components	budget [evaluations]	one evaluation [ms]
x1	one copy	24	62	100,000	2.40
x2	two copies	48	124	1,000,000	3.24
x4	four copies	96	248	500,000	5.76
full	the real three-area system	73	193	500,000	2.85

Both halves matter and they behave differently, which is Section 27.4.1. Lowering the floor from 10^{-5} to 10^{-7} enlarges the set of states worth considering by orders of magnitude. Raising the severity threshold from 10 to 50 MW shrinks the watchlist without touching that set at all.

27.3 Method

Every cell of the grid is given a complete certified reference: one streamed enumeration sweep of the system, carried far enough that the watchlist is known exactly rather than estimated.

Against enumeration stand fourteen baselines, chosen to be stronger than the usual straw man:

- crude Monte Carlo;
- inflated-Bernoulli importance sampling at four outage-rate multipliers, $\kappa = 2, 3, 5, 8$;
- the cross-entropy method at three sample sizes;
- subset simulation at three sample sizes;
- two screens — one ranking candidates by line outage distribution factor, one by probability times severity — and a pruning hybrid.

All are scored at the same budget of *unique* evaluations, which is the fair currency: a sampler that draws the same state twice has paid for one physics evaluation, not two, and enumeration never repeats. Each method is scored on coverage at budget and on evaluations needed to reach ninety, ninety-nine and one hundred percent coverage.

27.4 Results

27.4.1 The cost law

Figure 27.1 is the chapter's first result and it is a structural one. Within a system and a probability floor, the three severity thresholds produce watchlists differing in size by up to a factor of 4.3 — on the real system at 10^{-7}, 5,048 states at 10 MW against 1,169 at 50 MW — and enumeration needs the *same* number of evaluations to cover all three. In eight of the twelve system-and-floor pairs the count is identical to the unit; in the other four it varies by at most fourteen percent.

The reason is immediate once stated. Enumeration does not look for the watchlist. It walks every state above the floor, in probability order, and the watchlist is whatever it notices on the

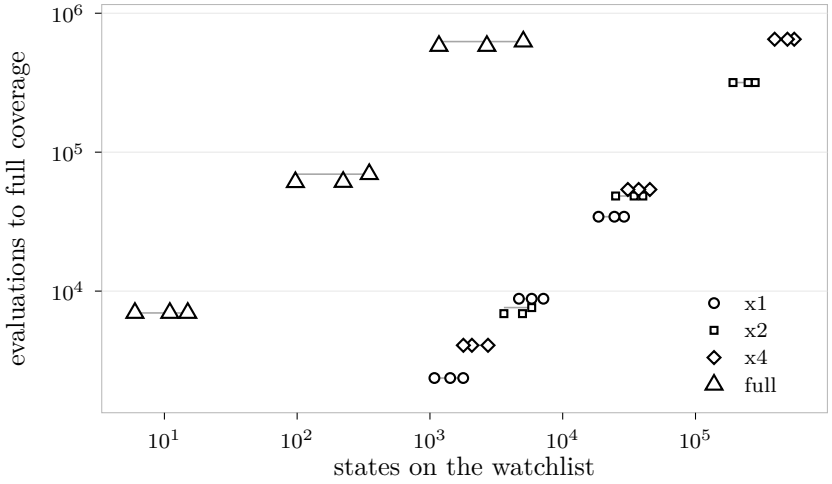


Figure 27.1: Evaluations enumeration needs for complete coverage, against the size of the watchlist it is covering. Each horizontal run of three points is one system at one probability floor, with the three severity thresholds; the watchlist varies along it and the cost does not. The cost of enumeration is set by the probability floor alone.

way. The states above the floor are the same states whichever watchlist you asked for, so the cost is

$$\text{evaluations to full coverage} \;\approx\; \big|\{\, s : p(s) \geq p_{\min} \,\}\big|,$$

independent of the severity threshold.

Principle 27.1. *Enumeration's cost is set by how rare an event you are willing to ignore, not by how many events you are looking for. A sampler's cost is set by the opposite. That is the whole of the trade, and it tells you which regime each belongs in before either is run.*

The practical reading is a decision rule. If the watchlist is a large fraction of the states above the floor — a congested system, a loose severity threshold — enumeration is finding something on almost every evaluation and sampling has nothing to exploit. If the watchlist is a tiny fraction, enumeration spends most of its budget on states it will discard, and a method that can guess where to look has room to win.

27.4.2 Where it dominates, and where it inverts

At probability floors of 10^{-5} and 10^{-6} there is no contest to report: on all four systems and all three severities, enumeration and the best baseline both reach complete coverage. Twenty-four of the thirty-six cells are ties at 1.0000.

Figure 27.2 shows the remaining twelve, at 10^{-7}.

On one copy and two copies enumeration covers the entire watchlist; the best of fourteen baselines reaches 0.918 to 0.945 on one copy and ties on two.

On four copies — the congested replica, 248 components — enumeration reaches 0.80 and the best baseline 0.34. A factor of 2.4. This is the regime the cost law predicts: the watchlist is a large fraction of what is above the floor, and there is nothing for a sampler to exploit.

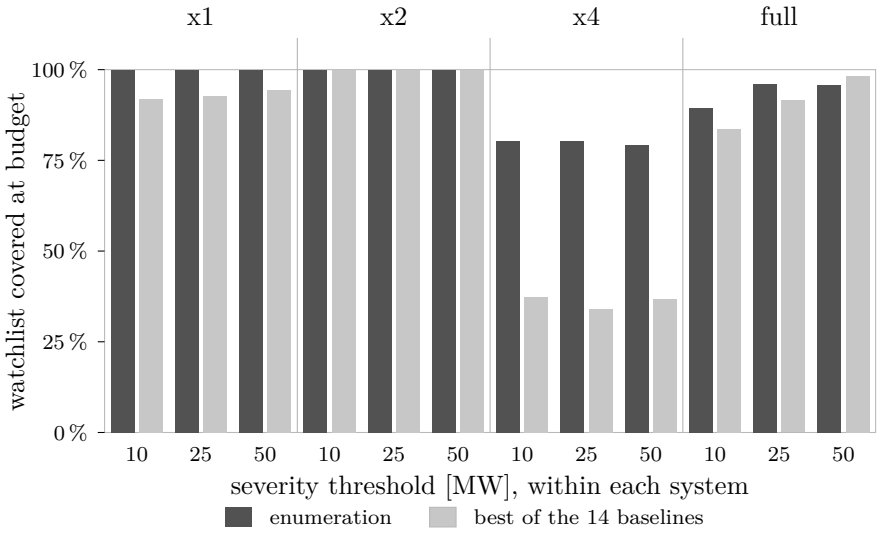


Figure 27.2: Coverage at budget at the deepest probability floor, 10^{-7}, where the contest is decided. At the two shallower floors every method on every system covers the whole watchlist, and the grid is uninformative. Here enumeration is complete on one and two copies, wins by a factor of two on four copies, and loses on the real system at the loosest watchlist.

On the real system it inverts. At the 50 MW severity threshold the probability-times-severity screen reaches 0.982 coverage where enumeration reaches 0.957. And the cost difference is not marginal:

- to reach ninety percent coverage, enumeration needs 422,823 evaluations and the screen needs 4,453 — a factor of 95;
- at the 25 MW threshold, 445,711 against 23,655 — a factor of 19;
- at 10 MW the screen never reaches ninety percent at all, and enumeration does.

Subset simulation also matches enumeration on the real system at 50 MW, reaching 0.977.

So the answer to the chapter's question is a boundary rather than a verdict. Enumeration is complete and cheap when the floor is shallow. It dominates on congested systems at deep floors. It is beaten, by up to two orders of magnitude in cost, on the real system at a deep floor and a demanding severity threshold.

27.4.3 It is not about size

The obvious reading of Section 27.4.2 is that enumeration works on small systems and fails on large ones. The grid rules that out.

The four-copy replica has 248 components. The real system has 193. The replica is the larger of the two, and it is the one where enumeration wins by a factor of 2.4; the smaller one is where it loses by a factor of 95 in cost. Size is not the variable.

What differs is where the risk sits. A replicated network has its congestion spread evenly by construction — every copy contributes its own share, and no small set of states dominates. The real three-area system does not: its risk concentrates, and a screen that ranks candidates by probability times an estimate of severity can find that concentration without evaluating much. On the replica there is no concentration to find, so the screen has no advantage and enumeration's exhaustiveness is worth what it costs.

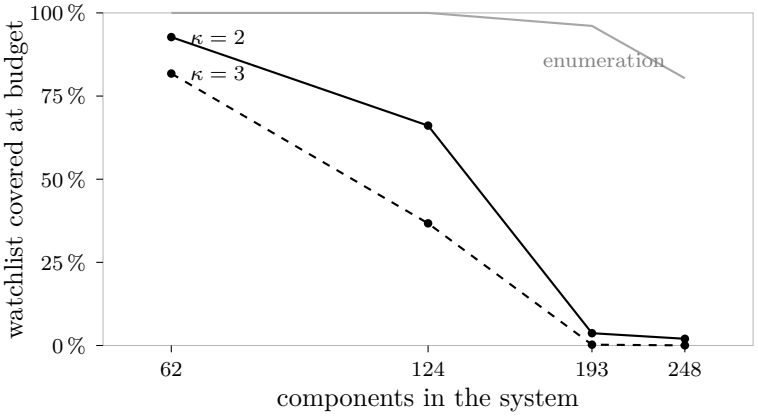


Figure 27.3: Inflated-Bernoulli importance sampling at two outage-rate multipliers, against component count, at the deepest floor and the middle severity. The collapse is not gradual. Between 124 and 193 components, coverage at a fixed multiplier falls from 0.66 to 0.04. The gray line is enumeration on the same cells.

> **Principle 27.2.** *Whether a screen beats an exhaustive search is a question about the distribution of risk in the system, not about the system's size. Replicating a network to study scaling changes the size and destroys the concentration, which is exactly the property the comparison turns on.*

That last sentence is also a warning about the experiment's own design. Three of the four systems here are replicas, and replication is what made the component count controllable. It is also what made three of the four systems unrepresentative of the thing being modelled. The real system had to be in the grid, and it is the one that produced the finding.

27.4.4 Importance sampling collapses with component count

Figure 27.3 isolates one baseline family, because its failure mode is the clearest illustration of why this comparison needed several systems rather than one.

Inflated-Bernoulli importance sampling multiplies every component's outage rate by a constant κ, samples from the inflated distribution, and reweights. At 62 components and $\kappa = 2$ it covers 0.93 of the watchlist. At 124 it covers 0.66. At 193 and 248 it covers 0.04 and 0.02.

A fixed multiplier cannot survive a growing component count, because the number of simultaneous outages it induces grows with the number of components while the states that matter do not. Raising κ makes it worse, not better: at 62 components, $\kappa = 2$ gives 0.93 and $\kappa = 8$ gives 0.33. The method has a tuning parameter that must be re-tuned per system and no principled way to set it, which is the honest reason it is a weak baseline — and the reason the study also carries cross-entropy and subset simulation, which adapt.

27.4.5 The DC watchlist, judged under AC

Everything above is computed on a direct-current approximation of the network. That approximation is what makes a 650,000-evaluation enumeration possible — one evaluation takes between 2.4 and 5.8 milliseconds — and it is also an assumption the whole chapter rests on.

Table 27.2: The DC watchlist judged by a full AC solve on sampled states, at the 25 MW severity threshold. On one copy the DC verdict on active power is exact. On the real system it misses 200 of the 2,000 sampled states, and never raises a false alarm. The apparent-power row is a different question and disagrees on both.

of the sampled states	one copy	the real system
both benign	2,809	1,629
both violating	5,970	13
DC benign, AC violating on active power	0	200
DC violating, AC benign on active power	0	0
DC benign, AC violating on apparent power	443	350
states sampled	11,336	2,000
AC solves that converged	81.6 %	99.6 %

Two checks. First the DC path itself: an independently written implementation reproduced 20,000 scenarios on the real system, and the largest difference in severity, shed power and shed cost is *zero*, with a 12,000-row flow subsample also at zero. The comparison of Chapter 26 applies here too.

Then the assumption. Table 27.2 re-judges sampled states with a full AC power flow. On one copy the DC watchlist is exact on active power: no state the DC model called benign is AC-violating, and none it called violating is AC-benign. On the real system it is conservative in one direction only — zero false alarms, and 200 of 2,000 sampled states, ten percent, that DC calls benign and AC calls violating.

So the DC screen does not invent risk; it misses some. Ten percent of a sampled population is not a small miss, and it is the honest limit on everything the chapter has measured: the watchlists are DC watchlists, and the coverage figures are coverage of those.

An AC solve takes 206 milliseconds at the median on the real system against 2.85 for the DC evaluation — a factor of 72 — which is why the grid is not computed that way and why an AC-exact version of this experiment is a different undertaking.

27.4.6 Cost

The certified references are the expensive part: one streamed sweep per system, between 1.0 and 20.8 million states, taking from 0.11 to 4.1 hours. Two of the four sweeps stopped on a state cap rather than on the probability floor — the four-copy replica reached cumulative probability mass 0.63 and the real system 0.89 — which is recorded because it bounds what the references can certify.

27.5 What surprised

The cost of an exhaustive search does not depend on what you are searching for. Stated after the fact this is obvious: enumeration walks every state above the floor regardless. Stated before, it is not how anyone reasons about search cost, and it has a sharp consequence — tightening the severity threshold, which is the natural way to make a study cheaper, buys enumeration nothing at all. Eight of twelve system-and-floor pairs give a cost identical to the unit across a watchlist that varies by a factor of four. The lever that works is the probability floor,

and that is a statement about what risk you are willing to ignore rather than a computational parameter.

The inversion is on the smaller system. Going in, the expected result was a scaling curve: enumeration wins below some component count and loses above it. What happened instead is that it wins on the 248-component replica and loses on the 193-component real system. The variable is whether risk concentrates, and replication — the device that made component count controllable — destroys concentration. Three quarters of this experiment's systems were therefore unrepresentative of the thing being modelled in exactly the respect that decides the answer, and only the one real system revealed it. That is a general hazard in scaling studies built on synthetic replication, and it is worth more than the specific numbers here.

A method that loses is still worth carrying. The obvious response to Section 27.4.2 is to prefer the screen on real systems. But the screen provides a ranked list, not a certificate; it reached 0.982 coverage and cannot say so without the enumeration that produced the reference. Every coverage figure in this chapter, for every baseline, is measured against an exhaustive sweep. The expensive method's role at that point is not to be the production method — it is to be the thing that makes the cheap method's claims checkable, which is a different and permanent job.

27.6 Reproduction

The full grid is hours of compute; the one-copy system reproduces the structure in minutes:

```
python docs/terra_graph_engine/case_studies/foundation_power_flow/\
       contingency_risk_at_scale/cras_run.py
```

It writes `frontier.json`, `eval_rates.json`, `ac_spike.json` and `crossval_full.json` among others beside itself. The chapter's numbers and figures are then reproduced from those four alone:

```
cd docs/books/transmission && python scripts/t8_at_scale.py
```

which prints every value quoted above — including the cost-law table of Section 27.4.1 and the evaluation ratios of Section 27.4.2, both derived from the recorded cells — and rewrites the five fragments in `figures/`. A representative line of its output:

```
  x4  1e-07  554 324 -> 649 611  492 868 -> 649 611  394 596 -> 649 611
      watchlist varies 1.40x, cost varies 1.000x
```

Notes

The test system and its replications are standard and public. Crude Monte Carlo, inflated-Bernoulli importance sampling, the cross-entropy method and subset simulation are all standard rare-event techniques and are used here as their literature defines them; the line-outage-distribution-factor screen is routine contingency-analysis practice. Probability-ordered enumeration with a certified truncation bound, and the argument that it yields an interval rather

than a confidence statement, are Chapter 15; the direct-current approximation and its severity accounting are standard.

What is this chapter's own is the experiment's shape and what it found: a controlled grid over component count, probability floor and severity threshold, with a complete certified reference in every cell, which is what makes coverage a measurable quantity rather than an estimate. From it come three results reported here — the cost law of Section 27.4.1, that enumeration's cost is set by the probability floor and is nearly independent of the watchlist; the inversion of Section 27.4.2, with the ninety-fivefold cost difference on the real system; and the observation of Section 27.4.3 that the inversion tracks the concentration of risk rather than the size of the network, so that a scaling study built on replication is systematically blind to the property that decides it.

Chapter 28

What a sensor is worth

Numbers in this chapter are produced by `scripts/t9_sensor_value.py` from two studies' committed results files. Nothing is retyped.

28.1 The question

A transmission line's rating is a limit on how much power it may carry, and it is set by how fast the conductor sheds heat — which depends on the wind. The conventional rating assumes a still day: a fixed 0.61 m/s, every hour of the season. A *dynamic* line rating uses the weather instead, and the case for it is a large one, because the wind is usually blowing.

That case is almost always made by comparing a dynamic rating computed from forecast weather against the static one. This chapter argues the comparison is the wrong one, and then measures how wrong.

The reason is that a rating a utility *publishes* carries an obligation. It is not a best estimate of what the line can carry; it is a number the true ampacity must not fall below more than a stated fraction ε of the time. Wind at a conductor span is not measured — it is inferred from a sparse network of weather stations — so the ampacity it implies is a distribution, and the publishable rating is a low quantile of that distribution.

A quantile is not a mean, and the difference is the chapter:

- **How much of the dynamic rating's benefit survives the quantile?**
- **What does a sensor buy — and is it the thing you would expect?**

Two studies answer them from opposite sides of the decision. One is a corridor with no conductor-mounted sensors at all, asking what a fleet would be worth before buying it. The other is a fleet already installed, asking what it is worth now that it is there.

28.2 The two systems

Table 28.1 gives them. The first is a 138 kV corridor carrying a 220 MW interconnection, thirty-four monitored circuits, evaluated over 5,881 hours against an $N-1$ bus-outage curtailment screen. Its wind comes from a network of 213 weather stations and nothing on the conductors. The question it asks is a procurement question: would eight conductor-mounted sensors pay for themselves, and where would they go?

The second has twenty-four units already installed across nine circuits, and 34,206 hours of them. Its question is an accounting one: what are they worth?

Table 28.1: The two studies. They differ in almost every way that matters — one is a single corridor with a hypothetical fleet, the other nine circuits with two dozen units already on the conductors — and they are held to the same exceedance bound.

	no conductor sensors	a deployed fleet
circuits	34	9
weather stations	213	87
conductor-mounted units	8 (hypothetical)	24 (installed)
hours in the window	5,881	34,206
median wind speed [m/s]	4.07	1.71
exceedance bound	0.05	0.05

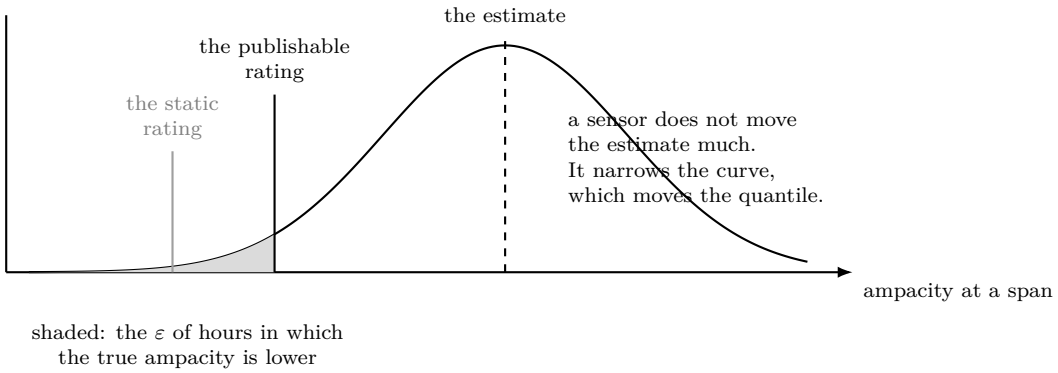


Figure 28.1: Why the published number is not the estimated one. At each hour the inferred ampacity is a distribution, and the rating a utility may publish is the quantile below which the truth falls no more than ε of the time. Narrowing the distribution raises that quantile without moving the estimate at all — which is what a conductor-mounted sensor is bought to do.

