Probabilistic Modeling of Battery Energy Storage Systems

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 battery energy storage system is a good system to point that idea at. Its hierarchy was drawn by engineers — cell inside parallel group inside module inside rack inside container — and the levels nest because the cuts nest. It conserves two quantities on one graph, charge and heat, coupled through the heat the current makes in every internal resistance. Its instruments look dense and are not: a management system reports a voltage and a few temperatures per module, and the quantities an operator needs — the capacity of each cell, the resistance that has grown in one connector, the temperature at the middle of a rack — are not among them. And its decisions have obligations and money attached: whether to accept a dispatch, whether to bypass a module, whether the site can deliver the power it is contracted for over the next two hours.

Every system worth modeling is a piece cut out of a larger one, trading a conserved quantity with an environment it does not model. In a storage system the pieces nest: a cell trades charge with the group it is connected in and heat with the coolant around it, a module trades both with its rack, a rack trades current and voltage with the container's bus and heat with the cooling plant, and the container trades power with the grid. At each level the boundary is narrow by construction — a series current, a bus voltage, a coolant temperature — and it 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 those few quantities are exactly what the rest of the system 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 batteries. The balances, closures and ports are the same whether the conserved quantity is charge, heat, mass or power, and Table 1.3 lines them up. Four companion editions take the same method text through electric power transmission, water distribution, district heating and cooling, and data centers. Each is complete in itself, and a reader of this one needs none of them.

Chapters end with exercises, which are worked with pencil and paper and whose answers are given for a selection of them.

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 a set of complete studies, each reproducible from the material in the earlier parts; its opening chapter says which are here and which are still being written. 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.

Preview edition. This is Parts I–III of the book: the objects, probability on a physical graph, and inference, through Chapter 11, with the appendices. Parts IV–VI — the operator's questions, time and hierarchy and scale, and the case studies — are in draft and are not printed here. Cross-references to their chapters are kept and resolve to chapter numbers this edition does not contain.

Contents

Chapter 0

Introduction

A storage site has been asked to discharge at full power for two hours, and forty minutes in, one rack drops out. Its management system reports a cell at the low-voltage limit, although the rack's state of charge says it should have had a quarter of its energy left. The list of explanations is short. One cell has lost capacity and empties before its neighbours. Or its resistance has risen, so it sags to the limit under load while holding charge it cannot deliver fast enough. Or a connector between modules has loosened, and the voltage the system is reading is partly across that joint. Or the sensor reading the low cell is biased and the cell is fine. The answer matters before the next dispatch, because the site is paid for energy it promised and penalised for energy it fails to deliver.

The engineer on call already has the physics of that rack: a charge balance at every node, a voltage relation across every cell and every joint, a heat balance wherever current meets resistance, and a coolant temperature the plant holds. Given the currents, a solver turns those equations into every cell's voltage and every temperature. What the solver cannot do is run the other way: take the few hundred readings a management system offers, one of which may be wrong, and say which cell is the weak one, how sure that is, what the rack can still deliver tomorrow, and whether one more sensor 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 storage 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 storage system looks better instrumented than most physical systems and is not: a management system reports a voltage per cell and a temperature per module, at a rate chosen for protection rather than for inference, and the quantities an operator needs are not among them. No instrument reads a cell's capacity, its internal resistance, or the temperature in the middle of a rack. The parameters that matter drift over years, and the drift is what the operator is paid to track. The boundary conditions — what the grid will ask for, what the coolant will do — are forecasts. And the system's parts fail one at a time, in a population of thousands. The questions an operator or an asset engineer actually asks are:

- What is the state of charge of this rack, given cell voltages I half trust and a model calibrated when it was new?
- Which cell's capacity has faded, and by how much?
- What is the temperature at the middle of the rack, where there is no sensor?
- Is this anomaly a weak cell, a biased sensor, a loose connector or a cooling fault, and which module is it in?
- How uncertain is each of those answers?
- Where should the next sensor go, and what would it be worth?
- What would happen if I bypassed that module, derated the site, or charged it more gently?
- Which uncertain cell or rack drives the risk of missing an obligation, and is the energy figure I am about to promise 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 cell, a module or a rack on a bus 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

balance at node 2:
$$Q_{2a} - Q_{12} = 0$$

the pad:
$$Q_{12} - k\,(T_1 - T_2) = 0$$

the plate, to the coolant:
$$Q_{2a} - hA\,(T_2 - T_a) = 0$$

a sensor on the plate:
$$T_2 = 30.4 \pm 0.5$$

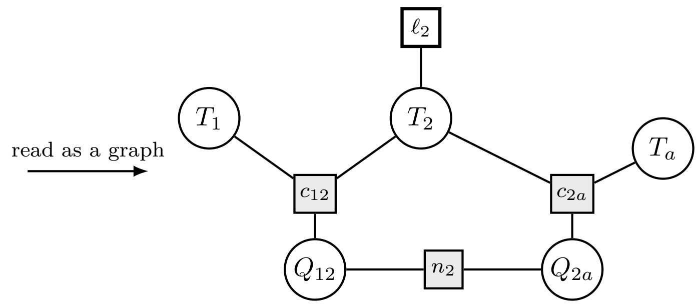


Figure 0.1: The graph comes from the equations. Left: three physics equations of the module heat path of Example 1.1 and one sensor reading, as an engineer writes them. Right: the same four statements as a factor graph. Each equation is a square joined to exactly the variables it reads; the sensor is 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

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 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 flow-and-potential pairs at ports, the same structure whether the domain is thermal, fluid, electrical or mechanical [41, 68, 71, 95]. 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 [6, 21, 65, 70]. 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 [17, 47, 49, 51, 79]. 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 control rooms since Schweppe and Wildes formulated it [83], in battery management, where Kalman and particle filters estimate a cell's state of charge and health from its voltage and current, and in process data reconciliation [1, 39, 59, 66, 89]. 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.

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

$$\boxed{\text{Do not put probability around the solver. Put it into the structure of the physical model.}} \tag{0.3}$$

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

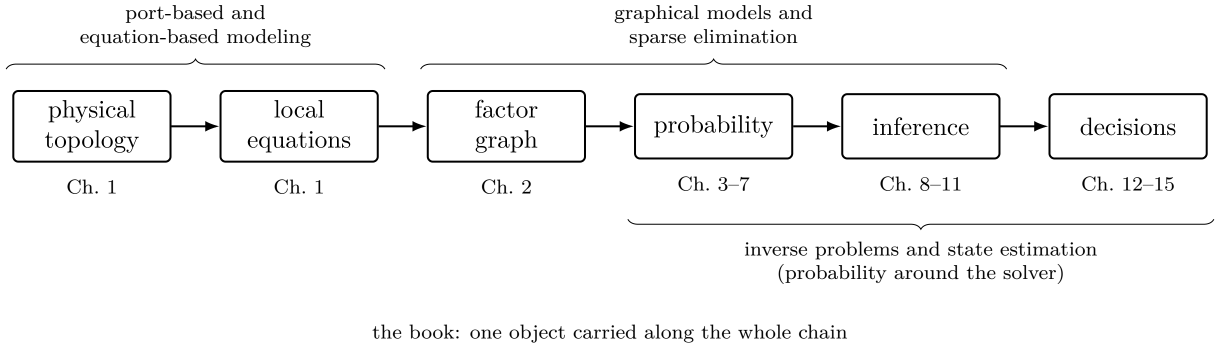
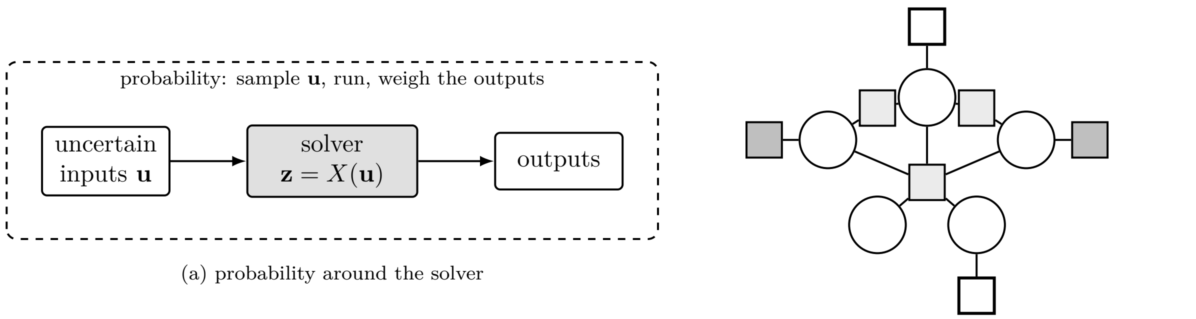


Figure 0.2: The chain of ideas the book follows, with the chapters that cover each link and the traditions that own parts of it. No one tradition carries the object from the topology to the decision.



(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.

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 biased sensor, the connector that has loosened, the cell whose capacity has faded (Chapters 4 and 12).

0.5 The boundary

Every system worth modeling is a piece cut out of a larger one. A cell trades charge with the group it sits in and heat with the coolant around it, a rack trades current with the container's bus, a container trades power with the grid. 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. A battery system was built with its separators in it. Cells in series share one current and nothing else; a module meets its rack through two terminals and a coolant channel; a rack meets the container through its breaker and its bus connection. Those few quantities, a flow and its conjugate potential at each port, are exactly the variables that separate a module's interior from the rest of the system (Chapter 5 proves it, and proves that nothing smaller does). So the instruction "find a separator" becomes "cut the system where it is already bolted together", and the mathematics then explains why that cut is also the cheap place to divide the computation (§8.6 works two racks and one bus).

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

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

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 temperature is evidence about everything the physics ties to it. On the module heat path that runs through the book, a single thermistor on the cold plate halves the uncertainty in the heat the cells are making, and predicts the unmeasured temperature inside the module to within about three quarters of a kelvin, before any second sensor is read (Chapter 3). That

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 [71, 95]	✓	—	—	✓	✓	—
multibody factor graphs [11]	✓	✓	limited	multibody	—	solve, estimate
bond-graph Bayesian networks [56]	✓	indirect	✓	general idea	—	fault diagnosis
factor-graph state estimation [15]	power grid	specialized	✓	power only	regional	state estimation
graphical-model texts [47, 51]	—	—	✓	abstract	graph-theoretic	✓
inverse-problem texts [39, 89]	equations	not generally	✓	✓	—	inverse only
this book	✓	✓	✓	✓	✓	✓

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 [11], which is the "relations, not assignments" reading in a single domain. Bond graphs have been converted into Bayesian networks for fault diagnosis [56], 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 [15], 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.

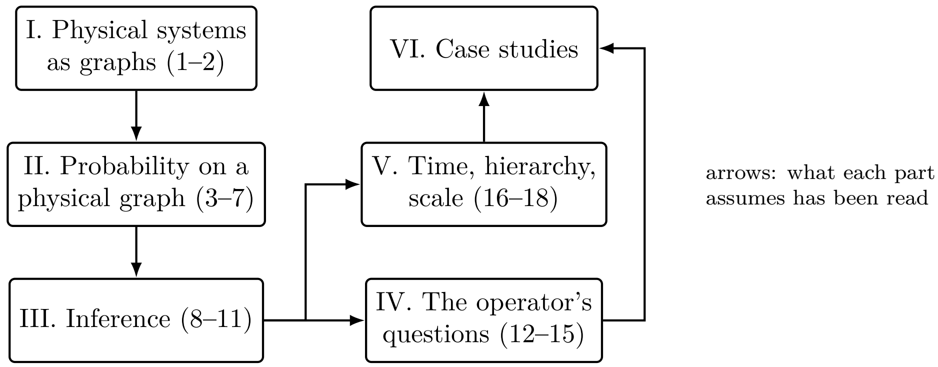


Figure 0.4: The parts of the book and their dependencies. Parts IV and V each need Part III and not each other; Part VI needs both.

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: a module's heat path, from the cells through a pad to a cold plate and into the coolant, and a ring bus with two racks hanging from it. 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. A third, a module carrying charge and heat on one graph, is the book's two-ledger example and is where the coupling between the ledgers is worked out. Larger systems appear where the small ones cannot show the point: a six-tap network for decomposition and failure, two racks for common cause, a mesh of cells 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 model the electrochemistry inside a cell: there are no porous electrodes and no solid-phase diffusion here, and the cell models are the equivalent-circuit and lumped-thermal ones an engineer builds a system from; a physics-based cell model appears only where it supplies a parameter or a truth to check against. It does not treat dynamics faster than the management system's own loop: cell impedance spectra, inverter switching 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 [11], Lo, Wong and Rad's bond-graph Bayesian networks [56], and Chavali and Nehorai's factor-graph state estimation [15], 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 [41], Benveniste, Caillaud and Malandain [6], Koller and Friedman [47], and Kaipio and Somersalo [39].

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 storage container trades power with the grid that dispatches it. A rack trades current with the bus it hangs from and heat with the coolant that cools it. A cell trades both with the module around it. 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 [65, 71], 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* [95]. 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. The meter in a power-conversion system, reading the total current it delivers to the racks behind it, reads a number that is identically their total draw, 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

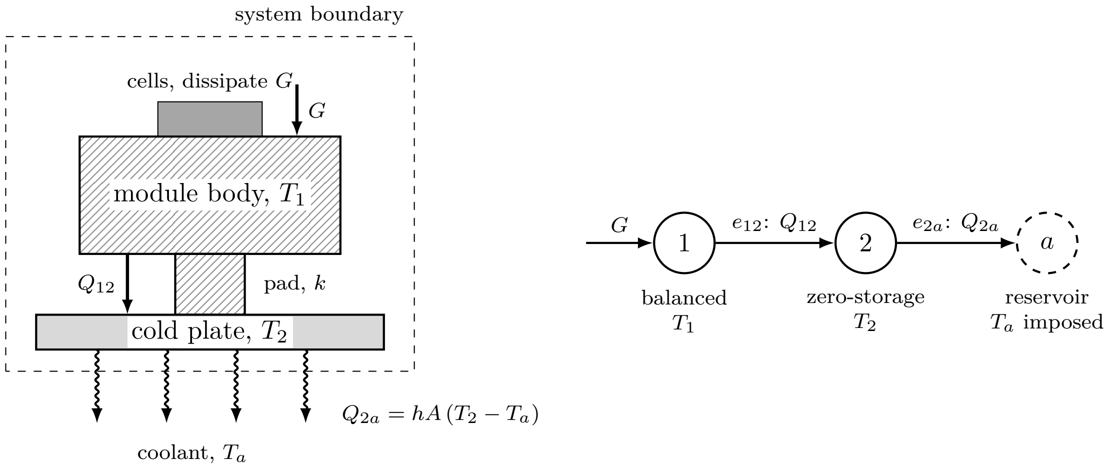


Figure 1.1: The module heat path of Example 1.1. Left: the physical arrangement. The cells dissipate G into the module body at T_1; the body conducts through the thermal pad, of conductance k, to the cold plate at T_2; the plate gives the heat up to the coolant at T_a. The dashed box is the system boundary at the plate's coolant channel; the loop 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 a, dashed, is the coolant 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 a chain and carries one conserved quantity; the second has a loop and is the running example of the book, acquiring uncertainty in Chapter 3 and a failed cable in Chapter 7; the third carries two domains on one graph. The method of this book is not particular to batteries, 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 a storage container, at three of its levels: a module, a rack bus, and the cells inside a module. Every number is from `ch01_examples.py` in the book's `scripts` directory.

Example 1.1 (A module's heat path to the coolant)**.** The cells of one module dissipate a known heat rate G into the module body (node 1): every ampere that passes through a cell's internal resistance makes heat there, and Example 1.3 works out how much. The body reaches the cold plate beneath it (node 2) through the thermal pad, a path of conductance k. The plate gives the heat up to the coolant flowing through it (node a), with coefficient-area product hA; the pump's flow rate is what sets hA. The coolant temperature T_a, the temperature at which coolant arrives from the chiller, is imposed: the coolant loop beyond the plate is not modeled. Figure 1.1 draws the graph.

Unknowns. Two flows, Q_{12} and Q_{2a}, and two potentials, T_1 and T_2. The coolant temperature is imposed and substituted out.

Incidence. Rows for nodes 1 and 2, columns for edges e_{12} and e_{2a}, 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:

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

$$r_2 = -(Q_{12} - Q_{2a}) = Q_{2a} - Q_{12} \qquad \text{(balance at node 2)}. \tag{1.10}$$

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

$$r_3 = Q_{12} - k\,(T_1 - T_2) \qquad \text{(closure on } e_{12}\text{)}, \tag{1.11}$$

$$r_4 = Q_{2a} - hA\,(T_2 - T_a) \qquad \text{(closure on } e_{2a}\text{)}. \tag{1.12}$$

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:

$$\mathbf{z} = \begin{pmatrix} Q_{12} \\ Q_{2a} \\ T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{T} \end{pmatrix}, \qquad \mathbf{F}(\mathbf{z}) = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} Q_{12} - G \\ Q_{2a} - Q_{12} \\ Q_{12} - k\,(T_1 - T_2) \\ Q_{2a} - hA\,(T_2 - T_a) \end{pmatrix}.$$

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:

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

The vector $\mathbf{s}$ holds the sources at the inside nodes, the diagonal matrix $\mathbf{K}$ holds one conductance per edge, and $\mathbf{A}_b$ is the dropped reservoir row, through which the imposed temperature T_a enters. The balance rows r_1, r_2 are $-\mathbf{A}\mathbf{q} - \mathbf{s}$. For the closure rows, look at the vector $\mathbf{A}^\mathsf{T}\mathbf{T} + \mathbf{A}_b^\mathsf{T} T_a$. Its first entry is $-T_1 + T_2$ and its second is $-T_2 + T_a$: it is the temperature rise along each edge, head minus tail, so its negative is the drop that drives the heat. The closure rows r_3, r_4 are therefore $\mathbf{q} + \mathbf{K}(\mathbf{A}^\mathsf{T}\mathbf{T} + \mathbf{A}_b^\mathsf{T} T_a)$, and

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

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 temperatures. The closure block depends on the flows through the identity and on the temperatures through $\mathbf{K}\mathbf{A}^\mathsf{T}$. So

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

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 $G = 20$ watts, $k = 5$ and $hA = 2$ watts per kelvin, and $T_a = 20$ degrees, one step from $\mathbf{z} = \mathbf{0}$ gives $\mathbf{q} = (20, 20)^\mathsf{T}$ and $\mathbf{T} = (34, 30)^\mathsf{T}$: both flows equal the source, the module body sits at 34 degrees and the cold plate at 30. 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_{2a}, 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 coolant temperature. Collect them in $\mathbf{u} = (G, T_a)^\mathsf{T}$; then each temperature is a fixed combination of them,

$$T_1 = \mathbf{h}_1^\mathsf{T}\mathbf{u}, \quad \mathbf{h}_1 = \begin{pmatrix} 1/hA + 1/k \\ 1 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 1 \end{pmatrix}; \qquad T_2 = \mathbf{h}_2^\mathsf{T}\mathbf{u}, \quad \mathbf{h}_2 = \begin{pmatrix} 1/hA \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}.$$

Each temperature is the boundary temperature plus the thermal resistances 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 coolant temperature become uncertain and a sensor on a temperature 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 T_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 (Three racks on a ring bus)**.** Three points of a container's direct-current bus are joined in a ring by cable runs of equal conductance g. Racks 2 and 3 are connected at two of them and carry known currents q_2 and q_3, negative while they charge. Point 1 is where the power-conversion system connects, and its potential is taken as the datum, $v_1 = 0$; every other potential in the example is a difference from it, in millivolts. Point 1 is a reservoir: the model does not claim to know the balance of the plant beyond it. Figure 1.2 draws the graph.

Unknowns. Three currents, I_{12}, I_{13} and I_{23}, and two potentials, v_2 and v_3; v_1 is imposed and substituted out.

Incidence. Rows for taps 2 and 3, columns for the three cables 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 taps 2 and 3, which are zero-storage, so there is no accumulation:

$$r_2 = -(I_{12} - I_{23}) - q_2 \qquad \text{(balance at tap 2)}, \tag{1.16}$$

$$r_3 = -(I_{13} + I_{23}) - q_3 \qquad \text{(balance at tap 3)}. \tag{1.17}$$

Three more come from the closures, one gradient closure on each cable, which is Ohm's law with the conductance written as g:

$$r_{12} = I_{12} - g\,(v_1 - v_2) = I_{12} + g\,v_2 \qquad \text{(closure on } e_{12}), \tag{1.18}$$

$$r_{13} = I_{13} + g\,v_3 \qquad \text{(closure on } e_{13}), \tag{1.19}$$

$$r_{23} = I_{23} - g\,(v_2 - v_3) \qquad \text{(closure on } e_{23}). \tag{1.20}$$

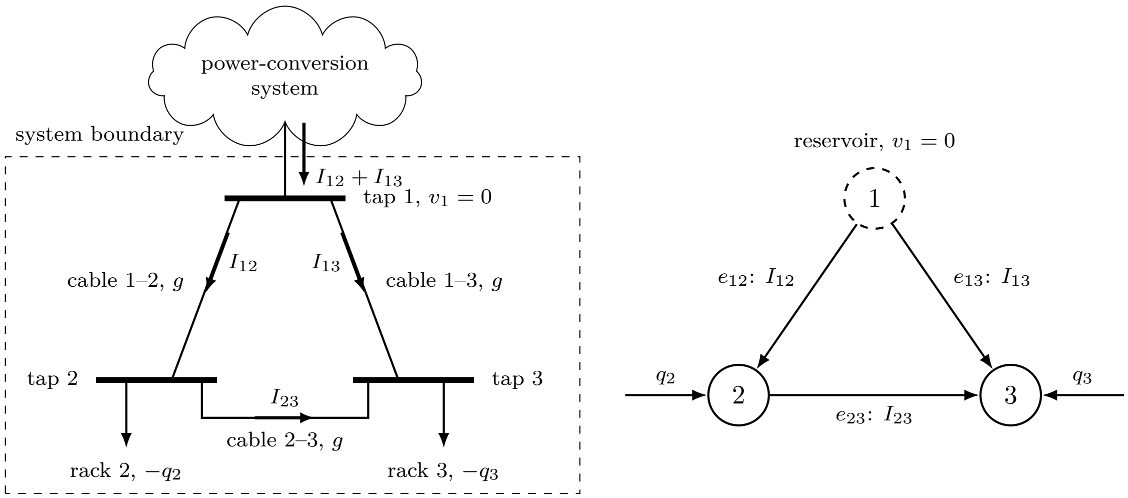


Figure 1.2: The ring bus of Example 1.2. Left: the arrangement. Three tap points are joined by three cable runs of equal conductance g; racks 2 and 3 hang from two of them; tap 1 carries the tie to the power-conversion system, which lies outside the dashed system boundary and supplies whatever the racks take. Arrows mark the current directions of the example. Right: the same bus as a conservation graph. Taps 2 and 3 are zero-storage — copper holds no charge — and tap 1 is a reservoir at the datum potential. The three cables form a loop, the first loop in this book. The rack currents q_2, q_3 are imposed.

Five rows, five unknowns.

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

$$\mathbf{z} = \begin{pmatrix} I_{12} \\ I_{13} \\ I_{23} \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \mathbf{i} \\ \mathbf{v} \end{pmatrix}, \qquad \mathbf{F}(\mathbf{z}) = \begin{pmatrix} r_2 \\ r_3 \\ r_{12} \\ r_{13} \\ r_{23} \end{pmatrix}.$$

With the rack currents $\mathbf{s} = (q_2, q_3)^\mathsf{T}$, one conductance per cable, $\mathbf{K} = g\,\mathbf{I}$, and the reservoir row of tap 1, $\mathbf{A}_b = (-1, -1, 0)$, with its potential v_1, the residual vector has the two blocks of equation (1.13), with potentials in place of temperatures: the balance rows are $-\mathbf{A}\mathbf{i} - \mathbf{s}$ and the closure rows are $\mathbf{i} + \mathbf{K}(\mathbf{A}^\mathsf{T}\mathbf{v} + \mathbf{A}_b^\mathsf{T} v_1)$. Differentiating block by block, as there, gives the Jacobian

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

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. Take $g = 10$ amperes per millivolt, a cable run of a tenth of a milliohm. With $q_2 = q_3 = -40$ amperes — two racks charging at the same rate — both potentials are -4 millivolts, which is $-120/(3g)$, and $\mathbf{i} = (40, 40, 0)^\mathsf{T}$. The cable between the two racks carries nothing, by symmetry. The current across the model's boundary follows as in Example 1.1: here too $\mathbf{1}^\mathsf{T}\mathbf{A} + \mathbf{A}_b = \mathbf{0}$, so $\mathbf{A}_b\mathbf{i} = \mathbf{1}^\mathsf{T}\mathbf{s}$, which reads $-(I_{12} + I_{13}) = q_2 + q_3$. The boundary current $I_{12} + I_{13} = 80$ amperes is the total the racks take: Proposition 1.1 again, with $S = \{2, 3\}$. That current is precisely the port flow of the boundary-exchanging system, and it is also the one current in the example that is metered anyway, because the conversion system measures what it delivers.

Now break the symmetry: $q_2 = -60$, $q_3 = -20$ amperes, one rack taking three times the other. Then $v_2 = -4.667$ and $v_3 = -3.333$ millivolts, and $\mathbf{i} = (46.667, 33.333, -13.333)^\mathsf{T}$. The third entry, I_{23}, is negative: thirteen and a third amperes flow from tap 3 to tap 2, against the drawn direction. The loop matters. Rack 2's extra draw is served partly by the direct cable from tap 1 and partly around the other side of the ring. There is no way to compute I_{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 nodal solve. Because every closure here is linear, the currents can also be eliminated by hand, and that is how an engineer usually solves such a bus. The closure rows give $\mathbf{i} = -\mathbf{K}(\mathbf{A}^\mathsf{T}\mathbf{v} + \mathbf{A}_b^\mathsf{T} v_1)$. Substituting this into the balance rows, with $v_1 = 0$, leaves two rows in the two potentials, and their solution:

$$g\begin{pmatrix}2 & -1 & -1 & 2\end{pmatrix}\begin{pmatrix}v_2 & v_3\end{pmatrix} = \begin{pmatrix}q_2 & q_3\end{pmatrix}, \qquad \begin{pmatrix}v_2 & v_3\end{pmatrix} = \frac{1}{3g}\begin{pmatrix}2 & 1 & 1 & 2\end{pmatrix}\begin{pmatrix}q_2 & q_3\end{pmatrix}, \tag{1.21}$$

after which $\mathbf{i} = -g\,\mathbf{A}^\mathsf{T}\mathbf{v}$. The matrix $g\,\mathbf{A}\mathbf{A}^\mathsf{T}$ on the left is the bus's conductance matrix, and this elimination is nodal analysis. The loop is visible in it. Without cable e_{23} the incidence matrix would be its first two columns alone, $\mathbf{A}\mathbf{A}^\mathsf{T}$ would be the identity, and each potential would follow from its own rack's current; the off-diagonal -1 is the cable that closes the ring, and it is why I_{23} depends on both rack currents 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 — as they do the moment a cell's open-circuit voltage enters in Chapter 7 — the shortcut is gone: only Newton's method on the five-row system remains.

Example 1.3 (A module carrying charge and heat on one graph)**.** Four cells are joined in series between a module's two terminals, cell to cell through bolted joints. The module is clamped to the cold plate of Example 1.1, each cell through its own pad. Two conserved quantities move through the same hardware. Charge runs along the series string, from one terminal to the other. Heat is made wherever that charge meets resistance — in the cells and in the joints — and leaves downwards, through the pads to the plate and from the plate to the coolant. The point of the example is that both live on one graph, and that the second ledger reads the first. Figure 1.3 draws the arrangement and its graph.

Unknowns. The charge ledger has the series current I and one potential drop per element, seven of them: four cells and three joints. The heat ledger has the heat leaving each cell through its pad, four of them, the four cell temperatures T_c, and the plate temperature T_p. The terminal potentials v_+ and v_- and the coolant temperature T_{cool} are imposed. Seventeen unknowns in all.

Residuals. The charge ledger closes around the string: the drops sum to the difference the terminals impose, and each drop obeys Ohm's law with that element's resistance R_e,

$$r_0 = \textstyle\sum_e \Delta v_e - (v_+ - v_-) \qquad \text{(the string closes)}, \tag{1.22}$$

$$r_e = \Delta v_e - I\,R_e \qquad \text{(closure on element } e\text{)}. \tag{1.23}$$

The heat ledger balances at each cell and at the plate. What a cell makes is its own I^2R and half of each neighbouring joint's, since a joint's heat divides between the two cells it touches;

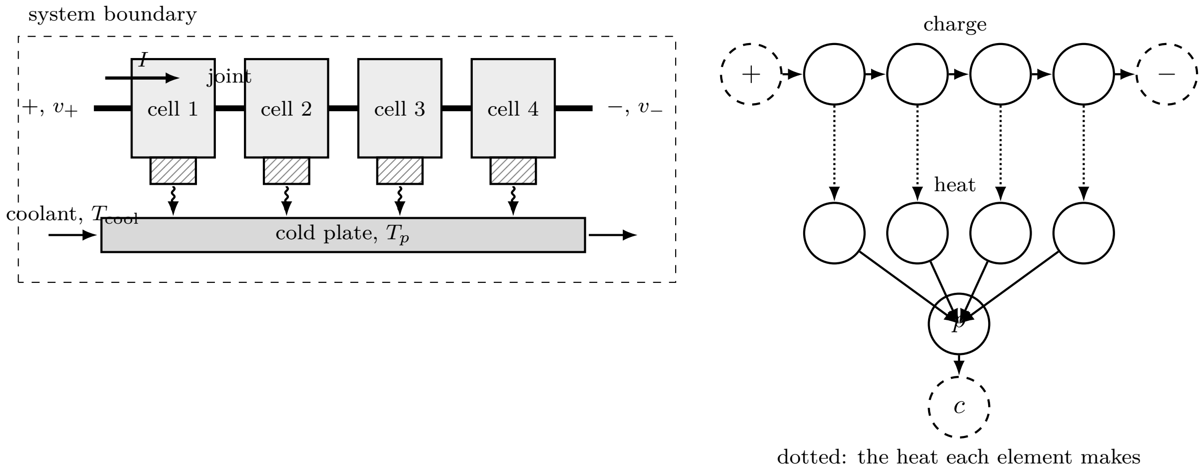


Figure 1.3: The two-ledger module of Example 1.3. Left: four cells in series on one cold plate, each on its own pad, with the coolant passing through the plate. Right: the conservation graph. The upper chain is charge, from terminal to terminal, and its two reservoirs are the terminals; the lower graph is heat, from each cell through its pad to the plate and from the plate to the coolant, whose temperature is imposed. The dotted arrows are not edges: they mark where the heat ledger's sources read the charge ledger's current, which is the coupling between the two.

what it loses is what its pad carries:

$$r_c = Q_c - \left(I^2 R_c + \tfrac{1}{2}\textstyle\sum_{j \ni c} I^2 R_j\right) \qquad \text{(balance at cell } c\text{)}, \tag{1.24}$$

$$r'_c = Q_c - k_c\,(T_c - T_p) \qquad \text{(pad closure)}, \tag{1.25}$$

$$r_p = \textstyle\sum_c Q_c - hA_p\,(T_p - T_{\text{cool}}) \qquad \text{(the plate to the coolant)}. \tag{1.26}$$

Seventeen rows for seventeen unknowns. Every row is one of the kinds this chapter has already introduced; what is new is only that the sources of one ledger are functions of another ledger's unknown.

Solution. Take cells of 0.25 and joints of 0.05 milliohms, so the string is 1.15 milliohms, and hold the terminals 60 millivolts apart. The charge ledger alone gives $I = 52.17$ amperes and a drop of 13.04 millivolts across each cell, 2.61 across each joint. Those currents make 0.681 watts in each cell and 0.136 in each joint, 3.130 watts in all. Splitting each joint's heat between its neighbours leaves 0.749 watts to carry for the two end cells and 0.817 for the two inner ones, which have a joint on each side. With pads of 0.5 watts per kelvin, a plate coupled to the coolant at 4 watts per kelvin, and coolant at 22 degrees, the plate sits at 22.78 degrees, the end cells at 24.28 and the inner cells at 24.42.

That the inner cells run warmer than the outer ones is the example's first useful fact, and nothing about it was put in by hand: it follows from where the joints are. It is also the smallest version of the question Part VI asks of a real pack, where the warmest cell is not the one an operator would have guessed and is not measured either.

The cut, twice. Proposition 1.1 holds on each ledger separately. On the charge ledger, what enters at the positive terminal leaves at the negative: 52.17 amperes, both ways, because no charge accumulates anywhere in the string. On the heat ledger, what crosses the plate's boundary into the coolant is 3.130 watts, which is exactly I^2 times the string's resistance — every watt

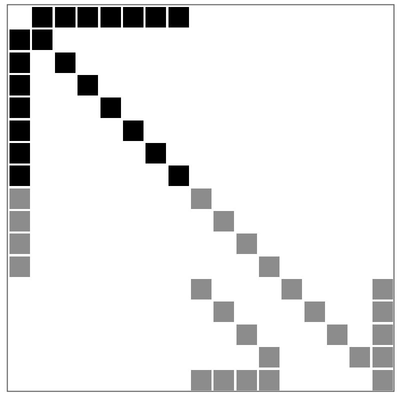


Figure 1.4: Nonzero pattern of the Jacobian of Example 1.3, rows in the fixed order of §1.5 and columns grouped by variable kind, flows first. Black entries are the constant ± 1 of the balance rows; grey entries carry parameters or the state. Fourteen percent of the matrix is nonzero, and the pattern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

the charge ledger made. The two statements are the same proposition applied to two ledgers, and together they say the module is neither storing charge nor storing heat at this instant.

One graph, one Jacobian. Figure 1.4 draws the nonzero pattern of the seventeen-by-seventeen Jacobian. It has 46 nonzeros, sixteen percent of its entries, and full rank. The charge rows, drawn dark, do not read a single temperature: charge does not care how hot the module is, in this model. The heat rows, drawn light, do read the current, in the column under I, and that column is the coupling made visible — one column of entries that would be absent if the ledgers were solved separately. Chapter 7 makes the coupling run both ways, by letting a cell's resistance depend on its temperature, and the pattern gains the mirror of that column.

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 pad conductance k and the cable conductance g were given; the coolant temperature and the rack currents were imposed. Nothing was measured. Nothing failed.

List where a real model's ignorance actually sits.

- *Parameters.* The conductance of a real pad is a nominal value from a data sheet and a clamping pressure nobody has measured since assembly; a cell's internal resistance is a datasheet number at one temperature and one state of charge, and it rises as the cell ages.
- *Boundary conditions.* The coolant temperature follows the chiller and the weather, and is at best a forecast. What the racks draw is a dispatch instruction that the grid may revise within the hour.
- *Model form.* The cable from tap 2 to tap 3 is either in service or it is not, and the closure is different in the two cases. A rack is either following its dispatch or sitting against a cell's voltage limit, and those are two different models.
- *Measurements.* A real system has sensors. Some of its temperatures, voltages and currents 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,

$$p(\mathbf{z}) \propto \prod_f \varphi_f(\mathbf{z}_f), \tag{1.27}$$

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 [71] and developed by Karnopp and Rosenberg [41]; what this chapter calls a port is their pair of effort and flow variables, and Table 1.3 is their table. Network thermodynamics [68] 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 [95]. 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 [65] 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 [6]. Structural rank and the matching of equations to unknowns go back to Dulmage and Mendelsohn [21]; the same structural analysis applied to differential-algebraic systems is Pantelides' algorithm [70], and the numerical theory of such systems is in Brenan, Campbell and Petzold [13].

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 [17] and Duff, Erisman and Reid [20] are the references, and the observation that power-network equations should be factored in an order chosen for sparsity is Tinney and Walker's [93]. The DC power flow of Example 1.2, its assumptions and its accuracy are reviewed by Stott, Jardim and Alsaç [88].

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 module's seventeen rows carry charge and heat together, with the heat ledger's sources reading the charge ledger's current, 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 coolant temperature by an imposed heat flow $Q_{2a} = \bar{Q}$ 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 charge domain to Example 1.1 by giving the module a series current, $I_e = g_e\,(v_{\text{tail}} - v_{\text{head}})$ on each element, with the module's negative terminal imposed and a known current into node 1. Write the incidence matrix with both domains and confirm it is the same matrix. Which nodes need a charge balance, and how many unknowns and rows does the model now have?

Exercise 1.3. In Example 1.2, remove cable e_{23}. Solve for the potentials and currents with $q_2 = -60$, $q_3 = -20$ amperes, and compare I_{12} with the looped case. Then remove cable e_{13} instead. Which removal changes the boundary current at tap 1, and why does Proposition 1.1 say it cannot?

Exercise 1.4. Make tap 1 of Example 1.2 an inside zero-storage node with a known current q_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 bus with no connection to anything outside it cannot fix its own datum.

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,

$$\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -k & k & 0 & 0 \\ 0 & 1 & 0 & -hA & 0 & hA \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

has rank five, and its null direction is $(0, 0, 1, 1, 0, 1)$: add the same constant to T_1, T_2 and T_a and every row is unchanged, because every temperature enters only through a difference. The flows are fully determined, $Q_{12} = Q_{2a} = G = \bar{Q}$, and so are the temperature differences $T_1 - T_2 = \bar{Q}/k$ and $T_2 - T_a = \bar{Q}/hA$; what is not determined is the level. One imposed potential, any one of the three temperatures, 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 thermal form of the floating-datum problem of the next exercise but one — the same failure in the charge 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 $q_2 = -60$ and $q_3 = -20$ amperes the looped bus of Example 1.2 has $I_{12} = 46.667$, $I_{13} = 33.333$ and $I_{23} = -13.333$. Remove e_{23}: each rack is fed by its own cable alone, $I_{12} = 60$, $I_{13} = 20$, and the potentials are $v_2 = -6$ and $v_3 = -2$ millivolts. Remove e_{13} instead: rack 3 is fed through tap 2, $I_{12} = 80$ and $I_{23} = 20$, with $v_2 = -8$ and $v_3 = -10$ millivolts, the far tap now the lowest. In every case the boundary current $I_{12} + I_{13}$ is 80 amperes, the total the racks take. 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 cable changes the internal structure and nothing else. What the removals change is how the two cables from the reservoir share the boundary current, from 46.7 and 33.3 to 60 and 20 or to 80 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.

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 module heat path 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 $T_2 \to c_{12} \to Q_{12} \to n_2 \to Q_{2a} \to c_{2a} \to T_2$. Each step is an edge of the figure. The cycle exists because there are two routes by which T_2 and Q_{2a} are tied together: directly, through the closure

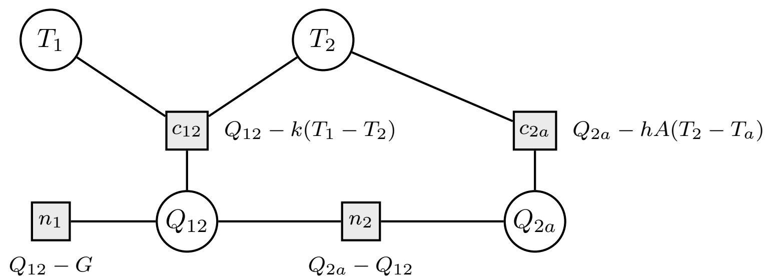


Figure 2.1: The module heat path of Example 1.1 as a factor graph. Circles are the four unknowns; squares are the four rows of $\mathbf{F}$. The two balance rows n_1, n_2 read only flows. The two closures c_{12}, c_{2a} read a flow and the temperatures at the ends of their path. The known rack current G and the imposed coolant temperature T_a are constants inside their factors, not nodes.

of the line into the coolant, and indirectly, through the closure of the line above it and the balance at tap 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.

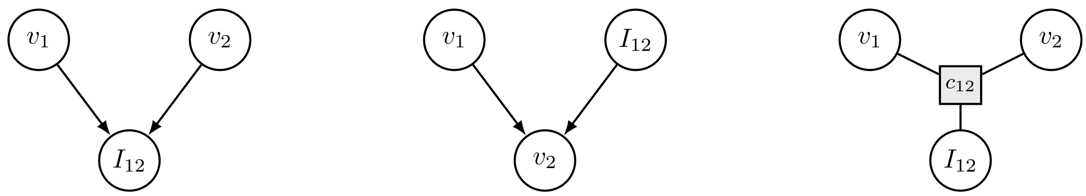
2.2.1 Directed graphs

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

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

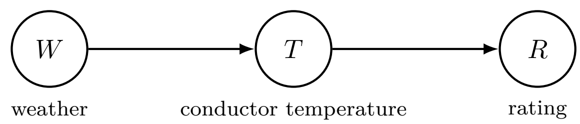
one factor per variable, each reading that variable and its *parents* $\mathrm{pa}(X_i)$, the sources of the arrows into it. The graph must have no directed cycle. When the factors are probabilities the form is $P(A)\,P(B \mid A)\,P(C \mid B)$ for the chain $A \to B \to C$, and the arrows read as "given".

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 potentials (b) potential from flow (c) the relation itself

Figure 2.2: Three drawings of one closure, $I_{12} - B(v_1 - v_2) = 0$. 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

$$I_{12} = B\,(v_1 - v_2). \tag{2.2}$$

The directed language needs one variable to be the child and the rest to be its parents. The textbook arrangement is $v_1 \to I_{12} \leftarrow v_2$, 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 $v_2 = v_1 - I_{12}/B$, which is $v_1 \to v_2 \leftarrow I_{12}$, or $B = I_{12}/(v_1 - v_2)$, 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 module heat path such an order exists. Read Figure 2.1: n_1 delivers $Q_{12} = G$; n_2 then delivers $Q_{2a} = Q_{12}$; c_{2a} then delivers T_2; c_{12} finally delivers T_1. The directed drawing is $G \to Q_{12} \to Q_{2a} \to T_2 \to T_1$, and it is the order in which the example was solved by hand. But now change the boundary condition. Impose the heat flow $Q_{2a} = \bar{Q}$ into the coolant instead of the coolant temperature (Exercise 1.1). Then c_{2a} must deliver T_a, n_2 must deliver Q_{12} from Q_{2a}, 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 ring bus no order exists at all. Try to build one. The balance at tap 2 reads I_{12} and I_{23} and must deliver one of them; the balance at tap 3 reads I_{13} and I_{23} and must deliver one of them. Suppose tap 2 delivers I_{12} and tap 3 delivers I_{13}. Then the closure c_{12} must deliver v_2 from I_{12}, the closure c_{13} must deliver v_3 from I_{13}, and c_{23} must deliver I_{23} from v_2 and v_3. But tap 2 needed I_{23} to deliver I_{12}, and I_{23} needs v_2, which needs I_{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 potentials, 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 module heat path has one assignment of outputs to rows, and its dependency graph is acyclic: the order found above is the only one. The ring bus has three assignments, and every one has a directed cycle; its five rows form a single irreducible block. The two-ledger module of Example 1.3, with seventeen rows, has one block of eight and nine blocks of one. The block of eight is the charge ledger: the row that closes the string and the seven closures that deliver its drops. It does not split, because no single row fixes the current — the drops divide among the seven elements in a proportion the loop row alone decides. The heat ledger, by contrast, is nine substitutions in sequence: once the current is known, each cell's balance delivers its heat, each pad closure its temperature, and the plate's closure the plate temperature. The charge block is solved first and the heat rows after it, because no charge row reads a temperature, which is what the one-way coupling means in the language of this chapter.

> **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
module heat path	4	4	8	1	$1, 1, 1, 1$	5
ring bus	5	5	11	2	5	7
two-ledger module	17	17	46	13	8, then 1 nine times	50

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

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

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 v_1, v_2, I_{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 ring bus, where two closures both read v_2 and two balances both read I_{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:

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

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 $\mathbf{J}$	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 T_a vanished into c_{2a} 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 ring bus. It has the cycle traced in thick lines, a second cycle through v_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 module heat path, which had no physical loop, had one. Counting physical loops does not predict the shape of the factor graph.

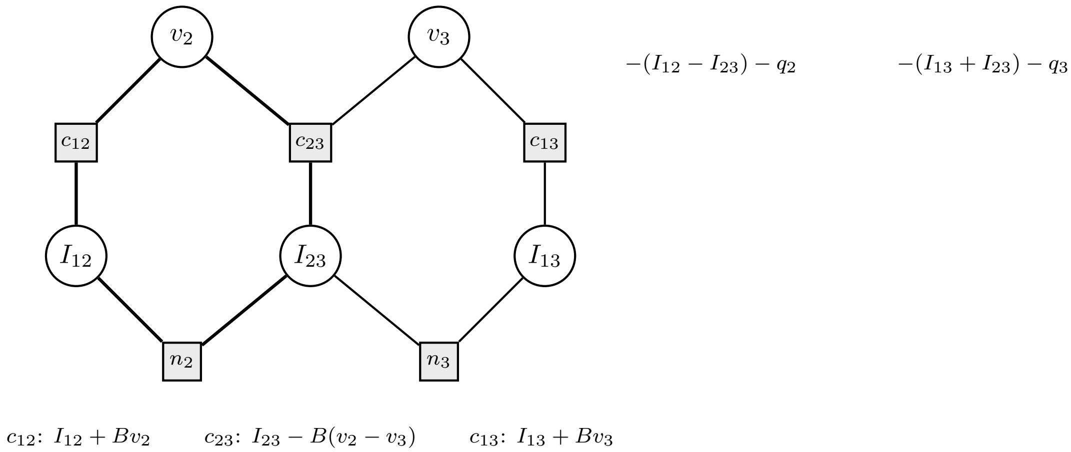


Figure 2.3: The ring bus of Example 1.2 as a factor graph: five unknowns, five rows. The datum potential $v_1 = 0$ and the rack currents q_2, q_3 are constants inside their factors. The thick edges trace one of the graph's cycles, $v_2 \to c_{12} \to I_{12} \to n_2 \to I_{23} \to c_{23} \to v_2$; there are others.

2.3.1 A node 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 tap on the bus of Example 1.2 has one unknown, its potential, 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. A cell is the opposite case. It has a potential and a temperature, a state of charge, a capacity and an internal resistance that both drift as it ages, and it carries a charge balance and a heat balance at once. A physical node becomes several variable nodes and one or more factor nodes.

A cable in the same example has one current and one closure. A cable whose conductance is uncertain adds a parameter variable. A cell whose resistance depends on its temperature, and whose temperature depends on what the coolant is doing, adds a temperature variable, a thermal closure and a resistance closure. A module that may be bypassed 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 two-ledger module of Example 1.3 shows the map at its richest. Its factor graph has thirty-four nodes, seventeen variables and seventeen factors, joined by forty-six edges, one per nonzero of the Jacobian of Figure 1.4. Figure 2.4 draws the part of it that belongs to one cell, together with the rows that read it. The pattern repeats for every cell.

The figure makes two points. The two ledgers are two layers of one graph, joined only through the series current, which every heat source reads; a measurement of one cell's temperature constrains I through that cell's balance, and I constrains every other cell's heat through theirs.

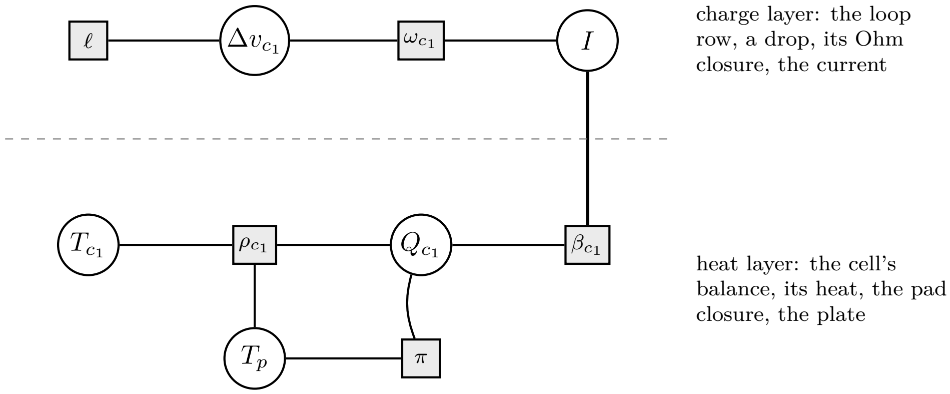


Figure 2.4: One cell of the two-ledger module as a factor graph. Above the dashed line, the charge ledger: the row ℓ that closes the string reads every drop, the closure ω_{c_1} reads this cell's drop and the series current. Below it, the heat ledger: the balance β_{c_1} says what the cell makes, the pad closure ρ_{c_1} reads that heat and the two temperatures it flows between, and the plate's row π reads every cell's heat and the plate temperature. The series current I is the only variable in both layers, and the heavy edge is the only one that crosses the line. It is where the ledgers couple, and it is why a temperature measurement can inform a current.

That path exists because both ledgers describe the same hardware, and it runs in one direction only: no charge row reads a temperature, so the coupling is a one-way street — until Chapter 7 makes a cell's resistance depend on its temperature, and the street runs both ways. And the graph is loopy: Proposition 2.1 found the charge ledger as one block of eight rows, the string's loop row and its seven closures, while every heat row falls in a block of its own. The cycle rank of the whole, edges minus nodes plus one, is thirteen.

2.4 The factor graph is not unique

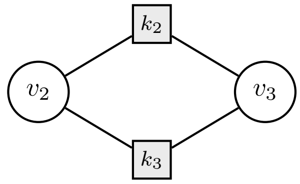
Definition 2.1 draws one factor per row of $\mathbf{F}$. That is the finest factorization the model offers, and it is the canonical one for constructing the model. It is not the only one, and it is not always the one to compute with.

Take the ring bus 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,

$$k_2(v_2, v_3) = 2Bv_2 - Bv_3 - q_2, \qquad k_3(v_2, v_3) = -Bv_2 + 2Bv_3 - q_3, \tag{2.5}$$

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 potentials that satisfy $k_2 = k_3 = 0$ are exactly the potentials 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 module heat path eliminates Q_{12} and Q_{2a} and leaves two rows in T_1, T_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 scope, and the heat path's reduced graph is finally a tree: one factor on $\{T_1, T_2\}$. The ring bus's reduced graph, merged, is a single factor on $\{v_2, v_3\}$, also a tree, and also trivial.



$$k_2 = 2Bv_2 - Bv_3 - q_2$$
$$k_3 = -Bv_2 + 2Bv_3 - q_3$$

Figure 2.5: The ring bus 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.

The two-ledger module shows the same operation on a larger graph, and shows what it does to the Markov graph. Eliminate the seven drops from the charge ledger: substitute each Ohm closure into the row that closes the string. What remains is one row on one unknown, the series current, which is the whole charge ledger reduced to $I = (v_+ - v_-)/\sum_e R_e$ — a resistance in series, which is what an engineer writes down first.

Now eliminate the four heat flows. Substitute each pad closure into its cell's balance and into the plate's row. What remains is five rows on the four cell temperatures and the plate's, and the gap between that graph and the one before it is the point. Before elimination the temperatures had four adjacencies, each cell to the plate and no cell to any other: the cells are independent given the plate, which is what a common cold plate means. After elimination they have ten. The plate's row now reads all four cell temperatures at once, and that single row makes every pair of cells adjacent: six new edges, one for each pair, and not one of them is a physical path. 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, and here it has a physical reading worth keeping: the plate is what the cells have in common, and eliminating a thing they have in common is exactly what couples them to each other.

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 ring bus produced its conductance matrix, which is what an 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.

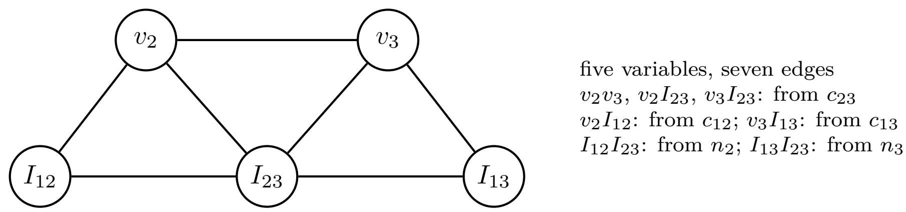


Figure 2.6: The Markov graph of the ring bus in row-per-factor form. Each edge is a pair of variables that some factor reads together; the closure c_{23}, with scope of size three, contributes a triangle. The graph is the sparsity pattern of $\mathbf{J}^\top\mathbf{J}$ for the Jacobian of Example 1.2, 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 $\mathbf{J}^\top\mathbf{J}$, 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 $\mathbf{J}^\top\mathbf{J}$ is not a coincidence.

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

Proof. The entry $(\mathbf{J}^\top\mathbf{J})_{ij} = \sum_k J_{ki}J_{kj}$ 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.) $\square$

Figure 2.6 draws the Markov graph of the ring bus: 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 $\mathbf{J}^\top\mathbf{J}$. Chapter 6 shows that $\mathbf{J}^\top\mathbf{J}$, 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 tap 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 [49]; Loeliger's tutorial [57] is the gentlest introduction, and Dellaert and Kaess [18] 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 [47] and Lauritzen [51]; 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 [21]; equation-based modeling languages use it, together with Pantelides' algorithm [70], to decide how to solve a model, and Benveniste, Caillaud and Malandain [6] 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 [17, 79]. 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 [34] 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 [37] 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 [18] 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 heat path has one order; the ring bus has three candidate assignments, all cyclic; the two-ledger module 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 one cell clamped between two plates. The cell sits at temperature T_j and carries a known current, so it makes a known heat G_j. Three paths leave it: two pads, to plates whose temperatures T_{p_1} and T_{p_2} are imposed, and one path to its neighbouring cell at T_n, itself imposed. Each path carries a gradient closure $Q_i = k_i(T_j - T_{s_i})$, and the cell carries one balance. Four unknowns, four rows. Identify every cycle, and say what would have to change for one to pass through two ledgers.

Exercise 2.2. Complete the argument of §2.2.2 for the ring bus: 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 heat path is a tree of n taps fed from a coolant reservoir, with one path into each node and a known heat at each tap. 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 heat path by hand. Then add a single tie line closing one loop and show that the order is destroyed.

Exercise 2.4. For the module heat path, form the reduced two-row system in (T_1, T_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 ring bus 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 $\{I_{12}, I_{23}\}$, n_3 on $\{I_{13}, I_{23}\}$, c_{12} on $\{v_2, I_{12}\}$, c_{13} on $\{v_3, I_{13}\}$ and c_{23} on $\{v_2, v_3, I_{23}\}$. An assignment gives each row one of its variables with no repeats. Since v_2 can be delivered only by c_{12} or c_{23} and v_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 potential. Enumerating: (1) $c_{12} \to v_2$, $c_{13} \to v_3$, $c_{23} \to I_{23}$, $n_2 \to I_{12}$, $n_3 \to I_{13}$; (2) $c_{12} \to I_{12}$, $c_{23} \to v_2$, $c_{13} \to v_3$, $n_2 \to I_{23}$, $n_3 \to I_{13}$; (3) the mirror image of (2) with the roles of taps 2 and 3 exchanged. There are three, and no others. Each has a directed cycle. In (1), c_{12} needs I_{12}, which n_2 delivers from I_{23}, which c_{23} delivers from v_2, which c_{12} was to deliver: $I_{12} \to v_2 \to I_{23} \to I_{12}$. In (2), n_2 delivers I_{23} from I_{12}, c_{12} delivers I_{12} from v_2, and c_{23} delivers v_2 from I_{23} and v_3: again $I_{23} \to v_2 \to I_{12} \to I_{23}$. 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 $Q_{12} = k(T_1 - T_2)$ from c_{12} into n_1 and $Q_{2a} = hA(T_2 - T_a)$ from c_{2a}, together with Q_{12}, into n_2:

$$k_1 = k\,(T_1 - T_2) - G, \qquad k_2 = hA\,(T_2 - T_a) - k\,(T_1 - T_2).$$

Setting $k_1 = 0$ gives $T_1 - T_2 = G/k$; substituting into $k_2 = 0$ gives $T_2 = T_a + G/hA$; and then $T_1 = T_a + G/hA + G/k$, which is the four-row solution, 30 and 34 degrees 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 $\{T_1, T_2\}$ leaves a tree of three nodes, cycle rank zero. The largest scope is three in the four-row form (c_{12} reads T_1, T_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.

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 rack current G in a model of the module heat path is what one would report as “the rack current 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

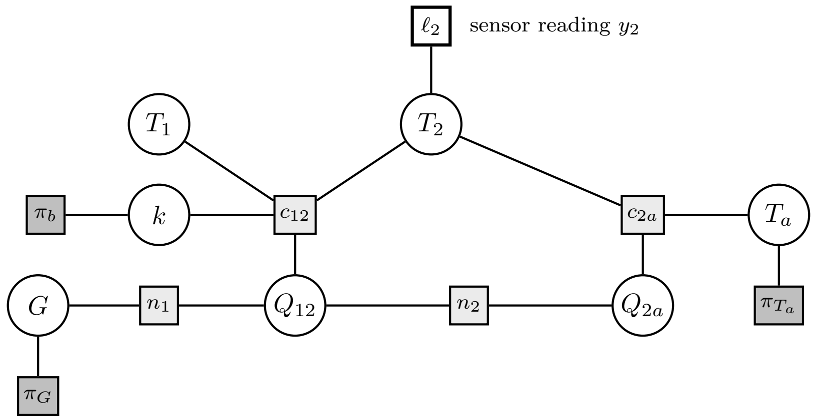


Figure 3.1: The module heat path with probability attached. Compared with Figure 2.1 three variables have been freed, the rack current G, the coolant temperature T_a and the conductance k, and each has acquired a prior (dark squares π). A sensor on T_2 has been added (open square ℓ_2). The four physics factors are the same squares as before, now soft. No edge of the earlier graph has moved.

dimension,

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

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 current at tap 2 takes each value, or as a degree of belief, how strongly the modeler expects the conductance 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 module heat path 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 conductance k 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,

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

which is $k \sim \mathcal{N}(5, 1^2)$ up to the constant. The coolant temperature is a set point with a deadband, $T_a \sim \mathcal{N}(20, 1^2)$ degrees. The heat the cells make 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 temperature, and a conductance that shifts as the conductor heats and sags. The heat that actually flows differs from $k(T_1 - T_2)$ 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 heat leaving a module, 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:

$$r_{23}(v_2, v_3, I_{23}, a_{23}) = I_{23} - a_{23}\, B\,(v_2 - v_3). \tag{3.6}$$

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 potentials: 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.

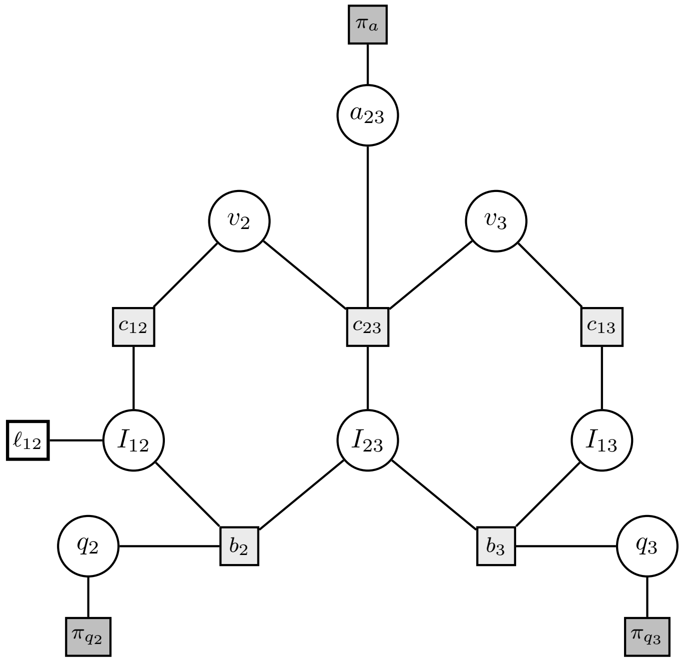


Figure 3.2: The ring bus with probability attached. The rack currents q_2, q_3 are freed and given priors; the line from tap 2 to tap 3 has an availability variable a_{23} with a failure prior, in the scope of its closure; a meter ℓ_{12} reads the flow on line 1–2. The structure of Figure 2.3 is intact inside it.

3.2.6 The complete list

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

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

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:

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

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 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. 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 temperature 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 heat path of §3.4 after its first reading, the heat sits 0.15 prior widths above nominal, the coolant temperature 0.08 above, and the sensor is fitted to a twenty-fifth of its accuracy; the sum of the squares, 0.030, is twice the objective at its minimum. After the second reading the four residuals are 0.19, 0.05, 0.04 and -0.06, and the sum is 0.042: two readings, both fitted to a tenth of an accuracy, at the cost of moving the source a fifth 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

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{k}\,(y - \mathbf{h}^\top\boldsymbol{\mu}), \qquad \boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \mathbf{k}\,\mathbf{h}^\top\boldsymbol{\Sigma}, \qquad \mathbf{k} = \frac{\boldsymbol{\Sigma}\mathbf{h}}{\mathbf{h}^\top\boldsymbol{\Sigma}\mathbf{h} + \sigma^2}. \tag{3.9}$$

Proof. Write $e = y - \mathbf{h}^\top\boldsymbol{\mu}$ for the *innovation*, what was read less what the prior predicts, and $s = \mathbf{h}^\top\boldsymbol{\Sigma}\mathbf{h} + \sigma^2$ 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 $(2\pi)^{-n/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp[-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})]$. Given $\mathbf{z}$, the number $\mathbf{h}^\top\mathbf{z}$ is fixed and ν is still $\mathcal{N}(0,\sigma^2)$, because it is independent of $\mathbf{z}$; so $y \mid \mathbf{z} \sim \mathcal{N}(\mathbf{h}^\top\mathbf{z}, \sigma^2)$, with density $(2\pi\sigma^2)^{-1/2}\exp[-(y - \mathbf{h}^\top\mathbf{z})^2/2\sigma^2]$. Multiplying the two densities and taking the negative logarithm,

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

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 $y - \mathbf{h}^\top\mathbf{z} = e - \mathbf{h}^\top\mathbf{x}$, and expanding the square with $(\mathbf{h}^\top\mathbf{x})^2 = \mathbf{x}^\top\mathbf{h}\mathbf{h}^\top\mathbf{x}$,

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

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

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

recovers the line above. A density proportional to $\exp[-\frac{1}{2}(\mathbf{x}-\mathbf{m})^\top\mathbf{P}(\mathbf{x}-\mathbf{m})]$ 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 $\mathbf{P}^{-1} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^\top\boldsymbol{\Sigma}/s$. Multiplying, and using that $\mathbf{h}^\top\boldsymbol{\Sigma}\mathbf{h}$ is a number,

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

because the bracket is $(s - \sigma^2 - \mathbf{h}^\top\boldsymbol{\Sigma}\mathbf{h})/(\sigma^2 s) = 0$. Since $\boldsymbol{\Sigma}\mathbf{h}\mathbf{h}^\top\boldsymbol{\Sigma}/s = \mathbf{k}\,\mathbf{h}^\top\boldsymbol{\Sigma}$, 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, $\boldsymbol{\Sigma}'\mathbf{h} = \boldsymbol{\Sigma}\mathbf{h} - \boldsymbol{\Sigma}\mathbf{h}\,(\mathbf{h}^\top\boldsymbol{\Sigma}\mathbf{h})/s = \boldsymbol{\Sigma}\mathbf{h}\,\sigma^2/s$. 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 $\mathbf{h}$ 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 temperature, 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 rack current and the coolant temperature, 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 module heat path, 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 conductances, $k = 5$ and $hA = 2$ watts per kelvin, and two changes. The heat the cells make and the coolant temperature are no longer known: they carry the priors $G \sim \mathcal{N}(20, 4^2)$ watts and $T_a \sim \mathcal{N}(20, 1^2)$ degrees, centred on the values Example 1.1 used. The first says the module's dissipation follows a dispatch that is planned but not measured; the second says the chiller holds its setpoint to about a degree. And a sensor on T_2, a thermistor on the cold plate with accuracy $\sigma_y = 0.5$ kelvin, reads $y_2 = 30.4$. 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:

$$\mathbf{z} = \begin{pmatrix} Q_{12} \\ Q_{2a} \\ T_1 \\ T_2 \\ G \\ T_a \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{v} \\ \mathbf{u} \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} G \\ T_a \end{pmatrix}.$$

Exact physics leaves no freedom beyond the inputs. Example 1.1 solved $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ for the flows and temperatures in terms of G and T_a: both flows equal G, and $T_1 = \mathbf{h}_1^\mathsf{T}\mathbf{u}$, $T_2 = \mathbf{h}_2^\mathsf{T}\mathbf{u}$. 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,

$$\boldsymbol{\mu}_u = \begin{pmatrix} 20 \\ 20 \end{pmatrix}, \qquad \boldsymbol{\Sigma}_u = \begin{pmatrix} 4^2 & 0 \\ 0 & 1^2 \end{pmatrix} = \begin{pmatrix} 4.0 \times 10^{-4} & 0 \\ 0 & 9.0 \times 10^{-6} \end{pmatrix},$$

where the zeros off the diagonal say that the rack current and the coolant temperature are independent a priori. It passes to the whole vector as $\boldsymbol{\mu}_z = \mathbf{S}\boldsymbol{\mu}_u = (20, 20, 34, 30, 20, 20)^\mathsf{T}$ and $\boldsymbol{\Sigma}_z = \mathbf{S}\boldsymbol{\Sigma}_u\mathbf{S}^\mathsf{T}$. The sensor on T_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 T_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, $1/hA = 0.5$, is how far the plate rises for each watt the cells make; the second, 1, is how far it rises for each degree the coolant rises.

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 T_2 at $\mathbf{e}_4^\mathsf{T}\boldsymbol{\mu}_z = \mathbf{h}_2^\mathsf{T}\boldsymbol{\mu}_u = 0.5 \cdot 20 + 1 \cdot 20 = 30.0$ degrees, with variance $\mathbf{h}_2^\mathsf{T}\boldsymbol{\Sigma}_u\mathbf{h}_2 = 0.5^2 \cdot 16 + 1^2 \cdot 1 = 5.0$, a spread of 2.24 kelvin. Most of that variance comes from the heat, 4 of the 5, against the coolant's 1. The same two products with $\mathbf{h}_1$, the third entries of $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$, give the prior for T_1, 34.00 ± 2.97.

After one reading. The reading $y_2 = 30.4$ gives the innovation $e = y_2 - \mathbf{h}_2^\mathsf{T}\boldsymbol{\mu}_u = 0.40$ kelvin. Proposition 3.1 then takes a handful of products:

$$\boldsymbol{\Sigma}_u\mathbf{h}_2 = \begin{pmatrix} 16 \cdot 0.5 \\ 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} 8.00 \\ 1.00 \end{pmatrix}, \qquad s = \mathbf{h}_2^\mathsf{T}\boldsymbol{\Sigma}_u\mathbf{h}_2 + \sigma^2 = 5.00 + 0.25 = 5.25,$$

$$\mathbf{k} = \frac{\boldsymbol{\Sigma}_u\mathbf{h}_2}{s} = \begin{pmatrix} 1.5238 \\ 0.1905 \end{pmatrix}, \qquad \boldsymbol{\mu}'_u = \boldsymbol{\mu}_u + \mathbf{k}\, e = \begin{pmatrix} 20 + 0.6095 \\ 20 + 0.0762 \end{pmatrix} = \begin{pmatrix} 20.610 \\ 20.076 \end{pmatrix},$$

$$\boldsymbol{\Sigma}'_u = \boldsymbol{\Sigma}_u - \mathbf{k}\,(\boldsymbol{\Sigma}_u\mathbf{h}_2)^\mathsf{T} = \begin{pmatrix} 3.8095 & -1.5238 \\ -1.5238 & 0.8095 \end{pmatrix},$$

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 heat's falls from 16 to 3.81, a spread of 1.95 watts, and the coolant temperature's from 1 to 0.81, a spread of 0.90 kelvin. The off-diagonal entry, zero before, is now -1.52, a correlation of -0.868. The posterior of the whole vector follows through $\mathbf{S}$:

$$\boldsymbol{\mu}'_z = \mathbf{S}\boldsymbol{\mu}'_u = (20.61,\ 20.61,\ 34.50,\ 30.38,\ 20.61,\ 20.08)^\mathsf{T},$$

and the diagonal of $\mathbf{S}\boldsymbol{\Sigma}'_u\mathbf{S}^\mathsf{T}$ gives the spreads $(1.95, 1.95, 0.74, 0.49, 1.95, 0.90)$. Both heat flows sit at 20.61 ± 1.95 watts, since each equals the source, and the plate at 30.38 ± 0.49 degrees. The posterior, in Table 3.2, is exactly this: the heat up by 0.61 watts and the coolant temperature by 0.08 kelvin; the heat moves more because it had more room to move. The heat's spread falls from 4.0 to 1.95 watts, a factor of two, from a single temperature reading taken on the plate below. The coolant temperature's spread barely moves, from 1.00 to 0.90. And the two, independent under the prior, are now correlated at -0.868: the posterior knows that $T_a + 0.5G$ is close to 30.4, so it knows that if the module is making more heat the coolant must be colder, without knowing which. Figure 3.3 draws the prior and the two posteriors as one-standard-deviation ellipses in the (G, T_a) plane; the first posterior is the thin diagonal ridge.