Both hold the same bound, $\varepsilon = 0.05$ — a rating the true ampacity may fall below in no more than five percent of hours.

28.3 What a publishable rating is

Figure 28.1 is the whole idea. Everything else in the chapter is measurement.

The rating is the minimum over the spans of a circuit, at each hour, of a low quantile of the posterior ampacity — and the posterior comes from conditioning a wind field on whatever is measured. With only weather stations, the posterior at a span far from any station is wide. With a sensor on the conductor, it is narrower there.

28.4 Results

28.4.1 Confidence eats the benefit

Figure 28.2 is the first result and it reverses the conventional case.

Treated as truth, the dynamic rating supports a capacity factor of 99.93 percent against the static rating's 98.81 — better by 1.11 percentage points, which is the benefit the literature

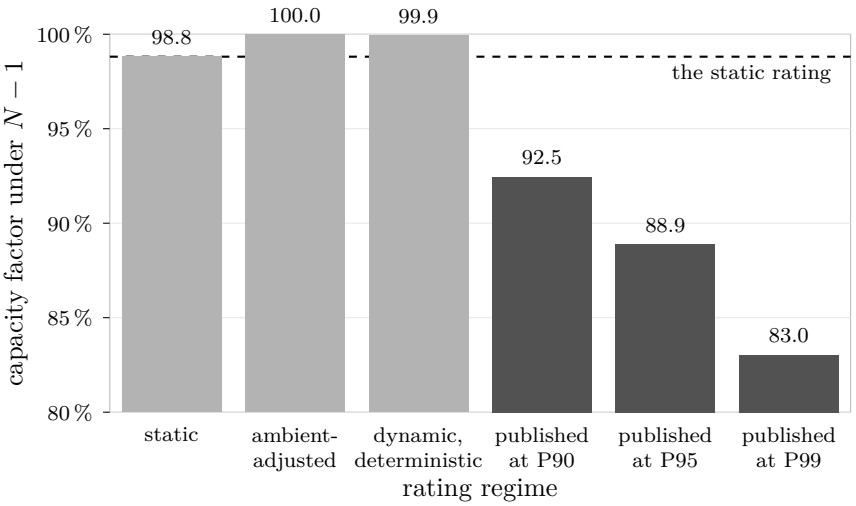


Figure 28.2: Capacity factor on the corridor under $N-1$, by rating regime. The three pale bars are ratings treated as known: the static seasonal rating, an ambient-adjusted one, and a dynamic rating computed from forecast weather as though the forecast were truth. The three dark bars are the same dynamic rating published with an exceedance bound. The dashed line is the static rating, which every published variant falls below.

reports.

Published with an exceedance bound, the same dynamic rating supports 92.45 percent at $P90$, 88.89 at $P95$ and 83.03 at $P99$. At the $P95$ level a utility would plausibly publish, the dynamic rating is 9.92 percentage points *worse* than the static one it was meant to improve on.

On the study's energy basis the same reversal reads as -10.04 percent of served energy against static, and the publishable dynamic rating is at least as large as the static rating in only 30.9 percent of hours.

> **Principle 28.1.** *A rating is a promise, not an estimate. Replacing an assumption with a measurement improves the estimate and does not, by itself, improve the promise — because the promise is a quantile, and a quantile depends on the width of the distribution as much as on its centre.*

There is nothing wrong with the dynamic rating here. It is a better estimate. It is the *quantile* of a wide distribution that loses to a conservative constant, and the distribution is wide because wind at a conductor span is being inferred from stations that are not at the conductor.

28.4.2 The bound is honest

Before any of that can be believed, the quantile itself has to be checked. Table 28.2 does it: configured exceedance levels of 0.01, 0.05 and 0.10 produce realized exceedances of 0.01003, 0.05051 and 0.10019. The bound is delivered to three digits.

The lower half of the table is the setup for the next section, and it is worth reading now. With no stations at all the wind posterior has a standard deviation of 1.091 m/s everywhere. The existing station network brings that to 0.573, 0.837 and 0.953 m/s at three representative

Table 28.2: Top: the realized exceedance against the configured level, over 20,000 draws — a check that the quantile machinery delivers the bound it claims, since every result in this chapter is meaningless if it does not. Bottom: the standard deviation of the wind posterior at three spans, under four states of knowledge.

configured exceedance	realized	draws
0.01	0.01003	20,000
0.05	0.05051	20,000
0.1	0.10019	20,000
wind posterior, standard deviation at three spans [m/s]		
prior, no stations		1.091, 1.091, 1.091
the existing station network		0.573, 0.837, 0.953
with eight conductor sensors		0.573, 0.803, 0.947
the deployed fleet		0.415, 0.492, 0.720

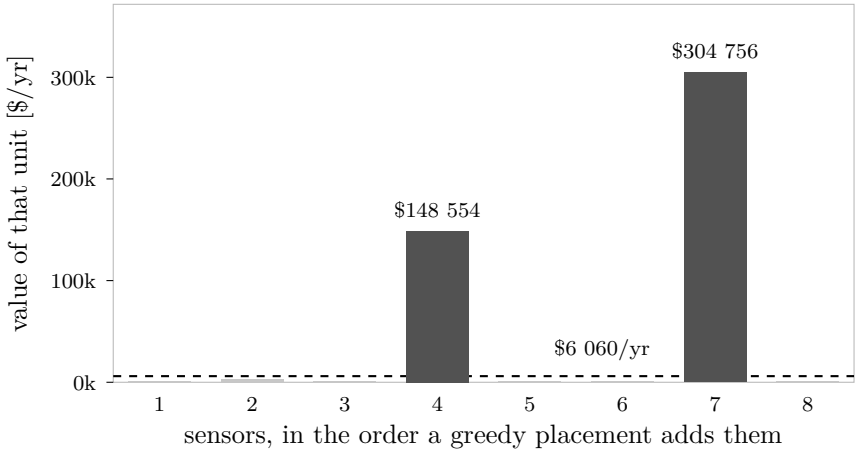


Figure 28.3: The eight sensors, in the order a greedy placement adds them, each valued at what it recovers in curtailed energy. Two of the eight pay for themselves; the dashed line is what a unit costs. This is not a diminishing-returns curve with a crossing point. It is a step function, and six of the eight steps are flat.

spans — good near a station, poor far from one. Adding eight conductor-mounted sensors brings the worst of the three from 0.953 to 0.947.

That is a narrowing of six tenths of one percent.

28.4.3 What each sensor is worth

Figure 28.3 is the chapter's second result.

The eight units, added greedily, are worth $2, $2,798, $2, $148,554, $1, $831, $304,756 and $319 a year. Against a unit cost of $6,060 a year, exactly two of them pay for themselves — the fourth and the seventh — and those two carry 99.1 percent of the fleet's entire value.

The fleet as a whole is worth $457,263 a year against $48,476 of cost, a factor of 9.4. That headline is true and it is also nearly uninformative, because it averages two units that return twenty-five and fifty times their cost with six that return almost nothing.

Now put the two sections together. The fleet narrows the widest wind posterior by *six tenths of one percent* and is worth nine times its cost. Those are not in tension; they explain each other. Value does not accrue in proportion to information. It accrues when the rating crosses the curtailment screen's threshold in an hour where it previously did not — a discrete event — and most of the narrowing happens in hours where no threshold is nearby.

> **Principle 28.2.** *The value of a measurement is not how much it tells you. It is whether what it tells you moves a decision across a threshold. A sensor that halves an uncertainty nobody acts on is worth nothing, and one that shifts a rating past a screening limit for a few hundred hours a year is worth its cost many times over.*

The estimate is sensitive to one assumption, and the chapter should say so. The recovered energy depends on the assumed correlation length of the wind field: at a quarter of the study's value the fleet recovers 43,587 MWh/yr, and at one and a half times it, 6,913. A factor of six across a plausible range. The headline $457,263 rests on a correlation length that was fitted, not measured.

28.4.4 A fleet that is already there

The second study removes the hypothetical. Twenty-four units are on the conductors of nine circuits, and the wind posterior at a span is conditioned on what they actually read.

At the same $\varepsilon = 0.05$, the published rating now averages 214.7 MVA with one sensor per circuit and 223.3 with two, against the static rating's 207.2 and a deterministic ceiling of 277.4. That is +3.63 and +7.80 percent over static — above it, where the corridor study's publishable rating was ten percent below.

The published rating exceeds the static one in 46.1 percent of hours with one sensor and 52.3 with two, and closes 13.8 percent of the gap to the deterministic ceiling. The twenty-four units cost 0.36 percent of the annual uplift they produce.

So the two studies do not disagree. They measure the same quantity at two points: a wind posterior conditioned only on distant stations gives a publishable rating below static, and one conditioned on conductor-mounted sensors gives a publishable rating above it. The sensors are what moves the quantile, and the corridor study's finding is what the deployed fleet's circuits looked like before they were installed.

28.4.5 One circuit loses, and the numbers hold up

Figure 28.4 reports the per-circuit result rather than the fleet average, which is where the honest detail lives. Uplift ranges from +24.78 percent on the best circuit to −1.96 percent on the worst. One of the nine circuits is made *worse* by having sensors on it: conditioning on its units narrows the posterior somewhere that does not matter and shifts the quantile the wrong way at the span that binds.

The robustness checks are reassuring rather than decisive. Fitting each unit's observation noise separately, and excluding readings that failed quality control, move the per-circuit uplift by at most 5.65 percentage points and change no circuit's sign. The sensors' readings are also strongly correlated with each other — pairwise correlation between 0.71 and 0.94 over distances of three to twenty-nine kilometres — which is the reason the second sensor on a circuit adds much less than the first.

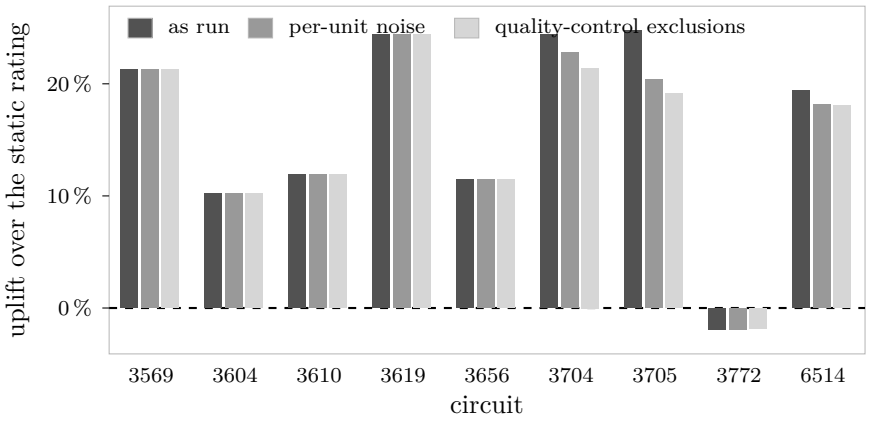


Figure 28.4: Uplift over the static rating, per circuit, under three treatments of the sensor data: as run, with per-unit observation noise fitted separately, and with quality-control-flagged readings excluded. Eight of the nine circuits gain; one loses. The three treatments agree closely enough that no circuit changes sign.

28.5 What surprised

Measuring the weather made the rating worse. The expected result was a smaller benefit than the deterministic case suggests, with the publishable dynamic rating somewhere between static and deterministic. Instead it is 9.9 percentage points below static — worse than the constant it was meant to replace, and worse at every confidence level tested. The dynamic rating is a better estimate of the ampacity and a worse promise about it, because the utility must publish a quantile and the quantile of a wide distribution can sit below a conservative constant. Any evaluation of a sensing programme that compares estimates rather than publishable quantities will overstate the case, and this one overstates it by enough to reverse the sign.

Six tenths of one percent of narrowing is worth nine times its cost. The eight-sensor fleet takes the widest wind posterior from 0.953 to 0.947 m/s. Read as an information gain that is nothing. Read as money it is $457,263 a year. The two readings are reconciled by the curtailment screen being a threshold: what matters is not how much the posterior narrows but whether the narrowing happens in the hours and at the spans where a rating is about to cross a limit. This is the sharpest instance in the edition of a general point — Chapter 21 makes it about gradients and Chapter 27 about screens — that the quantity worth computing is the one attached to a decision.

The average is the least useful number in the study. Two units of eight carry 99.1 percent of the value. A procurement decision taken on the fleet average — nine times cost, buy eight — would buy six units worth less than their price. The decision the analysis actually supports is: buy two, at these two locations. That a marginal-value curve turns out to be a step function rather than a smooth decline is not special to this corridor; it follows from the value being generated at a threshold, and it means the ordering matters more than the count.

28.6 Reproduction

Each study runs on its own committed data:

```
python docs/terra_graph_engine/case_studies/transmission/\
       dlr_rating_uncertainty/run.py
python docs/terra_graph_engine/case_studies/transmission/\
       wirewarrior_deployed_value/run.py
```

They write `results.json` and `ww_results.json` beside themselves. The chapter's numbers and figures are then reproduced from those two files alone:

```
cd docs/books/transmission && python scripts/t9_sensor_value.py
```

which prints every value quoted above — including the step structure of Section 28.4.3 and the posterior narrowing of Section 28.4.2, both derived from the recorded values — and rewrites the five fragments in `figures/`. A representative line of its output:

```
  2 of 8 units pay for themselves: 4, 7
  they carry 99.1 % of the fleet's total value
  the eight sensors narrow the widest of those three by 0.6 %
```

Notes

The corridor, its conductors, its curtailment screen and its $N-1$ bus-outage set are taken from a published deterministic non-curtailment study of the same interconnection, and the static, ambient-adjusted and deterministic dynamic ratings of Figure 28.2 are that study's; what is added here is the layer its own data manifest names as absent, a probabilistic envelope around the dynamic rating. The thermal model relating wind to ampacity is standard, as is the convention of a 0.61 m/s assumed wind in the static rating. Conditioning a spatial field on sparse observations and reading a quantile from the posterior is Chapter 6; the exceedance audit is the ordinary calibration check — generate truths, count how often the claim holds — applied to a quantile rather than to an interval.

What is this chapter's own is the pairing of the two studies — the same machinery run on a corridor with no conductor sensors and on a fleet already installed — and what the pair shows: that a publishable dynamic rating with only station data sits ten percent below the static rating it was meant to improve, that conductor-mounted sensors move it above, and that the value of those sensors is a step function in which two of eight units carry ninety-nine percent of the benefit while narrowing the wind posterior by less than one percent. The last of those is reported here because it is the result that changes what a utility should buy, and because the fleet average, which is the number such analyses usually report, conceals it.

Chapter 29

Beyond correctness

Numbers in this chapter are produced by `scripts/t10_beyond_correctness.py` from the study's committed results files. Nothing is retyped.

29.1 The question

Chapter 26 established that this workload's three annual numbers are right. Two independently written toolchains, 439,200 scenarios, agreement to fourteen digits. That is as strong a correctness result as the subject allows.

It is also not, by itself, an argument for anything. A spreadsheet could have produced those three numbers. The construction of this book carries a full model of the network — every bus, every branch, every conservation row — and if all that model yields is a loss-of-load expectation, the model was expensive and a simpler thing would have done.

So this chapter asks the other question. Given that the model is right, what *else* does it give you? Not from a second study, not from new data, and not from re-running the year — from the same rows, asked differently.

Two things, and they are different kinds of thing.

- **Which units failed?** The annual run walked forward: outages in, dispatch out. Walk the same rows backward — measurements in, outage out — and you have a diagnosis.
- **What is a megawatt worth, and where?** Every scenario was a linear program, and every linear program already computed dual variables. Those duals are the marginal value of capacity at each bus. They were sitting in the solver's output the whole time.

29.2 Which units failed

29.2.1 The setup, and three baselines

One thousand and thirteen hours are taken from the validated year: all 684 in which load went unserved, and 329 quiet ones as controls. Each is given noisy measurements — voltage to 0.005 per unit, power to 0.02 — and the question is which generating units were out.

The posterior is formed over outage hypotheses by the machinery of Chapter 9: each hypothesis is solved to its most probable state and scored by evidence.

The comparison that matters is against simpler rules using the same information, and three are run:

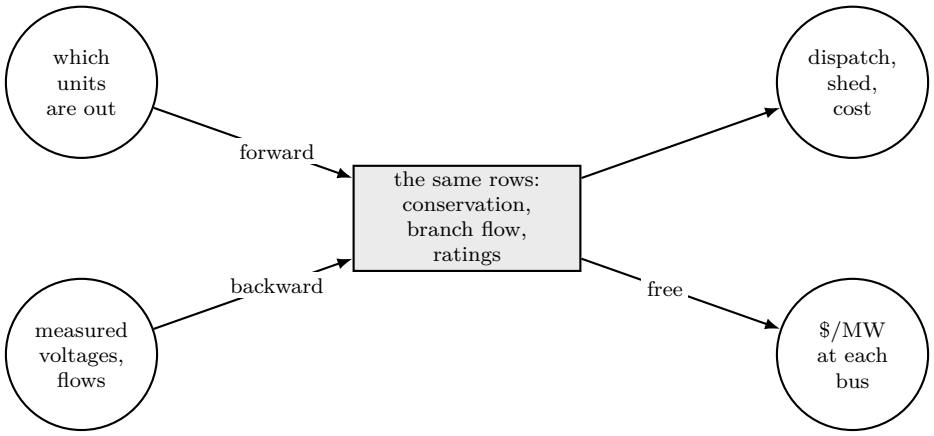


Figure 29.1: One model, three questions. The annual run of Chapter 26 is the top path: outages in, dispatch out. The same rows read backward turn measurements into a diagnosis, and the duals the solver produced along the way price capacity at every bus. Neither of the lower paths is a new model, and neither needs the year re-run.

Table 29.1: Naming the buses at which units failed, against three rules using the same measurements. "Precision" is the share of named buses that really failed; "recall" the share of failed buses that were named. The quiet hours are the controls: a method that hallucinates outages when nothing is wrong fails there.

	shortfall events		quiet hours	
naming the buses that failed	precision	recall	precision	recall
nothing but the prior	1.0000	0.0000	1.0000	0.0000
a deficit rule	0.4695	0.3409	0.0879	0.2065
deficit plus location	1.0000	0.7150	0.9793	0.9161
the inferred posterior	0.9994	0.9526	1.0000	0.9430

- *nothing but the prior* — name whatever is most likely a priori;
- *a deficit rule* — name units on buses where generation falls short;
- *deficit plus location* — the same, restricted to buses where the measurements say the shortfall actually is.