A virtual sensor. T_1 has not been measured, but it is the third entry of $\mathbf{z}$, and the posterior predicts it: $\mathbf{h}_1^\mathsf{T}\boldsymbol{\mu}'_u = 0.7 \cdot 20.610 + 20.076 = 34.50$, with variance $\mathbf{h}_1^\mathsf{T}\boldsymbol{\Sigma}'_u\mathbf{h}_1 = 0.543$, a spread of 0.74 kelvin. Its prior spread was 2.97. The reading on the plate has tightened the prediction inside the module by a factor of four, through the physics: T_1 and T_2 differ by G/k, and G is now known to ± 1.95 watts. This is the first virtual sensor in the book. Chapter 12 is about them. Figure 3.4 draws the three predictive densities of T_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.

Table 3.2: The module heat path: prior, posterior after the plate sensor, posterior after both sensors. Means and standard deviations, the heat in watts and every temperature in degrees; the last two rows are the temperatures predicted from each distribution.

	Prior	After $y_2 = 30.4$	After y_2 and $y_1 = 34.6$
G [W]	20.00 ± 4.00	20.61 ± 1.95	20.75 ± 1.47
T_a [°C]	20.00 ± 1.00	20.08 ± 0.90	20.04 ± 0.85
$\mathrm{corr}(G, T_a)$	0	-0.868	-0.920
T_2 predicted	30.00 ± 2.24	30.38 ± 0.49	30.42 ± 0.34
T_1 predicted	34.00 ± 2.97	34.50 ± 0.74	34.57 ± 0.41

A second reading. Now a sensor inside the module reads T_1: $y_1 = 34.6$. The reading falls within the virtual sensor's range, which is the first thing to check: had it read 39, six standard deviations from the prediction, the model rather than the heat 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 $34.6 - \mathbf{h}_1^\mathsf{T} \boldsymbol{\mu}'_u = 0.097$, and

$$\boldsymbol{\Sigma}'_u \mathbf{h}_1 = \begin{pmatrix} 1.143 \\ -0.257 \end{pmatrix}, \quad s = \mathbf{h}_1^\mathsf{T} \boldsymbol{\Sigma}'_u \mathbf{h}_1 + \sigma^2 = 0.543 + 0.25 = 0.793, \quad \mathbf{k} = \frac{\boldsymbol{\Sigma}'_u \mathbf{h}_1}{s} = \begin{pmatrix} 1.441 \\ -0.324 \end{pmatrix},$$

$$\boldsymbol{\Sigma}''_u = \boldsymbol{\Sigma}'_u - \mathbf{k}\,(\boldsymbol{\Sigma}'_u \mathbf{h}_1)^\mathsf{T} = \begin{pmatrix} 2.162 & -1.153 \\ -1.153 & 0.726 \end{pmatrix}.$$

The gain on the coolant temperature is negative although $\mathbf{h}_1$ has a positive entry for it. The first reading left the heat and the coolant temperature correlated, so a module temperature above its prediction is blamed on the heat, and the correlation pulls the coolant temperature down. The mean moves by $\mathbf{k}\,e = (0.140, -0.032)$, 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 heat is now 20.75 ± 1.47 watts, the coolant temperature 20.04 ± 0.85 degrees. The second sensor's lever arm on the heat is only the difference $T_1 - T_2 = G/k$, a slope of $1/k = 0.2$ kelvin per watt against two readings each accurate to half a kelvin, which is why the heat improves by a quarter and not more. The correlation is now -0.920: two readings of temperature, both of which are "coolant temperature plus something times heat", cannot fully separate the two. A sensor on the heat flow itself, or on the coolant, 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 ring bus with a shunt

The same operation on the electrical example shows what a loop does to it. The model is the one of Example 1.2, with $g = 10$ amperes per millivolt and the physics exact, and one change: the two rack currents are no longer known. They carry the priors $q_2 \sim \mathcal{N}(-60, 8^2)$ and $q_3 \sim \mathcal{N}(-20, 8^2)$ amperes, independent and centred on the second set of rack currents Example 1.2 used. Every number is from `ch03_triangle_posterior.py`.

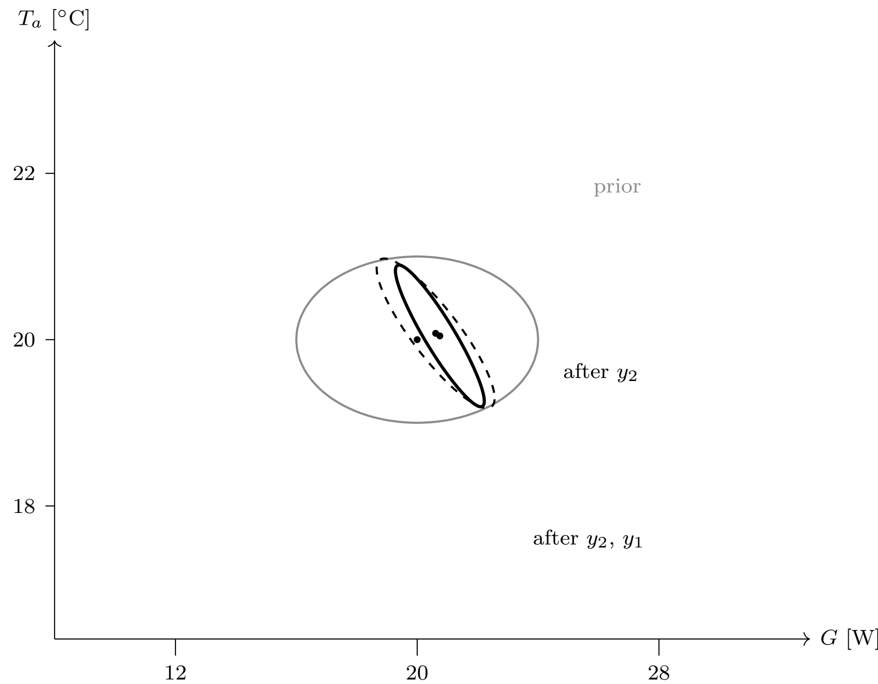


Figure 3.3: One-standard-deviation ellipses of the joint distribution of the heat and the coolant temperature: the prior (grey, axis-aligned, because the two are independent), the posterior after the plate reading (dashed, a ridge along $T_a + 0.5G = 30.4$), and the posterior after both readings (solid, shorter along the ridge). Dots mark the means. Generated by the same script as Table 3.2.

The example in vectors. As for the heat path, the unknown vector is the one of Example 1.2, currents first, with the rack currents appended:

$$\mathbf{z} = \begin{pmatrix} I_{12} \\ I_{13} \\ I_{23} \\ v_2 \\ v_3 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \mathbf{i} \\ \mathbf{v} \\ \mathbf{u} \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} q_2 \\ q_3 \end{pmatrix}.$$

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

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

and the boundary flow, $I_{12} + I_{13}$, is $-(q_2 + q_3)$. Under the prior the flows are 1.167 ± 0.224, 0.833 ± 0.224 and -13.33 ± 3.77, and the boundary current 80.0 ± 11.3. A shunt on line 1–2

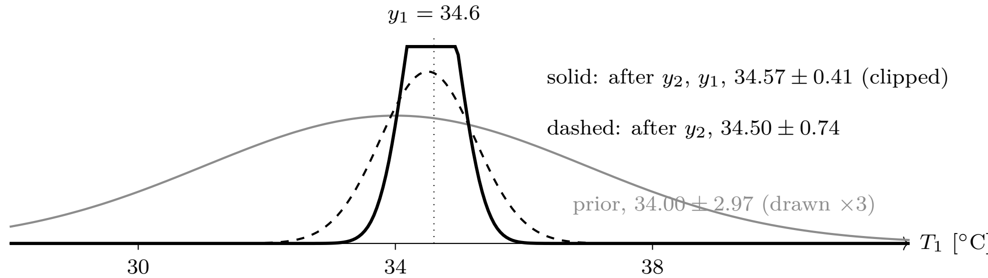


Figure 3.4: The predictive density of the unmeasured module temperature T_1 under the prior (grey), after the plate reading y_2 (dashed), and after both readings (solid, clipped in height). The dotted line is the reading $y_1 = 34.6$ when it arrives: inside the dashed prediction's range, so the model is not surprised.

Table 3.3: The ring bus: prior, posterior after the shunt on cable 1–2, posterior after a second meter on the boundary current. Rack currents and cable currents in amperes, means and standard deviations.

	prior	after $I_{12} = 47.0$	after I_{12} and boundary 81.0
q_2	-60.00 ± 8.00	-60.39 ± 3.70	-60.09 ± 2.29
q_3	-20.00 ± 8.00	-20.20 ± 7.17	-20.89 ± 2.61
$\mathrm{corr}(q_2, q_3)$	0	-0.976	-0.961
I_{12}	46.67 ± 5.96	46.99 ± 0.79	47.02 ± 0.74
I_{13}	33.33 ± 5.96	33.60 ± 3.63	33.96 ± 1.03
I_{23}	-13.33 ± 3.77	-13.40 ± 3.58	-13.07 ± 1.62
boundary current	80.00 ± 11.31	80.59 ± 3.85	80.98 ± 0.78

reads the first entry of $\mathbf{z}$: on $\mathbf{z}$ it is $\mathbf{h} = \mathbf{e}_1$, and on the rack currents it is the first row of $\mathbf{S}$, $(-\frac{2}{3}, -\frac{1}{3})^\mathsf{T}$. It reads $I_{12} = 47.0$ 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 rack currents, $\mathbf{S}$ carries it to the flows, Table 3.3 lists both, and Figure 3.5 draws the rack currents. The shunt reads the direction $(-0.894, -0.447)$ of the rack-current plane, mostly q_2; along it the spread falls from 8.00 to 1.06, and across it, in the direction $(0.447, -0.894)$, it stays at 8.00 exactly. The rack currents become $q_2 = -1.421 \pm 0.136$ and $q_3 = -0.460 \pm 0.269$, correlated at -0.947: the shunt has fixed two thirds of one plus one third of the other, so if one is lower the other must be higher. The unmetered currents move too, through the loop: I_{13} from 0.833 ± 0.224 to 0.780 ± 0.135, I_{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 shunt on one cable has narrowed the total draw by a factor of three, because the total is mostly what the shunt reads.

A second meter, on the boundary current at tap 1, reads 81.0 ± 0.8. The boundary flow is not 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 rack currents 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 rack currents: $q_2 = -60.09 \pm 2.29$, $q_3 = -20.89 \pm 2.61$, the current on the cable between them -13.07 ± 1.62. The prior had put q_3 at -20; the two meters have moved it by a tenth of a prior spread, to -20.9, because the boundary meter says the total is 81

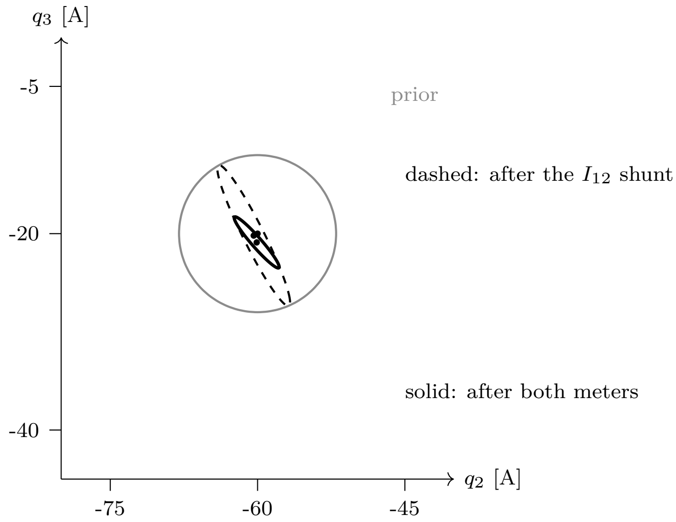


Figure 3.5: One-standard-deviation ellipses of the two rack currents of the ring bus: the prior (grey), the posterior after the shunt on cable 1–2 (dashed), and the posterior after a second meter on the boundary current (solid). The first meter reads one direction of the rack-current plane and flattens the ellipse along it only; the second reads a different direction and closes it.

and the shunt says most of it is not at tap 3. Neither meter reads q_3; both inform it through the loop. This is the meshed twin of the heat path's virtual sensor, and the loop is what makes it work: in a radial network a meter on one path says nothing about the draw 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 T_2. The rack current G is three squares away: through c_{2a} to Q_{2a}, 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 four watts wide. It reached T_a through one exact factor and found a prior one degree wide. It moved what was loosest.

The two priors π_G and π_{T_a} share no factor, and under the prior their variables are independent. After the reading they are correlated at -0.868. The reading is a factor on T_2, and T_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 $\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma}\mathbf{h}\mathbf{h}^{\mathsf{T}}\mathbf{\Sigma}/(a^2s_1^2 + b^2s_2^2 + \sigma^2)$, whose off-diagonal entry is $-abs_1^2s_2^2/D$ and whose diagonal entries are $s_1^2(b^2s_2^2+\sigma^2)/D$ and $s_2^2(a^2s_1^2+\sigma^2)/D$, 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 heat path, $a = 0.5$, $s_1 = 4$, $b = 1$, $s_2 = 1$, $\sigma = 0.5$: $\rho' = -0.5 \cdot 4 \cdot 1/\sqrt{(4.25)(1.25)} = -0.868$, the table's value. The sensor resolves $T_a + 0.5G$ 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 heat path's two temperatures left G and T_a at -0.98 and the triangle's two meters left the rack currents at -0.95. 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 [89] gives the mathematical framework and Kaipio and Somersalo [39] 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 [83] and developed in the books of Monticelli [66] and Abur and Gómez Expósito [1]; in chemical process engineering it is data reconciliation, of which Mah, Stanley and Downing [59] 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 [15]. 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 [58], 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 [72], 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 temperature reading a tap away from an uncertain rack current cut its spread by a factor of three and a half and predicted an unmeasured temperature to ± 0.74 kelvin. The second reading landed inside that prediction.
- On the ring bus, one shunt narrowed the total draw by a factor of three and moved an unmetered current 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 T_1 read first and the sensor on T_2 second. Show that the final posterior is the same and that the intermediate one is different. Which single sensor, on T_1 or on T_2, tells more about G, and why?

Exercise 3.2. Add a third sensor to the worked example that reads the coolant coolant temperature directly, first with accuracy 1.0 kelvin and then with 0.3. Compute the posterior in each case and show that the correlation between G and T_a 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 conductance k with the prior $\mathcal{N}(5, 1^2)$ and keep everything else. The map from inputs to T_1 is now nonlinear in (G, k). 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 heat path a width σ_n and suppose the two sensors read $y_2 = 30.4$ and $y_1 = 39.0$, 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 tap 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 $y_1 = 34.6$ first gives $G = 20.73 \pm 1.48$ and $T_a = 20.05 \pm 0.87$, correlated at -0.926; reading y_2 second gives $G = 20.75 \pm 1.47$, $T_a = 20.04 \pm 0.85$, 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 T_1 alone leaves G at ± 1.48, the sensor on T_2 alone at ± 1.95. The reason is the slope: $T_1 = T_a + 0.7G$ rises seven tenths of a kelvin for each watt and $T_2 = T_a + 0.5G$ five tenths, so at equal accuracy the T_1 sensor has the larger lever arm on the rack current. Both sensors share the same unit slope on T_a, which is why neither separates the two.

Exercise 3.2. With the two temperature sensors read, a coolant sensor of accuracy 1.0 kelvin reading 19.5 gives $G = 21.11 \pm 1.18$, $T_a = 19.82 \pm 0.65$, and a correlation of -0.873; with accuracy 0.3 it gives $G = 21.52 \pm 0.73$, $T_a = 19.56 \pm 0.28$, and -0.636. The first two sensors both read "coolant temperature plus a multiple of the rack current" and so both lie along the ridge $T_a + cG$; in Figure 3.1 they attach to T_2 and T_1, which are tied to T_a and G through the same chain of factors, and the ridge is what that chain cannot distinguish. The busbar sensor attaches to T_a alone, reads the direction $(0, 1)$ of the (G, T_a) 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 1.0 and mostly crossed at 0.3; a sensor on G itself, or on a heat flow, would cross it at right potentials.

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 module heat path 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

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

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 T_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 [23, 24]: 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 ring bus 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

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

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 $\mathbf{u} = (G, T_a)$; 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 heat path the map is explicit. With inputs $\mathbf{u} = (G, T_a)$ and solved variables $\mathbf{x} = (Q_{12}, Q_{2a}, T_1, T_2)$, flows first as in Example 1.1,

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

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 rack current, and the two temperatures 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 $X(\mathbf{u})$,

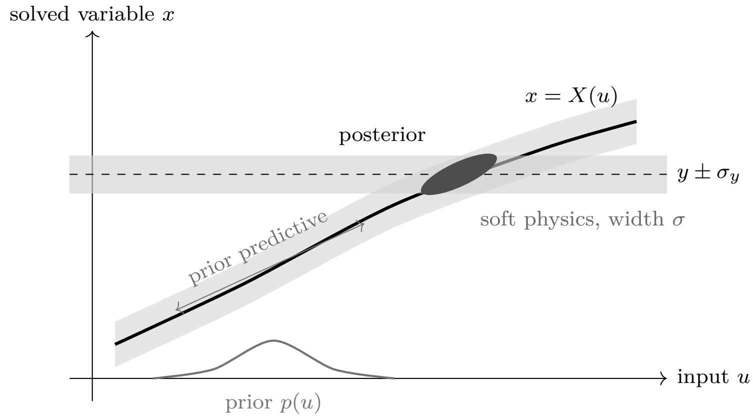


Figure 4.1: One input, one solved variable. The thick curve is the constraint manifold $x = X(u)$; the hard joint lives on it. The grey band around the curve is where the soft joint puts its mass, a distance of order σ from the curve. The prior on u sits on the horizontal axis and is carried up the curve 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 $\mathbf{h}_k\mathbf{h}_k^\top/\sigma^2$ for the residual $r_k = \mathbf{h}_k^\top\mathbf{z}$, 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 $\mathbf{h}_k\mathbf{h}_k^\top/\sigma^2$ 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.

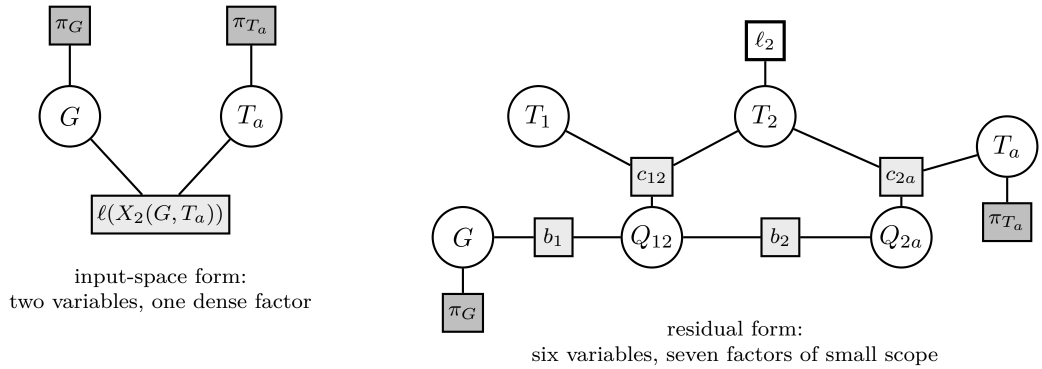


Figure 4.2: The same posterior in two forms. Left: the input-space form of §4.2, in which the physics has been solved and the likelihood is one factor reading every input. Right: the residual form of the soft joint, in which the physics rows remain as factors of small scope and the sensor reads one variable. 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	0.5 K thermistor; 0.8 A shunt
prior	spread of the input	data sheet, forecast, history	module heat ±4 W; rack current ±8 A; coolant ±1 K
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 conductance 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 conductance, 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 temperature, 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 [35]: 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 heat path, 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 $\mathbf{h}_k^\top \mathbf{z} - c_k \sim \mathcal{N}(0, \sigma_k^2)$ 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 $\mathbf{a}_k^\top$. The posterior mean minimizes $\|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2$, so it is the least-squares solution $\hat{\mathbf{z}} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$, and the standardized residual vector is $\mathbf{b} - \mathbf{A}\hat{\mathbf{z}} = (\mathbf{I} - \mathbf{P})\mathbf{b}$ with $\mathbf{P} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$ 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 $\boldsymbol{\epsilon}^\top(\mathbf{I} - \mathbf{P})\boldsymbol{\epsilon}$, 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. $\square$

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 heat path with soft physics

Take the module heat path of Figure 3.1 with all six unknowns, $\mathbf{z} = (Q_{12}, Q_{2a}, T_1, T_2, G, T_a)$, the priors $G \sim \mathcal{N}(20, 4^2)$ watts and $T_a \sim \mathcal{N}(20, 1^2)$ degrees, the conductances $k = 5$ and $hA = 2$ watts per kelvin known, and the four physics rows each given the same width σ, in watts. 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 $y_2 = 30.4$ of §3.4, Table 4.2 sweeps the width from 1 watt down to 10^{-4}. At $\sigma = 1$ watt, a twentieth of the flow, the physics is loose: the heat's posterior spread is 2.31 watts, the correlation with the coolant temperature is only -0.63, and the largest physics residual at the posterior mean is three hundredths of a watt. The reading has been partly absorbed by bending the physics rather than by moving the inputs. At $\sigma = 10^{-1}$

Table 4.2: The heat path with soft physics, one sensor $y_2 = 30.4$: posterior of the heat and the coolant temperature as the physics width shrinks. The last column is the condition number of the posterior precision matrix. Hard-physics reference from Table 3.2: $G = 20.61 \pm 1.95$ watts, T_a to ± 0.90 kelvin, correlation -0.868. Widths and heats in watts, temperatures in kelvin.

σ	mean G	sd G	sd T_a	corr	max $\lvert r_k \rvert$	cond
1	20.533	2.309	0.913	-0.632	3.3×10^{-2}	8.0×10^{2}
10^{-1}	20.609	1.956	0.900	-0.864	3.8×10^{-4}	6.6×10^{4}
10^{-2}	20.610	1.952	0.900	-0.868	3.8×10^{-6}	6.6×10^{6}
10^{-3}	20.610	1.952	0.900	-0.868	3.8×10^{-8}	6.6×10^{8}
10^{-4}	20.610	1.952	0.900	-0.868	3.8×10^{-10}	6.6×10^{10}

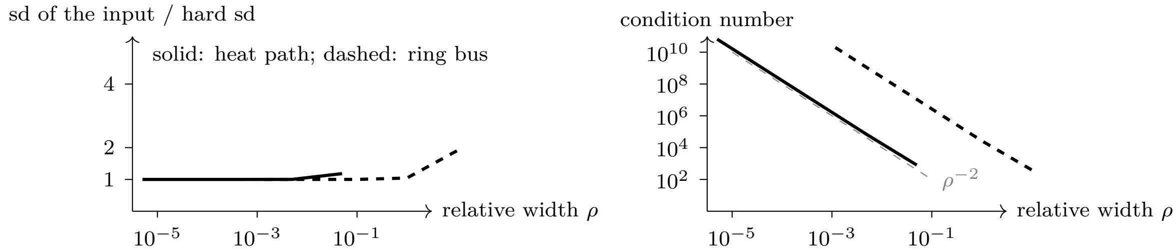


Figure 4.3: The width sweep on the heat path (solid) and on the ring bus of §4.7 (dashed), against the relative width, the physics width divided by the scale of the flows (20 watts for the heat path, eighty amperes for the ring bus). Left: the posterior spread of the uncertain input relative to its hard-physics 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 percent of the hard values. At $\sigma = 10^{-2}$ and below they agree with Chapter 3's hard posterior to every digit shown, the residuals are below 10^{-5}, 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^{-4} watts, which is a relative width of 5×10^{-6} against flows of twenty watts, the condition number is 7×10^{10} on a six-variable model; a larger model with the same relative width would be past the edge. A width of 10^{-2} to 10^{-1} watts gives the hard answer at a condition number a solver does not notice.

An inconsistent reading. Now keep every width at 10^{-5} watts and read both sensors, $y_2 = 30.4$ as before and $y_1 = 39.0$. In the exact model $T_1 - T_2 = G/k$, so the two readings imply $G = 43$ watts; but then $Q_{2a} = 43$ and $T_2 = T_a + 21.5$, which with $T_2 = 30.4$ puts the coolant at 8.9 degrees, eleven 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: heat 27.1 watts, coolant temperature 0.9804, the T_2 sensor missed by two thousandths and the T_1 sensor by one and a half. Every physics residual is zero. The model has kept its laws and thrown away its priors and its sensors: the coolant

Table 4.3: Inconsistent readings $y_2 = 30.4$, $y_1 = 39.0$: posterior as the width of the tap-2 balance is loosened while every other row stays at 10^{-3} watts. The leak estimate is the violation of that balance at the posterior mean, in watts. Last column: misfit at the posterior; about 2 is expected of an adequate model.

σ_{n_2}	draw $[10^{-3}]$	G	T_a	T_2, T_1 fitted	misfit
10^{-3}	0.00 ± 0.00	27.09 ± 1.47	18.62 ± 0.85	32.16, 37.58	25.5
10^{-1}	0.01 ± 0.10	27.10 ± 1.47	18.62 ± 0.85	32.16, 37.58	25.5
1	0.65 ± 0.97	27.47 ± 1.57	18.71 ± 0.86	32.12, 37.61	25.1
10	8.81 ± 3.57	32.17 ± 2.53	19.82 ± 0.98	31.51, 37.94	19.5

temperature 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 tap 2: the tap may be drawing heat 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^{-1} watts nothing changes; the row is still too stiff to absorb watts of heat. At 1 watt the posterior has begun to move: the balance is violated by 0.65, the coolant temperature has come back to 0.9931, the sensors are fitted to within a thousandth. At 10^{-1} the posterior is coherent: the balance at the plate is violated by 8.8 ± 3.6 watts, the heat is 32.2, the coolant temperature 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 rack current 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 tap 2 said heat in equals heat out. At the posterior, in exceeds out by nearly nine watts. That is heat leaving the plate by a path the model does not contain, and the model has estimated it, with an error bar, from two temperature 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, $y_2 = 30.4$ and $y_1 = 34.6$, through the same soft model with every width at 10^{-3} watts, give $G = 20.75 \pm 1.47$ and $T_a = 20.04 \pm 0.85$, identical to the hard result, with a misfit of 0.04. 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 ring bus with an unmodelled current

The ring bus of §3.5 carries the same lessons into the charge domain, where the violation is a current the graph does not show. Keep all seven unknowns of §3.5, currents first: the three cable currents, the two potentials and the two rack currents. Give the five physics rows a common width σ in amperes, the rack currents their priors -60 ± 8 and -20 ± 8, and read meters of accuracy 0.8 amperes. Every number is from `ch04_triangle_soft.py`.

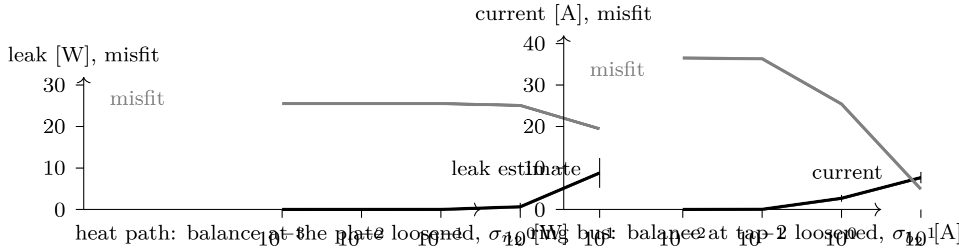


Figure 4.4: Loosening one balance row against inconsistent readings. Left: the heat path, the balance at the plate loosened; the leak estimate with its error bar (black) rises from nothing to 8.8 ± 3.6 watts as the width passes one watt, and the misfit (grey) falls from 25 toward the value an adequate model would give. Right: the ring bus of §4.7, the balance at tap 2 loosened; the unmodelled current rises to 7.7 amperes and the misfit falls from 37 to 4.9. In both, the row that is allowed to be violated is the row that reports the defect.

Table 4.4: The ring bus with soft physics, one shunt $I_{12} = 47.0$: posterior of the rack currents as the physics width shrinks, all in amperes. Hard reference from Table 3.3: $q_2 = -60.39 \pm 3.70$, $q_3 = -20.20 \pm 7.17$, correlation -0.947.

σ []	q_2	q_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 shunt $I_{12} = 47.0$ of Chapter 3, Table 4.4 sweeps the width from 10 amperes, an eighth of the boundary current, down to 10^{-3}. At 10 the posterior is loose, the correlation between the rack currents only -0.13, and the physics residuals are of order 10^{-1}: the shunt has been partly absorbed by bending the loop rather than by moving the rack currents. At 1 ampere they are within a percent of the hard values and at 0.1 they agree to every digit shown, with the correlation -0.947 of Table 3.3. The condition number grows by a hundred per decade, exactly as on the heat path, 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 cable 1–2 at 47.0, the current on cable 2–3 at -5.0, and the rack current at tap 2 itself, from the rack's own meter, at -60.0. The balance at tap 2 says that what arrives on the two lines is what the rack draws, $I_{12} - I_{23} = -q_2$; the meters say 52.0 on the left and 60.0 on the right. No state of the exact model satisfies all 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 cable currents are fitted at 49.49 and -7.90, three accuracies from their readings, the rack current at -57.40, three from its meter, and the misfit is 37 where an adequate model with ten factors and seven unknowns would give about three. Loosen the balance at tap 2 and the posterior finds the explanation: at a width of 0.1 the balance is violated by 2.7 ± 0.8 amperes and the misfit is 25; at a width of ten it is violated by 7.7 ± 1.4, every meter

Table 4.5: Inconsistent meters on the ring bus, $I_{12} = 47.0$, $I_{23} = -5.0$, $q_2 = -60.0$, each to 0.8 A: posterior as the width of the tap-2 balance is loosened while every other row stays at 0.001. The unmodelled current is the violation of that balance at the posterior mean. About 3 is expected of an adequate model.

σ_{b_2}	unmodelled current	q_2	q_3	I_{12}, I_{23} fitted	misfit
0.01	0.00 ± 0.01	-57.40 ± 0.65	-33.69 ± 1.69	49.49, −7.90	36.5
0.1	0.04 ± 0.10	-57.41 ± 0.65	-33.70 ± 1.69	49.48, −7.89	36.3
1	2.70 ± 0.81	-58.29 ± 0.70	-34.55 ± 1.71	48.58, −7.02	25.4
10	7.69 ± 1.37	-59.95 ± 0.79	-36.14 ± 1.74	46.89, −5.37	4.9

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
heat path, consistent readings, tight physics	8, 6	2 ± 2	0.07
heat path, inconsistent readings, tight physics	8, 6	2 ± 2	74
heat path, inconsistent, tap-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, tap-2 balance at 1	10, 7	3 ± 2.4	0.5

is fitted to within an accuracy, and the misfit is 0.5. The violation is a current of eight amperes at tap 2 that the graph does not show, a path on a heat path 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 heat path's.

Table 4.6 sets every misfit of the chapter beside its expectation. The consistent heat path 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 heat path, at 10.6 against 2 ± 2, sits four deviations above, which is the residual doubt the text described.

Two differences from the heat path are instructive. The ring bus's unmodelled current is found at a width of one ampere, a hundredth of the boundary current, whereas the heat path's needed ten watts, half a whole flow; the reason is that the ring bus's three meters pin the violation from both sides, while the heat path's two temperature sensors see the leak only through the prior on the heat. And the ring bus's misfit falls to 4.9, near its expected value, while the heat path's stops at 19.5; the ring bus's data are explained almost entirely by one unmodelled current, and the heat path's are not, which is the signal that the heat path 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 heat path's data were explained by a draw at tap 2 and not by anything at tap 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 [67] 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 [35].

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-rack current taps are handled either as hard constraints or as pseudo-measurements of very high weight [1, 66], and of process data reconciliation, where the residual of a balance is the signature of a leak or a bad meter [59]. 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 [39, 89]: 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 heat path, widths of 10^{-2} to 10^{-3} watts reproduce the hard posterior exactly; inconsistent readings through tight physics produce an absurd state with a misfit of 25; loosening one balance turns the inconsistency into an unmodelled leak of 8.8 ± 3.6 watts and a misfit of 19.5.
- On the ring bus the same sweep converges at a relative width of 10^{-3}, and three inconsistent meters become an unmodelled current of 7.7 ± 1.4 amperes at tap 2 with the misfit falling from 37 to 4.9. 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 module heat path 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} watts 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 tap 1 instead of tap 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 $Q - k(T_1 - T_2)$ solved for T_1 with k 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 $\mathbf{h}^\mathsf{T}\mathbf{\Lambda}^{-1}\mathbf{h} \leq \mathbf{h}^\mathsf{T}\mathbf{\Lambda}_1^{-1}\mathbf{h}$, 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 node 1 to 10 watts, with every other row at 10^{-3}, gives a violation of -6.9 ± 4.0 watts, a source of 21.1 ± 3.7, a coolant temperature of 18.12 ± 0.90, and a misfit of 22.5; at one watt the violation is -0.4, and at 10^{-1} nothing has moved from the tight solution. The violation's sign says that seven watts more than the cells make is leaving the module: a second, hidden source inside it. But the posterior has not been helped: the coolant temperature is still nearly two degrees below its setpoint, almost two 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 $T_1 - T_2$ near 0.0196 and hence $Q_{12} = 0.098$ whatever balance is loosened; a draw at tap 2 lets Q_{2a} be less than Q_{12}, so that $T_2 - T_a$ can be 0.027 instead of 0.049 and the coolant temperature can return to its set point, while a defect at tap 1 changes only where Q_{12} came from and leaves $Q_{2a} = Q_{12}$, so the coolant temperature must still sit near 0.980. The data decide: the tap-2 explanation reaches a misfit of 10.6 and the tap-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 leak is the whole story, since 19.5 is still above the expected 2; that residual doubt is about the prior on the heat, and only a meter on the module's current or a better duty schedule can remove it.

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 heat path 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.

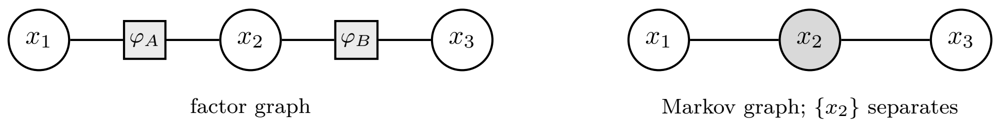


Figure 5.1: The three-variable chain. Left, as a factor graph. Right, its Markov graph: variables adjacent when they share a factor. Removing x_2 disconnects x_1 from x_3, so $x_1 \perp x_3 \mid x_2$. Nothing disconnects them otherwise, so in general $x_1 \not\perp x_3$.

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.

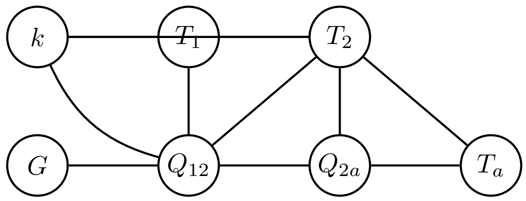
Definition 5.1 (Separator)**.** A set of variables S *separates* two disjoint sets A and B in the Markov graph if every path from a variable in A to a variable in B passes through some variable in S. Equivalently, removing the nodes of S leaves A and B in different connected components.

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.



$\{Q_{12}, Q_{2a}\}$ separates G from T_a
$\{T_1, T_2, Q_{12}\}$ separates k from T_a
no set of size one separates G from T_a

Figure 5.2: The Markov graph of the soft module heat path 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 rack current, the closures join each flow to the temperatures at its ends and, for c_{12}, to the conductance. The path from G to T_a 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 rack current G and the coolant temperature T_a, independent under the prior, were correlated at -0.985 after the reading of T_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 rack current is adjacent to Q_{12} through the balance n_1; the coolant temperature is adjacent to Q_{2a} and T_2 through the closure c_{2a}; and Q_{12}, Q_{2a}, T_2 are all adjacent to one another through n_2 and c_{12}. There is a path from G to T_a, 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 $G \to T_2 \leftarrow T_a$, 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 $T_a + 0.5\,G$, 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 rack current is paid for by a lower coolant temperature.

> **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.*

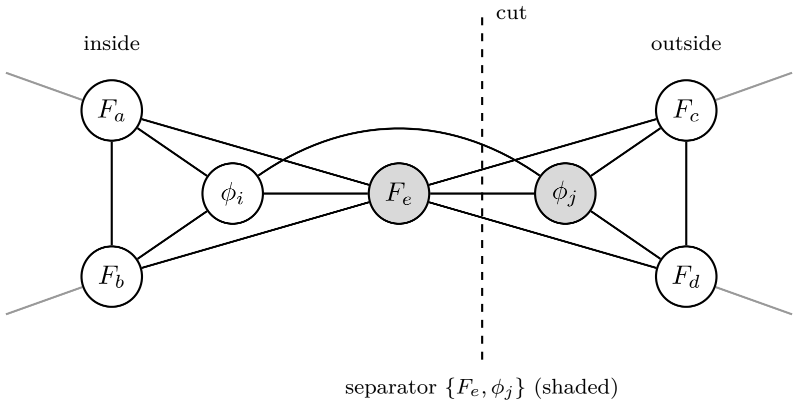


Figure 5.3: The Markov graph around one cut edge e from inside node i to outside node j. The closure c_e makes $\{F_e, \phi_i, \phi_j\}$ a clique; the balance at i joins F_e to the other inside flows F_a, F_b; the balance at j joins it to the outside flows F_c, F_d. Removing the shaded pair $\{F_e, \phi_j\}$ 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,

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

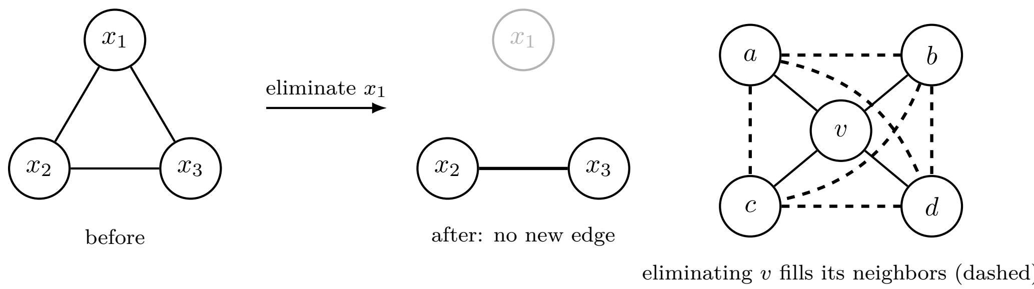


Figure 5.4: Elimination and fill. Left: eliminating x_1 from the triangle creates a factor on $\{x_2, x_3\}$, but they were already adjacent, so the graph gains no edge. Right: eliminating a variable with four neighbors that were not adjacent to each other creates six fill edges; the neighbors become 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 ring bus'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 module heat path 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 conductance 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 heat path 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 $O(\sqrt{n})$ [55], so their treewidth grows no faster than the square root of the number of taps 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 $n^{2/3}$ [63], 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 ring bus 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 potentials 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
potentials first	12	3	2
I_{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 ring bus 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 $v_2v_3I_{23}$ 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 potentials joined by the edge that c_{23} had already drawn. The twelve orders that eliminate the two potentials first all have width three and create two fill edges, I_{12} to v_3 and then I_{12} to I_{13}, joining flows that no factor read together. Flows first is the better family, but not uniformly: within it, eliminating I_{23} first, the flow on the loop, costs width four.

The six-tap 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 potentials 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 taps; eliminating the potentials 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 potential is not the cheapest *route*, and a heuristic that interleaves them is. The same table checks Proposition 5.3: removing the tie flow I_{34} and the potential v_4 disconnects the left triangle from the right, and removing either alone, or the two potentials at the tie's ends without the flow, does not.

Lemma 5.1 brackets these numbers. The six-tap 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;

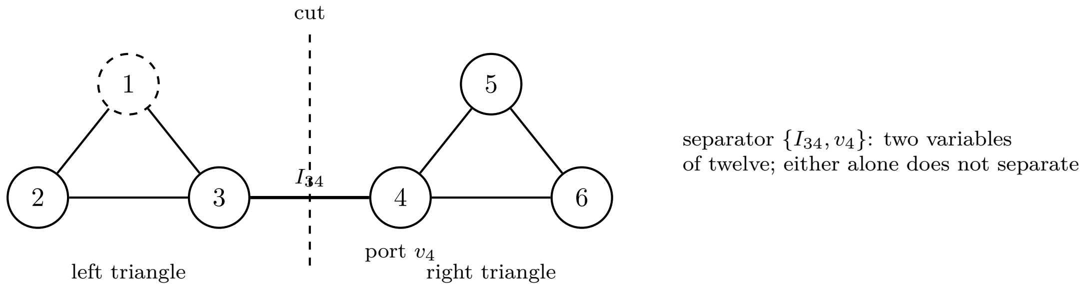


Figure 5.5: The six-tap network cut at its tie. The port pair of the cut edge, the tie flow and the potential on the far side, separates the two triangles in the row-per-factor Markov graph; the script confirms that neither variable alone does, nor the two end potentials without the flow.

Table 5.2: Elimination on the six-tap 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 potentials	6	14
all potentials first, then flows	8	26
minimum fill (interleaved)	4	3
removed	**left and right triangles**	
$\{I_{34}, v_4\}$	separated	
$\{I_{34}\}$ or $\{v_4\}$ alone	connected	
$\{v_3, v_4\}$	connected	

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

The two-ledger module. The two-domain model of Example 1.3 has a Markov graph of seventeen variables and fifty edges with treewidth at most seven by the heuristics and at least four by its largest clique, which has five variables — the plate's row, which reads all four cell heats and the plate temperature. Its interesting separator is not a physical cut but a ledger cut, and it is as small as a separator can be: remove the series current I, one variable, and the seven drops are disconnected from every heat and every temperature. Given the current, the charge ledger and the heat ledger are conditionally independent, which is the probabilistic statement of what Chapter 2's Figure 2.4 showed structurally: the ledgers couple only through the current that every heat source reads. A temperature measurement informs a drop only by way of what it says about the current. The minimum separator between one particular drop and one particular cell temperature is that same single variable, $\{I\}$ between Δv_{c_1} and T_{c_4}, which is the physical statement that what the coolant pushes into the ring reaches the far end's potential only through tap h.

The ring, around and across. A ring of twelve taps, each tied to its two neighbors, has treewidth two: eliminate the taps in order around the ring and every elimination joins the two neighbors of the eliminated tap, 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

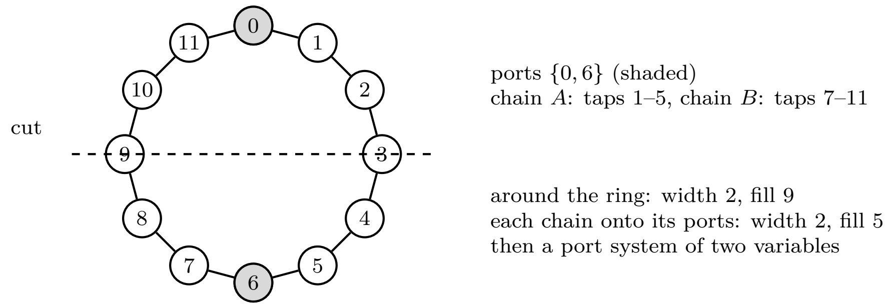


Figure 5.6: A ring of twelve taps cut at two opposite taps into two chains. Eliminating each chain's interior onto the shaded ports costs the same width as going around the ring, and leaves a two-variable port system that the chains share; the gain is independence and reuse, not width.

ring at two opposite taps, which are then the ports of two chains of five taps each; eliminate 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 taps 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-tap network and the two-ledger module sit far below the curve at their sizes. A uniform mesh is the hardest two-dimensional topology there is for elimination, because every tap 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 taps 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,

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

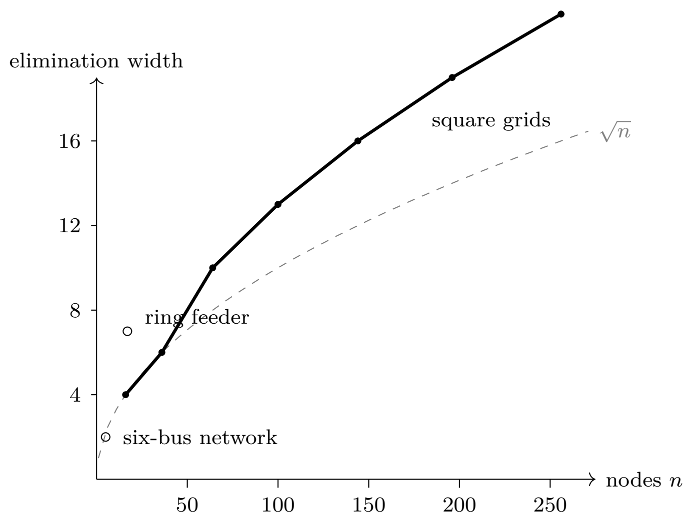


Figure 5.7: Elimination width of square grids of n nodes (filled points, minimum-fill ordering) against $\sqrt{n}$ (dashed). The six-tap network and the two-ledger module are marked at their sizes. A grid's width grows with the square root of its size; the networks' widths are set by their ports.

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.

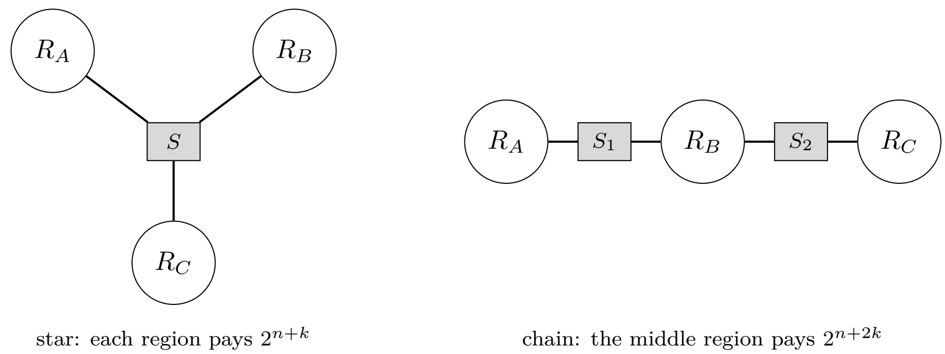


Figure 5.8: Three regions joined at one separator (left) and in a chain through two (right). In the chain the middle region's message must be computed jointly over both separators it touches, and its cost is exponential in their total size.

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 [51] and Koller and Friedman [47]; the theorem's first published proof is Besag's [8]. Explaining away, colliders and the d-separation rule that §5.3 paraphrases are Pearl's [72]. 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 [79]; that minimizing fill is NP-complete is Yannakakis's result [101], 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 [52]. Nested dissection, the strategy of §5.5's last paragraph, is George's [26]; the separator theorems that bound treewidth for planar and finite-element graphs are Lipton and Tarjan's [55] and Miller, Teng, Thurston and Vavasis's [63]. Davis [17] 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 [41, 71, 95], 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 [49]; 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 [34] 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 [18], 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-tap 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 two-ledger module's two ledgers are separated by one variable, the series current.
- 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 heat path of Figure 3.1 with all six variables and the sensor on T_2. Find the smallest separator between G and T_a, and the smallest between k and T_a. Verify by Proposition 5.1 that $G \perp T_a \mid \{Q_{12}, Q_{2a}\}$, and explain physically why knowing both flows makes the rack current and the coolant temperature 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 ring bus 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 taps, 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 taps 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: I_{12}, I_{13}, I_{23}, v_2, v_3. Eliminating I_{12} joins its neighbors v_2 (from c_{12}) and I_{23} (from b_2); they were not adjacent, so one fill edge appears and the front had two variables. Eliminating I_{13} likewise joins v_3 to I_{23}, a second fill edge, front of two. Eliminating

I_{23} now has neighbors v_2 and v_3, already adjacent through c_{23}: front of two, no fill. The two potentials remain, joined, and the order's width is two. No order of width one exists because the Markov graph contains the triangle v_2, v_3, I_{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 I_{23} first is worst, since its four neighbors v_2, v_3, I_{12}, I_{13} become a clique with three fill edges and a front of four; eliminating it last, after I_{12} and I_{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 potential-to-potential 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.

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 module heat path 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

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

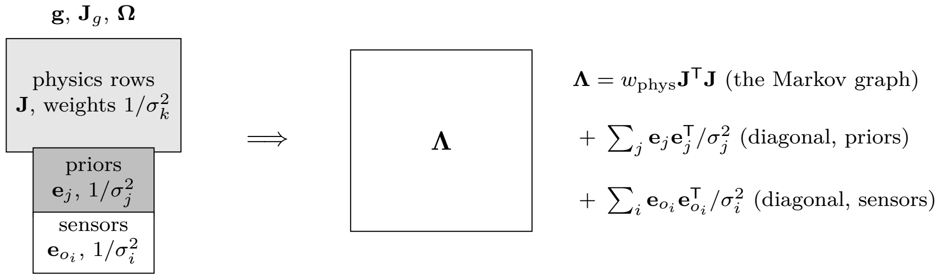


Figure 6.1: Assembling the precision. The three blocks of the stacked residual contribute three terms to $\boldsymbol{\Lambda}$: the physics rows give the sparse matrix whose pattern is the Markov graph, and priors and sensors add to the diagonal only. Removing the last two blocks and letting the physics 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)**.** $\boldsymbol{\Lambda}_{ij} \neq 0$ *only if some row of* $\mathbf{g}$ *reads both* z_i *and* z_j*. The off-diagonal sparsity pattern of* $\boldsymbol{\Lambda}$ *is the Markov graph of the factor graph.*

The proof is that $(\mathbf{J}_g^\mathsf{T}\boldsymbol{\Omega}\mathbf{J}_g)_{ij} = \sum_k w_k(\mathbf{J}_g)_{ki}(\mathbf{J}_g)_{kj}$, and a term is nonzero only when row k has nonzeros in both columns. Two consequences follow at once. The number of nonzeros of $\boldsymbol{\Lambda}$ 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 $\boldsymbol{\Lambda}$ verbatim, because the Markov graph is $\boldsymbol{\Lambda}$'s graph.

6.2 The most probable state

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

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

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

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

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:

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

This is the Gauss–Newton iteration. Because $\boldsymbol{\Lambda}$ is positive definite at a well-posed problem, $\boldsymbol{\delta}$ is a descent direction for C, and a step length $t \leq 1$ 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 ring bus. 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 $\boldsymbol{\Lambda}_{uu} - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\Lambda}_{xu}$ is the *Schur complement* of $\boldsymbol{\Lambda}_{xx}$. Its sparsity is that of $\boldsymbol{\Lambda}_{uu}$ 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 $\boldsymbol{\eta}_u - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\eta}_x$ as information vector. The Schur complement is positive definite when $\boldsymbol{\Lambda}$ is.*

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

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

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

Eliminating one variable at a time, in some order, is exactly what the Cholesky factorization $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^\top$ 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 $\boldsymbol{\Lambda}$ are the fill edges, the order of elimination is the order of the pivots, and the treewidth of $\boldsymbol{\Lambda}$'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 pack whose cells are wired in a mesh, 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 $\mathbf{L}^\top\mathbf{z}^\star = \mathbf{v}$;
- the marginal variance of z_k is $(\boldsymbol{\Lambda}^{-1})_{kk}$, and it is obtained by solving $\boldsymbol{\Lambda}\mathbf{x} = \mathbf{e}_k$ with the same two triangular solves and reading x_k, never by forming $\boldsymbol{\Lambda}^{-1}$, 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 heat path and the ring bus 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
heat path, 6 variables	inputs first	14	25
	flows first	20	50
ring bus, 7 variables	minimum degree	20	42
	cable currents 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, $\boldsymbol{\Lambda}(\mathbf{z}^\star) = \mathbf{J}_g(\mathbf{z}^\star)^\mathsf{T}\boldsymbol{\Omega}\mathbf{J}_g(\mathbf{z}^\star)$, is the curvature of C at the mode. The *Laplace approximation* replaces the posterior by the Gaussian with that mode and that curvature,

$$p(\mathbf{z} \mid \mathbf{y}) \;\approx\; \mathcal{N}(\mathbf{z}^\star,\; \boldsymbol{\Lambda}(\mathbf{z}^\star)^{-1}). \tag{6.9}$$

It is exact when every row is linear, since then $\boldsymbol{\Lambda}$ 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 heat path in matrix form

Take the soft module heat path of Chapter 4 with both sensors: six unknowns, four physics rows at width 10^{-3} watts, 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 $\boldsymbol{\Lambda} = \mathbf{J}_g^\mathsf{T}\boldsymbol{\Omega}\mathbf{J}_g$ and mark its nonzeros:

$$\boldsymbol{\Lambda} \sim \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \cdot \\ \bullet & \bullet & \cdot & \bullet & \cdot & \bullet \\ \bullet & \cdot & \bullet & \bullet & \cdot & \cdot \\ \bullet & \bullet & \bullet & \bullet & \cdot & \bullet \\ \bullet & \cdot & \cdot & \cdot & \bullet & \cdot \\ \cdot & \bullet & \cdot & \bullet & \cdot & \bullet \end{pmatrix} \qquad (Q_{12},\, Q_{2a},\, T_1,\, T_2,\, G,\, T_a). \tag{6.10}$$

The off-diagonal nonzeros are the eight edges T_1T_2, T_1Q_{12}, T_2Q_{12}, T_2Q_{2a}, T_2T_a, $Q_{12}Q_{2a}$, $Q_{12}G$, $Q_{2a}T_a$, 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 heat path with two sensors, and the variables each reads. The pattern of $\mathbf{J}_g$.

Row	Residual	Q_{12}	Q_{2a}	T_1	T_2	G	T_a
n_1	$Q_{12} - G$	•				•	
n_2	$Q_{2a} - Q_{12}$	•	•				
c_{12}	$Q_{12} - k(T_1 - T_2)$	•		•	•		
c_{2a}	$Q_{2a} - hA(T_2 - T_a)$		•		•		•
π_G	$G - 20$					•	
π_{T_a}	$T_a - 20$						•
ℓ_2	$T_2 - 30.4$				•		
ℓ_1	$T_1 - 34.6$			•			

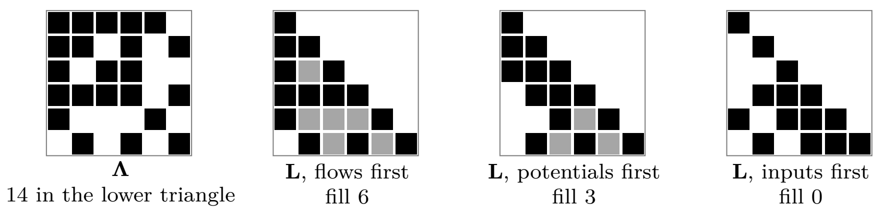


Figure 6.2: The heat path's precision $\mathbf{\Lambda}$ and its Cholesky factor $\mathbf{L}$ under three orderings, as nonzero patterns. Black entries of $\mathbf{L}$ are nonzeros of the lower triangle of the permuted $\mathbf{\Lambda}$; grey entries are fill. Inputs first has none; potentials first 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: $Q_{12} = Q_{2a} =$ 20.750 watts, $T_1 = 34.569$, $T_2 = 30.419$ degrees, $G = 20.750$ watts, $T_a = 20.045$ degrees. The two heat 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 1.470, 1.470, 0.414, 0.337, 1.470, 0.852 in the same order, and they agree with the diagonal of the dense inverse to six decimals, as they must. The Schur complement of $\mathbf{\Lambda}$ onto the two inputs (G, T_a) gives standard deviations 1.470 watts and 0.852 kelvin with correlation -0.920, 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
$Q_{12}, Q_{2a}, T_1, T_2, G, T_a$ (flows first, as listed)	20	6
$T_1, T_2, Q_{12}, Q_{2a}, G, T_a$ (potentials first)	17	3
$G, T_a, T_1, T_2, Q_{12}, Q_{2a}$ (inputs first)	14	0

Figure 6.2 draws the three factors. Inputs first creates no fill at all: G has one neighbor, T_a 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

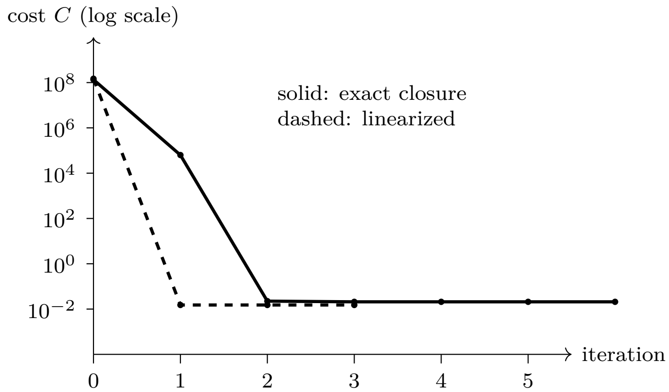


Figure 6.3: Gauss–Newton on the heat path with the natural-convection closure, from the cold start, one sensor: the cost at each iteration on a logarithmic scale (solid), against the linearized closure (dashed), which is exact in one step. The first step removes three orders of magnitude and the second six; the convergence is quadratic once the iterate is close.

graph of the heat path is chordal. Flows first is the worst of the three: Q_{12} has four neighbors, T_1, T_2, Q_{2a}, 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 linear plate closure $Q_{2a} = hA\,(T_2 - T_a)$ by the law the plate actually obeys when the coolant pump is off and the heat leaves by natural convection,

$$Q_{2a} = c\,(T_2 - T_a)^{1.25}, \qquad c = \frac{hA\,\Delta T_{\mathrm{nom}}}{\Delta T_{\mathrm{nom}}^{1.25}} = 1.1247\ \mathrm{W/K}^{1.25},$$

with c chosen so that the two closures agree at the nominal plate rise of $\Delta T_{\mathrm{nom}} = 10$ kelvin. The linear closure of Chapter 1 is this one with the film coefficient frozen at its nominal value; the row is now a fractional power of $T_2 - T_a$, and the slope it presents to the solver grows with the plate's rise. Start Gauss–Newton from a deliberately poor guess, both heat flows at 35 watts, $T_1 = 40$ and $T_2 = 36$ degrees, the dissipation at 30 watts and the coolant at 22. With one sensor the cost falls from 1.4×10^8 to 6.3×10^4 to 0.023 to 0.021, with full steps throughout, and the step length is below 10^{-6} by the third iteration. 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: Natural-convection closure: Laplace posterior at the Gauss–Newton mode against exact quadrature over (G, T_a). Dissipation in watts, coolant temperature in degrees Celsius.

Sensors		G [W]	T_a [°C]
y_2 only	Laplace	20.672 ± 2.301	20.106 ± 0.857
	exact	20.724 ± 2.296	20.097 ± 0.856
y_2, y_1	Laplace	20.830 ± 1.662	20.077 ± 0.806
	exact	20.849 ± 1.661	20.073 ± 0.806

Table 6.3 compares the Laplace posterior at the mode with the exact posterior, computed by quadrature on an 801×801 grid over the two inputs in the input-space form of Chapter 4. Laplace tracks the exact posterior to about two parts in a thousand in every spread, and the exact posterior's mean sits 0.013 of a standard deviation above its mode with one sensor and 0.011 with two: over the width of this posterior the closure is very nearly straight, and the Gaussian at the mode is an excellent account of it. What is *not* negligible is the difference between the exact closure and its linearization. At the mode the plate sits 10.27 kelvin above the coolant, where the convection law's slope is 2.52 watts per kelvin against the linear closure's $hA = 2$, twenty-six percent steeper. A steeper closure means the thermistor's lever arm on the dissipation is shorter: the same reading, read through the true law, leaves the dissipation less well known. The linearized model answers 20.61 ± 1.95 watts where the true one answers 20.67 ± 2.30, a spread fifteen percent too narrow. Linearizing did not move the answer much; it made the model too sure of it.

Fifteen percent is what linearizing costs at the operating point the model was built around. It gets worse away from it, and the model says how fast. Drive the same module harder — a higher charge rate, a higher current, more ohmic heat in the cells — and hold the linearization fixed at its nominal slope. Figure 6.4 plots the bias, the gap between the linearized model's estimate of the dissipation and the exact model's, in units of the exact posterior's own standard deviation. At 25 watts, with the cell surface at 32 degrees, the bias is 0.37 standard deviations and a modeler would not notice it. At 35 watts it is 1.2; at 50 watts, the surface at 41 degrees, it is 2.5; at 70 watts, past the 45-degree surface temperature at which this module would be derated, it is 4.4; at 160 watts, a surface at 73 degrees, it is 14. 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. A thermal model calibrated at a lazy 0.2C trickle and used to read a 2C discharge is making exactly this error.

One pair of rows in the table deserves a physical reading. With one sensor the linearized model's spread is fifteen percent too narrow; with two it is eleven. The gap shrinks because the second sensor is the thermistor inside the module, and what it reads passes to the dissipation through the pad closure $Q_{12} = k(T_1 - T_2)$, which is linear and stays linear. Only part of the evidence now travels through the nonlinear row, so only part of it is distorted by pretending that row is straight. This is worth remembering when a model is audited: the damage a bad closure does is not a property of the closure alone but of how much of the evidence has to pass through it.

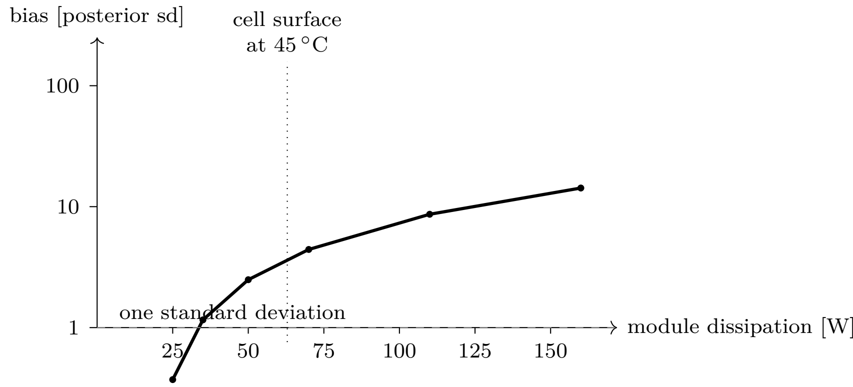


Figure 6.4: The cost of linearizing the plate closure, as the module is driven harder. The vertical axis is the gap between the linearized model's estimate of the dissipation and the exact model's, in units of the exact posterior's standard deviation, on a logarithmic scale. The dotted line marks the dissipation, 62.9 watts, at which the cell surface reaches 45 degrees and the module would be derated. Near the nominal rise the linearization costs a third of a standard deviation; at the derate point it costs four; at twice it, fourteen.

Table 6.4: Rows of $\mathbf{g}$ for the ring bus with uncertain rack currents and a shunt on cable 1–2, and the variables each reads. The pattern of $\mathbf{J}_g$.

Row	Residual	v_2	v_3	I_{12}	I_{13}	I_{23}	q_2	q_3
b_2	$I_{12} - I_{23} + q_2$			•		•	•	
b_3	$I_{13} + I_{23} + q_3$				•	•		•
c_{12}	$I_{12} + gv_2$	•		•				
c_{13}	$I_{13} + gv_3$		•		•			
c_{23}	$I_{23} - g(v_2 - v_3)$	•	•			•		
π_{q_2}	$q_2 + 60$						•	
π_{q_3}	$q_3 + 20$							•
ℓ_{12}	$I_{12} - 47.0$			•				

6.6 Second worked example: the ring bus in matrix form

The ring bus with uncertain rack currents and one shunt, the model of §3.5, in the same matrix form: seven unknowns, the two bus potentials, three cable currents and two rack currents; eight rows of $\mathbf{g}$, five physics rows at width 0.01 amperes, two rack priors and the shunt. 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 rack currents and the cable currents they balance. One solve gives the MAP, $q_2 = -60.393$, $q_3 = -20.196$, $I_{12} = 46.994$, $I_{13} = 33.595$, $I_{23} = -13.399$ amperes, which is the second column of Table 3.3, and selected inversion against the Cholesky factor gives the spreads 3.702, 7.171, 0.793, 3.634, 3.581, agreeing with the dense inverse to 10^{-9}. The condition number is 2.7×10^8, the value the sweep of Chapter 4 found at this width.

Fill. The lower triangle of $\mathbf{\Lambda}$ has 18 nonzeros. Potentials first gives a factor with 22, four fill entries; cable currents first 23, five; the rack currents first, then the potentials, then the cable

Table 6.5: Observability of the ring bus's rack currents: the ratio of the smallest to the largest eigenvalue of the precision, and the posterior of the rack currents in amperes, with and without rack priors, for one and two shunts.

priors and shunts	eigenvalue ratio	q_2 [A]	q_3 [A]
no priors; I_{12}	-7.2×10^{-17}	unobservable	unobservable
no priors; I_{12}, I_{13}	8.8×10^{-7}	-60.40 ± 1.79	-20.20 ± 1.79
no priors; I_{12}, I_{23}	4.5×10^{-7}	-60.40 ± 1.13	-20.20 ± 1.79
rack priors; I_{12}	5.0×10^{-8}	-60.39 ± 3.70	-20.20 ± 7.17

currents, 20, two, which is what the minimum-degree heuristic also finds. The rack currents are the leaves of this graph, each read by one balance and its prior, and eliminating leaves first never fills; the potentials then have two or three neighbors that are nearly cliques already. Cable currents first is again not the cheapest route, for the reason Chapter 5 counted: a cable current's neighbors are the potentials at its ends and the other cable currents at both its taps, and they are not adjacent to one another until that current joins them.

Observability. Remove the rack priors and keep the one shunt. The precision is then singular: its smallest eigenvalue is -7×10^{-17} of its largest — zero, to round-off, and of either sign — and the eigenvector belonging to it is the direction $\Delta q_2 = -0.5$, $\Delta q_3 = 1$, with the currents in cables 1–3 and 2–3 each moving by -0.5 and I_{12} not at all. That is the shift of draw from rack 2 to rack 3 that leaves the shunted current unchanged, since $I_{12} = -(2q_2 + q_3)/3$; the shunt cannot see it, no prior pins it, and the posterior is flat along it. The factorization fails, and the vector that makes it fail names the unobservable combination. Table 6.5 lists the cures. A second shunt on cable 1–3 or on cable 2–3 reads a different combination of the rack currents and restores a condition number near 10^6, with the racks known to between 1.1 and 1.8 amperes from the shunts alone; the rack priors restore it too, at the price of a condition number of 2×10^7 and of racks known only to 3.7 and 7.2 amperes, because a prior pins the unseen direction with its own width and no more. Observability, in the sense the word carries in state estimation, 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 3.71 and 7.20 amperes for the rack currents against the exact 3.70 and 7.17, and a sample correlation of -0.948 against the exact -0.947. 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 total string current exceeds the 88 amperes this bus is rated for is 0.0268 from the samples and 0.0272 exactly, and the probability that cable 2–3 carries more than 20 amperes backwards, toward rack 2, is 0.0343. Chapter 10 uses exactly this sampler inside a region.