The third is deliberately strong. A weak baseline makes any method look good, and the point of the chapter is what the model adds over what an engineer would reach for.

29.2.2 What it buys

Table 29.1 and Figure 29.2 give it.

The posterior names failed buses with precision 0.9994 and recall 0.9526 on the shortfall hours. On the quiet hours its precision is 1.0000: across 329 controls it never once names a bus that did not fail. It recovers the entire outage set exactly in 54.1 percent of events and the exact set of *buses* in 87.3 percent.

The honest comparison is the third baseline. Deficit plus location achieves precision 1.0000 and recall 0.7150. So a rule an engineer would write in an afternoon, given the same measurements, finds nearly three quarters of the failed buses and never guesses wrong.

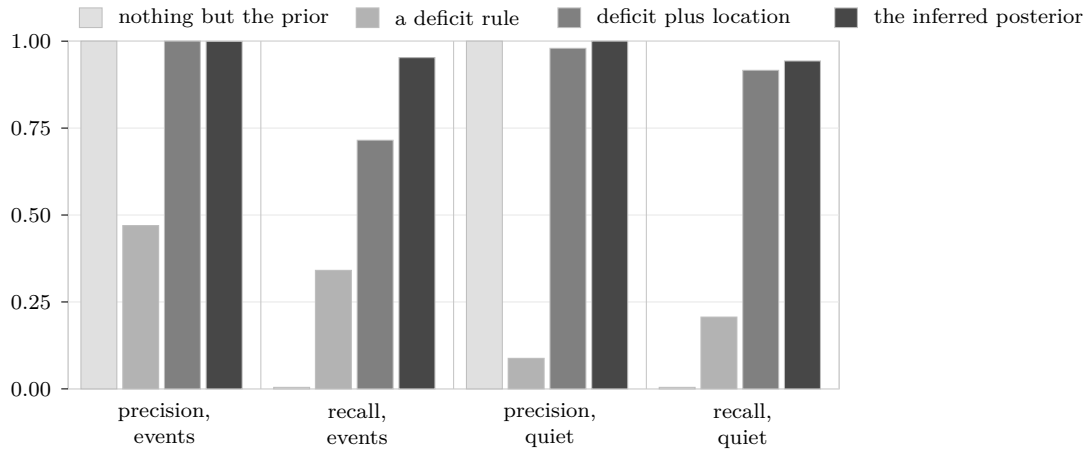


Figure 29.2: The same four methods, drawn. The prior alone names nothing, so its precision is vacuously perfect and its recall zero. The deficit rule names badly. Deficit plus location is a genuinely good rule and gets most of the way. The posterior's contribution is almost entirely recall.

What the model adds is the remaining quarter of the recall — 0.7150 to 0.9526 — at effectively unchanged precision, and a probability attached to each answer. It costs 1.47 hours over the 1,013 cases where all three baselines together cost 17.4 seconds, a factor of about 300. Whether that is worth a full network model is a judgement, and the chapter's job is to state the size of it on both sides rather than to inflate one.

29.2.3 Are the stated odds honest?

A probability is a claim, and Figure 29.3 checks it across 24,312 (bus, hour) pairs. The Brier score is 0.0268 and the expected calibration error 0.0081.

The shape is more informative than either number. The bucket below 0.1 holds 21,266 pairs, claims 0.0111 and delivers 0.0149. The bucket above 0.9 holds 1,512, claims 0.9999 and delivers 0.9967. Those two account for 93.7 percent of everything the posterior says, and both are calibrated.

The middle is not. Between 0.5 and 0.9 the posterior is consistently over-confident: the 0.8–0.9 bucket claims 0.8171 and delivers 0.5385, on 52 pairs. Where it hedges, it hedges too little.

Operationally that is the benign arrangement — the confident claims and the dismissals are trustworthy, and it is the shrugs that should be discounted — but it is worth saying plainly that a well-calibrated aggregate score can hide it. A Brier score of 0.0268 is dominated by the twenty-one thousand pairs near zero, which are the easy ones.

Principle 29.1. *An aggregate calibration score is an average over a population the posterior does not sample evenly. When most claims are near certainty or near zero, a good score says those extremes are honest and says almost nothing about the claims in between — which are exactly the ones a reader would want a probability for.*

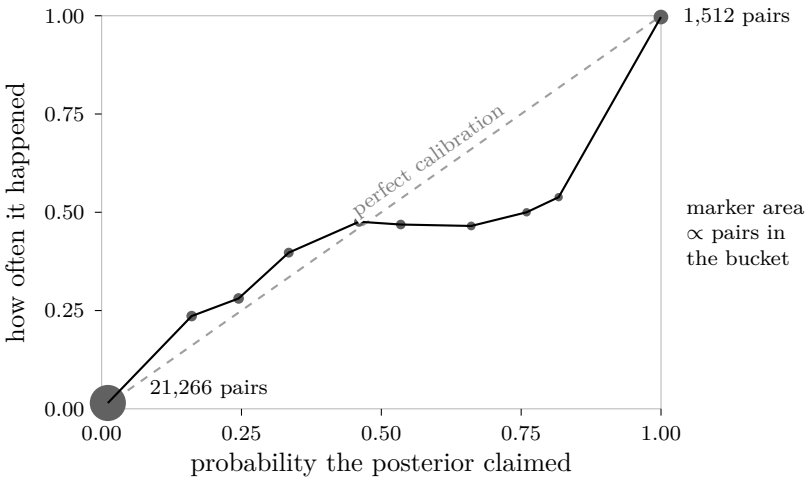


Figure 29.3: Every (bus, hour) pair the posterior assigned a probability to, bucketed by that probability and plotted against how often those buses actually failed. Marker area is the number of pairs in the bucket. The two buckets at the ends hold 93.7 percent of the pairs and sit on the diagonal; the sparse middle does not.

29.2.4 Does it survive a wrong model?

Two checks, in Table 29.2.

The inference assumes it knows the network. Perturbing impedances by two percent changes essentially nothing: precision 0.9994, recall 0.9521. At five percent, 0.9972 and 0.9478. At ten percent it degrades — precision falls to 0.9610 and the exact-bus-set rate from 0.8728 to 0.7690 — so the method tolerates the model error a real network model carries and does not tolerate an order more.

The second check is of the search rather than the model. The posterior is formed over a searched set of hypotheses, not an enumerated one, so on fifty sampled cases every outage set of three units or fewer was enumerated exhaustively and compared. The truth lay inside that space in 33 of the 50; there, the search found the exhaustive maximum every time. Across all fifty the search matched or beat the exhaustive maximum in every case — and every one of the fifteen nominal disagreements has a *negative* score gap, meaning the search returned a hypothesis the restricted enumeration could not reach because it was larger than three units.

Cost grows steeply with the candidate set rather than with the network. Six candidates cost 8 seconds and 22 hypotheses; twenty-four cost 151 seconds and 365. Quadrupling the network from 24 buses to 96 takes one evaluation from 0.61 to 2.37 seconds and the whole search from 151 seconds to 3,974.

29.3 What a megawatt is worth, and where

The second question needs no new solve at all.

Each of the 439,200 linear programs produced a dual variable at every bus: the rate at which the objective would improve given one more megawatt there. Averaged over the year with the right weights, that is the marginal value of capacity at that bus in dollars per megawatt-year — the quantity a capacity market is trying to price.

Table 29.2: Top: the inference run against a deliberately wrong network — branch impedances perturbed, load split jittered, reactive power ratios varied. Bottom: the hypothesis search audited against exhaustive enumeration of every outage set of three units or fewer, on fifty sampled cases.

the model the inference uses	bus precision	bus recall	exact, buses
exact	0.9994	0.9526	0.8728
perturbed, $\sigma = 2\,\%$	0.9994	0.9521	0.8713
perturbed, $\sigma = 5\,\%$	0.9972	0.9478	0.8611
perturbed, $\sigma = 10\,\%$	0.9610	0.9484	0.7690
the search, audited against exhaustive enumeration of subsets of three or fewer			
cases audited			50
truth inside the enumerated space			33
search matched the exhaustive maximum there			1.00
search matched or beat it overall			1.00

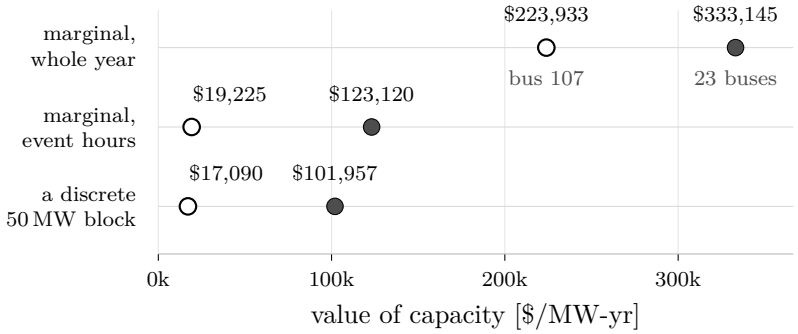


Figure 29.4: The value of capacity at the 24 buses under three pricing conventions, all of it read off the duals of one pass. Each row would hold twenty-four markers; it holds two, because under every convention the buses take exactly two distinct values. The open marker is bus 107 in all three rows — the bus behind the constrained branch that Chapter 26 found carrying the entire deliverability effect.

Figure 29.4 is the result and it has two halves.

The structure. Across twenty-four buses there are exactly *two* distinct marginal values: twenty-three at $333,145/MW-year and one — bus 107 — at $223,933, a ratio of 1.49. That is the same two-class structure Chapter 26's binding-line census produced from a completely different computation, and it arrives here for free, with the bus named. In the event scenario of the validation set, the two classes are stark: the shortage price at buses 101 and 113 is $9,000/MWh and at bus 107 it is $28.07 — capacity there cannot reach the load.

The split is the same two classes under every convention, and the convention decides how large it looks. Priced over the whole year the gap is 1.49-fold; priced over the event hours alone it is 6.40-fold ($19,225 against $123,120), and as a fifty-megawatt block 5.97-fold. Spreading the value across the great majority of hours in which the constraint does not bind dilutes the very effect being priced by a factor of more than four, which is worth knowing before choosing the convention a payment rests on.

The arithmetic is exact. At nine validation points spanning normal, congested and shortfall scenarios, the dual is compared against a finite difference taken by actually adding half a megawatt and re-solving. The largest disagreement anywhere is 5.3×10^{-8} \$/MWh. The duals are the derivative, not an approximation to it.

And it is cheaper. The whole twenty-four-bus accreditation curve comes from one pass in 298 seconds. Reproducing it by re-sweeping the year once per bus takes 723 seconds, 2.43 times as long — and the re-sweep does not answer quite the same question, which is the subject of the next section.

29.3.1 Three prices for the same megawatt

A megawatt at a typical bus is worth, over the same year:

- \$333,145/MW-year as a *marginal* value — the first megawatt, the derivative;
- \$123,120/MW-year counting only the hours in which load was actually unserved;
- \$101,957/MW-year if fifty megawatts are added at once and the benefit divided among them.

None of these is wrong and they differ by a factor of 3.3 between the first and the last. The first megawatt is worth 3.27 times the average of the first fifty, because adding capacity relieves the constraint that made it valuable.

> **Principle 29.2.** *A derivative is not a block. Where value comes from relieving a binding constraint, the marginal price overstates what a real increment is worth by however much the increment relaxes the constraint — and the quantity a procurement actually buys is the block.*

29.4 What surprised

A good rule gets three quarters of the way. The expectation, going into a comparison between a full probabilistic network model and a heuristic, was a wide margin. Deficit-plus-location finds 71.5 percent of failed buses and never names one wrongly. The model takes that to 95.3 percent and adds a probability. That is a real improvement and it is a quarter, not an order of magnitude, and reporting it that way is the difference between a result and a sales pitch. The place it is unambiguously worth the model is the 329 quiet hours, where precision is 1.0000: a diagnosis that never cries wolf is worth more operationally than one that is occasionally more complete.

A good calibration score can hide a bad region. Brier 0.0268 and expected calibration error 0.0081 read as a well-calibrated posterior, and for 93.7 percent of its claims it is. In the sparse middle it claims 0.82 and delivers 0.54. The aggregate is dominated by twenty-one thousand pairs near zero, which are easy. Any calibration claim should be read together with how the claims are distributed, and this one is only defensible because the reliability table is published alongside it.

The same structure arrived twice, from different computations. Chapter 26 found that one branch carried the entire deliverability effect by censusing which branches bound during shortfall events. This chapter found that one bus has a different capacity value by reading duals the solver had already computed. Neither knew about the other. Two independent routes to the same fact is the kind of internal corroboration that a single-purpose calculation cannot produce, and it is the most concrete answer this part gives to the question of what carrying the whole model is for.

29.5 Reproduction

Six scripts in the study directory produce the six results files this chapter reads, one file each. From

```
docs/terra_graph_engine/case_studies/foundation_power_flow/
```

and with `S=annual_ra_rtsgmlc_via_pgm`:

```
python $S/run_inference.py        # outage_inference_results.json
python $S/baselines.py            # baseline_results.json
python $S/run_accreditation.py    # accreditation_results.json
python $S/run_mismatch.py         # mismatch_results.json
python $S/search_audit.py         # search_audit.json
python $S/scaling_run.py          # scaling_results.json
```

The inference is the long part at 1.47 hours; all three baselines together take 17.4 seconds. The chapter's numbers and figures are then reproduced from the six committed files alone:

```
cd docs/books/transmission && python scripts/t10_beyond_correctness.py
```

which prints every value quoted above — including the reliability table of Section 29.2.3 and the three prices of Section 29.3.1, both derived from the recorded values — and rewrites the five fragments in `figures/`. A representative line of its output:

```
  the two buckets holding 93.7 % of the pairs are calibrated:
    0.0-0.1: claimed 0.0111, happened 0.0149
    0.9-1.0: claimed 0.9999, happened 0.9967
```

Notes

The workload, the scenario contract and the linear program are those of Chapter 26, unchanged. Posterior inference over discrete hypotheses by evidence is Chapter 9, and the Laplace approximation it rests on Chapter 6. That the dual variable of a capacity constraint is the marginal value of relaxing it is the standard Karush–Kuhn–Tucker identity and is textbook; using it to build a locational capacity-accreditation curve is established practice in resource adequacy, and the contribution here is that the curve falls out of a run already performed rather than out of a study designed to produce it.

What is this chapter's own is the pairing of the two — a diagnosis and a price, from one validated model, neither requiring the year to be re-run — together with three measurements

reported here: that a strong heuristic baseline reaches 71.5 percent bus recall where the posterior reaches 95.3, which is the honest size of the gain; that the posterior's aggregate calibration is carried by the ninety-four percent of its claims that are near certainty or near zero while the sparse middle is over-confident by a quarter; and that the two-class accreditation structure read from the duals reproduces, independently, the binding-branch structure that Chapter 26 obtained by census.

This chapter closes the part. The six chapters that open it established what the method computes and what it costs; Chapter 26 established that the computation is right; Chapter 27 established where it stops being the right computation to do; Chapter 28 established what it is worth paying to measure. This one asks the question those leave open — whether carrying a whole model earns its keep — and answers it in the only way available: by showing what the same rows produce when they are asked something they were not built for.

Appendix A

Notation

This appendix collects the book's notation. Section A.1 states the conventions that hold throughout. Section A.2 lists the symbols by subject, each with its meaning and the place that introduces it. Section A.3 lists the letters that carry more than one meaning, and where each meaning applies. A symbol used in only one example, one proof or one closure is defined where it is used and is not listed.

A.1 Conventions

Type. Scalars are italic: z_k, σ, G. Vectors are bold lowercase, $\mathbf{z}$, $\mathbf{h}$, and matrices bold capitals, $\mathbf{J}$, $\mathbf{K}$; Greek vectors and matrices are bold as well, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Lambda}$. A vector is a column, $\mathbf{z}^\mathsf{T}$ is its transpose, and $(\mathbf{a};\mathbf{b})$ stacks two columns into one. $\mathbf{I}$ is the identity, $\mathbf{0}$ and $\mathbf{1}$ are the vectors of zeros and ones, and $\mathbf{e}_k$ is the kth unit vector. Calligraphic capitals are sets or objects built from sets: $\mathcal{N}$, $\mathcal{E}$, the manifold $\mathcal{M}$, the regime $\mathcal{R}_b$. $\mathrm{diag}(\mathbf{w})$ is the diagonal matrix with $\mathbf{w}$ on its diagonal, and tr, det, $\mathbb{E}$, Var and Cov are the trace, determinant, expectation, variance and covariance.

The unknowns and the rows. The unknowns of a model are stacked into one vector $\mathbf{z} \in \mathbb{R}^n$, flows first and potentials after: $\mathbf{z} = (\mathbf{q};\mathbf{V})$ for the export feeder of Example 1.1, and line flows before bus angles for the DC triangle of Example 1.2. When a later chapter frees an input of an example, the input is appended to the example's $\mathbf{z}$, so the order a reader has learned is never changed. Flows first is an order for listing, not for elimination; Chapter 5 chooses elimination orders on other grounds. The residual $\mathbf{F}(\mathbf{z}) \in \mathbb{R}^m$ lists its rows in the order of §1.5: balances, edge closures, storage relations, freestanding closures, and boundary-condition rows. In an example the balance rows come first and the closure rows after, and the text says which is which.

Inputs and solved variables. The inputs $\mathbf{u}$ are the quantities a modeler places priors on; the solved variables $\mathbf{x}$ are the ones the physics then determines, through the solve map $\mathbf{x} = X(\mathbf{u})$ (§4.1). When the physics is linear the whole state is a linear image of the inputs, $\mathbf{z} = \mathbf{S}\mathbf{u}$ (§3.4).

Widths and weights. Every factor has a width σ, in the units of its residual, and a weight $w = 1/\sigma^2$; $\boldsymbol{\Omega}$ is the diagonal matrix of the weights of all factors (§6.1). A residual divided by its width is *standardized* (§3.3.1).

Gaussians. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is its density at $\mathbf{z}$. The information form has precision $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and information vector $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ (equations (B.1) and (B.2)); a density written with $(\boldsymbol{\eta}, \boldsymbol{\Lambda})$ in place of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is marked as such where it appears. A covariance may be singular (Definition B.1).

Decorations. A prime marks the result of conditioning on readings: $\boldsymbol{\mu}'$ and $\boldsymbol{\Sigma}'$ are a posterior mean and covariance, and $\boldsymbol{\Sigma}''$ follows a second reading. A star marks a mode: $\mathbf{z}^\star$ is the most probable state (§6.2) and $\mathbf{z}_a^\star$ the mode of branch a. A bar marks either an imposed value, $\bar{z}_k$, or a sample mean, $\bar{G}_N$. In Chapter 16 a superscript counts time steps, $\mathbf{x}^n$, and the subscript $n+1 \mid n$ marks a prediction from the readings up to step n. Chapters 13 and 15 use the prime for other purposes, listed in Table A.7.

Drawings. In a factor graph, circles are variables and squares are factors; a dark square is a prior and an open square is a sensor (Figure 3.1).