A nonlinear cable closure. A copper cable's conductance is not a constant: it falls as the cable heats, and the cable heats on its own current. Replace the fixed conductance by $g(T) = g_0/(1 + \alpha\kappa I^2)$, with $\alpha = 0.0039$ per kelvin for copper and $\kappa = 0.009$ kelvin per ampere squared for this cable's self-heating, so that a cable carrying 47 amperes runs about 20 kelvin warm and conducts seven percent worse than a cold one. Gauss–Newton from a flat start, potentials

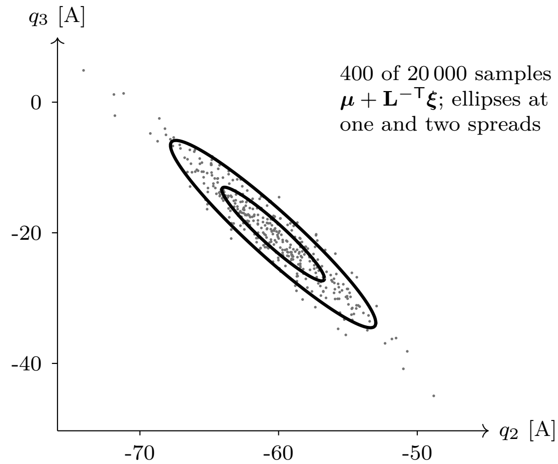


Figure 6.5: Four hundred posterior samples of the two rack currents, drawn from the Cholesky factor of the precision, with the exact one- and two-standard-deviation ellipses. The samples fill the ellipses in the proportions a Gaussian gives, because a triangular solve against the factor turns independent standard normals into draws with exactly the posterior covariance.

zero and rack currents at their priors, reduces the cost from 2.0×10^7 to 6.5×10^4 to 0.41 in three iterations, full steps throughout, and converges by the fifth. The mode has $v_2 = -5.062$ millivolts against the cold cable's -4.699: the same current now needs eight percent more potential to push it through. The Laplace posterior of the rack currents is $q_2 = -61.188 \pm 3.844$ and $q_3 = -20.624 \pm 7.102$ amperes; the exact posterior, by quadrature over the two rack currents with the hard warm-cable physics solved by Newton's method at each grid point, is -61.188 ± 3.843 and -20.573 ± 7.103. The cold-cable model gives -60.393 ± 3.702 and -20.196 ± 7.171. Seven percent of nonlinearity in the closure moves the racks' posterior by eight tenths of an ampere, a fifth of a standard deviation, and their spreads by four percent — and the Laplace posterior tracks the exact one to a thousandth of an ampere. That the shift is small is a fact about this bus at this loading, not a licence: the shunt reads 47 amperes either way, and it is the model, not the meter, that decides how much draw that reading implies.

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 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 tap temperatures and potentials 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 $\mathbf{\Lambda}$ 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 heat path, inputs first has no fill and flows first has six; on the ring bus, 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, watts and kelvins and amperes and millivolts, 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 [80] 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 [22], and it was invented for power networks. Sparse Cholesky, fill-reducing orderings and their graph theory are in Davis [17] and Duff, Erisman and Reid [20]; the first fill-reducing ordering for network equations is Tinney and Walker's [93].

Gauss–Newton with backtracking for nonlinear least squares, its descent property, and its quadratic convergence under small residuals are in Nocedal and Wright [67]. The Laplace approximation is classical; MacKay [58] and Bishop [10] 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 [40]; Särkkä and Svensson [82] 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 [83]; Monticelli [66] and Abur and Gómez Expósito [1] cover the Gauss–Newton iteration, sparse factorization, observability (the singular $\boldsymbol{\Lambda}$ of §6.2) and bad-data detection. Chavali and Nehorai [15] 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 [18].

Summary

- In information form, multiplying Gaussian factors is adding matrices. Every factor is a rank-one term on its scope; the posterior precision is $\boldsymbol{\Lambda} = \mathbf{J}_g^\top \boldsymbol{\Omega} \mathbf{J}_g$, and its sparsity is the Markov graph.
- The MAP solves $\boldsymbol{\Lambda}\mathbf{z}^\star = \boldsymbol{\eta}$; 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 heat path: the precision pattern is the Markov graph by inspection; inputs-first elimination has zero fill, flows-first six; the natural-convection closure converges in three Gauss–Newton steps from a cold start and its Laplace posterior matches quadrature to two parts in a thousand, but the *linearized* closure reports a spread fifteen percent too narrow

at the nominal rise, and drives the estimate off by fourteen standard deviations at eight times it.

- On the ring bus: eleven Markov edges, the MAP of Chapter 3 in one solve, selected inversion to 10^{-9}, leaves-first the cheapest order; a warm-cable closure converges in five iterations and its Laplace posterior matches quadrature to a thousandth of an ampere.

Exercises

Exercise 6.1. Show that the product of $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, as functions of $\mathbf{z}$, is proportional to a Gaussian with precision $\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}$ and information vector $\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2$. 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 $\boldsymbol{\Sigma}_{uu}$ of the joint covariance, and hence that $(\boldsymbol{\Lambda}_{uu} - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\Lambda}_{xu})^{-1} = \boldsymbol{\Sigma}_{uu}$: 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 heat path's $\boldsymbol{\Lambda}$ 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 ring bus with uncertain rack currents (Figure 3.2, $a_{23} = 1$ fixed) and one shunt is linear-Gaussian. Write its $\mathbf{g}$, its $\mathbf{J}_g$, and its $\boldsymbol{\Lambda}$, and find an elimination order with the least fill. Compare with the nodal form of Figure 2.5.

Exercise 6.5. Drive the module of the worked example harder and rerun the Laplace-versus-quadrature comparison with one sensor at dissipations of 35, 70 and 160 watts. The Laplace spread stays accurate at all three; what does change, and why does the exact posterior stay nearly Gaussian even when the plate runs fifty kelvin above the coolant? Relate the answer to the slope of the closure and to how sharply the thermistor reads it.

Solutions to selected exercises

Exercise 6.2. In covariance form the joint is $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the marginal of a block is obtained by dropping the other coordinates: $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_{uu})$, 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 $\boldsymbol{\Lambda}_{uu} - \boldsymbol{\Lambda}_{ux}\boldsymbol{\Lambda}_{xx}^{-1}\boldsymbol{\Lambda}_{xu}$, so that precision is $\boldsymbol{\Sigma}_{uu}^{-1}$: 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 $\boldsymbol{\Lambda}_{xx}$ and $|\mathbf{u}|$ solves against it, whereas $\boldsymbol{\Sigma}_{uu}$ by way of the full covariance needs the dense inverse of $\boldsymbol{\Lambda}$. 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 T_a, whose neighbors T_2 and Q_{2a} are adjacent through c_{2a}; then T_1, neighbors T_2 and Q_{12}, adjacent through c_{12}; then T_2, neighbors Q_{12} and Q_{2a}, adjacent through b_2; then Q_{12}, neighbor Q_{2a}; then Q_{2a}. 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 T_1, T_2, Q_{2a}, G, fills T_1Q_{2a}, T_1G, T_2G and $Q_{2a}G$; eliminating Q_{2a}, now with neighbors T_1, T_2, G, T_a, fills T_1T_a and GT_a; 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 $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is $-\frac{1}{2}\mathbf{z}^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}\mathbf{z} + \mathbf{z}^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$ plus a constant. The logarithm of the product is the sum, $-\frac{1}{2}\mathbf{z}^\mathsf{T}(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})\mathbf{z} + \mathbf{z}^\mathsf{T}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)$ plus a constant, which is the logarithm of a Gaussian with precision $\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}$ and information vector $\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2$: 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 $\mathrm{diag}(1/4^2,\ 1/1^2)$ and information $(20/4^2,\ 20/1^2)$. The thermistor reading $T_2 = 30.4$ degrees with accuracy 0.5 reads $T_2 = T_a + G/hA$, the linear function $\mathbf{h}^\mathsf{T}(G, T_a)$ with $\mathbf{h} = (\frac{1}{2}, 1)$, so it contributes precision $4\,\mathbf{h}\mathbf{h}^\mathsf{T}$ and information $4 \cdot 30.4\,\mathbf{h}$. In one line, the posterior has precision $\left(\begin{smallmatrix} 1.0625 & 2 \\ 2 & 5 \end{smallmatrix}\right)$ and information $(62.05,\ 141.6)$, whose solution is $G = 20.610 \pm 1.952$ watts and $T_a = 20.076 \pm 0.900$ degrees, the second column of Table 3.2.

Exercise 6.4. In row form the variables are the two potentials, the three flows and the two loads. The rows are the two balances, $I_{12} - I_{23} + q_2$ and $I_{13} + I_{23} + q_3$; the three closures, $I_{12} + gv_2$, $I_{13} + gv_3$ and $I_{23} - g(v_2 - v_3)$; the two rack priors; and the shunt on I_{12}. Each row's variables form a clique of $\boldsymbol{\Lambda}$'s pattern: $\{I_{12}, I_{23}, q_2\}$, $\{I_{13}, I_{23}, q_3\}$, $\{I_{12}, v_2\}$, $\{I_{13}, v_3\}$ and $\{I_{23}, v_2, v_3\}$, the priors and the shunt adding only diagonal entries. Eliminate the rack currents first: each has two neighbours already joined by its balance row. Then eliminate I_{12}, whose neighbours v_2 and I_{23} are joined by the closure of line 2–3, and then I_{13} likewise. What remains, v_2, v_3 and I_{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 cable currents are gone. The variables are v_2, v_3, q_2 and q_3; the balances read $q_2 = g(2v_2 - v_3)$ and $q_3 = g(2v_3 - v_2)$, each touching its rack current and both potentials; and the shunt reads $I_{12} = -gv_2$, touching v_2 alone. Eliminating the rack currents first is again perfect, and what remains is the pair of potentials. 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.

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 rack contactor that may have opened, detected from one shunt.

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 coolant loop that modulates until it saturates, a thermostat at the end of its band, a current limiter, a contactor: $y = f_b(x)$ 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, the Arrhenius rise of a cell's own heat generation, 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 conductance or a coulombic efficiency that must be positive, an availability that must lie in $[0, 1]$, a rack current that cannot reverse. 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 $\boldsymbol{\Lambda}_a$ 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,

$$\log Z_a \;=\; -\,C_a(\mathbf{z}_a^\star) \;-\; \tfrac{1}{2}\log\det\boldsymbol{\Lambda}_a \;+\; \text{const}, \tag{7.2}$$

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 $I_{23}/B - (v_2 - v_3)$ instead of $I_{23} - B(v_2 - v_3)$, 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 open contactor does, changes the determinant with it. On the ring bus the physics determinant with cable 2–3 carrying is three times its value with the contactor open, 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*

$$p(z_j \mid \mathbf{y}) \;=\; \sum_a \mathbb{P}(a \mid \mathbf{y})\, \mathcal{N}\big(z_j;\; (\mathbf{z}^\star_a)_j,\; (\boldsymbol{\Lambda}_a^{-1})_{jj}\big), \tag{7.5}$$

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.*

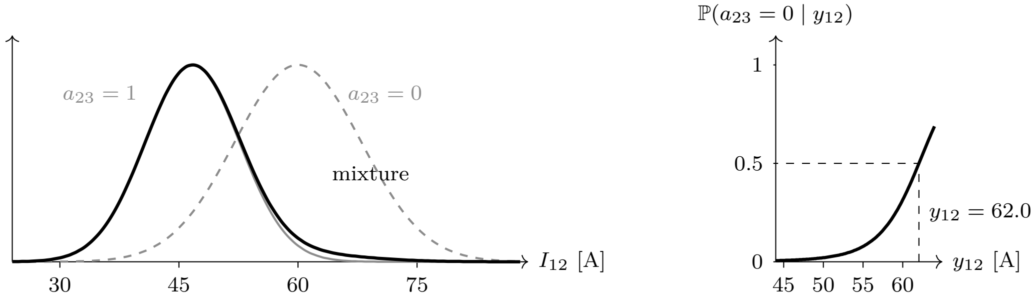
7.2.1 Worked example: a contactor that may have opened

Return to the ring bus of Figure 3.2. The rack currents are uncertain, $q_2 \sim \mathcal{N}(-60, 8^2)$ and $q_3 \sim \mathcal{N}(-20, 8^2)$ amperes, negative for draw; the cable conductances are $g = 10$ amperes per millivolt; the physics rows have width 0.01 amperes; the contactor on the cable from rack 2 to rack 3 has opened with prior probability 0.05; and one shunt reads the current in cable 1–2 with accuracy 0.8 amperes. Every number is from the script `ch07_hybrid_triangle.py` in the book's `scripts` directory.

Before the reading. With the contactor closed the shunted current is $I_{12} = -(2q_2 + q_3)/3$, which under the rack priors is 46.67 ± 5.96 amperes; the loop shares rack 2's draw with cable 1–3 by way of cable 2–3, which carries -13.33 ± 3.77. With the contactor open, rack 2 is fed by cable 1–2 alone, $I_{12} = -q_2 = 60.00 \pm 8.00$, and cable 2–3 carries nothing. Panel (a) of Figure 7.1 draws the prior predictive of the shunted current: the two branch Gaussians, weighted 0.95 and 0.05, and their sum. The open-contactor branch is a small shoulder on the right of the main peak.

After the reading. Table 7.1 sweeps the reading from 46.7 to 64 amperes. Two things happen at once. Within each branch the posterior is Gaussian and the shunt pins the current to ± 0.8; the branch's estimate of q_2 is then whatever draw explains that current under the branch's physics, -62.0 under the open branch when the reading is 62 amperes, and -78.1 under the closed branch, which needs a heavier draw at rack 2 to push that much current down cable 1–2 when part of it is being shared around the loop. Across branches the posterior probability that the contactor is open rises from 0.010 at a reading of 46.7, where the reading sits on the closed branch's peak, to 0.18 at 58 and 0.69 at 64, where it sits on the open branch's. Panel (b) draws the whole curve; it crosses one half at a reading of 62.0 amperes.

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



(a) prior predictive of the shunted current
(b) open-contactor probability after the reading

Figure 7.1: Open-contactor detection from one shunt. (a) The prior predictive of the shunted current I_{12}: the closed branch (solid grey) and the open 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 open branch is a shoulder of height about a twentieth of the main peak. (b) The posterior probability that the contactor is open, as a function of the shunt reading, with the branch evidence corrected by the physics determinant (Proposition 7.1); it crosses one half at $y_{12} = 62.0$ amperes.

Table 7.1: The ring bus with an uncertain contactor and one shunt: the open-contactor posterior and the per-branch and mixture posteriors of the rack current at rack 2, in amperes, as the shunt reading is swept.

y_{12} [A]	$\mathbb{P}(a_{23} = 0 \mid y)$	$q_2 \mid a_{23} = 1$	$q_2 \mid a_{23} = 0$	$\mathbb{E}[q_2 \mid y]$	$\mathrm{sd}[q_2 \mid y]$
46.7	0.010	-60.0 ± 3.70	-46.8 ± 0.80	-59.9	3.91
50.0	0.021	-63.9 ± 3.70	-50.1 ± 0.80	-63.6	4.16
54.0	0.059	-68.6 ± 3.70	-54.1 ± 0.80	-67.8	4.97
58.0	0.184	-73.4 ± 3.70	-58.0 ± 0.80	-70.5	6.83
62.0	0.496	-78.1 ± 3.70	-62.0 ± 0.80	-70.1	8.48
64.0	0.688	-80.4 ± 3.70	-64.0 ± 0.80	-69.1	7.93

determinant, the closed branch would have been understated threefold: the open-contactor probability at 58 amperes would have been reported as 0.40 instead of 0.18, at 62 as 0.75 instead of 0.50, and the crossover placed at 59.1 instead of 62.0. An alarm set at one half would have fired on readings between 59.1 and 62.0 amperes, which the correct posterior considers more likely normal than not.

What a single Gaussian would miss. Look at the last two columns at a reading of 62 amperes. The mixture's mean draw is -70.1 with a spread of 8.5. No branch believes that. The closed branch says -78.1 ± 3.7 and the open branch says -62.0 ± 0.8; 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 "rack 2 is drawing 70 amperes give or take 8" would be told something that is not true under any hypothesis. The correct report is "either the contactor is open and rack 2 draws 62 amperes, or it is closed and rack 2 draws 78, at even odds". 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 open 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 coolant loop is in its saturated regime when the demanded duty exceeds one, and in its modulating 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 coolant loop'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 modulating coolant loop

Put a thermostat on the module heat path. The coolant loop varies its pump speed to hold the cold plate at 30 degrees, and the plate conductance it can deliver ranges from 2 to 3 watts per kelvin as the pump runs from minimum flow to full. The plate closure then has three regimes. With the pump at its minimum, $Q_{2a} = 2(T_2 - T_a)$ and the plate is below the set point. With the pump modulating, $T_2 = 30$ and the conductance is whatever the controller needs, between the limits. With the pump at its maximum, $Q_{2a} = 3(T_2 - T_a)$ and the plate is above the set point. The first boundary is where the minimum flow just holds the plate at 30, $G = 2(30 - T_a)$; the second is where full flow does, $G = 3(30 - T_a)$. The priors of Chapter 3, $G \sim \mathcal{N}(20, 4^2)$ watts and $T_a \sim \mathcal{N}(20, 1^2)$ degrees, put the first boundary through the prior mean: the prior gives 0.500 to the pump at minimum, 0.477 to modulating, and 0.023 to the pump at maximum, which needs a dissipation of 30 watts. A thermistor inside the module reads T_1 with accuracy 0.2 kelvin. Every number is from `ch07_regimes.py`.

The map and its kinks. In the three regimes the module temperature is $T_a + 0.70\,G$, $30 + 0.20\,G$ and $T_a + 0.53\,G$. Panel (a) of Figure 7.2 draws the map at $T_a = 20$: continuous, with slopes 0.70, 0.20 and 0.53 kelvin per watt, and kinks at 20 and 30 watts. A central difference across the first kink gives 0.45 for steps of 8, 2 and 0.2 watts. It does not converge as the step shrinks: it sits at the average of the two slopes, which is the slope of no regime, and nothing in the number says so.

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 0.001 in every regime probability and to three thousandths of a watt in the dissipation's mean and spread. The shortcut is right away from the boundaries and wrong near them. At a reading of 34.0 degrees, where the exact posterior gives 0.22 to the pump at minimum and 0.78 to modulating, the shortcut reports the pump 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 36.0 degrees it reports 0.66 modulating where the exact value is 0.57.

What the regime does to the sensor. In the modulating regime the module temperature is $30 + 0.20\,G$ whatever the coolant does: the controller has absorbed the coolant temperature, and the reading says nothing about it. At a reading of 35.0 the posterior of the coolant temperature is its prior, 19.98 ± 0.93 against 20.0 ± 1.0. The dissipation is known to 0.96 watts there, which is about the thermistor's accuracy divided by the slope 0.20, while with the pump at its minimum the coolant temperature's spread adds to the reading's and the dissipation is known only to 1.40 watts. What a sensor tells is a property of the regime, and the regime is a property of the state.

Table 7.2: The coolant loop after one reading of the module temperature: regime probabilities and the posterior of the dissipation in watts, exact by quadrature and by Proposition 7.2, and the modulating probability under the active-set shortcut.

	exact				proposition		shortcut
y_1 [°C]	min	mod	max	G [W]	mod	G [W]	mod
33.0	0.969	0.031	0.000	18.69 ± 1.40	0.031	18.69 ± 1.40	0.000
33.6	0.565	0.435	0.000	19.33 ± 1.21	0.436	19.33 ± 1.21	0.000
34.0	0.217	0.782	0.000	20.45 ± 0.97	0.783	20.45 ± 0.97	0.000
34.4	0.061	0.937	0.002	22.03 ± 0.93	0.937	22.03 ± 0.93	1.000
35.0	0.004	0.972	0.024	24.69 ± 0.96	0.972	24.69 ± 0.96	1.000
35.6	0.000	0.842	0.158	27.28 ± 1.07	0.842	27.28 ± 1.07	0.794
36.0	0.000	0.574	0.426	28.56 ± 1.41	0.575	28.55 ± 1.41	0.658
36.4	0.000	0.219	0.781	29.12 ± 1.78	0.219	29.12 ± 1.78	0.000
37.0	0.000	0.013	0.987	29.71 ± 1.75	0.013	29.71 ± 1.76	0.000

Table 7.3: The module driven toward thermal runaway, with one thermistor on T_2 of accuracy 0.2 kelvin: Laplace posterior against exact quadrature as the balance's net slope collapses. Skewness of the exact posterior of the dissipation D in the last column. Dissipation in watts, temperature in degrees Celsius.

D [W]	T_2 [°C]	Laplace D	exact D	exact T_a	skew
4	22.09	4.00 ± 1.70	3.98 ± 1.63	20.00 ± 0.88	$+0.077$
17	30.47	17.00 ± 1.25	16.93 ± 1.25	20.03 ± 0.95	-0.114
38	60.90	38.00 ± 0.29	37.97 ± 0.29	20.06 ± 1.00	-0.177

7.4 Strong nonlinearity: when the Laplace posterior holds and when it does not

Chapter 6 showed that the natural-convection closure left the Laplace posterior accurate to two parts in a thousand at the nominal plate rise, where the closure is barely curved across the posterior's width. Curve it. Give the cell its own thermal feedback: its heat generation rises with its temperature, and to second order in the Arrhenius expansion the generated heat is $D\,(1 + \alpha\,\Delta T + \frac{1}{2}(\alpha\,\Delta T)^2)$ with D the cold-cell dissipation, $\Delta T = T_2 - T_a$ and $\alpha = 0.02$ per kelvin. The plate removes $hA\,\Delta T$, and the steady balance is a quadratic in ΔT whose two roots merge at $D^\star = hA/(\alpha(1 + \sqrt{2})) = 41.4$ watts. Past that, no steady temperature solves the row at all: the module runs away. This is the Semenov criterion, and the net slope of the balance, $hA - D\alpha - D\alpha^2\Delta T$, is what falls to zero at it. Table 7.3 repeats the one-sensor comparison of Table 6.3 at dissipations of 4, 17 and 38 watts — a tenth, two fifths and nine tenths of the way to the threshold.

The result is not what a reader expecting nonlinearity to break the approximation would predict. At 38 watts, three and a half watts from runaway, with the cell surface at 61 degrees and the balance's net slope fallen from 1.92 to 0.62 watts per kelvin, the Laplace posterior agrees with quadrature to two decimals in the mean and to under one percent in the spread, and the exact posterior's skewness is under a fifth. It is *further* from the threshold, at 4 watts, that Laplace is least accurate, overstating the spread by four percent. The reason is physical, and it is the mirror image of the one the thermal literature usually gives. A collapsing closure moves T_2 *more* per watt of dissipation: the slope $\partial T_2/\partial D$ grows from 0.54 kelvin per watt at 4 watts to 0.77 at 17 and 3.48 at 38. The thermistor's lever arm on the dissipation lengthens, the posterior narrows from ± 1.63 watts to ± 0.29, 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 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 contactor 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 sign ambiguity of a rack current read only through a squared loss, 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 potentials 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 conductance $k \sim \mathcal{N}(5, 1^2)$ puts a probability of about 3×10^{-7} on $k < 0$, which is harmless. A prior $\mathcal{N}(1, 1^2)$ on a coulombic 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.

Reparameterization. Write the positive conductance as $k = e^{\beta}$ 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 k 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 k. Its cost is that β's posterior, mapped back to k, 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 heat path shows where. Give the pad conductance a Gaussian prior $\mathcal{N}(3, 2^2)$ watts per kelvin, a thermal interface known only to lie somewhere between one and five, or a Gaussian prior on $\ln k$ with the same median and a spread of 0.6. With the two temperature sensors of Chapter 3 the posterior of k is 4.82 ± 0.88 under the first and 4.88 ± 1.02 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 module temperature, $T_1 = T_2 + G/k$, by sampling the three inputs. Under the Gaussian prior 6.6 percent of the draws have $k \leq 0$, and in 6.6 percent of all draws heat flows uphill, from the plate into the module; the fifth percentile of T_1 is 14.6 degrees, below the coolant itself, and 3.1 percent of the draws put the module above 80 degrees, where a cell vents, because k is near zero. Under the log-normal prior almost no draw is unphysical, and the fifth to ninety-fifth percentiles are 30.0 to 49.8 degrees. 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 open contactor 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 contactor 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 conductance, an efficiency or a rack current 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 [53]; 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 [50]. 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 [43] under the name of Bayes factors, and by MacKay [58], 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 [92]; Bishop [10] 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 self-heating sweep is its evidence. The quadratic balance of §7.4 is the second-order form of Semenov's thermal-ignition criterion [84], which in the battery literature is the standard lumped account of when a cell's own heat generation outruns its cooling [5].

Detecting an open contactor or a lost connection from measurements is a standard problem in battery-system operation, usually posed as a hypothesis test over a list of candidate faults; 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 [58].

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 ring bus the uncorrected evidence understates the closed branch threefold and would raise a fault alarm too early.
- One shunt on the ring bus detects an open contactor: the open-contactor probability rises from 0.010 to 0.69 across a reading sweep, and at the crossover the rack-current 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 coolant loop to 0.001; keeping branches whose modes are in their regions fails near the boundaries. A finite difference across a kink sits at the average of two slopes, which is the slope of no regime.
- In the coolant loop's modulating regime the sensor says nothing about the coolant temperature: what a reading tells depends on the regime.
- Driving the module to within three and a half watts of thermal runaway did not break the Laplace posterior, because the collapsing closure lengthened the thermistor's lever arm and narrowed the posterior faster than it curved the map; Laplace was least accurate at the *lightest* load, where the posterior is widest. 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 nearly the same posterior, but a Gaussian prior on a positive pad conductance sends 6.6 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 contactor example, move the shunt to cable 2–3 instead of cable 1–2. Compute the prior predictive of its reading under both branches and explain why this shunt detects the open contactor from a single reading with near certainty, whatever the rack currents. Then move it to cable 1–3 and explain why it is exactly as informative as the shunt on cable 1–2.

Exercise 7.3. Add a second uncertain contactor to the ring bus, on cable 1–3, so that there are four branches. Compute the prior probability of each, the predictive density of each at a reading $y_{12} = 60$ amperes, and the posterior. Which two branches can this shunt not tell apart, and what does their posterior ratio equal?

Exercise 7.4. A coolant loop removes $Q = Q_{\max} u$ for a duty 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 self-heating sweep of Table 7.3 with the thermistor coarsened from 0.2 to 2.0 kelvin, so that the posterior on D is no longer concentrated. Find the dissipation at which the Laplace standard deviation 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 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.

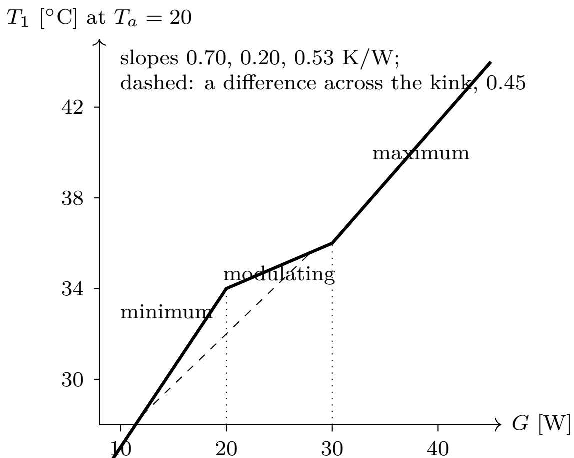
Exercise 7.4. The two regimes are two rows for the same closure, each with its consistency condition: modulating, $Q - Q_{\max} u = 0$ with $u \leq 1$, and saturated, $Q - Q_{\max} = 0$ with $u \geq 1$. The script `ch07_compensator.py` takes $Q_{\max} = 40$ watts, a prior $u \sim \mathcal{N}(1.05, 0.1^2)$ for a loop usually asked for slightly more than it can deliver, and a heat-balance reading $Q = 39.2$ watts with accuracy 1.2. Each regime is a linear-Gaussian branch. The modulating 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 0.076. The saturated branch does not read u at all, since the heat removed no longer depends on it. Its posterior is the prior, with mode 1.05, inside its region, and its predictive density is 0.266. 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 modulating and 0.692 for saturated, 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, T_a) by quadrature and the Laplace posterior from the curvature at the input-space mode. With the thermistor coarsened to 2.0 kelvin the answer is the opposite of the one the question expects. Near runaway, at 38 watts, the Laplace standard deviation of D is 0.634 against the exact 0.648, two percent out. It is at the *lightest* dissipation, 4 watts, that it fails: 2.867 against 2.393, twenty percent too wide, with the mean 0.38 watts low — a sixth of a spread — and a skewness of $+0.34$. The reason is the third failure mode of §7.4, read in the right direction. The reading confines the posterior to a band about $T_2(D, T_a) = y$, and what matters is not how curved that band is but how far the posterior extends along it. At 38 watts the thermistor's lever arm is 3.5 kelvin per watt, so even a 2-kelvin reading pins D to two thirds of a watt and the posterior sees almost none of the curvature. At 4 watts the lever arm is 0.54, the posterior spans more than half of D itself, and it runs into the constraint that no steady temperature exists for $D \leq 0$: the curvature survives the concentration because there is no concentration. Steepness is not curvature, and a wide posterior on a gentle closure is the more dangerous of the two. Chapter 12's probes are how to tell them apart, and the check costs two objective evaluations.

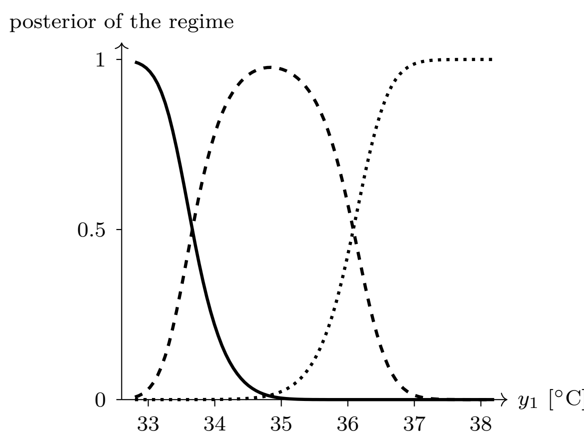
Exercise 7.2. With the contactor closed the current in cable 2–3 is $I_{23} = (q_2 - q_3)/3$, so under the rack priors its prediction is -13.33 ± 3.86 amperes with the shunt's accuracy added; with the contactor open it is exactly zero, 0.00 ± 0.80. The two predictions are separated by more than three of the wider spread, and a reading near either is decisive. A reading of zero gives $\mathbb{P}(\text{open}) = 0.990$ despite the prior of 0.05, and a reading of -6.67, halfway between the two, gives less than 10^{-4}: the open branch says the reading must be zero to within 0.8, so anything sixteen of its spreads away rules it out. The shunt reads the quantity the fault changes most, the current in the cable itself. The shunt on cable 1–3 reads $I_{13} = -(q_2 + 2q_3)/3$, predicted 33.33 ± 6.02 closed and $-q_3 = 20.00 \pm 8.04$ open: the same separation and the same spreads as the shunt on cable 1–2, mirrored. At its open-branch prediction it gives $\mathbb{P}(\text{open}) = 0.315$, exactly what the cable 1–2 shunt gives at its own. It is as informative because the open contactor moves rack 2's draw onto cable 1–2 and rack 3's onto cable 1–3, by amounts that the racks' symmetric priors make equal.

Exercise 7.3. Under each branch I_{12} is a linear combination of the rack currents: $-(2q_2+q_3)/3$ with both contactors closed, $-q_2$ with cable 2–3 open, $-(q_2 + q_3)$ with cable 1–3 open, since rack 3 is then fed through rack 2, and $-q_2$ with both open, since rack 3 is islanded and offline. The predictive densities at 60 amperes are 0.0057, 0.0496, 0.0074 and 0.0496; 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 shunt cannot tell "cable 2–3 open" from "both open": in both, rack 2 is fed by cable 1–2 alone and $I_{12} = -q_2$. Their predictive densities are identical, so their posterior ratio is their prior ratio, 19 to 1, whatever the reading. The second open contactor is invisible to this shunt because it lies on the far side of the first; a shunt on cable 1–3 would see it at once. Pruning by weight alone would discard "both open" at a threshold of 0.02, and that is safe only because its consequence, rack 3 offline, is one an operator would see by other means.



(a) the map from dissipation to module temperature



(b) the regime after one reading

Figure 7.2: The modulating coolant loop. (a) The module temperature as a function of the dissipation at a coolant temperature of 20 degrees: continuous, with kinks where the pump reaches each end of its range, 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 module reading.

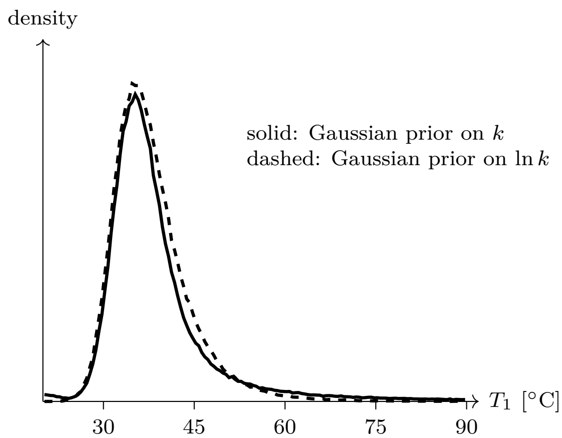


Figure 7.3: The prior predictive density of the module temperature under a Gaussian prior on the pad conductance (solid) and a Gaussian prior on its logarithm (dashed), both with median 3 watts per kelvin. The solid curve's heavy right tail comes from conductances near zero; its 6.6 percent of draws with negative conductance, all with the module below the plate, lie off the left of the figure.

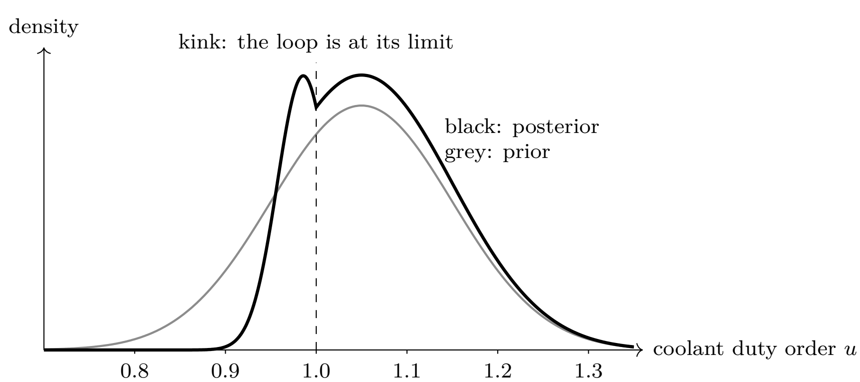


Figure 7.4: The posterior of the coolant duty order after a reading of 39.2 watts with accuracy 1.2 (black), against the prior (grey). Below $u = 1$ the loop modulates and the reading pins the order; above it the loop is at its limit and the reading says nothing about the order. The posterior is continuous with a kink at $u = 1$, a notch between the two regimes' modes.

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 pack cut at one tie cable, 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:

$$m_{x\to f}(x) \;=\; \prod_{g\neq f} m_{g\to x}(x). \tag{8.1}$$

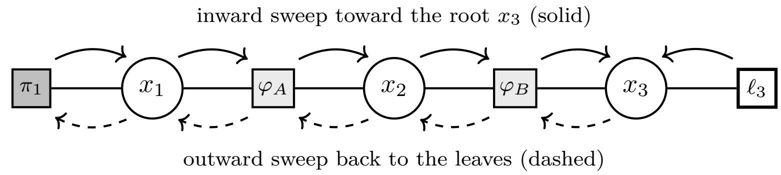


Figure 8.1: Sum-product on the three-variable chain with a prior on x_1 and a sensor on x_3. The inward sweep carries one message along each edge toward the root; the outward sweep carries one back. After both, every variable has received a message from every neighbor, and its marginal is their product (Proposition 8.1). Twelve messages, one per edge in each direction, give all three marginals.

A factor f with scope $\{x, y_1, \ldots, y_k\}$ sends to x its own function times the messages from its other variables, integrated over those variables:

$$m_{f\to x}(x) \;=\; \int \varphi_f(x, y_1, \ldots, y_k) \prod_i m_{y_i\to f}(y_i)\, \mathrm{d}y_1 \cdots \mathrm{d}y_k. \tag{8.2}$$

Equation (8.2) is one elimination step: it removes $y_1, \ldots, y_k$ 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 $y_1, \ldots, y_k$, 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 T_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 ring bus in nodal form is a single cluster of two variables. The heat path 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 I_{12}, I_{13}, I_{23}, v_2, v_3 that Chapter 5 found to have width two. Eliminating I_{12} forms the clique $\{I_{12}, v_2, I_{23}\}$; eliminating I_{13} forms $\{I_{13}, v_3, I_{23}\}$; eliminating I_{23} forms $\{I_{23}, v_2, v_3\}$; 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 I_{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.*

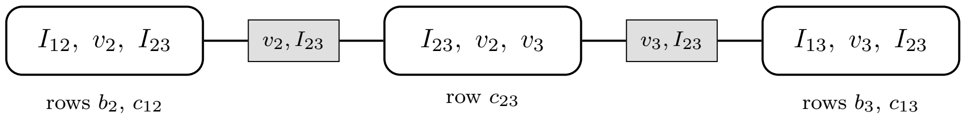


Figure 8.2: A junction tree for the ring bus in row-per-factor form. Clusters (rounded) are the cliques of the elimination order $I_{12}, I_{13}, I_{23}, v_2, v_3$; separators (shaded) are their intersections. Every row is assigned to a cluster containing its scope, and the loop of the network has become 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 $I_1, \ldots, I_m$ 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:

$$\mathbf{\Lambda} \;=\; \begin{pmatrix} \mathbf{\Lambda}_{11} & & & \mathbf{\Lambda}_{1S} \\ & \mathbf{\Lambda}_{22} & & \mathbf{\Lambda}_{2S} \\ & & \ddots & \vdots \\ \mathbf{\Lambda}_{S1} & \mathbf{\Lambda}_{S2} & \cdots & \mathbf{\Lambda}_{SS} \end{pmatrix}, \qquad \mathbf{\Lambda}_{SS} = \sum_r \mathbf{\Lambda}_{SS}^{(r)} + \mathbf{\Lambda}_{SS}^{\mathrm{cut}}, \tag{8.4}$$

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 $\mathbf{\Lambda}^{(r)}$ and information $\boldsymbol{\eta}^{(r)}$ on $I_r \cup S$. Their product, marginalized over I_r, is a Gaussian on S with*

$$\mathbf{S}_r \;=\; \mathbf{\Lambda}_{SS}^{(r)} - \mathbf{\Lambda}_{SI}^{(r)} \big(\mathbf{\Lambda}_{II}^{(r)}\big)^{-1} \mathbf{\Lambda}_{IS}^{(r)}, \qquad \mathbf{h}_r \;=\; \boldsymbol{\eta}_S^{(r)} - \mathbf{\Lambda}_{SI}^{(r)} \big(\mathbf{\Lambda}_{II}^{(r)}\big)^{-1} \boldsymbol{\eta}_I^{(r)}. \tag{8.5}$$

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 $\sum_r \mathbf{S}_r + \mathbf{\Lambda}_{SS}^{cut}$ and information $\sum_r \mathbf{h}_r + \boldsymbol{\eta}_S^{cut}$. Given the port posterior $(\boldsymbol{\mu}_S, \mathbf{\Sigma}_S)$, the posterior of region r's interior is Gaussian with*

$$\boldsymbol{\mu}_I = \big(\mathbf{\Lambda}_{II}^{(r)}\big)^{-1} \big(\boldsymbol{\eta}_I^{(r)} - \mathbf{\Lambda}_{IS}^{(r)} \boldsymbol{\mu}_S\big), \qquad \mathbf{\Sigma}_I = \big(\mathbf{\Lambda}_{II}^{(r)}\big)^{-1} + \mathbf{X}\, \mathbf{\Sigma}_S\, \mathbf{X}^{\mathsf{T}}, \quad \mathbf{X} = \big(\mathbf{\Lambda}_{II}^{(r)}\big)^{-1} \mathbf{\Lambda}_{IS}^{(r)}. \tag{8.6}$$

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), $\mathbf{\Lambda}_{I_r I_s} = \mathbf{0}$ 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.

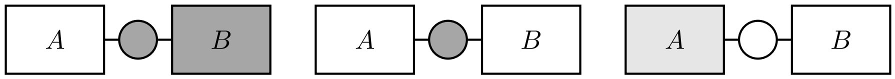
This organization has three names in three literatures. In sparse direct methods it is *nested dissection* or *substructuring*: eliminate the interiors, solve the reduced system on the separators, back-substitute. In electrical networks it is *Kron reduction*: eliminating the interior taps of a network leaves an equivalent network among the boundary racks, with equivalent cables and equivalent rack currents, and the reduced conductance 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 potential of region A to an potential of region B directly, a comparison of v_3 with v_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



(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.

region's group 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 shunt 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 rack currents 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

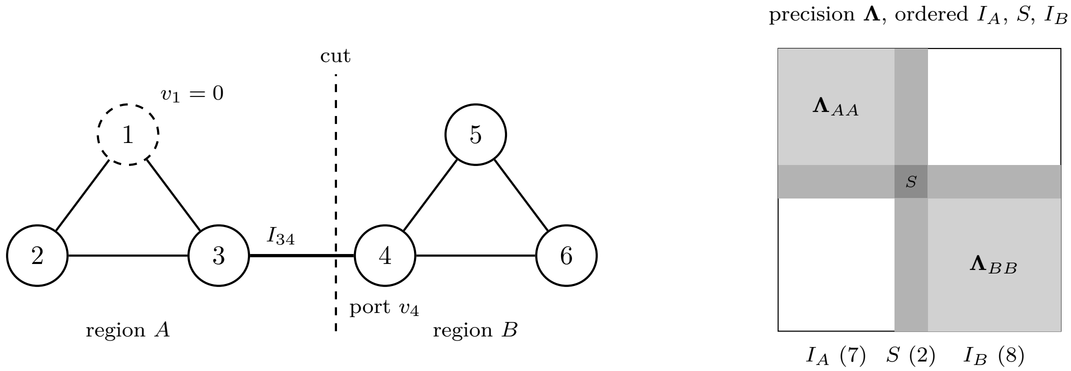


Figure 8.4: Two regions joined by one tie cable. Left: the network; the dashed line is the cut, whose port pair is the tie current I_{34} and the potential v_4 on the B side. Right: the precision matrix with region A's seven interior variables first, the two ports in the middle, and region B's eight 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 containers, one tie cable

Two copies of the ring bus are joined by a tie cable. Region A is racks 1, 2, 3 with rack 1's bar the reference; region B is racks 4, 5, 6; the tie is cable 3–4. Every cable has conductance $g = 10$ amperes per millivolt. The five rack currents $q_2, \ldots, q_6$ are uncertain with Gaussian priors of spread 8 amperes about -48, -32, -20, -40, -24. Physics rows have width 0.01 amperes. Three shunts read I_{12}, I_{56} and the tie current I_{34}, with accuracy 0.8, at 70.0, -6.0 and 86.0, values a little off the currents the nominal racks 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 $\{I_{34}, v_4\}$, with v_4 assigned to region B and the cut closure c_{34}, which also reads v_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 currents eliminated, the balance at rack 4 would read v_3, v_4, v_5, v_6 and q_4 together, making them a clique, and the smallest separator at the same cut would have three or four variables. The current 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 $I_{34} = 85.986 \pm 0.798$ amperes and $v_4 = -17.960 \pm 0.380$ millivolts. 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 eight: the rack current at rack 3, for instance, is -31.217 ± 7.173 either way. Proposition 8.3, checked.

The messages, read physically. Region A's message has precision

$$\mathbf{S}_A \approx \begin{pmatrix} 1.715 & 1.064 \\ 1.064 & 7.574 \end{pmatrix} \quad \text{on } (I_{34}, v_4): \tag{8.7}$$

it constrains both the tie current and the potential at rack 4, because rack 4 is tied through cable 3–4 and the ring to the reference, and because the tie shunt sits on the cut side of the split and was assigned to A. Region B's message is

$$\mathbf{S}_B \approx \begin{pmatrix} 5.21 \times 10^{-3} & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbb{E}[I_{34} \mid v_4] = 84.00 + 0 \cdot v_4, \quad \text{sd } 13.86. \tag{8.8}$$

Region B says nothing about the potential at its own port, because nothing in B reaches the reference except the tie itself: its potentials are defined only relative to v_4. What it says about the current is that the tie must carry 84 amperes into B with a spread of 13.86, which is the sum of its three rack priors, $20 + 40 + 24$, with their spreads added in quadrature, $8\sqrt{3}$. That is Proposition 1.1, exactness at a cut, delivered as a message with an error bar: the current across the boundary equals the net draw 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 v_4, the equivalent conductance seen from rack 4, and the intercept would become an equivalent rack current: the classical Ward equivalent, with the zero spread that classical equivalents assume replaced by the spread the priors imply.

Reuse, two ways. Move the shunt on I_{56} from -6.0 to 0.0 amperes. Inside B the estimates shift: q_5 from -41.62 to -33.01 and q_6 from -23.71 to -32.32, the shunt redistributing draw between the two racks. Region B's message does not change at all, to eleven decimals: the shunt reads a difference of rack currents, and the message carries only their sum. Nothing outside B needs to be told. Now instead change the prior mean of q_5 from -40 to -52 amperes. Region B's intercept moves from 84.00 to 96.00; 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 $I_{34} = 86.025$, $v_4 = -17.965$, 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 containers with their own references

Region B of the previous example said nothing about the potential 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 storage installation that is the ordinary case: each container's power conversion system regulates its own DC bus, and holds it whether or not the tie cable is there.

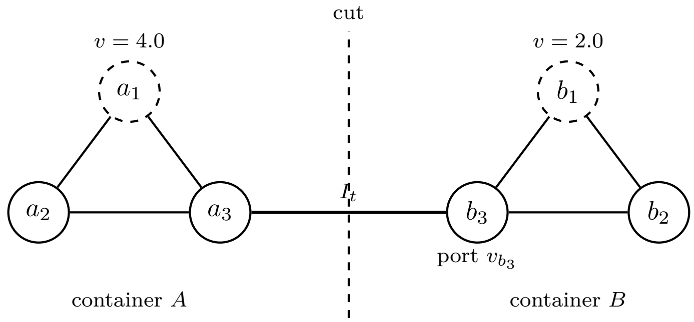


Figure 8.5: Two containers joined by one tie cable, each with a converter holding the potential at one of its racks. The reservoir symbol marks the regulated rack. The cut is the tie; its port pair is the tie current I_t and the potential at b_3. Unlike the previous example, container B's potentials are defined without the tie, so its message constrains the port potential as well as the port current.

Table 8.1: The two containers: the port posterior from the full model and from the reduced port system, and container B's message read as a Thevenin equivalent, $\mathbb{E}[v_{b_3} \mid I_t] = v_{\text{eq}} + r_{\text{eq}} I_t$, with and without the shunt in container B, against the Kron reduction of container B's conductance matrix.

	full model	reduced port system
I_t [A]	10.721 ± 0.768	10.721 ± 0.768
v_{b_3} [mV]	0.8689 ± 0.1023	0.8689 ± 0.1023
container B's message	v_{eq} [mV]	r_{eq} [mV/A]
with its shunt	0.6414 ± 0.1129	0.02015
without its shunt	0.7667 ± 0.1863	0.03333
Kron reduction at b_3	–	0.03333

Every busbar carries $I_{ij} = g_{ij}(v_i - v_j)$ and the converter pins v at its terminal. The two containers of Figure 8.5 are each a ring: container A is racks a_1, a_2, a_3 with a_1's converter holding 4.0 millivolts above the nominal bus, container B is b_1, b_2, b_3 with b_1 holding 2.0, and the tie joins a_3 to b_3. Every busbar inside a container has conductance 20 amperes per millivolt; the tie cable is thinner, at 6. The four rack currents are uncertain with Gaussian priors of spread 5 amperes about 25, 18, 30 and 22. Physics rows have width 10^{-2} amperes. One shunt of accuracy 0.4 amperes sits in each container, reading 25.20 and 30.50. The model has fifteen unknowns and seventeen factors. Every number is from `ch08_two_areas.py`.

Exactness. The port pair is $\{I_t, v_{b_3}\}$. Container A's interior is its two free potentials, its three busbar currents and its two rack currents; container B's is the same less the port potential. The Markov graph has no edge between the two interiors. The full posterior gives $I_t = 10.72 \pm 0.77$ amperes, flowing from the higher-potential container A into B, and $v_{b_3} = 0.869 \pm 0.102$ millivolts. The reduced port system, the sum of the two containers' messages, gives the same two numbers to 4×10^{-10} and their covariance to 4×10^{-12}, and back-substitution into container B reproduces its six interior marginals to 2×10^{-11}. Table 8.1 collects the numbers.

The message is a Thevenin equivalent. Container B has its own converter, so its potentials are defined without the tie, and its message constrains the port potential as well as the port current. Read as the conditional of the port potential given the tie current, it is $\mathbb{E}[v_{b_3} \mid I_t] = 0.6414 + 0.02015\, I_t$ with a conditional spread of 0.113 millivolts: an equivalent source and an equivalent resistance, with an error bar on the first. This is the Thevenin equivalent an engineer would draw at b_3, and the resistance is not quite the network's. Kron reduction of container B's conductance matrix at b_3, with the regulated rack grounded, eliminates b_2 and leaves $2g - g^2/2g = 1.5g = 30$ amperes per millivolt, a resistance of 0.03333 millivolts per ampere — thirty-three microhms; removing container B's shunt from its message recovers that slope to 1×10^{-12}, with the equivalent source 0.767 known only to 0.186. The shunt has done two things to the message. It has narrowed the equivalent source from 0.186 to 0.113 millivolts, and it has stiffened the apparent resistance from 0.03333 to 0.02015, because a busbar current held near its reading pushes back against a change in the tie current 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 container A's shunt by 1.0 ampere, to 26.20. Container B's message does not change at all — not to within a tolerance, but in the same floating-point bits, because no factor of container B was touched — and solving the two-variable port system with B's cached message gives $I_t = 10.546$ and $v_{b_3} = 0.864$, the full re-solve's values to 2×10^{-10}.

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 open contactors by the hundred.

The two-region pack shows the count at a size small enough to list. Make three cables uncertain, each open with probability 0.05: cable 2–3 in region A, and cables 4–5 and 5–6 in region B. Region A has two branches, region B four, the pack eight. Each branch is linear in the rack currents, so its weight is its prior times the predictive density of the three shunts (Proposition 7.1), and Table 8.2 lists them, from `ch08_messages.py`. Before the shunts the prior needs four components to reach ninety-nine percent of the weight: all cables carrying, at 0.857, and each single opening, at 0.045. The prior mixture on the port potential, drawn in Figure 8.6, is visibly spread: opening cable 2–3 moves v_4 from -17.7 to -20.0 millivolts, and losing both cables at rack 5, which islands it and takes its draw off the bus, moves it to -11.1.

After the shunts one component carries 0.99904 of the weight. The shunt on cable 5–6 reads -6.0 amperes, and a branch with that cable open predicts a reading of zero to within the shunt's accuracy; its predictive density is 10^{-16}, and every branch with cable 5–6 open is gone. Opening cable 2–3 keeps 0.00096, which is the 0.001 that Chapter 9 finds for region A's mixture message

Table 8.2: The two-region pack with three uncertain cables: the eight branches, their prior probabilities, the predictive density of the three shunt readings under each, the posterior, and the port potential v_4 in millivolts before and after the readings. The two branches that island rack 5 take its draw off the bus.

cables open	prior	predictive density	posterior	v_4 prior	v_4 posterior
none	0.857	1.9×10^{-4}	0.99904	-17.73 ± 2.39	-17.960 ± 0.380
2–3	0.045	3.4×10^{-6}	0.00096	-20.00 ± 2.88	-20.399 ± 0.816
4–5	0.045	1.1×10^{-10}	$< 10^{-8}$	-17.73 ± 2.39	-17.936 ± 0.380
5–6	0.045	5.6×10^{-16}	$< 10^{-8}$	-17.73 ± 2.39	-17.960 ± 0.380
2–3, 4–5	0.0024	2.0×10^{-12}	$< 10^{-8}$	-20.00 ± 2.88	-20.365 ± 0.816
2–3, 5–6	0.0024	1.0×10^{-17}	$< 10^{-8}$	-20.00 ± 2.88	-20.399 ± 0.816
4–5, 5–6	0.0024	7.3×10^{-19}	$< 10^{-8}$	-11.07 ± 1.98	-17.932 ± 0.380
all three	0.0001	1.3×10^{-20}	$< 10^{-8}$	-12.00 ± 2.40	-20.358 ± 0.816

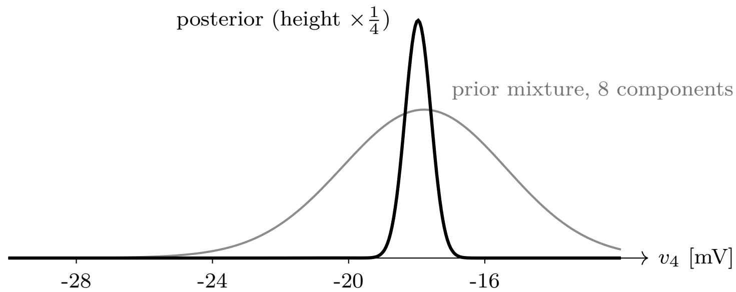


Figure 8.6: The port potential v_4 of the two-region pack with three uncertain cables. Grey: the prior mixture over the eight branches, with four components carrying ninety-nine percent of the weight. Black: the posterior after the three shunts, one component, drawn at a quarter of its height.

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 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; container B's shunt stiffened the equivalent resistance by a factor of 1.65 and narrowed the equivalent source'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 [49]. The junction tree, the running intersection property and exact local propagation on loopy graphs are Lauritzen and Spiegelhalter's [52]; Koller and Friedman [47] give the correspondence between junction trees and elimination orders in full. For Gaussian models, Rue and Held [80] 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 [17, 26]; its iterative descendants are the domain decomposition methods of Toselli and Widlund [94]. In electrical networks the same Schur complement is Kron reduction [48], given its modern graph-theoretic treatment by Dörfler and Bullo [19], and the reduced network with equivalent rack currents is Ward's equivalent [97], still the standard external-system model in power-system analysis. Proposition 8.2 is Kron reduction of the posterior precision rather than of the conductance 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 [15] is the closest published instance of §8.4 in a power-system setting.

In water networks, Han, Eguchi, Mehrotra and Venkatasubramanian [34] 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 [37] 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 [18] 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-ring pack the reduced system agrees with the full model to nine decimals; region B's message is its total draw 84.0 ± 13.9 amperes and nothing about the potential, which is exactness at a cut with an error bar; an interior shunt moves B's rack currents and leaves its message untouched.
- With three uncertain cables the two-region pack 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 containers joined by a tie cable, a container with its own converter sends a Thevenin equivalent: an equivalent source with an error bar and an equivalent resistance. Without the container's shunt the resistance is the Kron reduction, 0.03333 millivolts per ampere; with it, 0.02015, and the equivalent source's spread falls from 0.186 to 0.113 millivolts.

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 ring bus 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 cable from rack 6 to rack 1. The cut now has two edges and four ports. Recompute region B's message and show that its precision on the potentials is no longer zero; read off the equivalent conductance between the two port racks and compare it with the Kron reduction of the conductance matrix of region B alone.

Exercise 8.4. Prove the claim of §8.5 for the shunt on I_{56}: show that, with B's only connection to the reference through the tie, the tie current is a function of the sum of B's rack currents alone, and hence that any interior shunt whose reading is a combination of rack currents with zero coefficient sum leaves the message unchanged. Find an interior shunt that does change it.

Exercise 8.5. Give region B two uncertain cables, 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 cables 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 T_2, the only row that ties the uncertain quantities to the reading is $T_2 = T_a + G/hA$; the other rows determine T_1 and the flows, which nothing reads. The graph is then a star, one physics factor on (T_2, G, T_a) with three leaves: the prior on G, with information pair $(\lambda, \eta) = (2500,\ 125)$; the prior on T_a, $(1.111 \times 10^5,\ 1.111 \times 10^5)$; and the sensor on T_2, $(4 \times 10^6,\ 4.108 \times 10^6)$. Each leaf sends itself. The physics factor's message to G integrates T_2 and T_a against their incoming messages, $T_2 \sim \mathcal{N}(1.027, 0.0005^2)$ and $T_a \sim \mathcal{N}(1.000, 0.003^2)$: then $G/2 = T_2 - T_a \sim \mathcal{N}(0.027,\ 9.25 \times 10^{-6})$, 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 T_a is $T_a = T_2 - G/2$ 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 T_a is 1.000165 ± 0.002874. Both are the second column of Table 3.2. The message to G is the Schur complement of the (T_2, T_a) block in the precision of the physics factor times the messages from T_2 and T_a: 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 racks, 4 and 6, and four port variables: the tie currents I_{34} into rack 4 and I_{61} out of rack 6, and the two potentials. Kron reduction of B's own conductance matrix, the three cables 4–5, 4–6, 5–6 of conductance 10, onto racks 4 and 6 eliminates rack 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 $\left(\begin{smallmatrix} 20 & -10 \\ -10 & 20 \end{smallmatrix}\right) - \frac{1}{20}\left(\begin{smallmatrix} 100 & 100 \\ 100 & 100 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 15 & -15 \\ -15 & 15 \end{smallmatrix}\right)$, which the script computes. The equivalent conductance between the two port racks is 15 amperes per millivolt: the direct cable of 10 in parallel with the path through rack 5, two cables of 10 in series, which is 5. Region B's message now constrains the tie currents given the port potentials, and its slope is the Kron-reduced conductance, because B has no sensor and its rack priors enter only the intercept. What B still cannot say is the level of its potentials: 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 potentials is no longer zero, but it is singular along the direction that shifts both port potentials together.

Exercise 8.4. With one tie, region B's three racks have balances $I_{34} - I_{45} - I_{46} + q_4 = 0$, $I_{45} - I_{56} + q_5 = 0$ and $I_{46} + I_{56} + q_6 = 0$; their sum is $I_{34} + q_4 + q_5 + q_6 = 0$, so the tie current is minus the sum of the rack currents whatever the internal currents. The message on (I_{34}, v_4) is therefore a statement about the sum $\Sigma = q_4 + q_5 + q_6$ alone, and a shunt changes it only if it changes the posterior of Σ given the port. A shunt reading a combination $\sum_i c_i q_i$ with $\sum_i c_i = 0$ is, under priors of equal spread, uncorrelated with Σ a priori, since $\mathrm{Cov}(\sum c_i q_i, \Sigma) = \sigma^2 \sum c_i = 0$, and for Gaussians uncorrelated is independent; conditioning on it leaves Σ's distribution unchanged. The current in cable 5–6 is such a combination: within B, $I_{56} = (q_6 - q_5)/3$ plus a term in I_{34} that the port already fixes. The script confirms it: a shunt on I_{56} changes the message by 10^{-11}, and a shunt on $q_5 - q_6$ by 2×10^{-10}, both leaving the implied tie current at 84.00 ± 13.86 amperes. A shunt on the sum changes it completely, narrowing the tie current to 84.00 ± 0.80; a shunt on one rack, q_5, changes it partly, to 82.20 ± 8.46, because one rack'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 rack currents fixed, the order I_{12}, I_{13}, I_{23}, v_2, v_3 has width two, and its cliques are the three clusters of the figure, with separators $\{v_2, I_{23}\}$ and $\{v_3, I_{23}\}$. The running intersection property holds: I_{23} is in all three clusters, and each potential is in two adjacent ones. If the rack currents are uncertain variables, as in the ring bus of Exercise 5.3, eliminate them first. q_2 forms the cluster $\{q_2, I_{12}, I_{23}\}$ and q_3 the cluster $\{q_3, I_{13}, I_{23}\}$, and each joins the cluster holding its current through the separator of that current and I_{23}. The width stays two, since the new clusters also have three variables, and the property still holds: I_{23} is now in all five clusters, and each rack current is in one. With the availability a_{23} added, the closure of cable 2–3 becomes $I_{23} - a_{23} g(v_2 - v_3)$, and a_{23} joins the cluster that holds that row, the middle one, which becomes $\{I_{23}, v_2, v_3, a_{23}\}$. 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: cable 2–3 uncertain in region A, and cables 4–5 and 5–6 in region B, each open with probability 0.05. Table 8.2 lists the eight components with their weights. Before the shunts, 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 cables 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 cables 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.

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 cable of a pack may be open. 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: $\eta_i = P_i\mu_i - \sum_k m_{ki} = \mathbf{\Lambda}_{ii}\mu_i + \sum_k (P_{ki}\mu_i + \mathbf{\Lambda}_{ki}\mu_{k\setminus i})$. 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 module heat path with two sensors, the ring bus with uncertain rack currents and one shunt, and the two-region pack 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 potentials-only nodal form of the

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 currents eliminated; the last block has the rack currents eliminated as well. Physics widths as each chapter set them.

Model	Form	unknowns	$\rho(\|\mathbf{R}\|)$	converged
module heat path	row	6	1.64	no, with or without damping
ring bus	row	7	1.73	no
two-region pack	row	17	2.22	no
module heat path	nodal	4	1.80	no
ring bus	nodal	4	1.54	no
two-region pack	nodal	10	2.24	no
module heat path	temperatures only	2	0.71	yes, exact (a tree)
ring bus	potentials only	2	0.18	yes, exact (a tree)
two-region pack	potentials only	5	1.01	no

two-region pack, five variables joined in two rings 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 ring bus in row form the error of the belief means is 1.0 ampere after the first sweep and 1.0 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 heat path reduced to its two temperatures and on the two-region pack reduced to its port pair, both trees, the first sweep is exact to 10^{-10} and to machine precision, 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 ring bus in row form from 0.01 to 10 amperes, the last being physics so loose as to be nearly absent: the radius falls from 1.73 to 1.30 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 current in one closure and two balances, a potential in the closures of its cables, 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 currents folds rows together and changes the ties a little, but a balance at a rack still ties that rack's potential to each neighbor's as tightly as the conductances say, and the pack's own cycles remain.

One pair shows the mechanism in numbers. The current in cable 1–2 and the potential at rack 2 share the closure c_{12} and nothing else, and each sits in two physics rows. Without the shunt their partial correlation is -0.500 at every width, which is $-1/\sqrt{2 \cdot 2}$. The shunt on I_{12}

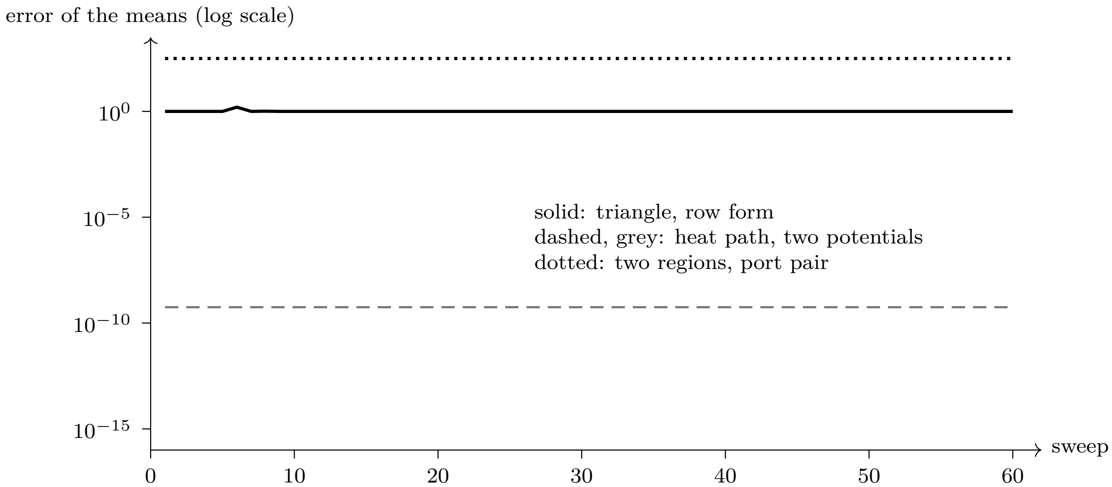


Figure 9.1: Gaussian belief propagation sweep by sweep: the largest error of the belief means against the exact posterior, on a logarithmic scale. Solid: the ring bus in row form, radius 1.73; the error wanders at about one hundred percent and never settles. Dashed: the heat path reduced to its two temperatures, radius 0.71. Dotted: the two-region pack reduced to its port pair, radius 0.30. Both reductions are trees and are exact after one sweep.

adds to that variable's diagonal a weight the physics does not scale, and dilutes the tie to -0.498 at a width of 0.1 and to -0.056 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 1.30.

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, $\mathrm{KL}(q\|p) = \frac{1}{2}\left[\sum_i \boldsymbol{\Lambda}_{ii} s_i^2 + (\mathbf{m} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Lambda} (\mathbf{m} - \boldsymbol{\mu}) - \sum_i \log s_i^2\right]$, 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 heat path after one reading the two inputs are correlated at -0.868. Mean field reports the dissipation to ± 0.970 watts and the coolant temperature to ± 0.447 kelvin, against the exact 1.952 and 0.900: the factor 0.497 on each, which Figure 9.2 draws. On all six variables of the soft heat path it is far worse. Every physics 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 dissipation to ± 0.0010 watts, two thousand 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.

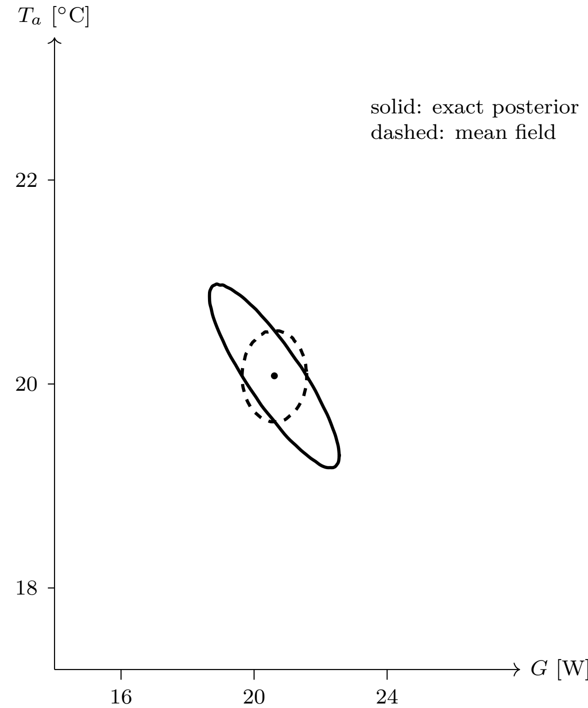


Figure 9.2: The heat path's posterior over the dissipation and the coolant temperature after one reading (solid) and the mean-field approximation to it (dashed), both at one standard deviation. The exact posterior is a ridge at a correlation of -0.868; mean field is the axis-aligned ellipse that fits inside it, too narrow in each coordinate by $\sqrt{1-0.868^2} = 0.497$.

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,

$$m_r(S) \;\approx\; \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(S;\ \boldsymbol{\mu}_S^{(i)},\ \boldsymbol{\Sigma}_S^{(i)}), \tag{9.1}$$

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 pack and make cable 2–3 uncertain, its contactor open with prior probability 0.05. Region A's message on its ports (I_{34}, v_4) is a two-component mixture, one Gaussian per branch, weighted by the branch's evidence from the factors inside A, which include the shunts on I_{12} and I_{34}, corrected for the physics determinant as Proposition 7.1 requires. The script computes both components. On the potential v_4 they are -17.961 ± 0.380 millivolts with weight 0.999 and -20.400 ± 0.816 with weight 0.001: the shunts inside A have all but ruled the open contactor out, and the two components are three standard deviations apart. Collapsing the mixture to one Gaussian by moment matching gives -17.963 ± 0.388, 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.