Local symbols. The ring feeder of Example 1.3, the entries of Appendix D (§D.1) and the case studies of Part VI define symbols of their own where they use them. The case studies keep the book's symbols where the meaning is the same.

A.2 Symbols

Tables A.1 to A.6 give the symbols by subject, in the order the book introduces the subjects. The last column names the first place a symbol is defined; many are used throughout.

Table A.1: Graphs, sets and indices.

symbol	meaning	introduced
$\mathcal{N}$, $\mathcal{E}$, $\mathcal{D}$	nodes, edges, and conservation domains of a physical graph	§1.2
n, e, d	a node, an edge, a domain	equation (1.1)
$\mathcal{E}(n)$	the edges incident on node n	equation (1.1)
$A_{n,e}$, $\mathbf{A}$	incidence: $+1$ or -1 by the orientation of edge e at node n, 0 otherwise; one row per inside node	§1.2.3
$\mathbf{A}_b$	the incidence rows of the reservoirs	Example 1.1
reservoir, port	a node whose condition is imposed; a flow and potential pair at a boundary	§1.4, Table 1.3
S, ∂S	a set of inside nodes; the edges with one end in it	Proposition 1.1, equation (17.2)
N, E, S, Θ, C, R	counts of inside nodes, edge channels, storages, freed parameters, freestanding closures and boundary rows	Proposition 1.2
φ_f, $\mathbf{z}_f$	a factor and its scope, the variables it reads	§2.5
$\mathrm{pa}(i)$	the parents of variable i in a directed graph	equation (2.1)
$x_1 \perp x_3 \mid x_2$	x_1 and x_3 are independent given x_2	equation (5.1)
A, B, S	sets of variables; S separates A from B	Definition 5.1
$\mathcal{C}$	the set of cut edges	Proposition 5.3
width, treewidth	the largest scope an elimination order creates, less one; its minimum over orders	Definition 5.2

Table A.1, continued

symbol	meaning	introduced
R_A	the variables of region A	§5.7
r, I_r, S	a region, its interior variables, the port set	§8.4
n_r, k_r, w_r	the interior variables, ports and elimination width of region r	§18.1

Table A.2: The physical model.

symbol	meaning	introduced
$X_{n,d}$	the extensive state of domain d at node n	equation (1.1)
$P_{d,e}$	the flow of domain d along edge e; written Q for reactive power and F for DC active power in the examples	equation (1.1)
$G_{n,d}$, $L_{n,d}$	a source and a loss at a node	equation (1.1)
$r(\mathbf{z}_{\text{loc}}; \theta) = 0$, θ	a closure in residual form; its parameters	§1.3
$\mathbf{z} \in \mathbb{R}^n$	the unknowns, flows first	§1.5
$\bar{z}_k$	a value imposed at a reservoir	§1.5
$\mathbf{F}(\mathbf{z}) = \mathbf{0}$, m	the residual system; its number of rows	equation (1.7)
$\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{z}$	the Jacobian	§1.5
$\mathbf{q}$, $\mathbf{V}$, $\mathbf{s}$	the reactive flows, the bus voltages, and the injections at the inside buses	Example 1.1
b_{12}, b_{2s}, V_s	the line's susceptance; the substation tie's susceptance; the substation voltage	Example 1.1
$\mathbf{K}$	the diagonal matrix of susceptances or conductances	equation (1.13)
$\mathbf{F}$, $\boldsymbol{\theta}$, B	the line flows and bus angles of the DC triangle; a susceptance	Example 1.2
C	a storage capacity, in a storage row $E = C\,T$	§1.5, equation (16.3)
$\mathbf{u}$, $\mathbf{x}$	inputs; solved variables	§4.1
$X(\mathbf{u})$	the solve map from inputs to solved variables	equation (4.1)
$\mathcal{M}$	the constraint manifold, the states that satisfy the physics	equation (4.1)
$\mathbf{J}_x$, $\mathbf{J}_u$	the Jacobian's blocks for solved variables and for inputs	§4.1
$\mathbf{S}$	the linear map from inputs to unknowns, $\mathbf{z} = \mathbf{S}\mathbf{u}$	§3.4
$\boldsymbol{\lambda}$	the adjoint vector	§14.2
$\mathbf{F}(\mathbf{z}, \dot{\mathbf{z}}, t) = \mathbf{0}$	the differential-algebraic system	equation (16.2)

Table A.3: Probability and factors.

symbol	meaning	introduced
$p(\mathbf{z})$, $p(\mathbf{z}_1 \mid \mathbf{z}_2)$	a density; a conditional density	§3.1
$\mathbb{P}$, $\mathbb{E}$	probability; expectation	Chapter 3
Z	the normalizer of a density	§3.1
$\mathcal{N}(\mu, \sigma^2)$	the Gaussian with mean μ and variance σ^2	equation (3.1)
π_j, μ_j, σ_j	a prior factor on z_j; its mean and width	§3.2

Table A.3, continued

symbol	meaning	introduced
φ_k, σ_k	a physics factor; its width	equation (3.3)
y_i, z_{o_i}	a reading; the variable it observes	equation (3.5)
ν_i, σ_i	a sensor's noise; its accuracy, σ_y for a single reading	equation (3.5)
ℓ_i	the likelihood factor of reading i	equation (3.5)
a, q	the availability of a component, 0 or 1; its failure rate	equation (3.6)
$p(\mathbf{z} \mid \mathbf{y})$	the posterior	equation (3.7)
$\mathbf{y}$, $\mathbf{H}$, $\mathbf{R}$	the stacked readings, the sensor matrix, the noise covariance: $\mathbf{y} = \mathbf{H}\mathbf{z} + \boldsymbol{\nu}$	Proposition 3.2
$\mathbf{h}$	the sensor vector of one reading, $y = \mathbf{h}^\mathsf{T}\mathbf{z} + \nu$	Proposition 3.1
χ^2, $m_g - n$	the misfit; its degrees of freedom	Proposition 12.1
Z_a, C_a	the evidence of branch a; its objective	equations (7.1) and (7.2)
$\mathcal{R}_b$, $p_b(\mathbf{y})$	the region of input space where branch b holds; the branch's predictive density of the readings	Proposition 7.2
$\mathrm{KL}(q \,\|\, p)$	the Kullback–Leibler divergence	Proposition 9.2, equation (B.12)
L, S, k	the number of components that can fail; a failure set; its size	§15.1
$\mathbb{P}(S)$, q_i	the prior of a failure set; the failure probability of component i	§15.1, equation (15.1)

Table A.4: Linear-Gaussian inference.

symbol	meaning	introduced
$\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$	a mean and a covariance	Proposition 3.1
$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$	the precision; the information vector	Proposition 3.2
$\boldsymbol{\mu}'$, $\boldsymbol{\Sigma}'$, $\boldsymbol{\Lambda}'$, $\boldsymbol{\eta}'$	the same after conditioning on readings	equations (3.9) and (3.10)
e, s	the innovation $y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}$; its predicted variance. Indexed e_i, s_i by reading and e_n, s_n by time step	Proposition 3.1
$\mathbf{k}$, $\mathbf{s}$	the gain of one reading; the vector $\boldsymbol{\Sigma}\mathbf{h}$	equation (11.1)
$\mathbf{S}$, $\mathbf{K}$	the innovation covariance $\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^\mathsf{T} + \mathbf{R}$ of several readings; their gain	equation (B.6)
w, $\boldsymbol{\Omega}$	a factor's weight $1/\sigma^2$; the diagonal matrix of weights	§6.1
$\mathbf{g}(\mathbf{z})$, m_g, $\mathbf{J}_g$	every residual stacked, physics, prior and sensor; its length; its Jacobian	equation (6.3)
$C(\mathbf{z})$	the cost, $\frac{1}{2}\mathbf{g}^\mathsf{T}\boldsymbol{\Omega}\mathbf{g}$, the negative log posterior up to a constant	equation (6.4)
$\mathbf{z}^\star$	the most probable state	§6.2
$\boldsymbol{\delta}$	a Gauss–Newton step	equation (6.6)
$\nabla^2 C$	the Hessian of the cost	equation (6.7)
$\mathbf{L}$	the Cholesky factor, $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^\mathsf{T}$	§6.3
$\boldsymbol{\xi}$	a vector of independent standard normals	Proposition 6.4
$\mathbf{M}/\mathbf{D}$	the Schur complement of the block $\mathbf{D}$ in $\mathbf{M}$	Lemma B.4
$m_{x\to f}$, $m_{f\to x}$	sum-product messages	equations (8.1) and (8.2)
$\mathbf{S}_r$, $\mathbf{h}_r$	the precision and information of region r's message on the ports	equation (8.5)
$\mathbf{X}$	the sensitivity of a region's interior to its ports	equation (8.6)

Table A.4, continued

symbol	meaning	introduced
EIG	the expected information gain of a reading	equation (12.2)
$\mathbf{F}$, $\mathbf{Q}$, $\mathbf{w}$	the transition matrix, process-noise covariance and process noise of a linear dynamic model	Proposition 16.3

Table A.5: Time.

symbol	meaning	introduced
t, t_n, Δt	time; the nth time point; the step	§16.2
X^n, $\mathbf{z}^{n+1}$	a superscript counts time steps	equation (16.4)
$\boldsymbol{\mu}_{n+1\|n}$, $\boldsymbol{\Sigma}_{n+1\|n}$	the prediction of step $n+1$ from readings to step n	equation (16.6)
$\log Z$	the evidence of a record of readings in time	equation (16.8)

Table A.6: Consequences and attribution.

symbol	meaning	introduced
$G(\mathbf{u})$	a scalar consequence of the physics as a function of the inputs	equation (10.1), §14.1
$C(S)$	the consequence of failure set S	§15.1
S_i	the first-order Sobol index of input i	§14.3
$L(\mathbf{u})$, $\mathbf{s}_j$	a cutting-plane surrogate below G; a subgradient of G at census point j	equation (14.3), Proposition 14.2
PTDF, LODF	power-transfer and line-outage distribution factors	equation (15.2)

A.3 Letters with more than one meaning

A book that spans physics, probability and computation runs out of letters. Table A.7 lists the letters that carry different meanings in different places, with the scope of each meaning. Within a chapter a letter has one meaning unless the chapter says otherwise.

Table A.7: Letters with more than one meaning, and the scope of each.

letter	meanings and where they apply
A, $\mathbf{A}$	the incidence (Chapter 1 on); a generic matrix in lemmas and in Appendix B; sets or regions A, B (Chapters 5, 8, 13, 15); the area in hA
C	a storage capacity; a count in Proposition 1.2; the cost $C(\mathbf{z})$ of Chapter 6; the consequence $C(S)$ of Chapter 15; the census size in Chapter 14; the compression C_r of Chapter 18
e, $\mathbf{e}_k$	an edge (Chapter 1); the innovation (Chapters 3, 11, 16); the unit vector $\mathbf{e}_k$
F, $\mathbf{F}$	the residual $\mathbf{F}(\mathbf{z})$ throughout; the line flows F_e and $\mathbf{F}$ of DC networks; the transition matrix $\mathbf{F}$ of Proposition 16.3
G, $\mathbf{G}$	a source at a node, including the feeder's reactive injection G in every chapter that uses the feeder; the consequence $G(\mathbf{u})$ of Chapters 10 and 14 and the case studies; the branch susceptances $\mathbf{G}$ of Example 1.3; the blocks $\mathbf{G}_x$, $\mathbf{G}_u$ of Chapter 13

Table A.7, continued

letter	meanings and where they apply
$\mathbf{g}$	the stacked residual of every factor, Chapter 6 on
h, $\mathbf{h}$	the sensor vector $\mathbf{h}$ (Chapter 3 on); the convection coefficient in hA; the region information $\mathbf{h}_r$ (Chapter 8); enthalpy in Appendix D; a step size in Chapters 10 and 14
H, $\mathbf{H}$	the sensor matrix; the Fisher information in Chapter 12; the entropy $H[p]$ of equation (B.11)
J, $\mathbf{J}$	the Jacobian throughout; a slope jump in Chapters 10 and 14; the objective $J(A)$ of Chapter 23
K, k	the susceptance matrix $\mathbf{K}$ and a susceptance k; the gain $\mathbf{k}$ of one reading and $\mathbf{K}$ of several; the size k of a failure set (Chapter 15); the ports k_r of a region (Chapters 8 and 18); the number of inputs K in Chapters 21 and 22
L, $\mathbf{L}$	a loss at a node; the Cholesky factor $\mathbf{L}$; the number of components L (Chapter 15); the surrogate $L(\mathbf{u})$ (Chapter 14)
P, $\mathbf{P}$	a flow $P_{d,e}$; a bus injection; a projector
Q, q	a reactive flow; the process noise $\mathbf{Q}$ (Chapter 16); the shed load Q of Chapter 23; a failure rate q; the flows $\mathbf{q}$; a variational density q (Chapter 9); a nodal source q_n (Chapter 17)
R, r	the noise covariance $\mathbf{R}$; a count in Proposition 1.2; a residual r; a region r (Chapter 8 on)
S, $\mathbf{S}$, s	a set of nodes (Chapter 1), of variables (Chapter 5), of ports (Chapter 8) or of failed components (Chapter 15); the map $\mathbf{z} = \mathbf{S}\mathbf{u}$; the innovation covariance $\mathbf{S}$ and variance s; the region message $\mathbf{S}_r$; the Sobol index S_i (Chapter 14); the sources $\mathbf{s}$ of Chapter 1; the vector $\mathbf{s} = \mathbf{\Sigma h}$ (Chapter 11); a subgradient $\mathbf{s}_j$ (Chapter 14)
T, $^\mathsf{T}$	a temperature, where one is modeled; the transpose
w	a weight $1/\sigma^2$; the width of an elimination order and w_r of a region; process noise $\mathbf{w}$ (Chapter 16); a scenario weight w_s (Chapter 23)
X, x	the extensive state $X_{n,d}$; the solve map $X(\mathbf{u})$; the variables X_i and scopes X_f of a generic graphical model in Chapter 2; the interior sensitivity $\mathbf{X}$ (Chapter 8); the solved variables $\mathbf{x}$, and the state $\mathbf{x}^n$ of Chapter 16
Z	a normalizer; the evidence Z_a of a branch or hypothesis
θ	the parameters of a closure; the bus angles of a DC network
ν	a sensor's noise
φ	a factor; a potential φ_i at a port (Chapters 5 and 17)
λ, ρ, ε	each is defined where it is used: $\boldsymbol{\lambda}$ the adjoint (Chapter 14), an eigenvalue, a scalar precision; ρ a correlation, a relative width, a spectral radius; ε a noise, a discarded weight (Proposition 9.3)
prime	a posterior (Chapters 3 and 11, Appendix B); a replaced mechanism or the intervened world (Definition 13.1); the post-outage flow F'_e and the reduced susceptance $\mathbf{B}'$ (Proposition 15.2)

Appendix B

Gaussian identities

This appendix collects the facts about Gaussian distributions that the book uses, each with a short proof. Most chapters need only a few of them, and the table in §B.8 says which chapter uses which. Where a chapter already proves an identity in the form it needs, the appendix states the identity and points to that proof. Every identity is checked numerically, on randomly drawn matrices, by the script `appB_gaussian_identities.py` in the book's `scripts` directory.

Throughout, $\mathbf{z} \in \mathbb{R}^n$, a covariance $\boldsymbol{\Sigma}$ is symmetric positive semidefinite, and a precision $\boldsymbol{\Lambda}$ is symmetric positive definite. Bold capitals are matrices, $\mathbf{I}$ is the identity, and tr and det are the trace and the determinant.

B.1 The two forms of a Gaussian

A Gaussian with mean $\boldsymbol{\mu}$ and positive definite covariance $\boldsymbol{\Sigma}$ has the density

$$\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; (2\pi)^{-n/2} \, (\det \boldsymbol{\Sigma})^{-1/2} \exp\!\big(-\tfrac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{z}-\boldsymbol{\mu})\big). \tag{B.1}$$

This is the *moment form*. Expanding the quadratic with $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ gives the *information form* of Chapter 6,

$$\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \exp\!\big(-\tfrac{1}{2}\mathbf{z}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{z} - \kappa\big), \qquad \kappa = \tfrac{1}{2}\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} + \tfrac{n}{2}\log 2\pi - \tfrac{1}{2}\log\det\boldsymbol{\Lambda}. \tag{B.2}$$

The two carry the same information. The moment form makes marginals and predictions easy, and the information form makes products and conditioning easy, which is the division of labour §B.3 makes precise.

Lemma B.1 (The Gaussian integral)**.** *For $\boldsymbol{\Lambda}$ positive definite and any $\boldsymbol{\eta}$,*

$$\int_{\mathbb{R}^n} \exp\!\big(-\tfrac{1}{2}\mathbf{z}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{z}\big)\,\mathrm{d}\mathbf{z} \;=\; (2\pi)^{n/2}\,(\det\boldsymbol{\Lambda})^{-1/2} \exp\!\big(\tfrac{1}{2}\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}\big).$$

Proof. Complete the square: with $\mathbf{m} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$ the exponent is $-\tfrac{1}{2}(\mathbf{z}-\mathbf{m})^{\mathsf{T}}\boldsymbol{\Lambda}(\mathbf{z}-\mathbf{m}) + \tfrac{1}{2}\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$. Factor $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^{\mathsf{T}}$ by Cholesky and substitute $\mathbf{w} = \mathbf{L}^{\mathsf{T}}(\mathbf{z}-\mathbf{m})$, so that $\mathrm{d}\mathbf{z} = \mathrm{d}\mathbf{w}/\det\mathbf{L}$ and the quadratic becomes $\mathbf{w}^{\mathsf{T}}\mathbf{w}$. The integral of $\exp(-\tfrac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w})$ is a product of n one-dimensional integrals, each $\sqrt{2\pi}$, and $\det\mathbf{L} = (\det\boldsymbol{\Lambda})^{1/2}$. □

The lemma is what normalizes equations (B.1) and (B.2), and it has a corollary used on almost every page of Part III: *a density whose logarithm is $-\tfrac{1}{2}\mathbf{z}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{z}$ plus a constant is the*

Gaussian with precision $\mathbf{\Lambda}$ *and mean* $\mathbf{\Lambda}^{-1}\boldsymbol{\eta}$. To find a posterior, collect the quadratic and linear terms of its logarithm and read the two off.

A Gaussian need not have a density. When the physics is exact, as in the examples of Chapter 3, the unknown vector is a linear function of a few inputs, $\mathbf{z} = \mathbf{S}\mathbf{u}$, and its covariance $\mathbf{S}\mathbf{\Sigma}_u\mathbf{S}^\mathsf{T}$ is singular. The general definition goes through the moment generating function.

Definition B.1 (Gaussian vector)**.** A random vector $\mathbf{z}$ is Gaussian with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$, written $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, if $\mathbb{E}[\exp(\mathbf{t}^\mathsf{T}\mathbf{z})] = \exp(\mathbf{t}^\mathsf{T}\boldsymbol{\mu} + \frac{1}{2}\mathbf{t}^\mathsf{T}\mathbf{\Sigma}\mathbf{t})$ for every $\mathbf{t} \in \mathbb{R}^n$. It has the density (B.1) exactly when $\mathbf{\Sigma}$ is positive definite.