9.4.2 Worked example: a container as a simulated region

Take container 1 of the site-adequacy example of Chapter 10 as a region. Its interior holds three rack power limits and an export-path limit, its port is the power it delivers to the site, S_1, and with the coolant-inlet 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 container 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 site combines two such messages, one per container, and asks for the probability of a shortfall against its commitment, $\mathbb{P}(S_1 + S_2 < 2{,}100)$ kilowatts. Every number is from the script `ch09_region.py`.

At the heat-wave inlet 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 racks bind in 55 percent of the draws and the export path in 45, so the message is the minimum of two comparable quantities, skewed by $+0.33$, and the site'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 site 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

Table 9.2: Container 1 as a simulated region: the site'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.

	inlet 26.5°C		inlet 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

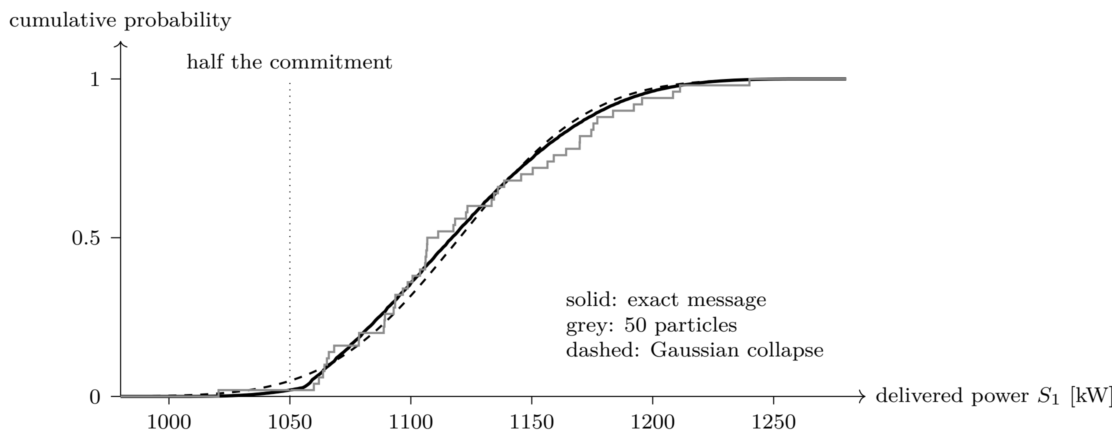


Figure 9.3: Container 1's message on its port at an inlet temperature of 21°C, as a cumulative distribution. Solid: the exact message. Grey: a particle message of 50 draws. Dashed: the Gaussian with the same mean and spread. The exact message is skewed by the competition between the rack and export-path limits, and its lower tail, which decides whether two containers together fall below the commitment, is thinner than the Gaussian's.

at 5,000. Table 9.2 gives the convergence at both inlet 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 inlet 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 56 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.

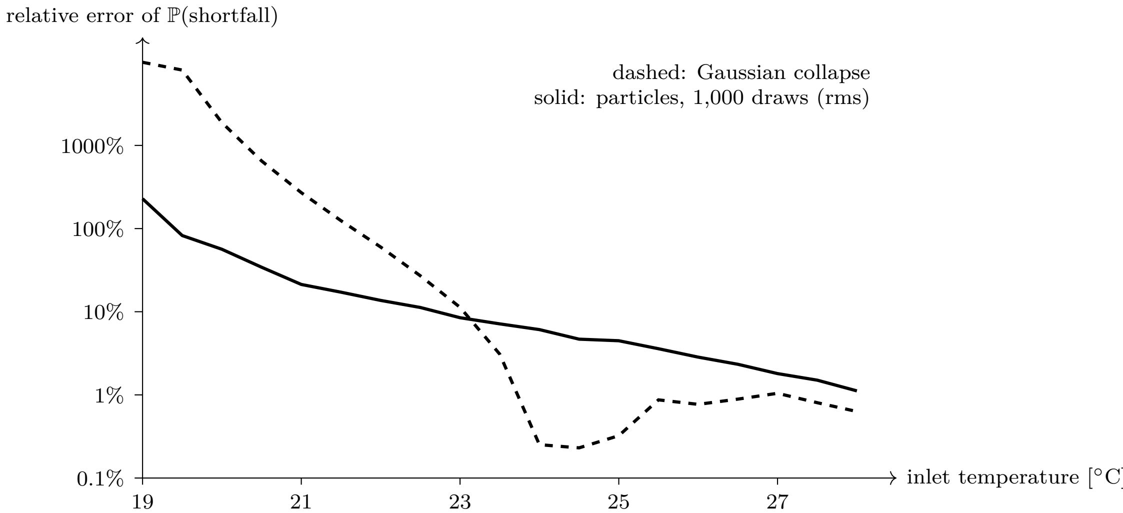


Figure 9.4: The relative error of the site's shortfall probability across inlet temperatures, on a logarithmic scale clipped at 10,000 percent. Dashed: the Gaussian collapse of the two containers' exact messages. Solid: particle messages of 1,000 draws each, rms over 100 repetitions. The collapse is better where shortfalls are common and fails without bound where they are rare.

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 cables, each with its contactor open with probability 0.01. Keeping every branch with up to three cables open 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 cables open, 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

Table 9.3: Pruning a hundred cables, each open with probability 0.01, by prior weight: the prior weight discarded by keeping every branch with at most k cables open, 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}

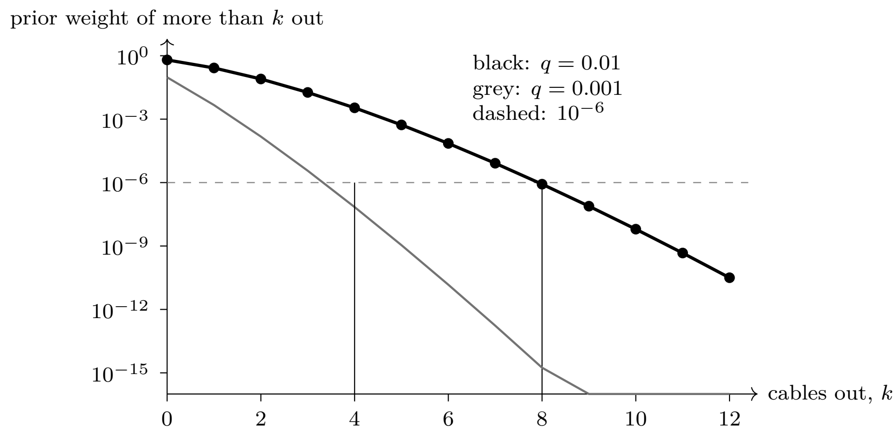


Figure 9.5: The prior weight of more than k of 100 cables open, each independently with probability q. Black: $q = 0.01$, which crosses 10^{-6} (dashed) at $k = 8$. Grey: $q = 0.001$, which crosses it at $k = 4$. Pruning by prior weight alone keeps every branch with k or fewer cables open.

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 cables cross the tolerance at $k = 8$ or at $k = 4$, and the branch count differs by a factor of fifty thousand.

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.

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

9.7 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, two thousand times too narrow on the heat path. 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 pack 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. A container whose message was one percent off at one inlet 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 [98] proved that the means are exact at any fixed point and gave the first sufficient conditions for convergence; Malioutov, Johnson and Willsky [60] 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 $\boldsymbol{\Lambda}\mathbf{z} = \boldsymbol{\eta}$, is Shental, Siegel, Wolf, Bickson and Dolev's [85]. 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 [15] operating between regions rather than between variables.

Variational inference and its relation to belief propagation are set out by Wainwright and Jordan [96], with the divergence argument for why mean-field approximations are too narrow. Expectation propagation is Minka's [64]; Bishop [10] presents both in textbook form and MacKay [58] gives the Laplace approximation its place in the family. Particle representations of messages come from the sequential Monte Carlo literature; Robert and Casella [77] is the general reference and Särkkä and Svensson [82] 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 [27]; 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 0.497 on the heat path's two inputs and a factor of two thousand 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 ring bus 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 container's message collapses to a Gaussian within one percent on a hot afternoon and misstates the site'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 cables 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 I_{12} and v_2 in the ring bus's row form with and without the sensor on I_{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 conductance 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 cable and uncertain conductances, 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 v_4 is within one percent of its spread?

Exercise 9.5. For L cables each open independently with probability q, find the number of cables open 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 $-1/\sqrt{d_id_j}$. On the ring bus, I_{12} sits in b_2 and c_{12} with unit coefficients, and v_2 in c_{12} and c_{23} with coefficient g: $\mathbf{\Lambda}_{I_{12}I_{12}} = 2w$, $\mathbf{\Lambda}_{v_2v_2} = 2g^2w$, $\mathbf{\Lambda}_{I_{12}v_2} = gw$, and the partial correlation is $-gw/\sqrt{2w \cdot 2g^2w} = -\frac{1}{2}$, independent of w and of g, which is the script's -0.500 at every width. The width drops out because every entry scales with w. With the shunt of weight w_m on I_{12}, $\mathbf{\Lambda}_{I_{12}I_{12}} = 2w + w_m$ and the partial correlation is $-1/\sqrt{2(2 + w_m/w)}$; at a physics width of 0.5 amperes and a shunt of 0.8, $w_m/w = 0.39$ and the value is $-1/\sqrt{4.78} = -0.457$. Now the width matters, because the shunt's weight does not scale with it; but only the ties at shunted variables are diluted.

Exercise 9.5. The number of cables open is binomial with $L = 100$ and $q = 0.01$, so the prior weight of having more than k open 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 pack 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 at machine precision after the first sweep and nothing moves afterward. The walk-summability radius of the port-level precision is 0.30, 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 racks 4, 5 and 6 and its three cables, rack 6 of each copy is tied to rack 4 of the next, rack 4 of the first copy is the reference, and every rack carries a prior of spread 8 amperes. Eliminating each region's interior rack leaves the port-level precision on the potentials at racks 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 conductance from 1 to 30 amperes per millivolt, and the iteration diverges in every case. A region's two ports are tied tightly through its own cables, 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 rack priors couple each potential 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.

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 storage containers that share a coolant inlet, 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.

One note on the first example before it starts. Its numbers are read from a published study of a three-line transmission model, which is where this comparison was first run and measured carefully enough to quote. It is kept here in its own domain, for two reasons. The counts being compared — grid nodes, draws, physics solves — are properties of the conservation graph and its separators, not of what flows along the edges, so a storage reader loses nothing by reading them on a network of lines; and a comparison of methods is worth more when its measurements come from a study that was run, published and can be rerun than when they are re-derived on a fresh model to make the domain match. The second example, in §10.7, is a storage site.

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

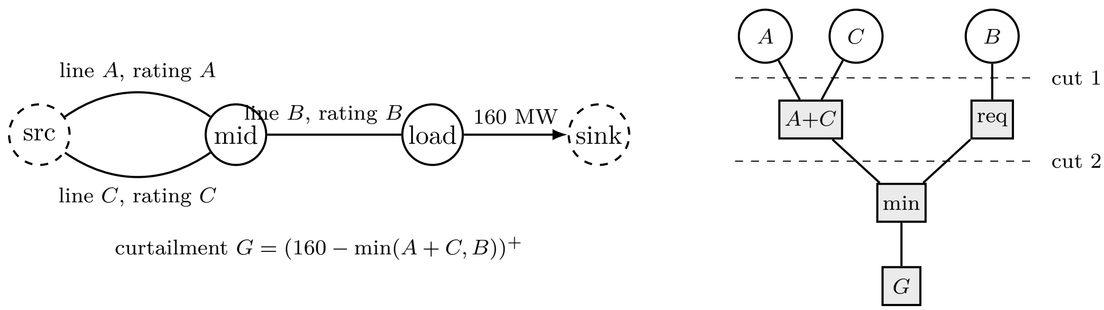


Figure 10.1: Left: the three-line network. Two lines in parallel feed a middle tap; one line feeds the load. Right: the dependency graph of the compiled model, inputs at the top and the curtailment at the bottom. Cut 1 separates the three ratings from everything below; cut 2 separates (A+C, req), two quantities, from the curtailment. Each cut is a separator in the sense of Chapter 5, and §10.6 counts what each saves.

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.

10.2 The problem

Two lines, A and C, run in parallel from a source to a middle tap; a third line, B, runs from the middle tap 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

$$G \;=\; \bigl(160 - \min(A + C,\; B)\bigr)^+, \tag{10.1}$$

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 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.

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 $G_1, \ldots, G_N$ 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.*

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]	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

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 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

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 $x = (x_1, \ldots, x_d)$ 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*

$$\mathbb{E}[G \mid X_u = x_u] \;=\; \sum_{x_{-u}} \Bigl(\prod_{i \notin u} w_i(x_i) \Bigr) G(x_u, x_{-u}), \tag{10.2}$$

a partial sum over values already solved, and $\operatorname{Var}\bigl(\mathbb{E}[G \mid X_u]\bigr)$ *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.*

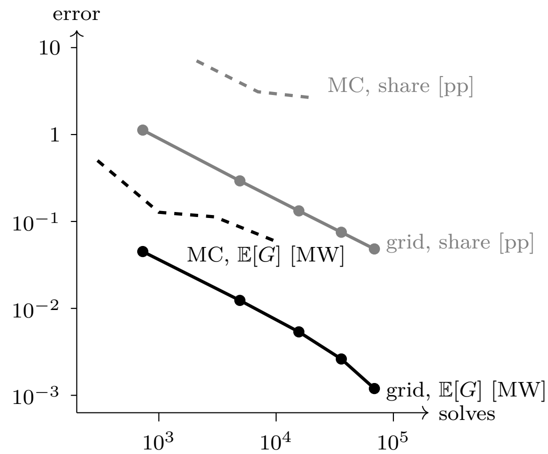


Figure 10.2: Error against solves on the three-line problem, from the committed study results. Solid: product grid; dashed: Monte Carlo. Black: error of the mean in megawatts; grey: error of line B's Shapley share in percentage points. The grid's attribution rides on its mean because it costs no further 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 tap, 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*

$$\mathbb{E}[\phi(G)] \;=\; \sum_{z} \phi\big(h(z)\big)\, q(z), \qquad q(z) = \sum_{x:\, c(x)=z} p(x), \tag{10.3}$$

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
$\{A{+}C,\ \mathrm{req}\}$	2	85	297	637	1,105	1,701
$\{A{+}C,\ B\}$	2	153	561	1,225	2,145	3,321
$\{A,\ C,\ \mathrm{req}\}$	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 storage containers

A site is committed to 2,100 kilowatts, supplied by two storage containers. Each container has three racks in parallel, each able to deliver between 350 and 450 kilowatts, uniform and independent. The racks are lithium-ion, whose deliverable power falls as the coolant they are fed warms, here by 1.5 percent per degree above 18°C — the temperature above which this site's chillers stop holding the cells at their reference and the battery management system begins cutting the current limit. The inlet temperature T is uncertain, uniform on $[18, 28]$°C, and it is the same for both containers, which share one chiller loop. Each container delivers through its own export path, whose rating L_k is uncertain, uniform on [1,056, 1,257] kilowatts. A container delivers the smaller of its derated rack power and its export rating, and the site's balance leaves

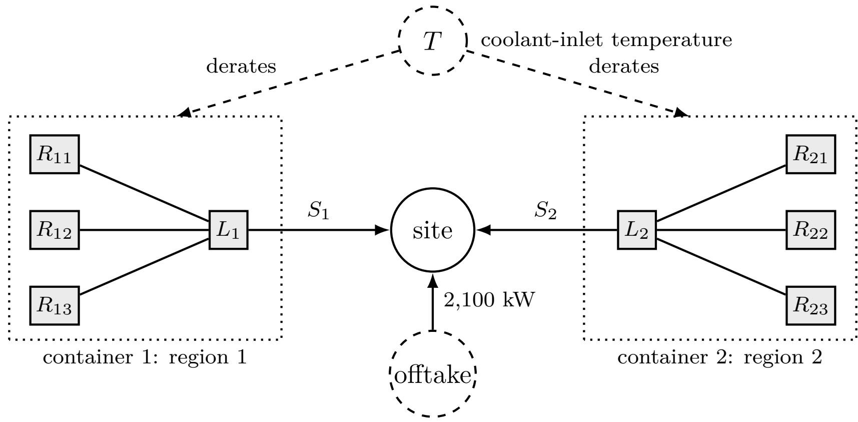


Figure 10.3: Two storage containers serving one site commitment. Each container has three racks in parallel, deliverable powers R_{kj}, and an export path whose rating L_k caps what it delivers; S_k is the container's delivered power and its port to the site. The coolant-inlet temperature T derates every rack in both containers. Given T the containers are independent, so the separator between them is $\{T, S_1, S_2\}$, not the ports alone.

the rest unmet:

$$S_k = \min\Big((1 - \alpha(T - 18)) \sum_{j=1}^{3} R_{kj},\ L_k\Big), \qquad U = (2{,}100 - S_1 - S_2)^+. \tag{10.4}$$

The shortfall U is power the site has promised and cannot deliver, and it shows up as a missed dispatch. There are nine uncertain inputs: six rack limits, two export ratings and the inlet temperature. Figure 10.3 draws the containers and the cut.

The dependency graph has the shape of the three-line model's, twice over, with one node more. Inside each container the rack limits enter only through their sum, a cut of one dimension. The container's delivered power is its port to the site, and the inlet temperature feeds both containers. Given the inlet temperature the containers are independent, and each container's message to the site 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*

$$\mathbb{E}[\phi(G)] \;=\; \sum_w p(w) \sum_{s_1, s_2} \phi(h(s_1, s_2))\, q_1(s_1 \mid w)\, q_2(s_2 \mid w), \qquad q_k(s \mid w) = \sum_{x_k:\ s_k(w, x_k) = s} p_k(x_k). \tag{10.5}$$

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

Table 10.4: The two storage containers. For each grid size: the solves of the full product grid on nine inputs; the container 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 container solves, by Proposition 10.2.

n	full grid	container solves	$\mathbb{E}[U]$ error [kW]	$\mathbb{P}(U > 0)$ error	Monte Carlo rms [kW]
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

inside each region, which gives $\mathbb{E}[\phi(G) \mid W = w]$ as the inner double sum; averaging over w 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 $2n^2(3n - 2)$ container solves, 4,050 at $n = 9$. For each of the n inlet values each container is enumerated on its own cut: $3n - 2$ values of the rack sum, from two convolutions, times n export ratings. The combination $U = (2{,}100 - S_1 - S_2)^+$ is the site's power 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 $\mathbb{E}[U] = 12.588677026$ kilowatts; the region route gives the same number from 650 container solves, to 2.5×10^{-14}. The reference is a region computation at $n = 161$: $\mathbb{E}[U] = 10.248$ kilowatts, $\mathbb{P}(U > 0) = 0.195$ and a standard deviation of 27.6 kilowatts. Twenty million Monte Carlo draws agree with it to 0.008 kilowatts.

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 kilowatts 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 container solves, and a container 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 inlet temperature and two per container once the rack limits are summed. That is past the crossover at four, and the shortfall 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 inlet nodes with 33 points on the other inputs still leave 0.40 kilowatts, and 33 inlet 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 inlet 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 containers never fall short. At 24.25°C the expected shortfall is 4.5 kilowatts with a 15 percent chance of any; at 28°C it is 66 kilowatts with an 84 percent chance. The nine values cost nothing beyond the 4,050 container solves, and they are within 0.27 kilowatts rms of the

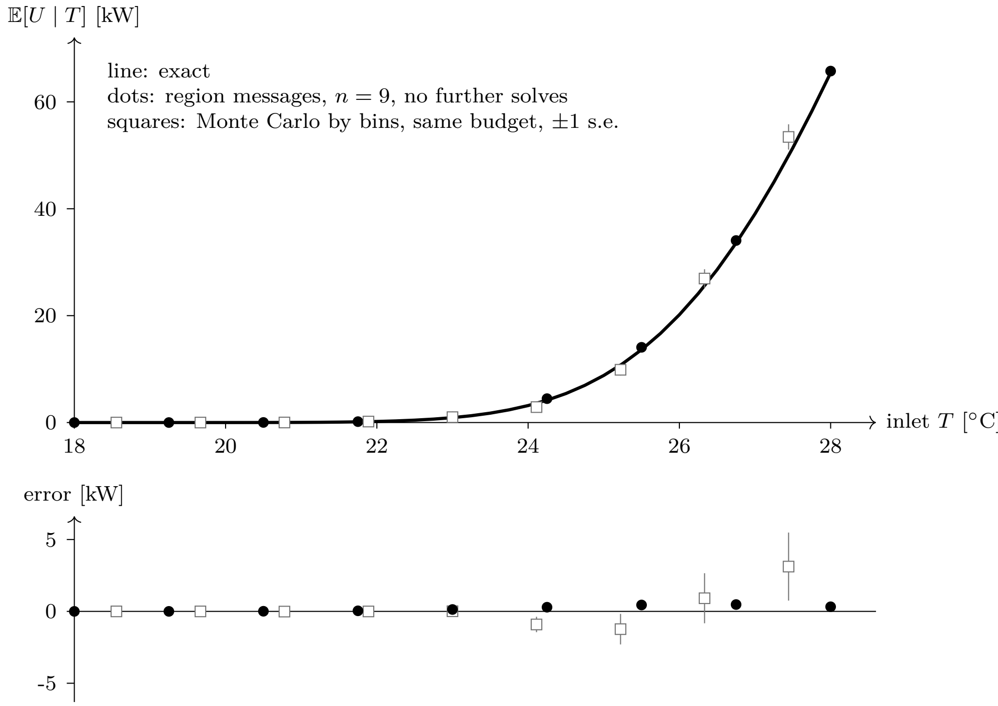


Figure 10.4: The expected shortfall given the coolant-inlet temperature, for the two storage containers. Top: the line is exact, from a region computation at $n = 33$. The dots are the nine values held by the region computation at $n = 9$, which cost nothing beyond its 4,050 container solves. The squares are Monte Carlo at the same 4,050 draws sorted into nine inlet 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.

fine computation. A Monte Carlo estimate of the same curve sorts its draws into nine inlet bins 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 kilowatts, 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 $t = 100$ kilowatts 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 $t = 150$ kilowatts, 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 inlet temperature out of it: compute each container'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 kilowatts 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 container's delivered power has mean 1,092 kilowatts and standard deviation 56.0; the part of its variance that the inlet temperature explains is 1,324 square kilowatts, so the two containers' deliveries are correlated at 0.42. The standard deviation of their sum is 94.5 kilowatts with the shared inlet temperature and 79.2 without it. The shortfall is the tail of that sum below 2,100 kilowatts, so a thinner tail means a smaller shortfall. A warm inlet derates both containers at once. A model that lets one container's bad hour 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 a case study of Part VI runs it on a network with 73 taps 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 two containers; no further solves against $6N$ for the second question; about thirteen to one for the containers' 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 temperature 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 two-container 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 inlet temperature, from nine nodes to 33, cuts the containers' error of the mean from 0.59 to 0.23 kilowatts at 14,850 container 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 a case study of Part VI, 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 [77] treat both sides. Variance-based attribution and the Shapley share are Sobol's [86], Owen's [69] and Song, Nelson and Staum's [87]; the pick-freeze estimator and its $(d+4)N$ or $7N$ solve count for $d = 3$ are Saltelli's [81]. 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 two containers of §10.7 were built for this chapter; their constants are typical of a utility-scale storage site 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-tap network.
- On two storage containers with nine inputs, region messages conditioned on the shared inlet 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 containers' 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 kilowatts 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 two containers 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 kilowatts 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 container 2's grid over its export rating by the exact uniform distribution, so that given the inlet, container 2's rack sum and container 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 container 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 $\mathbb{P}(B \geq 160 - t) = (20 + t)/40$. 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 $70 + ih$, $i = 0, \ldots, n - 1$, with $h = 50/(n - 1)$. A sum $a_i + c_j = 140 + (i + j)h$ depends only on $i + j$, which runs over $0, \ldots, 2n - 2$: $2n - 1$ 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}.

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 $\boldsymbol{\Lambda}$, information $\boldsymbol{\eta}$, mean $\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$, covariance $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$, with $\boldsymbol{\Lambda}$ 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* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *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*

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{k}\,(y - \mathbf{h}^\mathsf{T}\boldsymbol{\mu}), \qquad \boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \mathbf{k}\,\mathbf{s}^\mathsf{T}, \qquad \mathbf{s} = \boldsymbol{\Sigma}\mathbf{h}, \quad \mathbf{k} = \frac{\mathbf{s}}{\sigma_y^2 + \mathbf{h}^\mathsf{T}\mathbf{s}}. \tag{11.1}$$

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 $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \mathbf{s}\mathbf{s}^\mathsf{T}/S$ inverts $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda} + w\mathbf{h}\mathbf{h}^\mathsf{T}$. Multiply, using $\boldsymbol{\Lambda}\boldsymbol{\Sigma} = \mathbf{I}$ and $\boldsymbol{\Lambda}\mathbf{s} = \mathbf{h}$:

$$\boldsymbol{\Lambda}'\boldsymbol{\Sigma}' \;=\; \mathbf{I} + w\,\mathbf{h}\mathbf{s}^\mathsf{T} - \frac{(1 + w\,\mathbf{h}^\mathsf{T}\mathbf{s})}{S}\,\mathbf{h}\mathbf{s}^\mathsf{T} \;=\; \mathbf{I} + w\,\mathbf{h}\mathbf{s}^\mathsf{T} - w\,\mathbf{h}\mathbf{s}^\mathsf{T} \;=\; \mathbf{I},$$

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, $\boldsymbol{\mu}' = \boldsymbol{\Sigma}'(\boldsymbol{\eta} + wy\mathbf{h})$. The first part is $\boldsymbol{\Sigma}'\boldsymbol{\eta} = \boldsymbol{\mu} - \mathbf{s}\,(\mathbf{h}^\mathsf{T}\boldsymbol{\mu})/S$. The second uses

$\mathbf{\Sigma}'\mathbf{h} = \mathbf{s} - \mathbf{s}\,(\mathbf{h}^\mathsf{T}\mathbf{s})/S = \mathbf{s}\,\sigma_y^2/S$, so $wy\,\mathbf{\Sigma}'\mathbf{h} = y\,\mathbf{s}/S$. 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 $\mathbf{s} = \mathbf{\Sigma}\mathbf{h}$, and it is not computed by forming $\mathbf{\Sigma}$. It is the solution of $\mathbf{\Lambda}\mathbf{s} = \mathbf{h}$ 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 $\mathbf{\Sigma}$ implicit: hold the factorization of the original $\mathbf{\Lambda}$ and a list of the pairs $(\mathbf{k}_i, \mathbf{s}_i)$ from the updates so far. Then any product $\mathbf{\Sigma}'\mathbf{v}$ 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 $y_1, \ldots, y_r$ 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*

$$\log p(y_1, \ldots, y_r) \;=\; -\tfrac{1}{2}\sum_{i=1}^{r}\Big(\log 2\pi s_i + \frac{e_i^2}{s_i}\Big). \tag{11.2}$$

Proof. By the chain rule of probability, $p(y_1, \ldots, y_r) = \prod_i p(y_i \mid y_1, \ldots, y_{i-1})$. 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 heat path, one reading at a time. Return to the module heat path 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, T_2 is predicted at 30.000 ± 2.291 degrees. The reading is 30.4: innovation $+0.400$, standardized $+0.17$, unremarkable. One rank-one update gives $G = 20.610 \pm 1.952$ watts and $T_a = 20.076 \pm 0.900$ degrees, the second column of Table 3.2. Now T_1 is predicted at 34.503 ± 0.890, the virtual sensor of §3.4 with its spread now including the instrument's. The reading is 34.6: innovation $+0.097$, standardized $+0.11$. The second update gives $G = 20.750 \pm 1.470$, $T_a = 20.045 \pm 0.852$, the third column. The batch posterior with both readings agrees with the two sequential updates to seven decimals in the mean and eight in the covariance, the residual being the conditioning of the tight physics rows and not the method.

Figure 11.1 adds a third reading, $T_a = 19.85$ with accuracy 0.5, and folds the three in three orders. In every order the dissipation ends at 20.980 ± 0.894 watts, the batch posterior with all three readings. The paths differ. Read the coolant first and the dissipation does not move at all, 20.000 ± 4.000, because under the prior the coolant temperature and the dissipation are independent and the reading carries nothing about G. Read T_2 next and the dissipation drops to ± 1.272, better than the ± 1.952 that T_2 gave alone, because the coolant reading has pinned the one quantity that T_2 confounds with the dissipation. Read T_1 first and the dissipation is at 20.739 ± 1.483, after which the coolant reading removes 60 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.

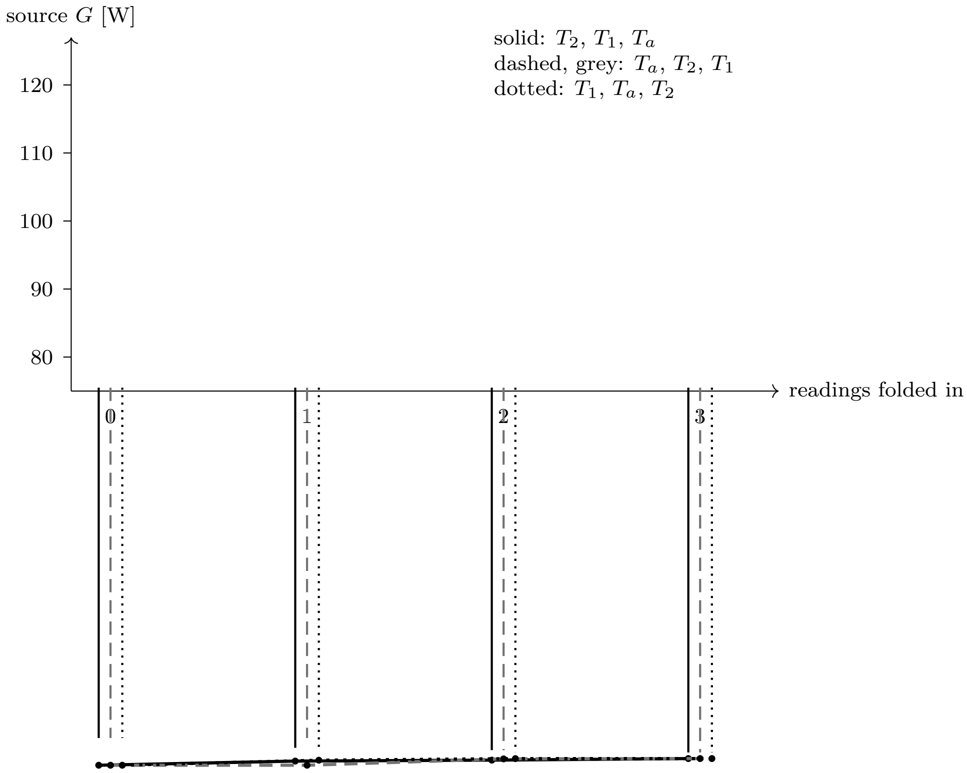


Figure 11.1: The dissipation of the module heat path as three readings are folded in, in three orders: $T_2 = 30.4$, $T_1 = 34.6$ and $T_a = 19.85$, each of accuracy 0.5 kelvin. Dots are posterior means and bars one posterior standard deviation; the three orders are offset slightly for legibility. Every order ends at the batch posterior. Read first, the coolant moves nothing, because under the prior the coolant temperature and the dissipation are independent.

11.5 Worked example: a stream of shunt readings

The two-region pack of Chapter 8 carries six shunts, on cables 1–2, 2–3, 3–4, 4–5, 4–6 and 5–6, each of accuracy 0.8 amperes. They report in rounds, one after another, and the rack currents do not change between rounds: twenty-four readings of a static state. The state is a draw from the prior, with rack currents at racks 2 through 6 of −55.52, −24.11, −12.04, −52.93 and −11.14. From the third round the shunt on cable 4–5 reads 4.0 amperes 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 7.8 amperes, and ordinary in their own, none beyond 1.8 spreads, because the predicted spread of a first reading is mostly the prior's uncertainty about the current: 7.6 amperes for cable 1–2. By the second round every current is known to about the shunts' accuracy, and the predicted spreads have fallen to between 0.98 and 1.13. The drifting shunt's first biased reading, the sixteenth, lands 4.4 spreads high, and its second, the twenty-second, 4.0. No other reading in the stream exceeds 1.8.

The running sum tells the same story less sharply. It crosses the quantile at the sixteenth

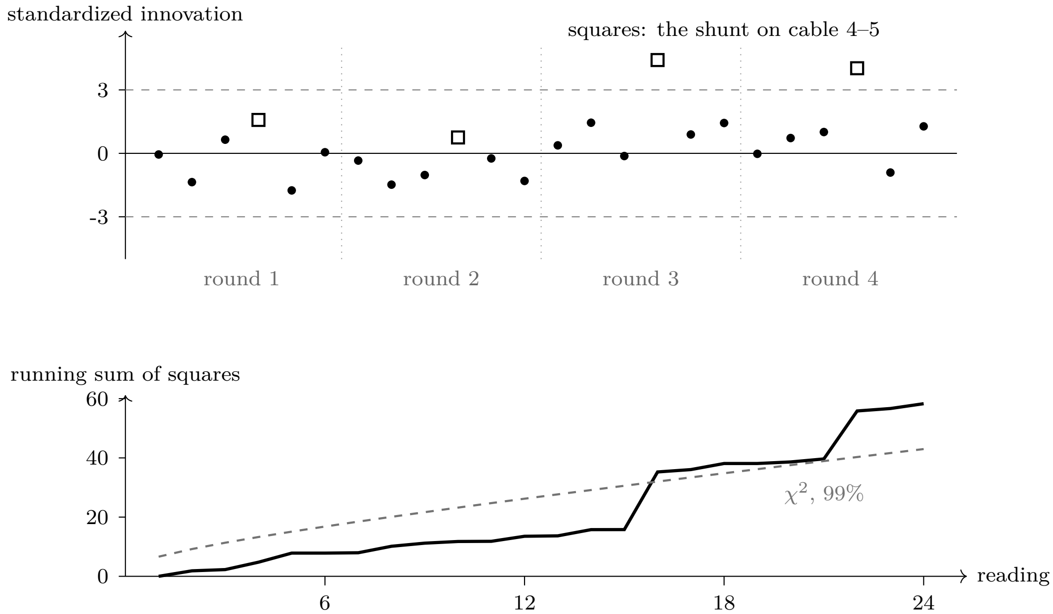


Figure 11.2: The two-region pack's six shunts read in four rounds, a static state. Top: the standardized innovation of each reading; squares mark the shunt on cable 4–5, which reads 4.0 amperes high from the third round, and the dashed lines are at ± 3. Bottom: the running sum of squared standardized innovations (solid) against the 99 percent point of the chi-squared distribution with as many degrees of freedom as readings (dashed).

reading and stays above it for the rest of the stream. 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 rack current at rack 5 at -54.36 ± 0.56, 2.5 standard deviations from the true -52.93. Withholding the two flagged readings gives -52.24 ± 0.66, 1.0 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 -59.201164. 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 ring bus and the linearized heat path of this book are, the check always passes and the update is always exact. For a model with a curved closure the check fails only when a reading moves the state by a sizable fraction of the region over which the closure curves, which under normal operation is rarely.