For a positive definite $\mathbf{\Sigma}$ the definition agrees with the density: multiplying equation (B.2) by $\exp(\mathbf{t}^\mathsf{T}\mathbf{z})$ and integrating by Lemma B.1, with $\boldsymbol{\eta}$ replaced by $\boldsymbol{\eta} + \mathbf{t}$, gives the stated function. A moment generating function that is finite everywhere determines its distribution, so the definition names each Gaussian exactly once.

Lemma B.2 (Affine maps)**.** *If* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ *and* $\mathbf{w} = \mathbf{A}\mathbf{z} + \mathbf{b}$ *for an* $m \times n$ *matrix* $\mathbf{A}$*, then* $\mathbf{w} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b},\ \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\mathsf{T})$*. In particular every block of a Gaussian vector is Gaussian, with the corresponding blocks of* $\boldsymbol{\mu}$ *and* $\mathbf{\Sigma}$*.*

Proof. $\mathbb{E}[\exp(\mathbf{s}^\mathsf{T}\mathbf{w})] = \exp(\mathbf{s}^\mathsf{T}\mathbf{b})\,\mathbb{E}[\exp((\mathbf{A}^\mathsf{T}\mathbf{s})^\mathsf{T}\mathbf{z})] = \exp\big(\mathbf{s}^\mathsf{T}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}) + \frac{1}{2}\mathbf{s}^\mathsf{T}\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\mathsf{T}\mathbf{s}\big)$. A block is the map $\mathbf{A} = (\mathbf{0}\ \ \mathbf{I})$. □

Lemma B.3 (Uncorrelated blocks are independent)**.** *If* $(\mathbf{a}, \mathbf{b})$ *is Gaussian and* $\operatorname{Cov}(\mathbf{a}, \mathbf{b}) = \mathbf{0}$*, then* $\mathbf{a}$ *and* $\mathbf{b}$ *are independent.*

Proof. With a zero cross-covariance the joint moment generating function is

$$\mathbb{E}[\exp(\mathbf{s}^\mathsf{T}\mathbf{a} + \mathbf{t}^\mathsf{T}\mathbf{b})] = \exp\big(\mathbf{s}^\mathsf{T}\boldsymbol{\mu}_a + \tfrac{1}{2}\mathbf{s}^\mathsf{T}\mathbf{\Sigma}_{aa}\mathbf{s}\big)\,\exp\big(\mathbf{t}^\mathsf{T}\boldsymbol{\mu}_b + \tfrac{1}{2}\mathbf{t}^\mathsf{T}\mathbf{\Sigma}_{bb}\mathbf{t}\big),$$

the product of the two marginal ones, which is the moment generating function of an independent pair. □

B.2 Block matrices

Partition a square matrix as $\mathbf{M} = \left(\begin{smallmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{smallmatrix}\right)$. When $\mathbf{D}$ is invertible, the *Schur complement* of $\mathbf{D}$ in $\mathbf{M}$ is $\mathbf{M}/\mathbf{D} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$, and when $\mathbf{A}$ is invertible, the Schur complement of $\mathbf{A}$ is $\mathbf{M}/\mathbf{A} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$.

Lemma B.4 (Block factorization)**.** *If* $\mathbf{D}$ *is invertible,*

$$\mathbf{M} = \begin{pmatrix}\mathbf{I} & \mathbf{B}\mathbf{D}^{-1}\\ \mathbf{0} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{M}/\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{pmatrix}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{D}^{-1}\mathbf{C} & \mathbf{I}\end{pmatrix},$$

so $\det\mathbf{M} = \det(\mathbf{M}/\mathbf{D})\,\det\mathbf{D}$*, and when* $\mathbf{M}/\mathbf{D}$ *is invertible,*

$$\mathbf{M}^{-1} = \begin{pmatrix}(\mathbf{M}/\mathbf{D})^{-1} & -(\mathbf{M}/\mathbf{D})^{-1}\mathbf{B}\mathbf{D}^{-1}\\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{M}/\mathbf{D})^{-1} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}(\mathbf{M}/\mathbf{D})^{-1}\mathbf{B}\mathbf{D}^{-1}\end{pmatrix}.$$

The same holds with the roles of the blocks exchanged: if $\mathbf{A}$ *is invertible,* $\det\mathbf{M} = \det\mathbf{A}\,\det(\mathbf{M}/\mathbf{A})$ *and the lower right block of* $\mathbf{M}^{-1}$ *is* $(\mathbf{M}/\mathbf{A})^{-1}$*.*

Proof. Multiplying the last two factors gives $\left(\begin{smallmatrix}\mathbf{M/D} & \mathbf{0}\\ \mathbf{C} & \mathbf{D}\end{smallmatrix}\right)$, and the first factor adds $\mathbf{BD}^{-1}$ times the second row to the first, restoring $\mathbf{A} = \mathbf{M/D} + \mathbf{BD}^{-1}\mathbf{C}$ and $\mathbf{B}$. The outer factors are triangular with unit diagonal, so their determinants are one and their inverses change the sign of the off-diagonal block. Inverting the product in reverse order gives $\mathbf{M}^{-1}$ as stated. The exchanged version is the same argument with the factorization pivoting on $\mathbf{A}$. □

Two identities follow from computing one block of $\mathbf{M}^{-1}$ in two ways. They are the algebra behind every rank-one update in the book.

Lemma B.5 (Woodbury and the determinant lemma)**.** *Let* $\mathbf{A}$ *$(n\times n)$ and* $\mathbf{W}$ *$(k\times k)$ be invertible, and* $\mathbf{U}$ *$(n\times k)$ and* $\mathbf{V}$ *$(k\times n)$ arbitrary. If* $\mathbf{W}^{-1}+\mathbf{VA}^{-1}\mathbf{U}$ *is invertible, then*

$$(\mathbf{A}+\mathbf{UWV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\,(\mathbf{W}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\,\mathbf{VA}^{-1}, \tag{B.3}$$

$$\det(\mathbf{A}+\mathbf{UWV}) = \det\mathbf{A}\,\det\mathbf{W}\,\det(\mathbf{W}^{-1}+\mathbf{VA}^{-1}\mathbf{U}). \tag{B.4}$$

For a rank-one change, $\mathbf{U}=\mathbf{h}$, $\mathbf{V}=\mathbf{h}^{\mathsf T}$ *and* $\mathbf{W}=w$, *these read*

$$(\mathbf{A}+w\,\mathbf{hh}^{\mathsf T})^{-1} = \mathbf{A}^{-1} - \frac{w\,\mathbf{A}^{-1}\mathbf{hh}^{\mathsf T}\mathbf{A}^{-1}}{1+w\,\mathbf{h}^{\mathsf T}\mathbf{A}^{-1}\mathbf{h}}, \qquad \det(\mathbf{A}+w\,\mathbf{hh}^{\mathsf T}) = \det\mathbf{A}\,(1+w\,\mathbf{h}^{\mathsf T}\mathbf{A}^{-1}\mathbf{h}),$$

the Sherman–Morrison formula and the matrix determinant lemma.

Proof. Apply Lemma B.4 to $\mathbf{M} = \left(\begin{smallmatrix}\mathbf{A} & \mathbf{U}\\ -\mathbf{V} & \mathbf{W}^{-1}\end{smallmatrix}\right)$. Pivoting on the lower right block, $\mathbf{M/W}^{-1} = \mathbf{A}+\mathbf{UWV}$, so the upper left block of $\mathbf{M}^{-1}$ is $(\mathbf{A}+\mathbf{UWV})^{-1}$ and $\det\mathbf{M} = \det(\mathbf{A}+\mathbf{UWV})\,\det\mathbf{W}^{-1}$. Pivoting on the upper left block, $\mathbf{M/A} = \mathbf{W}^{-1}+\mathbf{VA}^{-1}\mathbf{U}$, so $\det\mathbf{M} = \det\mathbf{A}\,\det(\mathbf{W}^{-1}+\mathbf{VA}^{-1}\mathbf{U})$, and the upper left block of $\mathbf{M}^{-1}$, computed from that factorization, is the right side of equation (B.3). Equating the two expressions for each quantity gives both identities. □

Lemma B.6 (Push-through)**.** *For* $\mathbf{\Lambda}$ *and* $\mathbf{R}$ *positive definite and any* $\mathbf{H}$,

$$(\mathbf{\Lambda}+\mathbf{H}^{\mathsf T}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathsf T}\mathbf{R}^{-1} = \mathbf{\Lambda}^{-1}\mathbf{H}^{\mathsf T}(\mathbf{R}+\mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^{\mathsf T})^{-1}.$$

Proof. $(\mathbf{\Lambda}+\mathbf{H}^{\mathsf T}\mathbf{R}^{-1}\mathbf{H})\,\mathbf{\Lambda}^{-1}\mathbf{H}^{\mathsf T} = \mathbf{H}^{\mathsf T}+\mathbf{H}^{\mathsf T}\mathbf{R}^{-1}\mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^{\mathsf T} = \mathbf{H}^{\mathsf T}\mathbf{R}^{-1}(\mathbf{R}+\mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^{\mathsf T})$. Multiply on the left by the inverse of the first factor and on the right by the inverse of the last. □

B.3 Marginals and conditionals

Partition $\mathbf{z}=(\mathbf{x},\mathbf{u})$, and partition $\boldsymbol{\mu}$, $\mathbf{\Sigma}$, $\boldsymbol{\eta}$ and $\mathbf{\Lambda}$ to match. Chapter 6 proved the information-form results: the marginal of $\mathbf{u}$ has the Schur complement $\mathbf{\Lambda}_{uu}-\mathbf{\Lambda}_{ux}\mathbf{\Lambda}_{xx}^{-1}\mathbf{\Lambda}_{xu}$ as its precision (Proposition 6.3), and conditioning on $\mathbf{x}$ keeps the block $\mathbf{\Lambda}_{uu}$ and shifts the information vector (Proposition 6.5). The moment form is the mirror image.

Proposition B.1 (Marginal and conditional, moment form)**.** *Let* $\mathbf{z}=(\mathbf{x},\mathbf{u})\sim\mathcal{N}(\boldsymbol{\mu},\mathbf{\Sigma})$ *with* $\mathbf{\Sigma}_{xx}$ *positive definite. The marginal of* $\mathbf{u}$ *is* $\mathcal{N}(\boldsymbol{\mu}_u,\mathbf{\Sigma}_{uu})$, *and the conditional of* $\mathbf{u}$ *given* $\mathbf{x}$ *is*

$$\mathbf{u}\mid\mathbf{x} \;\sim\; \mathcal{N}\big(\boldsymbol{\mu}_u+\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}(\mathbf{x}-\boldsymbol{\mu}_x),\;\; \mathbf{\Sigma}_{uu}-\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xu}\big). \tag{B.5}$$

Proof. The marginal is Lemma B.2. For the conditional, let $\mathbf{e}=\mathbf{u}-\boldsymbol{\mu}_u-\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}(\mathbf{x}-\boldsymbol{\mu}_x)$. The pair $(\mathbf{e},\mathbf{x})$ is an affine map of $\mathbf{z}$, hence Gaussian, and $\operatorname{Cov}(\mathbf{e},\mathbf{x}) = \mathbf{\Sigma}_{ux}-\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xx} = \mathbf{0}$, so by Lemma B.3 $\mathbf{e}$ is independent of $\mathbf{x}$. It has mean zero and covariance $\mathbf{\Sigma}_{uu}-2\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xu}+\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xx}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xu} = \mathbf{\Sigma}_{uu}-\mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xu}$. Given $\mathbf{x}$, therefore, $\mathbf{u}$ is a fixed vector plus the independent Gaussian $\mathbf{e}$, which is equation (B.5). □

Table B.1: Marginalizing and conditioning in the two forms. What is a selection of blocks in one form is a Schur complement in the other.

	moment form $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	information form $(\boldsymbol{\eta}, \boldsymbol{\Lambda})$
marginal of $\mathbf{u}$	$\boldsymbol{\mu}_u,\ \boldsymbol{\Sigma}_{uu}$	$\boldsymbol{\eta}_u - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\eta}_x,\ \boldsymbol{\Lambda}_{uu} - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\Lambda}_{xu}$
conditional on $\mathbf{x}$	$\boldsymbol{\mu}_u + \boldsymbol{\Sigma}_{ux}\boldsymbol{\Sigma}_{xx}^{-1}(\mathbf{x} - \boldsymbol{\mu}_x),\ \boldsymbol{\Sigma}_{uu} - \boldsymbol{\Sigma}_{ux}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xu}$	$\boldsymbol{\eta}_u - \boldsymbol{\Lambda}_{ux}\mathbf{x},\ \boldsymbol{\Lambda}_{uu}$

The proof never inverts $\boldsymbol{\Sigma}$ or $\boldsymbol{\Sigma}_{uu}$, so the proposition holds for any covariance whose conditioning block $\boldsymbol{\Sigma}_{xx}$ is positive definite. When $\boldsymbol{\Sigma}$ itself is positive definite the two forms agree through Lemma B.4. The upper left block of $\boldsymbol{\Sigma}^{-1}$, ordered as $(\mathbf{u}, \mathbf{x})$, is $\boldsymbol{\Lambda}_{uu} = (\boldsymbol{\Sigma}_{uu} - \boldsymbol{\Sigma}_{ux}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xu})^{-1}$, the inverse of the conditional covariance, as Proposition 6.5 requires. Symmetrically, $\boldsymbol{\Sigma}_{uu}$ is the inverse of the Schur complement of $\boldsymbol{\Lambda}_{xx}$, as Proposition 6.3 requires. Table B.1 sets the four results side by side.

B.4 Products and linear-Gaussian models

Proposition B.2 (Products of Gaussians)**.** *As functions of* $\mathbf{z}$, $\prod_i \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ *is proportional to the Gaussian with precision* $\boldsymbol{\Lambda} = \sum_i \boldsymbol{\Sigma}_i^{-1}$ *and information vector* $\boldsymbol{\eta} = \sum_i \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$*. For two factors the constant of proportionality is*

$$\int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\, \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\, \mathrm{d}\mathbf{z} \;=\; \mathcal{N}(\boldsymbol{\mu}_1;\ \boldsymbol{\mu}_2,\ \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2).$$

Proof. In information form the exponents add, and the corollary of Lemma B.1 reads off the product. For the constant, let $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ and $\mathbf{y} = \mathbf{z} + \boldsymbol{\nu}$ with an independent $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1)$. The density of $\mathbf{y}$ at $\boldsymbol{\mu}_1$ is $\int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\, \mathcal{N}(\boldsymbol{\mu}_1; \mathbf{z}, \boldsymbol{\Sigma}_1)\, \mathrm{d}\mathbf{z}$, which is the integral above, since $\mathcal{N}(\boldsymbol{\mu}_1; \mathbf{z}, \boldsymbol{\Sigma}_1) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$. By Lemma B.2 $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$. □

The product rule is the one sentence Chapter 6 builds on: every factor adds its precision and its information vector.

Proposition B.3 (The linear-Gaussian model)**.** *Let* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$*, with* $\boldsymbol{\Sigma}$ *positive semidefinite, and let* $\mathbf{y} = \mathbf{H}\mathbf{z} + \mathbf{c} + \boldsymbol{\nu}$ *with* $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ *independent of* $\mathbf{z}$ *and* $\mathbf{R}$ *positive definite. Write* $\mathbf{S} = \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}} + \mathbf{R}$ *and* $\mathbf{K} = \boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}\mathbf{S}^{-1}$*, the covariance of the innovation* $\mathbf{y} - \mathbf{H}\boldsymbol{\mu} - \mathbf{c}$ *and the gain. (This* $\mathbf{S}$ *is not the map* $\mathbf{z} = \mathbf{S}\mathbf{u}$ *of §B.1.) Then:*

1. *the pair* $(\mathbf{z}, \mathbf{y})$ *is Gaussian with cross-covariance* $\mathrm{Cov}(\mathbf{z}, \mathbf{y}) = \boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}$*;*
2. *the* predictive *distribution of the reading is* $\mathbf{y} \sim \mathcal{N}(\mathbf{H}\boldsymbol{\mu} + \mathbf{c},\ \mathbf{S})$*;*
3. *the posterior is*
$$\mathbf{z} \mid \mathbf{y} \;\sim\; \mathcal{N}(\boldsymbol{\mu} + \mathbf{K}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu} - \mathbf{c}),\quad \boldsymbol{\Sigma} - \mathbf{K}\mathbf{H}\boldsymbol{\Sigma}); \tag{B.6}$$
4. *if* $\boldsymbol{\Sigma}$ *is positive definite, the posterior has precision* $\boldsymbol{\Lambda} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ *and information vector* $\boldsymbol{\eta} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{c})$*, with* $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ *and* $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$*.*

Proof. The pair $(\mathbf{z}, \boldsymbol{\nu})$ is Gaussian with block diagonal covariance, and $(\mathbf{z}, \mathbf{y})$ is an affine map of it, so items 1 and 2 follow from Lemma B.2; the cross-covariance is $\mathbb{E}[(\mathbf{z} - \boldsymbol{\mu})(\mathbf{H}(\mathbf{z} - \boldsymbol{\mu}) + \boldsymbol{\nu})^{\mathsf{T}}] = \boldsymbol{\Sigma}\mathbf{H}^{\mathsf{T}}$. Item 3 is Proposition B.1 with $\mathbf{x} = \mathbf{y}$ and $\mathbf{u} = \mathbf{z}$, whose conditioning block is $\mathbf{S}$, positive definite because $\mathbf{R}$ is. For item 4, the posterior precision claimed inverts to the posterior covariance of item 3 by equation (B.3) with $\mathbf{A} = \boldsymbol{\Lambda}$, $\mathbf{U} = \mathbf{H}^{\mathsf{T}}$, $\mathbf{W} = \mathbf{R}^{-1}$ and $\mathbf{V} = \mathbf{H}$:

$(\mathbf{\Lambda} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H})^{-1} = \mathbf{\Sigma} - \mathbf{\Sigma}\mathbf{H}^\mathsf{T}\mathbf{S}^{-1}\mathbf{H}\mathbf{\Sigma} = \mathbf{\Sigma} - \mathbf{K}\mathbf{H}\mathbf{\Sigma}$. The mean is that covariance times the claimed information vector. Its first part is $(\mathbf{\Sigma} - \mathbf{K}\mathbf{H}\mathbf{\Sigma})\mathbf{\Lambda}\boldsymbol{\mu} = \boldsymbol{\mu} - \mathbf{K}\mathbf{H}\boldsymbol{\mu}$, and by Lemma B.6 its second part is $(\mathbf{\Lambda} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{c}) = \mathbf{K}(\mathbf{y} - \mathbf{c})$. Their sum is the mean of item 3. □

With one reading, $\mathbf{H} = \mathbf{h}^\mathsf{T}$ and $\mathbf{R} = \sigma^2$, item 3 is Proposition 3.1 and Proposition 11.1, and item 4 is Proposition 3.2. The proof here conditions the joint distribution rather than completing a square, so it never inverts $\mathbf{\Sigma}$: the one-reading formulas hold for a singular covariance such as the rank-two $\mathbf{\Sigma}_z$ of §3.4, exactly and not only as a limit.

B.5 Rank-one changes

A reading of $\mathbf{h}^\mathsf{T}\mathbf{z}$ with weight $w = 1/\sigma^2$ changes the precision by the rank-one term $w\,\mathbf{h}\mathbf{h}^\mathsf{T}$. With $\mathbf{s} = \mathbf{\Sigma}\mathbf{h}$, Lemma B.5 gives the new covariance and determinant at the cost of one solve:

$$\mathbf{\Sigma}' = \mathbf{\Sigma} - \frac{w\,\mathbf{s}\mathbf{s}^\mathsf{T}}{1 + w\,\mathbf{h}^\mathsf{T}\mathbf{s}}, \qquad \det \mathbf{\Lambda}' = \det \mathbf{\Lambda}\,(1 + w\,\mathbf{h}^\mathsf{T}\mathbf{s}). \tag{B.7}$$

A reading can also be removed, which is how a leave-one-out residual or a rejected meter is computed without refactoring. Removing it subtracts the same term, $\mathbf{\Lambda} = \mathbf{\Lambda}' - w\,\mathbf{h}\mathbf{h}^\mathsf{T}$, and with $\mathbf{s}' = \mathbf{\Sigma}'\mathbf{h}$ equation (B.3) with $\mathbf{W} = -w$ gives

$$\mathbf{\Sigma} = \mathbf{\Sigma}' + \frac{w\,\mathbf{s}'\mathbf{s}'^\mathsf{T}}{1 - w\,\mathbf{h}^\mathsf{T}\mathbf{s}'}, \tag{B.8}$$

valid exactly when $1 - w\,\mathbf{h}^\mathsf{T}\mathbf{s}' > 0$, which is the condition that the reduced precision stays positive definite. The determinant identity gives what a reading is worth before it is taken, measured in information rather than variance. With the entropy of equation (B.11) below, the entropy of $\mathbf{z}$ falls by

$$\tfrac{1}{2}\log\det\mathbf{\Sigma} - \tfrac{1}{2}\log\det\mathbf{\Sigma}' \;=\; \tfrac{1}{2}\log(1 + \mathbf{h}^\mathsf{T}\mathbf{\Sigma}\mathbf{h}/\sigma^2), \tag{B.9}$$

the mutual information between $\mathbf{z}$ and the reading. It depends on the reading only through the ratio of the variance it predicts, $\mathbf{h}^\mathsf{T}\mathbf{\Sigma}\mathbf{h}$, to its noise, and it is the information counterpart of the variance reduction of equation (11.3).

B.6 Integrals, quadratic forms and divergences

Proposition B.4 (The evidence of a quadratic objective)**.** *If* $C(\mathbf{z}) = C^\star + \frac{1}{2}(\mathbf{z} - \mathbf{z}^\star)^\mathsf{T}\mathbf{\Lambda}(\mathbf{z} - \mathbf{z}^\star)$ *with* $\mathbf{\Lambda}$ *positive definite, then* $\int \exp(-C(\mathbf{z}))\,\mathrm{d}\mathbf{z} = \exp(-C^\star)\,(2\pi)^{n/2}\,(\det\mathbf{\Lambda})^{-1/2}$.

Proof. Lemma B.1, with the square already completed. □

For a linear-Gaussian model the negative log posterior of equation (6.4) is exactly such a quadratic. The joint density of the unknowns and the readings is $\exp(-C)$ times the normalizing constants of the factors, so the evidence is $\exp(-C(\mathbf{z}^\star))$, times $(2\pi)^{n/2}(\det\mathbf{\Lambda})^{-1/2}$, times those constants. Its logarithm is equation (7.2) of Chapter 7, whose constant collects $\frac{n}{2}\log 2\pi$ and the logarithms of the factors' normalizers.

Proposition B.5 (Quadratic forms and the chi-squared law)**.** *Let* $\mathbf{z}$ *have mean* $\boldsymbol{\mu}$ *and covariance* $\boldsymbol{\Sigma}$*. For any symmetric* $\mathbf{A}$ *and any* $\mathbf{m}$*,*

$$\mathbb{E}\big[(\mathbf{z}-\mathbf{m})^{\mathsf{T}}\mathbf{A}(\mathbf{z}-\mathbf{m})\big] = \operatorname{tr}(\mathbf{A}\boldsymbol{\Sigma}) + (\boldsymbol{\mu}-\mathbf{m})^{\mathsf{T}}\mathbf{A}(\boldsymbol{\mu}-\mathbf{m}). \tag{B.10}$$

If $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ *and* $\mathbf{P}$ *is a symmetric idempotent matrix of rank* r*, then* $\boldsymbol{\epsilon}^{\mathsf{T}}\mathbf{P}\boldsymbol{\epsilon}$ *is chi-squared with* r *degrees of freedom, with mean* r *and variance* $2r$*.*

Proof. Write $\mathbf{z}-\mathbf{m} = (\mathbf{z}-\boldsymbol{\mu})+(\boldsymbol{\mu}-\mathbf{m})$. The cross term has zero mean, and $\mathbb{E}[(\mathbf{z}-\boldsymbol{\mu})^{\mathsf{T}}\mathbf{A}(\mathbf{z}-\boldsymbol{\mu})] = \operatorname{tr}(\mathbf{A}\,\mathbb{E}[(\mathbf{z}-\boldsymbol{\mu})(\mathbf{z}-\boldsymbol{\mu})^{\mathsf{T}}]) = \operatorname{tr}(\mathbf{A}\boldsymbol{\Sigma})$, since a scalar equals its trace and the trace is invariant under cyclic permutation. For the second part, a symmetric idempotent matrix has eigenvalues zero and one, so $\mathbf{P} = \mathbf{Q}\mathbf{D}\mathbf{Q}^{\mathsf{T}}$ with $\mathbf{Q}$ orthogonal and $\mathbf{D}$ diagonal with r ones. By Lemma B.2, $\boldsymbol{\delta} = \mathbf{Q}^{\mathsf{T}}\boldsymbol{\epsilon}$ is again standard Gaussian, and $\boldsymbol{\epsilon}^{\mathsf{T}}\mathbf{P}\boldsymbol{\epsilon} = \sum_{i\le r}\delta_i^2$, a sum of r independent squared standard normals. Each has mean one and, since $\mathbb{E}[\delta_i^4] = 3$, variance two. □

The second part is the last step of Proposition 4.2 and of Proposition 12.1. There the matrix is the projector $\mathbf{I}-\mathbf{P}$ onto the complement of the column space of the standardized Jacobian, of rank $m-n$.

The *entropy* of a density is $H[p] = -\mathbb{E}_p[\log p]$, and the *Kullback–Leibler divergence* of q from p is $\mathrm{KL}(q\,\|\,p) = \mathbb{E}_q[\log q - \log p]$, which is non-negative and zero only when $q = p$.

Proposition B.6 (Entropy and divergence of Gaussians)**.** *For* $\boldsymbol{\Sigma}$ *positive definite,*

$$H\big[\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\big] = \tfrac{1}{2}\log\det(2\pi e\,\boldsymbol{\Sigma}), \tag{B.11}$$

and for two Gaussians $\mathcal{N}_0 = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ *and* $\mathcal{N}_1 = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ *on* $\mathbb{R}^n$*,*

$$\mathrm{KL}(\mathcal{N}_0\,\|\,\mathcal{N}_1) = \tfrac{1}{2}\Big[\operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^{\mathsf{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) - n + \log\det\boldsymbol{\Sigma}_1 - \log\det\boldsymbol{\Sigma}_0\Big]. \tag{B.12}$$

Proof. From equation (B.1), $-\log\mathcal{N}_1(\mathbf{z}) = \frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\boldsymbol{\Sigma}_1 + \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_1)^{\mathsf{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{z}-\boldsymbol{\mu}_1)$. Its expectation under $\mathcal{N}_0$ is, by equation (B.10), $\frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\boldsymbol{\Sigma}_1 + \frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + \frac{1}{2}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathsf{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$. With $\mathcal{N}_1 = \mathcal{N}_0$ the trace is n, which gives the entropy, $\frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\boldsymbol{\Sigma}_0 + \frac{n}{2}$. The divergence is the second expectation minus the entropy of $\mathcal{N}_0$. □

Proposition 9.2 minimizes equation (B.12) with $\mathcal{N}_1$ the posterior, over $\mathcal{N}_0$ a product of independent Gaussians. The terms that depend on $\mathcal{N}_0$ are $\operatorname{tr}(\boldsymbol{\Lambda}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}-\boldsymbol{\mu}_0)^{\mathsf{T}}\boldsymbol{\Lambda}(\boldsymbol{\mu}-\boldsymbol{\mu}_0) - \log\det\boldsymbol{\Sigma}_0$, and with a diagonal $\boldsymbol{\Sigma}_0$ the trace keeps only the diagonal of $\boldsymbol{\Lambda}$.

B.7 The Laplace approximation

Let a posterior be $p(\mathbf{z}\mid\mathbf{y}) \propto \exp(-C(\mathbf{z}))$ with C twice differentiable, a minimum at $\mathbf{z}^\star$, and a positive definite Hessian $\nabla^2 C^\star = \nabla^2 C(\mathbf{z}^\star)$ there. Because the gradient vanishes at the minimum, Taylor's theorem gives $C(\mathbf{z}) = C(\mathbf{z}^\star) + \frac{1}{2}(\mathbf{z}-\mathbf{z}^\star)^{\mathsf{T}}\,\nabla^2 C^\star\,(\mathbf{z}-\mathbf{z}^\star) + O(\|\mathbf{z}-\mathbf{z}^\star\|^3)$. Dropping the cubic remainder and applying the corollary of Lemma B.1 and Proposition B.4 gives the *Laplace approximation* to the posterior and to the evidence,

$$p(\mathbf{z}\mid\mathbf{y}) \approx \mathcal{N}\big(\mathbf{z}^\star,\ (\nabla^2 C^\star)^{-1}\big), \qquad \log\int \exp(-C)\,\mathrm{d}\mathbf{z} \approx -C(\mathbf{z}^\star) + \tfrac{n}{2}\log 2\pi - \tfrac{1}{2}\log\det\nabla^2 C^\star. \tag{B.13}$$

Table B.2: The identities of this appendix and where the book uses them.

identity	used in
Gaussian integral, reading a Gaussian off its exponent (Lemma B.1)	every posterior of Chapters 3 to 9; evidence, equation (7.2)
affine maps, singular covariances (Definition B.1, Lemma B.2)	$\mathbf{z} = \mathbf{S}\mathbf{u}$ in §3.4; sampling, Proposition 6.4; misfit law
block factorization, Schur complement (Lemma B.4)	marginalization and region messages, Chapters 6 and 8
Woodbury, Sherman–Morrison, determinant lemma (Lemma B.5)	Propositions 3.1 and 11.1; sequential updates, Chapter 11
moment-form conditional (Proposition B.1)	conditioning a joint prediction on a reading; virtual sensors
product rule (Proposition B.2)	information form, Proposition 3.2; Exercise 6.1
linear-Gaussian model (Proposition B.3)	predictive checks and innovations, Chapters 11 and 12; one reading with a singular prior
rank-one update, downdate, information gain (§B.5)	the value of a reading, equation (11.3); removing a meter
quadratic forms, chi-squared (Proposition B.5)	the misfit law, Propositions 4.2 and 12.1
entropy and divergence (Proposition B.6)	mean field, Proposition 9.2
Laplace approximation, Gauss–Newton Hessian (§B.7)	equation (6.9); branch evidence, Chapter 7

Both are exact when C is quadratic. Otherwise their accuracy depends on how much of the posterior's mass lies where the cubic remainder is not small against the quadratic, which shrinks as the posterior concentrates [90].

The book's objective has the least-squares form $C = \frac{1}{2}\mathbf{g}^{\mathsf{T}}\boldsymbol{\Omega}\mathbf{g}$ of equation (6.4), with $\boldsymbol{\Omega} = \operatorname{diag}(w_k)$. Differentiating twice,

$$\nabla C = \mathbf{J}_g^{\mathsf{T}}\boldsymbol{\Omega}\mathbf{g}, \qquad \nabla^2 C = \mathbf{J}_g^{\mathsf{T}}\boldsymbol{\Omega}\mathbf{J}_g + \sum_k w_k\, g_k\, \nabla^2 g_k. \tag{B.14}$$

The first term is the Gauss–Newton matrix $\boldsymbol{\Lambda} = \mathbf{J}_g^{\mathsf{T}}\boldsymbol{\Omega}\mathbf{J}_g$ that the book's solver assembles and that equation (6.9) uses; the second is dropped. For row k the dropped term is smaller than the kept one by roughly the factor $|g_k|\,\|\nabla^2 g_k\|/\|\nabla g_k\|^2$: the row's residual at the mode times its curvature, over its squared slope. It vanishes for linear rows. It is small for rows that are nearly satisfied at the mode, such as physics rows held to a small width, and for readings that the posterior fits to within their noise, and §6.2.1 discusses when that fails. The Gauss–Newton matrix is also positive semidefinite by construction, which the full Hessian need not be away from the mode.

B.8 Where the book uses each identity

Table B.2 lists, for each result of this appendix, the places in the book that rely on it.

Appendix C

From the book to the engine

The constructions of this book are implemented in TERRA, a domain-independent engine for multi-conservation graphs, written by the author. Nothing in Chapters 1 to 18 depends on it: their scripts import no module of the engine and run on a numerical library alone. The case studies of Part VI were run with it. This appendix says what the engine is, shows a model written in it, maps each construction of the book to the module that implements it, and says what the case studies used and how to rerun them.

C.1 What TERRA is

TERRA is the Python package `terra_graph_engine`. It needs Python 3.11 or later, NumPy, SciPy and Jinja2. It has two layers.

The *engine* knows nothing of any domain. It holds the objects of Part I: nodes with roles, edges with one channel per conserved quantity, closures, boundary conditions, and the compiled residual system with its sparse Jacobian. It solves that system by Newton's method in steady state and by a BDF integrator in time. On the same rows it builds the probabilistic model of Parts II and III, together with counterfactual branching and hierarchical decomposition.

The *plugins* carry what is specific to one domain or one method. There are foundation plugins for substance physics and electrical quantities, and domain plugins for power flow, transmission grids, HVAC, batteries and data centers. There are also method plugins for reliability enumeration and for the census and its surrogate. A plugin supplies closures, factories that build standard networks, and algorithms that sit on top of the engine. Some plugin algorithms, such as the DC dispatch linear programs of the power-flow plugin, solve their own problems directly and never call the engine's Newton solver.

The objects a reader meets first are these.

- `Model` is the builder. Nodes, edges, closures and initial values are added to it, and `build()` compiles it.
- `CompiledModel` is the frozen result: a registry of unknowns, the residual rows of Chapter 2, and their Jacobian.
- `tge.solve` solves a compiled model in steady state, by Newton's method with a backtracking line search. `tge.simulate` integrates it in time.
- `InferenceModel` views a compiled model as the Gaussian factor graph of Chapter 3. Its physics rows are weighted by a physics width, `physics_sigma`, which is the soft-constraint width of Chapter 4. `observe` attaches a sensor, `prior` a belief, and `add_latent_fault` an

additive unknown source on a balance.

- `infer` returns a `Posterior`: the MAP state by Gauss–Newton weighted least squares, and the Laplace covariance from the information matrix, as in Chapter 6.

A prior is either a datum or a regularizer. A datum is a belief whose error is a draw at its stated spread, such as a forecast. A regularizer is a broad prior that only keeps an unknown identifiable. Both act the same way in the solve. The adequacy test of Chapter 12 counts only data, for the reason Chapter 25 measures.

C.2 A model in a dozen lines

The smallest model with physics and redundancy has one closure, $y = 2x$, three readings of x, and nothing observed about y except a broad regularizing prior.

```
import terra_graph_engine as tge
from terra_graph_engine import Model, NodeRole, PropertyClosure
from terra_graph_engine.inference import InferenceModel, infer, goodness_of_fit

m = Model()
m.add_node("n", role=NodeRole.RESERVOIR)
m.add_closure(PropertyClosure(out="y", inputs=["x"], f=lambda x: 2.0 * x,
                              analytic_jacobian=lambda x: {"x": 2.0}))
m.set_initial("x", 10.0)
m.set_initial("y", 20.0)
cm = m.build()

im = InferenceModel(cm, physics_sigma=1e-3)
for reading in (11.0, 9.0, 10.5):
    im.observe("x", reading, noise=1.0)
im.prior("y", mean=20.0, std=50.0, regularizer=True)
post = infer(im)
fit = goodness_of_fit(im, post)
```

The posterior gives $x = 10.167 \pm 0.577$, which is the mean of the three readings with spread $1/\sqrt{3}$. It gives $y = 20.333 \pm 1.154$, carried through the closure with twice the spread. The closure's row has put y on the same footing as x with no reading of y; this is the virtual sensor of Chapter 12. The misfit is 2.167 on 2 degrees of freedom, a p-value of 0.338. The count is one physics row plus three readings less two unknowns. The regularizing prior on y counts in neither the misfit nor the degrees of freedom.

C.3 Where each construction lives

Tables C.1 and C.2 map the chapters of Parts I to V to the modules that implement them. Engine modules are named relative to the package, and plugin modules by plugin. Where a chapter's construction has no module, its script carries it out directly.

Table C.1: Parts I to III: the constructions and the modules that implement them.

ch.	construction	modules
1	nodes and roles, edges and channels, boundary conditions, sources, losses, storage; conserved domains; closure patterns	`model/` (`node`, `edge`, `boundary`, `model`); `domains/`; `closure/` (carrier flow, gradient flow, property, piecewise)
2	incidence, the registry of unknowns, residual rows, the Jacobian, the Newton solve	`topology/graph`; `variables/registry`; `model/compiled_model`; `assembly/residual`, `assembly/jacobian`; `numerics/solver/newton`; `entry` (`solve`)
3	sensors, priors, latent faults on a balance	`inference/model` (`InferenceModel`); `inference/factors`; `model/compiled_model` (`with_latent_fault`)
4	physics as soft rows of a set width; bounds on a posterior	`inference/model` (`physics_sigma`); `inference/map_solve`; `inference/constrained`
5	separators and elimination on the information matrix	`inference/gaussian_bp`; `hierarchy/coordinator`
6	MAP by weighted least squares; the Laplace posterior	`inference/map_solve`; `inference/result` (`Posterior`)
7	hybrid states and branch mixtures; checks and corrections of the Laplace posterior	`closure/patterns/piecewise`, `numerics/regimes`; `inference/validity`, `inference/laplace_is`, `inference/particle`, `inference/escalation`; plugin `uncertainty` (`mixture`)
8	Schur complements over an interface; elimination orders	`inference/gaussian_bp`; `hierarchy/coordinator`
9	sampling corrections; iteration between regions	`inference/particle`, `inference/laplace_is`; `hierarchy/coupling`, `hierarchy/ports`
10	forward sampling through the physics; batched and swept solves	`inference/propagate`; `assembly/batch` (`solve_batch`); `inference/map_batch`; `numerics/sweep`
11	rank-one conditioning on new readings; particle filtering; the value of a reading	`inference/incremental`; `inference/particle`; `inference/online_identify`; `inference/experiment_design`

C.4 The case studies

Table C.3 lists, for each chapter of Part VI, what the study used and what one engine solve is in it. Three studies run the engine's own solver or inference, and three run plugin algorithms whose inner problem is a linear program or a linear network flow. The chapters count cost in engine solves in both cases, and each says which kind it counts.

Each study lives in its own directory under `docs/terra_graph_engine/case_studies/` in the repository that accompanies the book. It is run from the repository root with the package source on the Python path,

```
PYTHONPATH=src python \
  docs/terra_graph_engine/case_studies/<group>/<study>/run.py
```

and it writes a results file next to its script. The book's script for the chapter reads that file and runs no physics:

```
cd docs/book
python scripts/ch25_bad_data.py
```

Table C.2: Parts IV and V: the constructions and the modules that implement them.

ch.	construction	modules
12	the misfit law, studentized residuals, fault hypotheses by evidence, identifiability, virtual sensors	`inference/goodness_of_fit`; `inference/model_comparison`; `inference/identifiability`; `inference/queries` (`locate_faults`)
13	interventions on the model; stacked branches and their aggregation	`counterfactual/` (`perturbation`, `branch`, `reduce`)
14	sensitivities at a solution; the census of values and gradients, supporting planes, the certificate, the questions answered from it	`numerics/sensitivity`; `assembly/input_jacobian`; plugin `uncertainty` (`evaluator`, `pieces`, `queries`, `attribution`, `control_variate`, `sampling`)
15	component reliability, common cause, probability-ordered enumeration with certified bounds, interdiction	plugin `reliability` (`component_reliability`, `common_cause`, `enumeration`, `interdiction`, `risk`, `screening`, `rare_event`)
16	time integration and events; linearization, observers and model-predictive control	`entry` (`simulate`); `numerics/integrator/`; `assembly/dynamic`; `numerics/linearize`, `numerics/observer`, `numerics/mpc`; `controls/`; `inference/forecast`
17	subsystems with ports, interface coupling, Schur-complement coordination, composition	`hierarchy/` (`subsystem`, `ports`, `coupling`, `coordinator`, `compose`)
18	region messages, separator values and compression, region trees, certified boxing, boundary equivalents	plugin `reliability` (`local_global`, `compressibility`, `region_tree`, `boxing`, `environment_k`, `regional_contingency`, `thermal_adapter`); plugin `power_flow` (`regional_dc`, `partitioning`)

Table C.3: The case studies of Part VI: the engine and plugin pieces each used, and what one engine solve is.

ch.	engine and plugins	one engine solve
20	engine: `Model`, `PropertyClosure`, `solve`	a Newton solve of the seven-unknown model
21	plugin `power_flow` (`rating_evaluator`, `min_shed`); plugin `uncertainty`	a DC minimum-load-shed linear program, with its duals as the gradient
22	plugin `power_flow` (AC model on the engine); plugin `thermo` (duct heater); plugin `uncertainty`	a Newton solve and one factorization for the gradient
23	plugin `power_flow` (`min_shed`); plugin `reliability` (`interdiction`, `enumeration`, `common_cause`)	a DC minimum-load-shed linear program
24	plugin `reliability` (`local_global`, `compressibility`, `region_tree`, `environment_k`, `thermal_adapter`); plugin `power_flow` (`regional_dc`); plugin `hvac` (`hall_chain`)	a region-local linear network flow under a fixed dispatch rule
25	plugin `power_grid` (IEEE 14-bus factory, sensors, default priors); engine: `infer`, `goodness_of_fit`, `identifiability`, `incremental`, `model_comparison`	a MAP solve of the AC model with its latent loads

Each chapter's last section gives its exact commands, its run time, and the tests that pin its committed results. The engine and its plugins carry their own test suite, under `tests/terra_graph_engine` and `tests/plugins`.

Appendix D

A catalogue of closures

Table 1.2 named four closure patterns. This appendix fills them in. It gathers the closures of the domains this book touches, electrical and storage first and then thermal, fluid, moist air and plant equipment, and gives each one in the same short form. Some are implemented in the plugins of TERRA, and the entry names the class or function; the forms given are the ones the code implements. Others are standard relations the plugins do not yet carry, and the entry cites a textbook. Either way the entry says what a modeler needs before writing the relation into a graph.

D.1 How to read an entry

Each entry gives the relation in residual form, zero when the relation holds, and then eight short fields.

- *Pattern*: carrier, gradient, property or piecewise, as in Table 1.2, or a custom closure when none fits.
- *Unknowns*: the variables the closure reads. Any of them can be an unknown; which ones are is a property of the question (Principle 1.2).
- *Parameters*: constants fixed when the closure is built, with units.
- *Jacobian*: the partial derivatives of the residual, analytic or by finite differences.
- *Kinks*: where the relation, or its derivative, is not smooth. A kink is where Newton's method, the Laplace approximation and the certificates of Chapter 14 need care (§14.5).
- *Shape*: whether the relation is monotone, linear, convex or none of these, in the variables that matter. Convexity is what the supporting planes of Proposition 14.2 need, and monotonicity is what the floors of Proposition 14.4 need.
- *Prior*: the parameter a study usually leaves uncertain, and why. This is a modeling judgment, not a property of the code.
- *Source*: the plugin and class that implements it, or the reference for a textbook relation.

Three kinds of physics look like closures and are not. A *storage relation* is the accumulation term of a balance, $\mathrm{d}X/\mathrm{d}t$, and belongs to the node (Chapter 16). A *pin* fixes a variable at a setpoint and is a one-variable row. And an *outer loop* changes the model between solves, as a generator that hits its reactive limit does. The entries say when a relation is one of these.

Table D.1 indexes the entries.

Table D.1: The closures of this appendix: pattern, whether the relation kinks, its shape, and where it comes from (P: a plugin of TERRA; T: a textbook relation).

closure	pattern	kinks	shape	src
Electrical				
DC branch	gradient	no	linear	P
AC branch in polar form	custom	no	trigonometric	P
transformer with a tap	custom	no	trigonometric	P
slack and voltage-controlled buses	pin	—	—	P
reactive limits of a generator	outer loop	yes	—	P
transformer loss	property	no	convex	P
UPS loss	property	no	convex	P
PDU loss	property	no	linear	P
fixed share of a flow	custom	no	linear	P
generator fuel curve	property	gated	affine	P
Storage and electrochemistry				
battery energy with efficiencies	storage	no	linear	P
signed power port	property	yes	piecewise linear	P
equivalent-circuit battery	property	no	monotone in charge	T
state of health	storage	no	linear	P
Thermal				
conductance, $Q = UA\,\Delta T$	gradient	no	linear	P
series conduction	gradient	no	linear	P
convection with a correlation	gradient	yes	monotone in ΔT	T
radiation between two surfaces	custom	no	monotone, not convex	P
counterflow exchanger, fixed rates	custom	no	linear in T	P
counterflow exchanger, variable rates	custom	yes	not evident	P
cooling tower	gradient	yes	non-decreasing	P
waterside economizer	property	yes	non-decreasing	P
enthalpy carried by mass; throttle	carrier; custom	no	bilinear; linear	P
temperature-dependent load	property	no	linear	P
Fluid				
quadratic resistance; damper	gradient	yes	increasing, odd	P
pipe friction (Darcy–Weisbach)	gradient	yes	increasing	T
laminar pipe (Hagen–Poiseuille)	gradient	no	linear	T
fan or pump curve	gradient	yes	decreasing in flow	P
fan and pump power	property	no	convex (cubic)	P
valve with a loss coefficient	gradient	yes	increasing in flow	T
check valve	piecewise	yes	monotone	T
isothermal gas pipe	gradient	yes	increasing	T
ideal-gas mass flow	property	no	linear	P
mixing of two streams	property	no	bilinear	P
Moist air				
humidity ratio and moist-air enthalpy	property	no	monotone	P
coil condensation	property	yes	non-decreasing	P
moisture carried by mass	carrier	no	bilinear	P
Plant equipment				
chiller power from load	property	yes	increasing	P
chiller staging	piecewise	yes	—	P
thermal storage mode	piecewise	yes	—	P

D.2 The four patterns in the engine

Carrier flow. $r = P - q\,\varepsilon$, the flow of one quantity carried by the flow q of another at intensity ε. *Jacobian*: analytic, $\{P : 1,\ q : -\varepsilon,\ \varepsilon : -q\}$. *Shape*: bilinear, so not convex in q and ε together; a census over both needs the conditions of §14.7. *Source*: engine, `CarrierFlow`.

Gradient flow. $r = P - g(\phi_i, \phi_j; \theta)$, with g any function of the two potentials. *Jacobian*: analytic if the modeler supplies it, otherwise a central difference with step $10^{-6}\max(1, |\phi|)$. *Kinks*: not detected; whatever g has. *Source*: engine, `GradientFlow`.

Property. $r = y - f(x_1, \ldots, x_k)$, a relation among variables at one place. *Jacobian*: analytic if supplied, otherwise a central difference. When the Jacobian is differenced the engine probes for a kink at every evaluation: it compares the second difference with the first, and warns when their ratio exceeds 0.1. *Source*: engine, `PropertyClosure`.

Piecewise. $r = y - f_b(x)$ on the active branch b. Each branch may carry a guard, a margin that is non-negative where the branch is admissible. The engine solves with the branches frozen, then asks each closure which branch its guards now prefer. The current branch stays while it is admissible, and otherwise the first admissible branch in construction order is taken, so overlapping guards give hysteresis. The loop repeats at most eight times, and a closure that flips more than three times is relaxed and reported as a failure to converge. This is the regime loop of §7.3; the probabilistic model instead weighs the branches (Proposition 7.2). *Source*: engine, `PiecewiseClosure`, and `numerics/regimes`.

D.3 Electrical

DC branch.

$$r = F - b\,(\theta_i - \theta_j), \qquad b = 1/X.$$

Pattern: gradient. *Parameters*: the susceptance b [p.u.]. *Jacobian*: analytic, $\mp b$. *Shape*: linear. *Prior*: b for a line whose impedance is in doubt; more often none, since the DC model's error is structural. *Source*: `electrical`, `dc_branch`; the approximation is [86].

AC branch in polar form. With $\delta = \theta_i - \theta_j$, series admittance $G + jB = 1/(R + jX)$, half line charging B_{sh} and tap t, the four end flows are

$$\begin{aligned}
P_{\text{from}} &= V_i^2 G/t^2 - V_i V_j\,(G\cos\delta + B\sin\delta)/t,\\
Q_{\text{from}} &= -V_i^2(B + B_{sh})/t^2 - V_i V_j\,(G\sin\delta - B\cos\delta)/t,\\
P_{\text{to}} &= V_j^2 G - V_i V_j\,(G\cos\delta - B\sin\delta)/t,\\
Q_{\text{to}} &= -V_j^2(B + B_{sh}) + V_i V_j\,(G\sin\delta + B\cos\delta)/t,
\end{aligned}$$

and each closure is $r = F - (1 + f)\,F(V, \theta)$ for its flow. *Pattern*: custom. *Unknowns*: the flow, both voltages and angles, and an optional latent admittance shift f. *Jacobian*: analytic; every partial is scaled by $1 + f$, and $\partial r/\partial f = -F$. *Shape*: trigonometric, neither monotone nor convex. *Prior*: f, a latent fault of the kind of Chapter 3, zero-mean with a small width. *Source*: `electrical`, `ACBranchPolar`; the equations are in [98]. Each line meets its two buses through a midpoint node that absorbs the loss $P_{\text{from}} + P_{\text{to}}$.

Transformer with a tap. The AC branch with $t \neq 1$. It carries no latent fault in the current plugin. *Source*: `electrical`, `transformer_polar`; `power_flow`, `attach_ac_line` with a tap.

Slack and voltage-controlled buses. These are pins, not closures: $r = \theta - 0$ and $r = V - V_{\text{set}}$ at the slack, $r = V - V_{\text{set}}$ at a voltage-controlled bus. The generator's injections are free flows from a reservoir, closed by the bus balances alone. A pinned variable stays in the list of unknowns, so a study can observe it or put a prior on it. *Source*: `electrical`, `bus_residuals`; `power_flow`, `attach_bus`.

Reactive limits of a generator. An outer loop, not a closure. After a solve, a voltage-controlled bus whose generator exceeds its reactive band is rewritten as a load bus with the reactive output held at the violated limit, and the model is solved once more. There is one such pass, and no return to voltage control. *Kinks*: the switch is a kink in every quantity downstream. *Source*: `power_flow`, `solve_with_q_limits`.

Transformer loss.

$$r = L - L_0 - L_{cu}\,\frac{(\sum g)^2}{S^2}.$$

Pattern: property. *Parameters*: no-load loss L_0, copper loss at rating L_{cu}, rating S [kW]. *Shape*: convex in the load $\sum g$. *Prior*: L_{cu}, a calibrated coefficient. *Source*: `facility`, `TransformerLoss`.

UPS loss.

$$r = L - \ell_f \sum g - \ell_q\,\frac{(\sum g)^2}{S} - L_0.$$

Pattern: property. *Shape*: convex. *Prior*: the coefficients ℓ_f, ℓ_q, calibrated to an efficiency curve. *Source*: `facility`, `UpsLoss`.

PDU loss.

$$r = L - L_0 - p \sum g.$$

Pattern: property. *Shape*: linear. *Source*: `facility`, `PduLoss`.

Fixed share of a flow.

$$r = s_{\text{ref}}\,F - s_F\,F_{\text{ref}}.$$

Pattern: custom. *Shape*: linear. Written without dividing by s_{ref}, so the residual is well defined for any share. *Source*: `facility`, `SplitShare`.

Generator fuel curve.

$$r = \dot{V}_{\text{fuel}} - (a + b\,\lambda)\,g,$$

with λ the load fraction and $g \in \{0, 1\}$ the run state. *Pattern*: property. *Shape*: affine and non-decreasing in λ. *Kinks*: the start and stop are held outside the solver: g is set per segment by a switching driver, so each solve sees a smooth relation. *Source*: `facility`, `GensetFuelCurve`.

D.4 Storage and electrochemistry

Battery energy with efficiencies.

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \eta_c P_c - P_d/\eta_d, \qquad L = (1-\eta_c)P_c + (1/\eta_d - 1)P_d, \qquad \mathrm{SoC} = E/E_{\mathrm{cap}}.$$

Pattern: a storage node for the energy, with property closures for the loss and the state of charge. *Parameters*: the efficiencies η_c, η_d, and the capacity. *Shape*: linear. A steady solve must pin the energy or the state of charge, since a storage node in steady state has no row of its own (§16.1). *Prior*: the efficiencies. *Source*: `battery_storage`, `attach_battery_pack`. The plugin's pack is this bookkeeping model; it has no voltage curve and no internal resistance.

Signed power port.

$$P_c = \max(-s, 0), \qquad P_d = \max(s, 0).$$

Pattern: property. *Kinks*: at $s = 0$. The kink is harmless only because the setpoint s is pinned and never a Newton unknown; if a study makes s uncertain, this closure is where the Laplace approximation fails. *Source*: `battery_storage`, power port.

Equivalent-circuit battery.

$$r = V - \mathrm{OCV}(\mathrm{SoC}) + R_0\, I,$$

with further resistor–capacitor pairs for the slow voltage response. *Pattern*: property. *Unknowns*: terminal voltage, current, state of charge. *Parameters*: the open-circuit voltage curve, the series resistance R_0. *Shape*: OCV increasing in the state of charge, flat in the middle of the range, which makes the state of charge weakly identifiable from voltage there. *Prior*: R_0 and the capacity, both of which age. *Source*: textbook [73]; not in the plugin.

State of health.

$$\frac{\mathrm{d}D}{\mathrm{d}t} = \dot{a}, \qquad \mathrm{SoH} = \mathrm{SoH}_0 - D, \qquad C_{\mathrm{eff}} = C_{\mathrm{nom}}\,\mathrm{SoH},$$

with an aging rate $\dot{a}$ from throughput or from an Arrhenius calendar law,

$$\dot{a} = \dot{a}_{\mathrm{ref}}\, \exp\!\left[\frac{E_a}{R}\left(\frac{1}{T_{\mathrm{ref}}} - \frac{1}{T}\right)\right].$$

Pattern: a storage node for the accumulated damage D, with property closures. *Shape*: linear in D; the capacity relation is bilinear. *Prior*: the reference rate $\dot{a}_{\mathrm{ref}}$, the least known number in battery aging. *Source*: `battery_storage`, `attach_soh_evolution`.

D.5 Thermal

Conductance.

$$r = Q - UA\,(T_h - T_c).$$

Pattern: gradient. *Unknowns*: Q, T_h, T_c. *Parameters*: UA [W/K]. *Jacobian*: analytic, constant. *Shape*: linear. *Prior*: UA, which is seldom known well at the start and drifts as surfaces foul. *Source*: `thermo`, `sensible_heat`; it is equation (1.5).

Series conduction.

$$r = Q - \frac{T_h - T_c}{\sum_i R_i}.$$

Pattern: gradient. *Parameters*: layer resistances R_i [K/W]. *Jacobian*: analytic, constant, $\mp 1/\sum_i R_i$ on the temperatures. *Shape*: linear. *Prior*: the least known layer, often a contact resistance; the sum is identifiable, the layers separately are not (§12.7). *Source*: `thermo`, `SeriesConduction`.

Convection with a correlation.

$$r = Q - hA\,(T_s - T_\infty), \qquad h = \frac{k}{L}\,C\,\mathrm{Re}^m\,\mathrm{Pr}^n.$$

Pattern: gradient. *Unknowns*: Q, T_s, T_∞, and the velocity inside Re when it is solved for. *Parameters*: A, the length L, the fluid properties, and the correlation constants C, m, n. *Kinks*: where the correlation changes, at the laminar-to-turbulent transition, C, m and n jump. *Shape*: linear in the temperature difference at fixed velocity, increasing and concave in velocity. *Prior*: C, since empirical correlations carry errors that can reach about twenty-five percent [6]. *Source*: textbook [6].

Radiation between two surfaces.

$$r_i = q_i - k\,(T_i^4 - T_j^4) - G_i\,(T_i - T_{\mathrm{amb}}), \qquad r_j = q_j + k\,(T_i^4 - T_j^4) - G_j\,(T_j - T_{\mathrm{amb}}).$$

Pattern: custom, two residuals from one closure. *Unknowns*: q_i, q_j, T_i, T_j. *Parameters*: $k = \sigma\,\epsilon\,A$ with view factor [W/K^4], and ambient conductances G_i, G_j [W/K]. *Jacobian*: analytic, $4kT^3$ on each temperature. *Shape*: increasing in the hotter temperature, not convex over both. *Prior*: the emissivity inside k, which depends on the surface's condition. *Source*: `thermo`, `RadiativeCouple2Node`; the law is in [6].

Counterflow exchanger, fixed capacity rates.

$$r = Q - \varepsilon\,C_{\min}\,(T_{h,\mathrm{in}} - T_{c,\mathrm{in}}), \quad \varepsilon = \frac{1-e}{1-C_r e},\ e = \exp[-\mathrm{NTU}(1-C_r)],$$

with $\mathrm{NTU} = UA/C_{\min}$, $C_r = C_{\min}/C_{\max}$, and $\varepsilon = \mathrm{NTU}/(1+\mathrm{NTU})$ when $C_r = 1$. *Pattern*: custom. *Parameters*: UA, C_h, C_c [W/K], so ε is fixed when the closure is built. *Jacobian*: analytic, constant, $\mp\varepsilon C_{\min}$. *Shape*: linear in the inlet temperatures; the product $\varepsilon C_{\min}$ increases with UA. *Prior*: UA, for fouling. *Source*: `thermo`, `HeatExchangerEpsNTU`; the effectiveness relation is in [43].

Counterflow exchanger, variable capacity rates. The same residual with C_h and C_c as unknowns, so the exchanger responds to its flows. *Jacobian*: analytic in the temperatures; the capacity entries are differenced inside the closure. *Kinks*: at $C_h = C_c$, where $C_{\min}$ and $C_{\max}$ swap and the effectiveness formula changes. The closure refuses a capacity rate at or below a floor. *Shape*: not evident from the relation. *Source*: `thermo`, `HeatExchangerEpsNTUVariable`.

Cooling tower.

$$r = Q - \varepsilon_{\text{eff}}(a)\, C_{cw}\, \max(0,\, T_{cw} - T_{wb}), \qquad \varepsilon_{\text{eff}}(a) = \varepsilon_d\, \min(1, \max(0, a)),$$

with a the air-flow ratio when it is modeled, and $\varepsilon_{\text{eff}} = \varepsilon_d$ when it is not. *Pattern*: gradient. *Unknowns*: Q, the water temperature to the tower T_{cw}, the wet-bulb temperature T_{wb}, and a. *Parameters*: C_{cw} [W/K], design effectiveness $\varepsilon_d \in [0, 1]$. *Kinks*: at $T_{cw} = T_{wb}$, $a = 0$ and $a = 1$. *Shape*: non-decreasing in both the approach and the air flow. *Prior*: ε_d, which is a fit to a manufacturer's curve. *Source*: `thermo`, `CoolingTowerEffectiveness`.

Waterside economizer.

$$r = Q - k\, \max(0,\, T_{chw,r} - T_{cw,s}), \qquad k = \varepsilon\, C_{\min}.$$

Pattern: property. *Kinks*: at the crossover $T_{chw,r} = T_{cw,s}$, below which the economizer does nothing and the Jacobian is zero. *Shape*: non-decreasing and convex in the temperature difference. *Source*: `datacenter`, condenser loop.

Enthalpy carried by mass; throttle.

$$r = P - \dot{m}\, h, \qquad r = h_{\text{out}} - h_{\text{in}}.$$

Pattern: carrier; custom. *Shape*: bilinear; linear. The throttle is exact: an adiabatic valve conserves enthalpy. *Source*: `thermo`, `enthalpy_advection` and `Throttle`; the duct heater, `make_duct_heater`, is two carrier edges around a node with a source and a loss.

Temperature-dependent load.

$$r = P - P_{\text{base}} - \alpha\, (T - T_{\text{ref}}).$$

Pattern: property. *Parameters*: α [W/K], T_{ref} [K]; P_{base} may be a schedule. *Shape*: linear. *Prior*: α, a calibrated fan-and-leakage slope. *Source*: `datacenter`, rack load.

D.6 Fluid

Quadratic resistance; damper.

$$r = (p_i - p_j) - k\, \dot{m}\, |\dot{m}| - \epsilon\, \dot{m}, \qquad k(\theta) = k_{\text{open}} / \max(\theta, \theta_{\min})^2.$$

Pattern: gradient. *Unknowns*: $\dot{m}$, p_i, p_j, and the damper position θ. *Parameters*: k [Pa/(kg/s)^2]; an optional linear term ϵ. *Jacobian*: analytic, $-2k|\dot{m}| - \epsilon$ in $\dot{m}$, which is zero at no flow unless $\epsilon > 0$; that is what the linear term is for. *Kinks*: the second derivative jumps at $\dot{m} = 0$; the damper kinks at $\theta_{\min}$. *Shape*: increasing and odd in $\dot{m}$. *Prior*: k, which is set by fittings nobody measured. *Source*: `thermo`, `FlowResistance` and `DamperResistance`.

Pipe friction.

$$r = (p_i - p_j) - f(\text{Re},\, e/D)\, \frac{L}{D}\, \frac{\rho v^2}{2}, \qquad f = 64/\text{Re (laminar)},\ \text{Colebrook (turbulent)}.$$

Pattern: gradient. *Parameters*: length L, diameter D, roughness e, density ρ, viscosity. *Kinks*: at the transition near $\text{Re} \approx 2300$ the friction factor jumps; a smooth blend is the usual remedy. *Shape*: increasing in flow; between linear (laminar) and quadratic (fully rough). *Prior*: the roughness e, which grows with age. *Source*: textbook [97].

Laminar pipe.

$$r = \dot{V} - \frac{\pi D^4}{128\,\mu L}\,(p_i - p_j).$$

Pattern: gradient. *Shape*: linear, the fluid analogue of a conductance. *Prior*: the viscosity μ, through its temperature dependence. *Source*: textbook [97].

Fan or pump curve.

$$r = (p_i - p_j) - \left(h_0\, s^2 - h_2\, \dot{m}\,|\dot{m}|\right),$$

with p_i and p_j the pressures at the edge's two ends in the plugin's orientation, so that the curve's head is the pressure difference along the edge. *Pattern*: gradient. *Unknowns*: $\dot{m}$, the two pressures, and the speed ratio s. *Parameters*: shutoff head h_0 [Pa], curvature h_2 [Pa/(kg/s)^2]. *Jacobian*: analytic, $+2h_2|\dot{m}|$ in $\dot{m}$ and $-2h_0 s$ in s. *Kinks*: C^1 at no flow. *Shape*: head decreasing in flow and increasing in speed squared, the affinity laws at fixed curve shape. *Prior*: h_0 and h_2, which wear moves. *Source*: `thermo`, `FanCurve`.

Fan and pump power.

$$r = P - P_{\text{rated}}\,(\dot{m}/\dot{m}_{\text{rated}})^3 \quad \text{or} \quad r = P - P_{\text{nom}}\,u^3.$$

Pattern: property. *Shape*: convex and increasing, the cube law of the affinity relations. *Prior*: the rated power, since the cube law holds only near the design point. *Source*: `thermo`, `pump_work`; `hvac`, fan power.

Valve with a loss coefficient.

$$r = (p_i - p_j) - K(\theta)\,\frac{\rho v^2}{2}.$$

Pattern: gradient. *Unknowns*: the flow, the pressures, the stem position θ. *Parameters*: the characteristic $K(\theta)$, linear or equal-percentage in the opening. *Kinks*: at full closure, where $K \to \infty$; a floor on the opening keeps the residual finite. *Shape*: increasing in flow, decreasing in opening. *Source*: textbook [97]; the damper above is the same relation with $K \propto 1/\theta^2$.

Check valve. Two branches: open, $r = (p_i - p_j) - k\,\dot{m}|\dot{m}|$ with guard $\dot{m} \geq 0$; closed, $r = \dot{m}$ with guard $p_j - p_i \geq 0$. *Pattern*: piecewise. *Kinks*: the branch switch is the whole point. *Shape*: monotone. The regime loop settles it in one or two passes; the probabilistic model weighs the two branches when the pressure difference is uncertain (§7.2). *Source*: textbook; any valve in [97] with a one-way guard.

Isothermal gas pipe.

$$r = (p_i^2 - p_j^2) - K\,\dot{m}\,|\dot{m}|.$$

Pattern: gradient, with the square of pressure as the potential. *Parameters*: K, from length, diameter, friction factor, temperature and the gas constant. *Kinks*: C^1 at no flow. *Shape*: increasing in flow. The squared potential makes the relation linear in p^2 for a fixed flow, which is why gas networks are usually solved in p^2. *Prior*: the friction factor inside K. *Source*: textbook [97].

Ideal-gas mass flow.

$$r = \dot{m} - \frac{p}{R\,T}\,\dot{V}.$$

Pattern: property, with the density fixed when the closure is built. *Shape*: linear in $\dot{V}$. *Prior*: the volume flow, which the code's own documentation suggests pinning or giving a prior. *Source*: `thermo`, `ideal_gas_mass_flow`.

Mixing of two streams.

$$r = T_{\mathrm{mix}} - f\,T_{oa} - (1-f)\,T_{ra}, \qquad r = \dot{m}_{oa} - f\,\dot{m}_{\mathrm{supply}}.$$

Pattern: property. *Shape*: bilinear in the fraction f and the temperatures. *Prior*: f, the outdoor-air fraction, which dampers set and few sensors report; a carbon-dioxide balance on the zone, $G = (\dot{m}_{oa}/\rho)(C_{\mathrm{zone}} - C_{oa})$, makes it identifiable. *Source*: `hvac`, mixed-air closures.

D.7 Moist air

Humidity ratio and moist-air enthalpy.

$$W = 0.62198\,\frac{p_w}{p - p_w}, \qquad h = 1006\,t + W\,(2.501 \times 10^6 + 1860\,t),$$

with t in degrees Celsius and the saturation pressure from the Magnus form

$$p_{ws} = 0.61094\,\exp\!\left[\frac{17.625\,t}{t + 243.04}\right]\ \mathrm{kPa}.$$

Pattern: property, when written as closures. *Shape*: increasing in p_w and in t. *Source*: `thermo`, the functions of `psychro`, which the plugins call inside other closures; the relations are in [4].

Coil condensation.

$$r = \dot{m}_{\mathrm{cond}} - \dot{m}_{\mathrm{air}}\,(1 - \mathrm{BF})\,\max\!\big(0,\, W_{\mathrm{zone}} - W_{\mathrm{adp}}\big),$$

with W_{adp} the saturated humidity ratio at the coil's apparatus dew point, the chilled-water temperature plus an approach. The latent duty is $\dot{m}_{\mathrm{cond}}$ times the heat of vaporization. *Pattern*: property. *Parameters*: bypass factor BF, approach [K]. *Jacobian*: differenced. *Kinks*: at $W_{\mathrm{zone}} = W_{\mathrm{adp}}$, where the coil starts to condense. *Shape*: non-decreasing in the zone's humidity. *Prior*: the bypass factor. *Source*: `datacenter`, the CRAH zone.

Moisture carried by mass.

$$r = \dot{m}_w - \dot{m}\,W.$$

Pattern: carrier. *Source*: `thermo`, `moisture_advection`.

D.8 Plant equipment

Chiller power from load.

$$r = P - \frac{Q}{\mathrm{COP}}, \qquad \mathrm{COP} = \max\bigl(\mathrm{COP}_{\min},\ \mathrm{COP}_{\mathrm{pk}} - c\,(x - x_{\mathrm{opt}})^2\bigr),$$

with x the load per running unit as a fraction of one unit's capacity, and $P = 0$ at no load. *Pattern*: property. *Jacobian*: differenced. *Kinks*: at zero load, at the COP floor, and wherever the stage count changes, since the stage enters through rounding. *Shape*: increasing in load at fixed stage for the default curve. A condenser-loop variant also derates the COP linearly with the condenser-water temperature. *Prior*: the peak COP and the curvature, fitted to a part-load curve. *Source*: `datacenter`, `ChillerBankPowerClosure` and the condenser loop.

Chiller staging. A piecewise closure with one branch per stage, $r = \mathrm{stage} - k$. Stage k is admissible between $(k-1)$ units at their stage-down fraction and k units at their stage-up fraction, 0.70 and 0.90 of capacity by default. *Kinks*: every stage change. *Shape*: the admissible bands of neighbouring stages overlap, and since the current stage is kept while admissible, the overlap is a deadband that stops the plant from cycling. *Source*: `datacenter`, `ChillerBankStageClosure`.

Thermal storage mode. A piecewise closure with branches charge, hold and discharge, $r = \mathrm{mode} - m$ with $m \in \{+1, 0, -1\}$. Guards combine the plant load with the tank's state of charge: charge below a load threshold while the tank is not full, discharge above another while it is not empty, hold otherwise. *Kinks*: every mode change. *Source*: `datacenter`, `TesModeClosure`.

Bibliography

[1] Ali Abur and Antonio Gomez Exposito. *Power System State Estimation: Theory and Implementation*. Marcel Dekker, 2004.

[2] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, 2005.

[3] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic and Discrete Methods*, 8(2):277–284, 1987.

[4] ASHRAE. *ASHRAE Handbook—Fundamentals*. ASHRAE, Atlanta, 2021.

[5] Albert Benveniste, Benoît Caillaud, and Mathias Malandain. The mathematical foundations of physical systems modeling languages. *Annual Reviews in Control*, 50:72–118, 2020.

[6] Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. Wiley, 7th edition, 2011.

[7] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–236, 1974.

[8] Roy Billinton and Ronald N. Allan. *Reliability Evaluation of Power Systems*. Plenum Press, New York, 2nd edition, 1996.

[9] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] José Luis Blanco-Claraco, Antonio Leanza, and Giulio Reina. A general framework for modeling and dynamic simulation of multibody systems using factor graphs. *Nonlinear Dynamics*, 105(3):2031–2053, 2021.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.

[13] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[14] Phani Chavali and Arye Nehorai. Distributed power system state estimation using factor graphs. *IEEE Transactions on Signal Processing*, 63(11):2864–2876, 2015.

[15] Roy R. Craig and Mervyn C. C. Bampton. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1313–1319, 1968.

[16] Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[17] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2):1–139, 2017.

[18] Florian Dörfler and Francesco Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, 2013.

[19] Iain S. Duff, Albert M. Erisman, and John K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, 2nd edition, 2017.

[20] A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

[21] A. M. Erisman and W. F. Tinney. On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3):177–179, 1975.

[22] Lawrence C. Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.

[23] Herbert Federer. *Geometric Measure Theory*. Springer, 1969.

[24] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23 (2):298–305, 1973.

[25] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[26] Claudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. *ACM Computing Surveys*, 51(3):1–33, 2018.

[27] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F: Radar and Signal Processing*, 140 (2):107–113, 1993.

[28] Teoman Guler, George Gross, and Minghai Liu. Generalized line outage distribution factors. *IEEE Transactions on Power Systems*, 22(2):879–881, 2007.

[29] Kjell Gustafsson, Michael Lundh, and Gustaf Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28(2): 270–287, 1988.

[30] Robert J. Guyan. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380, 1965.

[31] William W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989.

[32] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 2nd edition, 1996.

[33] Qing Han, Ronald T. Eguchi, Sharad Mehrotra, and Nalini Venkatasubramanian. Enabling state estimation for fault identification in water distribution systems under large disasters. In *Proceedings of the 37th IEEE Symposium on Reliable Distributed Systems*, pages 161–170, 2018.

[34] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

[35] IEEE Power and Energy Society. IEEE standard for calculating the current-temperature relationship of bare overhead conductors. Technical Report IEEE Std 738-2012, IEEE, 2013.

[36] Paul Irofti, Luis Romero-Ben, Florin Stoican, and Vicenç Puig. Factor graph optimization for leak localization in water distribution networks. arXiv:2509.10982, 2025.

[37] Simon J. Julier and Jeffrey K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference*, volume 4, pages 2369–2373, 1997.

[38] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.

[39] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[40] Dean C. Karnopp and Ronald C. Rosenberg. *Analysis and Simulation of Multiport Systems: The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.

[41] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[42] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[43] William M. Kays and Alexander L. London. *Compact Heat Exchangers*. McGraw-Hill, 3rd edition, 1984.

[44] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[45] Brian W. Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, 1970.

[46] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[47] Gabriel Kron. *Tensor Analysis of Networks*. Wiley, New York, 1939.

[48] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[49] Steffen L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

[50] Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[51] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–224, 1988.

[52] Steffen L. Lauritzen and Nanny Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17(1): 31–57, 1989.

[53] Dennis V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[54] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[55] C. H. Lo, Y. K. Wong, and A. B. Rad. Bond graph based Bayesian network for fault diagnosis. *Applied Soft Computing*, 11(1):1208–1212, 2011.

[56] Hans-Andrea Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

[57] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[58] Richard S. H. Mah, Gregory M. Stanley, and Dennis M. Downing. Reconciliation and rectification of process flow and inventory data. *Industrial & Engineering Chemistry Process Design and Development*, 15(1):175–183, 1976.

[59] Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7: 2031–2064, 2006.

[60] Albert W. Marshall and Ingram Olkin. A multivariate exponential distribution. *Journal of the American Statistical Association*, 62(317):30–44, 1967.

[61] Mihajlo D. Mesarović, D. Macko, and Yasuhiko Takahara. *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York, 1970.

[62] Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Geometric separators for finite-element meshes. *SIAM Journal on Scientific Computing*, 19(2):364–386, 1998.

[63] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001.

[64] Modelica Association. Modelica: A unified object-oriented language for systems modeling. language specification, version 3.6. `https://specification.modelica.org`, 2023.

[65] Alcir Monticelli. *State Estimation in Electric Power Systems: A Generalized Approach*. Kluwer Academic, Boston, 1999.

[66] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[67] George Oster, Alan Perelson, and Aharon Katchalsky. Network thermodynamics. *Nature*, 234:393–399, 1971.

[68] Art B. Owen. Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.

[69] Constantinos C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, 1988.

[70] Henry M. Paynter. *Analysis and Design of Engineering Systems*. MIT Press, Cambridge, MA, 1961.

[71] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[72] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[73] Gregory L. Plett. *Battery Management Systems, Volume I: Battery Modeling*. Artech House, 2015.

[74] Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.

[75] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[76] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

[77] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

[78] Donald J. Rose, R. Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[79] Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, 2005.

[80] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, 2002.

[81] Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2nd edition, 2023.

[82] Fred C. Schweppe and J. Wildes. Power system static-state estimation, part I: Exact model. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(1):120–125, 1970.

[83] Ori Shental, Paul H. Siegel, Jack K. Wolf, Danny Bickson, and Danny Dolev. Gaussian belief propagation solver for systems of linear equations. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 1863–1867, 2008.

[84] Ilya M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1–3):271–280, 2001.

[85] Eunhye Song, Barry L. Nelson, and Jeremy Staum. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4 (1):1060–1083, 2016.

[86] Brian Stott, Jorge Jardim, and Ongun Alsaç. DC power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300, 2009.

[87] Andrew M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[88] Kaarthik Sundar, Carleton Coffrin, Harsha Nagarajan, and Russell Bent. Probabilistic N-k failure-identification for power systems. *Networks*, 71(3):302–321, 2018.

[89] Kaarthik Sundar, Andrew Mastin, Manuel Garcia, Russell Bent, and Jean-Paul Watson. Exact and heuristic approaches for the stochastic N-k interdiction in power grids. arXiv:2402.00217, 2024.

[90] Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

[91] William F. Tinney and John W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, 55(11):1801–1809, 1967.

[92] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods: Algorithms and Theory*. Springer, 2005.

[93] Arjan J. van der Schaft and Bernhard M. Maschke. Port-Hamiltonian systems on graphs. *SIAM Journal on Control and Optimization*, 51(2):906–937, 2013.

[94] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[95] J. B. Ward. Equivalent circuits for power-flow studies. *Transactions of the American Institute of Electrical Engineers*, 68(1):373–382, 1949.

[96] Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[97] Frank M. White. *Fluid Mechanics*. McGraw-Hill Education, 8th edition, 2016.

[98] Allen J. Wood, Bruce F. Wollenberg, and Gerald B. Sheblé. *Power Generation, Operation, and Control*. Wiley, 3rd edition, 2013.

[99] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981.