A worked example on a nonlinear closure. Take the plate closure of Chapter 6, $G = c\,\Delta T^{1.25}$ with $c = 1.1247$ watts per kelvin to the 1.25, a prior of 20 ± 4 watts on the dissipation, a physics row of width 0.01, and a thermistor reading of the plate rise ΔT with accuracy 0.1 kelvin. Before any reading the Laplace posterior sits at $\Delta T = 10.000$, $G = 20.00 \pm 4.00$, ΔT to within ± 1.60, and the two are correlated at 1.0000: the Gaussian lies along the tangent to the convection law at the mode. Figure 11.3 draws three readings of the rise, 10.3, 13.0 and 18.0 kelvin. For the first, the rank-one update gives $G = 20.747$ against the exact new mode 20.750 ± 0.252, an error of a hundredth of a posterior standard deviation. For the second it gives 27.471 against 27.728 ± 0.267, one standard deviation off. For the third it gives 39.922 against 41.585 ± 0.289, nearly six off. The update moves along the tangent, and the convection law has bent away from it. In every case one Gauss–Newton step from the updated mean lands on the exact mode to 10^{-5} watts, a ten-thousandth of a posterior spread.

The two checks of the procedure coincide here, because the reading is of ΔT itself and much sharper than the prior: the standardized innovations are 0.19, 1.87 and 4.99, and so are the shifts in units of the prior's standard deviation of ΔT. The first of the bad cases would not raise an alarm as a fault. The shift is what matters, and a shift of 1.87 prior standard deviations already cost a posterior one. Re-linearize whenever the shift exceeds about a half 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 $g = \mathbf{c}^\top \mathbf{z}$, the posterior variance after a reading of $\mathbf{h}^\top \mathbf{z}$ with accuracy σ_y is

$$\mathrm{Var}'(g) \;=\; \mathrm{Var}(g) - \frac{(\mathbf{c}^\top \boldsymbol{\Sigma} \mathbf{h})^2}{\sigma_y^2 + \mathbf{h}^\top \boldsymbol{\Sigma} \mathbf{h}}, \tag{11.3}$$

whatever the reading turns out to be. It follows from the covariance update of equation (11.1) by multiplying on both sides by $\mathbf{c}$: $\mathbf{c}^\top \boldsymbol{\Sigma}' \mathbf{c} = \mathbf{c}^\top \boldsymbol{\Sigma} \mathbf{c} - (\mathbf{c}^\top \mathbf{k})(\mathbf{s}^\top \mathbf{c})$, and $\mathbf{c}^\top \mathbf{k} = \mathbf{c}^\top \boldsymbol{\Sigma} \mathbf{h} / (\sigma_y^2 + \mathbf{h}^\top \boldsymbol{\Sigma} \mathbf{h})$. 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 covariances before and after. A single reading gives

$$\tfrac{1}{2} \log \frac{\det \boldsymbol{\Sigma}}{\det \boldsymbol{\Sigma}'} \;=\; \tfrac{1}{2} \log \frac{\det \boldsymbol{\Lambda}'}{\det \boldsymbol{\Lambda}} \;=\; \tfrac{1}{2} \log\Bigl(1 + \frac{\mathbf{h}^\top \boldsymbol{\Sigma} \mathbf{h}}{\sigma_y^2}\Bigr). \tag{11.4}$$

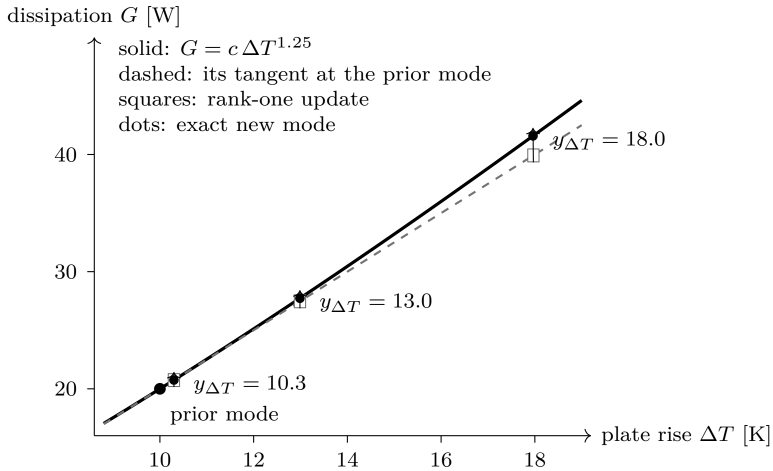


Figure 11.3: The plate closure $G = c\,\Delta T^{1.25}$ and three thermistor readings of the plate rise. The Laplace posterior before the readings lies along the tangent at the prior mode (dashed). The rank-one update of that Gaussian moves along the tangent (squares), while the exact new modes lie on the curve (dots). The error grows with the shift, from a hundredth to one and then nearly six posterior standard deviations. One Gauss–Newton step from each square reaches its dot.

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 heat path, a sensor on the coolant is worth nothing to the dissipation 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 temperature is worth most of the dissipation's variance, and a direct heat-flow meter, even a poor one, nearly all of it. Now read T_2 and ask again. The dissipation's spread is 1.952 watts, and the sensor that would reduce it most, among the two temperatures, is the one on the coolant, by 58 percent against the module thermistor's 43, because the reading has correlated the dissipation with the coolant temperature at -0.868 and a reading of one is now a reading of the other. The sensor that was worthless has become the better of the two. 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 pack the target is the rack current at rack 5, and the candidates are shunts on each cable. The shunt on the cable into rack 5 from rack 4 is worth 79 percent of that rack current's variance; the shunt on the tie cable, two racks away, a third; the shunt on cable 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 shunt has been read. After the shunt on cable 4–5, the shunt on cable 5–6, worth 48 percent of the prior variance, is worth 91 percent of what remains, and the shunt on cable 4–6 is worth 78 percent; the shunts in region A fall to a few percent. The balance at rack 5 makes its draw the difference of the two currents at the rack. With one current known, the other is the draw, and the shunt that reads it directly becomes decisive. The whole-state measure of equation (11.4) orders the shunts differently. The shunt on

Table 11.1: The value of the next sensor, computed before any reading. Top: the heat path, target G, prior spread 4 watts. Bottom: the two-region pack, target q_5, prior spread 8 amperes, shunts of accuracy 0.8.

heat path, sensor on	sd of G after [W]	reduced by
under the prior		
T_1 (acc. 0.5 K)	1.483	86%
T_2 (acc. 0.5 K)	1.952	76%
T_a (acc. 0.5 K)	4.000	0%
Q_{2a} (acc. 0.2 W)	0.200	100%
after T_2 is read (sd 1.952)		
T_1	1.470	43%
T_a	1.272	58%
Q_{2a}	0.199	99%

pack, shunt on	sd of q_5 after [A]	reduced by
I_{12}	7.489	12%
I_{23}	7.171	20%
I_{34}	6.537	33%
I_{45}	3.702	79%
I_{46}	7.171	20%
I_{56}	5.777	48%

the tie cable is worth 2.85 nats to the whole state, the most of the six, and the shunt on cable 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 heat path has six unknowns and the comparison is not worth timing. Take instead a mesh of $m \times m$ cells, each tied to its four neighbours through a conductance and to the cold plate through another, with an uncertain heat source in every cell; eliminating the heat flows leaves one soft row per cell on the temperatures, and the precision is sparse with the mesh's own structure, which in two dimensions has fill. Ten thermistors at random cells report in turn. Table 11.2 times the two ways of folding them in.

One rank-one update costs between a thirtieth and a fiftieth of a factorization across three decades of size, the ratio being that of a solve to a factorization, which for a two-dimensional

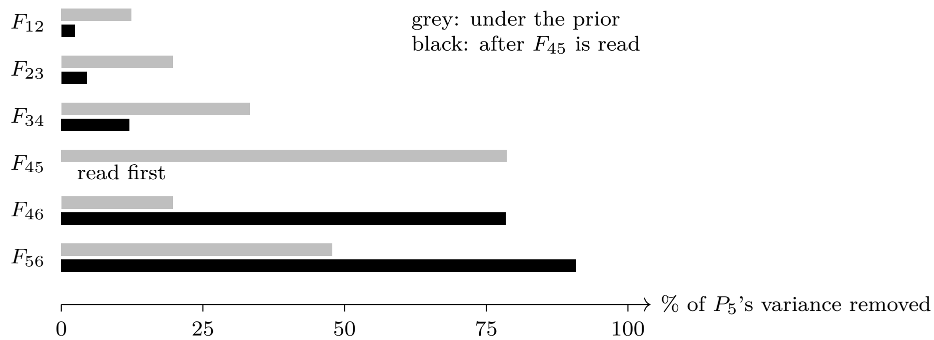


Figure 11.4: The value of each shunt to the rack current at rack 5, as the percentage of its current variance the shunt would remove, computed before any reading. Grey: under the prior. Black: after the shunt on cable 4–5 has been read. The shunt on cable 5–6 goes from second to decisive, because the draw at rack 5 is the difference of the two currents at the rack.

Table 11.2: A mesh of $m \times m$ cells: 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.

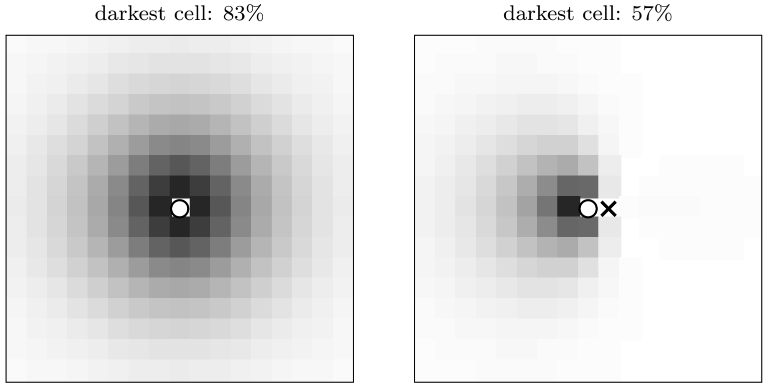
m	unknowns	factor nonzeros	one factorization	one update	ratio	ten readings, ratio
30	900	7.0×10^4	4.8 ms	0.15 ms	33	7.8
100	10,000	2.0×10^6	119 ms	3.0 ms	39	10.3
300	90,000	3.3×10^7	4.46 s	85 ms	52	7.0

sparse structure grows slowly with size. Over ten readings the factor-once route is about eight times faster than refactoring, the first factorization being paid either way; over a hundred it would be some forty 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 under a tenth of a second, which is faster than a battery management system reports.

11.9 Worked example: where to put the next thermistor

A cell in the middle of a large module has no thermistor of its own — there is nowhere to put one — and its temperature is the target. Model the module as the mesh of §11.8 at 30×30 cells with stronger coupling: conductance 4 watts per kelvin between neighbouring cells, a further 0.25 watts per kelvin from each cell to the cold plate, and an uncertain heat generation of standard deviation 0.5 watts in every cell. The uninstrumented cell is at the centre. Thermistors of accuracy 0.025 kelvin can go on any other cell. Under the prior the centre's temperature has a standard deviation of 0.144 kelvin, and it is correlated with each neighbour at 0.92.

Figure 11.5(a) maps the value of a thermistor at each cell 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 cell two steps away 74.0, three steps 59.1, and seven steps 23.3. The mesh spreads the correlation over several cells, so the value falls off slowly with distance. Greedy placement takes a neighbour first, and the centre's standard deviation



(a) under the prior (b) after the first shunt

Figure 11.5: The value of a thermistor on each cell near the centre of a 30×30 module, for the temperature of the centre cell (ringed), where no thermistor 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 thermistor 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.

drops to 0.060 kelvin.

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 cells beside the first thermistor have lost most of their value, and those on the far side have kept it. The first reading told the model the temperature on one side of the centre, and what remains uncertain is the gradient across it, which the opposite side measures. The second thermistor brings the centre to 0.040 kelvin. 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 thermistor's value can rise as others are placed: readings on opposite sides of a cell are worth more together than apart, the explaining away of Proposition 5.2 again.

The four thermistors leave the centre at 0.033 kelvin, and no thermistor outside the centre can do much better. With its four neighbours known exactly the centre would still be uncertain by 0.030 kelvin, and with every other cell of the module known exactly, by 0.028. What remains is the centre cell's own heat generation. It enters only the centre's balance, where it is divided by the total conductance at the cell, $4 \times 4 + 0.25$ watts per kelvin, which sets the scale: $0.5/16.25$ is 0.031 kelvin. 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 uninstrumented cell is limited not by where the thermistors 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 fifty 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 shunt on cable 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 crossed its quantile only at the sixteenth reading, where the per-reading test had already flagged the fault outright. 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 temperature where there is no sensor, is this reading a fault or a bad thermistor, what would happen if that cable were opened, which input drives the risk, which components 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 [32]. The update (11.1) is the measurement update of the Kalman filter [40], whose gain and innovation are the same objects; Särkkä and Svensson [82] 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 [22]; 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 [54] defined the value of an experiment as its expected information; Chaloner and Verdinelli [14] review the Bayesian theory, and the greedy variance-reduction criterion of §11.7 is its simplest instance. The reading of the criterion on the heat path, that the coolant sensor's value goes from nothing to the best of the three 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 [82] 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 heat path'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 heat path the coolant sensor is worth nothing before T_2 is read and 58 percent of the remaining variance after; on the pack the shunt nearest the target is worth most.
- On a mesh of ninety thousand unknowns one update costs a fiftieth of a factorization and takes under a tenth of a second.
- Standardized innovations of an adequate model are independent standard Gaussians, and their log densities sum to the evidence. On the pack's stream a shunt drifting by five times its accuracy was flagged at 4.4 spreads on its first biased reading; folding it in would have moved the rack current at rack 5 by 2.5 standard deviations.
- On a nonlinear closure the rank-one update errs by nearly six posterior standard deviations when a reading shifts the state by 5 prior ones. One Gauss–Newton step repairs it.
- On a mesh of cells the best place for a thermistor near an uninstrumented cell is a neighbour, and after it the opposite neighbour. Four thermistors bring the uninstrumented cell to 0.033 kelvin, against a floor of 0.028 that no outside thermistor can pass.
- A sensor's value depends on the target and on what has been read: after the shunt on cable 4–5, the shunt on cable 5–6 goes from 48 to 91 percent of that rack current's remaining variance.

Exercises

Exercise 11.1. Derive equation (11.1) from the information-form update $\mathbf{\Lambda}' = \mathbf{\Lambda} + w\mathbf{h}\mathbf{h}^\top$, $\boldsymbol{\eta}' = \boldsymbol{\eta} + wy\mathbf{h}$ 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 heat path, read T_a first (say 19.85 with accuracy 0.5), then T_2, then T_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 pack, choose two shunts greedily to minimize the posterior variance of q_5, then choose the best pair by exhaustive search over the fifteen pairs, and compare. Repeat with the target changed to the tie current I_{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 $\mathbf{S} = \mathbf{R} + \mathbf{H\Sigma H}^\mathsf{T}$. The claim is that $\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma H}^\mathsf{T}\mathbf{S}^{-1}\mathbf{H\Sigma}$ inverts $\mathbf{\Lambda}' = \mathbf{\Lambda} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H}$. Multiplying,

$$\begin{aligned}\mathbf{\Lambda}'\mathbf{\Sigma}' &= \mathbf{I} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H\Sigma} - \mathbf{H}^\mathsf{T}\mathbf{S}^{-1}\mathbf{H\Sigma} - \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}\mathbf{H\Sigma H}^\mathsf{T}\mathbf{S}^{-1}\mathbf{H\Sigma} \\ &= \mathbf{I} + \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}(\mathbf{S} - \mathbf{R} - \mathbf{H\Sigma H}^\mathsf{T})\mathbf{S}^{-1}\mathbf{H\Sigma} \;=\; \mathbf{I},\end{aligned}$$

where the second line inserts $\mathbf{SS}^{-1}$ 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: $\boldsymbol{\mu}' = \boldsymbol{\mu} + \mathbf{\Sigma H}^\mathsf{T}\mathbf{S}^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu})$. The cost is r solves for $\mathbf{\Sigma H}^\mathsf{T}$ 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 $\mathbf{\Lambda} + \sum_i w_i \mathbf{h}_i \mathbf{h}_i^\mathsf{T}$ 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 dissipation ends at 20.980 ± 0.894 watts and the coolant temperature at 19.900 ± 0.431 degrees, the batch posterior with all three rows; Figure 11.1 draws the three paths. The values for the dissipation, as percentages of its current variance, are as follows. Under the prior, T_1 is worth 86.2, T_2 76.2 and T_a nothing, so greedy placement takes T_1. After it, T_a is worth 60.5 and T_2 1.7, so greedy takes T_a. After both, T_2 removes 7.9 percent. The greedy order is T_1, T_a, T_2, and the sensor worth nothing at first is chosen second, because the first reading made the coolant temperature the largest remaining uncertainty about the dissipation. In the order the exercise prescribes, T_a first, the dissipation's variance is unchanged by the first reading, as its value of zero said it would be.

Exercise 11.4. For the rack current at rack 5, greedy placement takes the shunt on cable 4–5 first, worth 79 percent, and then the shunt on cable 5–6, which leaves a standard deviation of 1.118 amperes. Exhaustive search over the fifteen pairs finds the same pair best; the next are 4–5 with 4–6 at 1.720 and 4–6 with 5–6 at 1.744. For the tie current I_{34} greedy takes the shunt on the tie cable itself and then any other: every pair containing it leaves 0.797 or 0.799 amperes, about the shunt's own accuracy, and the second shunt hardly matters. On this pack greedy placement of two shunts is optimal for all seventeen variables as targets, for sums and differences of rack currents, and for the whole-state measure on the five rack currents.

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 shunt on a current that carries none of the target can be worthless alone and decisive in a pair. When candidates come in such pairs, search over pairs.

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, $\boldsymbol{\mu}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$. 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 $\mathcal{R}_b$. $\operatorname{diag}(\mathbf{w})$ 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 $w = 1/\sigma^2$; $\mathbf{\Omega}$ 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
$\mathrm{KL}(q \,\Vert\, p)$	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 a case study of Part VI
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 Part VI and Part VI
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 a case study of Part VI; 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 (a case study of Part VI)
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

$$\mathcal{N}(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma}) \;=\; (2\pi)^{-n/2}\,(\det\boldsymbol{\Sigma})^{-1/2}\exp\!\big(-\tfrac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\big). \tag{B.1}$$

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,

$$\mathcal{N}(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma}) \;=\; \exp\!\big(-\tfrac{1}{2}\mathbf{z}^\top\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^\top\mathbf{z} - \kappa\big), \qquad \kappa = \tfrac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} + \tfrac{n}{2}\log 2\pi - \tfrac{1}{2}\log\det\boldsymbol{\Lambda}. \tag{B.2}$$

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 $\boldsymbol{\eta}$,*

$$\int_{\mathbb{R}^n}\exp\!\big(-\tfrac{1}{2}\mathbf{z}^\top\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^\top\mathbf{z}\big)\,\mathrm{d}\mathbf{z} \;=\; (2\pi)^{n/2}\,(\det\boldsymbol{\Lambda})^{-1/2}\exp\!\big(\tfrac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}\big).$$

Proof. Complete the square: with $\mathbf{m} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$ the exponent is $-\frac{1}{2}(\mathbf{z}-\mathbf{m})^\top\boldsymbol{\Lambda}(\mathbf{z}-\mathbf{m}) + \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$. Factor $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^\top$ by Cholesky and substitute $\mathbf{w} = \mathbf{L}^\top(\mathbf{z}-\mathbf{m})$, so that $\mathrm{d}\mathbf{z} = \mathrm{d}\mathbf{w}/\det\mathbf{L}$ and the quadratic becomes $\mathbf{w}^\top\mathbf{w}$. The integral of $\exp(-\frac{1}{2}\mathbf{w}^\top\mathbf{w})$ 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 $-\frac{1}{2}\mathbf{z}^\top\boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\eta}^\top\mathbf{z}$ plus a constant is the*

Gaussian with precision $\boldsymbol{\Lambda}$ *and mean* $\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$. 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 $\mathbf{S}\boldsymbol{\Sigma}_u\mathbf{S}^\top$ 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 $\boldsymbol{\Sigma}$, written $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if $\mathbb{E}[\exp(\mathbf{t}^\top\mathbf{z})] = \exp(\mathbf{t}^\top\boldsymbol{\mu} + \frac{1}{2}\mathbf{t}^\top\boldsymbol{\Sigma}\mathbf{t})$ for every $\mathbf{t} \in \mathbb{R}^n$. It has the density (B.1) exactly when $\boldsymbol{\Sigma}$ is positive definite.

For a positive definite $\boldsymbol{\Sigma}$ the definition agrees with the density: multiplying equation (B.2) by $\exp(\mathbf{t}^\top\mathbf{z})$ 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* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *and* $\mathbf{w} = \mathbf{A}\mathbf{z} + \mathbf{b}$ *for an* $m \times n$ *matrix* $\mathbf{A}$*, then* $\mathbf{w} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b},\ \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$*. In particular every block of a Gaussian vector is Gaussian, with the corresponding blocks of* $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$*.*

Proof. $\mathbb{E}[\exp(\mathbf{s}^\top\mathbf{w})] = \exp(\mathbf{s}^\top\mathbf{b})\,\mathbb{E}[\exp((\mathbf{A}^\top\mathbf{s})^\top\mathbf{z})] = \exp\big(\mathbf{s}^\top(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}) + \frac{1}{2}\mathbf{s}^\top\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top\mathbf{s}\big)$. 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* $\mathrm{Cov}(\mathbf{a}, \mathbf{b}) = \mathbf{0}$*, then* $\mathbf{a}$ *and* $\mathbf{b}$ *are independent.*

Proof. With a zero cross-covariance the joint moment generating function is

$$\mathbb{E}\big[\exp(\mathbf{s}^\top\mathbf{a} + \mathbf{t}^\top\mathbf{b})\big] = \exp\big(\mathbf{s}^\top\boldsymbol{\mu}_a + \tfrac{1}{2}\mathbf{s}^\top\boldsymbol{\Sigma}_{aa}\mathbf{s}\big)\ \exp\big(\mathbf{t}^\top\boldsymbol{\mu}_b + \tfrac{1}{2}\mathbf{t}^\top\boldsymbol{\Sigma}_{bb}\mathbf{t}\big),$$

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 $\left(\begin{smallmatrix}\mathbf{M}/\mathbf{D} & \mathbf{0}\\ \mathbf{C} & \mathbf{D}\end{smallmatrix}\right)$, and the first factor adds $\mathbf{B}\mathbf{D}^{-1}$ times the second row to the first, restoring $\mathbf{A} = \mathbf{M}/\mathbf{D} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ 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 $\mathbf{W}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U}$ is invertible, then*

$$(\mathbf{A} + \mathbf{U}\mathbf{W}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\,(\mathbf{W}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\,\mathbf{V}\mathbf{A}^{-1}, \tag{B.3}$$

$$\det(\mathbf{A} + \mathbf{U}\mathbf{W}\mathbf{V}) = \det\mathbf{A}\,\det\mathbf{W}\,\det(\mathbf{W}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U}). \tag{B.4}$$

For a rank-one change, $\mathbf{U} = \mathbf{h}$, $\mathbf{V} = \mathbf{h}^\top$ and $\mathbf{W} = w$, these read

$$(\mathbf{A} + w\,\mathbf{h}\mathbf{h}^\top)^{-1} = \mathbf{A}^{-1} - \frac{w\,\mathbf{A}^{-1}\mathbf{h}\mathbf{h}^\top\mathbf{A}^{-1}}{1 + w\,\mathbf{h}^\top\mathbf{A}^{-1}\mathbf{h}}, \qquad \det(\mathbf{A} + w\,\mathbf{h}\mathbf{h}^\top) = \det\mathbf{A}\,(1 + w\,\mathbf{h}^\top\mathbf{A}^{-1}\mathbf{h}),$$

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, $\mathbf{M}/\mathbf{W}^{-1} = \mathbf{A} + \mathbf{U}\mathbf{W}\mathbf{V}$, so the upper left block of $\mathbf{M}^{-1}$ is $(\mathbf{A}+\mathbf{U}\mathbf{W}\mathbf{V})^{-1}$ and $\det\mathbf{M} = \det(\mathbf{A}+\mathbf{U}\mathbf{W}\mathbf{V})\,\det\mathbf{W}^{-1}$. Pivoting on the upper left block, $\mathbf{M}/\mathbf{A} = \mathbf{W}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U}$, so $\det\mathbf{M} = \det\mathbf{A}\,\det(\mathbf{W}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})$, 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 $\mathbf{H}$,*

$$(\mathbf{\Lambda} + \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{R}^{-1} = \mathbf{\Lambda}^{-1}\mathbf{H}^\top(\mathbf{R} + \mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^\top)^{-1}.$$

Proof. $(\mathbf{\Lambda} + \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H})\,\mathbf{\Lambda}^{-1}\mathbf{H}^\top = \mathbf{H}^\top + \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^\top = \mathbf{H}^\top\mathbf{R}^{-1}(\mathbf{R} + \mathbf{H}\mathbf{\Lambda}^{-1}\mathbf{H}^\top)$. 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*

$$\mathbf{u} \mid \mathbf{x} \;\sim\; \mathcal{N}(\boldsymbol{\mu}_u + \mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}(\mathbf{x} - \boldsymbol{\mu}_x),\;\; \mathbf{\Sigma}_{uu} - \mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xu}). \tag{B.5}$$

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 $\mathrm{Cov}(\mathbf{e}, \mathbf{x}) = \mathbf{\Sigma}_{ux} - \mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xx} = \mathbf{0}$, 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

$$\tfrac{1}{2}\log\det\mathbf{\Sigma} - \tfrac{1}{2}\log\det\mathbf{\Sigma}' \;=\; \tfrac{1}{2}\log\bigl(1 + \mathbf{h}^\mathsf{T}\mathbf{\Sigma}\mathbf{h}/\sigma^2\bigr), \tag{B.9}$$

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* $\mathbf{m}$*,*

$$\mathbb{E}\big[(\mathbf{z}-\mathbf{m})^\top \mathbf{A}(\mathbf{z}-\mathbf{m})\big] = \operatorname{tr}(\mathbf{A}\boldsymbol{\Sigma}) + (\boldsymbol{\mu}-\mathbf{m})^\top \mathbf{A}(\boldsymbol{\mu}-\mathbf{m}). \tag{B.10}$$

If $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ *and* $\mathbf{P}$ *is a symmetric idempotent matrix of rank* r*, then* $\boldsymbol{\epsilon}^\top \mathbf{P}\boldsymbol{\epsilon}$ *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 $\mathbb{E}[(\mathbf{z}-\boldsymbol{\mu})^\top \mathbf{A}(\mathbf{z}-\boldsymbol{\mu})] = \operatorname{tr}(\mathbf{A}\,\mathbb{E}[(\mathbf{z}-\boldsymbol{\mu})(\mathbf{z}-\boldsymbol{\mu})^\top]) = \operatorname{tr}(\mathbf{A}\boldsymbol{\Sigma})$, 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 $\mathbf{P} = \mathbf{Q}\mathbf{D}\mathbf{Q}^\top$ with $\mathbf{Q}$ orthogonal and $\mathbf{D}$ diagonal with r ones. By Lemma B.2, $\boldsymbol{\delta} = \mathbf{Q}^\top\boldsymbol{\epsilon}$ is again standard Gaussian, and $\boldsymbol{\epsilon}^\top \mathbf{P}\boldsymbol{\epsilon} = \sum_{i\le r} \delta_i^2$, 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* $\mathbb{R}^n$*,*

$$\mathrm{KL}(\mathcal{N}_0 \,\|\, \mathcal{N}_1) = \tfrac{1}{2}\Big[\operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) - n + \log\det\boldsymbol{\Sigma}_1 - \log\det\boldsymbol{\Sigma}_0\Big]. \tag{B.12}$$

Proof. From equation (B.1), $-\log\mathcal{N}_1(\mathbf{z}) = \frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\boldsymbol{\Sigma}_1 + \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_1)^\top\boldsymbol{\Sigma}_1^{-1}(\mathbf{z}-\boldsymbol{\mu}_1)$. Its expectation under $\mathcal{N}_0$ is, by equation (B.10), $\frac{n}{2}\log 2\pi + \frac{1}{2}\log\det\boldsymbol{\Sigma}_1 + \frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + \frac{1}{2}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^\top\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$. 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 $\operatorname{tr}(\boldsymbol{\Lambda}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Lambda}(\boldsymbol{\mu}-\boldsymbol{\mu}_0) - \log\det\boldsymbol{\Sigma}_0$, 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 $C(\mathbf{z}) = C(\mathbf{z}^\star) + \frac{1}{2}(\mathbf{z}-\mathbf{z}^\star)^\top\,\nabla^2 C^\star\,(\mathbf{z}-\mathbf{z}^\star) + O(\|\mathbf{z}-\mathbf{z}^\star\|^3)$. 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 [92].

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 a case study of Part VI 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

Each chapter of Part VI says which engine and plugin pieces its study used, and what one engine solve is in it: some studies run the engine's own solver or inference, and others run a plugin algorithm whose inner problem is a linear program or a linear network flow. Cost is counted in engine solves in both cases, and each chapter says which kind it counts. A table collecting them returns here when Part VI is complete.

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`)

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 [88].

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 [100]. 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 [74]; 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 [7]. *Source*: textbook [7].

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 [7].

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 [44].

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 [99].

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 [99].

Fan or pump curve.

$$r = (p_i - p_j) - \big(h_0\, s^2 - h_2\, \dot{m}\, |\dot{m}|\big),$$

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 [99]; 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 [99] 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 [99].

Ideal-gas mass flow.

$$r = \dot{m} - \frac{p}{RT}\,\dot{V}.$$

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.

$$r = T_{\text{mix}} - f\,T_{oa} - (1-f)\,T_{ra}, \qquad r = \dot{m}_{oa} - f\,\dot{m}_{\text{supply}}.$$

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, $G = (\dot{m}_{oa}/\rho)(C_{\text{zone}} - C_{oa})$, 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

$$p_{ws} = 0.61094\,\exp\left[\frac{17.625\,t}{t + 243.04}\right]\ \text{kPa}.$$

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.

$$r = \dot{m}_{\text{cond}} - \dot{m}_{\text{air}}\,(1 - \text{BF})\,\max(0,\,W_{\text{zone}} - W_{\text{adp}}),$$

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 $\dot{m}_{\text{cond}}$ times the heat of vaporization. *Pattern*: property. *Parameters*: bypass factor BF, approach [K]. *Jacobian*: differenced. *Kinks*: at $W_{\text{zone}} = W_{\text{adp}}$, 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.

$$r = P - \frac{Q}{\mathrm{COP}}, \qquad \mathrm{COP} = \max(\mathrm{COP}_{\mathrm{min}},\ \mathrm{COP}_{\mathrm{pk}} - c\,(x - x_{\mathrm{opt}})^2),$$

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] Todd M. Bandhauer, Srinivas Garimella, and Thomas F. Fuller. A critical review of thermal issues in lithium-ion batteries. *Journal of The Electrochemical Society*, 158(3):R1–R25, 2011.

[6] Albert Benveniste, Benoît Caillaud, and Mathias Malandain. The mathematical foundations of physical systems modeling languages. *Annual Reviews in Control*, 50:72–118, 2020.

[7] Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. Wiley, 7th edition, 2011.

[8] 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.

[9] Roy Billinton and Ronald N. Allan. *Reliability Evaluation of Power Systems*. Plenum Press, New York, 2nd edition, 1996.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] 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.

[12] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.

[14] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[15] Phani Chavali and Arye Nehorai. Distributed power system state estimation using factor graphs. *IEEE Transactions on Signal Processing*, 63(11):2864–2876, 2015.

[16] Roy R. Craig and Mervyn C. C. Bampton. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1313–1319, 1968.

[17] Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[18] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2):1–139, 2017.

[19] 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.

[20] Iain S. Duff, Albert M. Erisman, and John K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, 2nd edition, 2017.

[21] A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

[22] 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.

[23] Lawrence C. Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.

[24] Herbert Federer. *Geometric Measure Theory*. Springer, 1969.

[25] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23 (2):298–305, 1973.

[26] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[27] Claudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. *ACM Computing Surveys*, 51(3):1–33, 2018.

[28] 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.

[29] Teoman Guler, George Gross, and Minghai Liu. Generalized line outage distribution factors. *IEEE Transactions on Power Systems*, 22(2):879–881, 2007.

[30] 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.

[31] Robert J. Guyan. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380, 1965.

[32] William W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989.

[33] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 2nd edition, 1996.

[34] 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.

[35] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

[36] 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.

[37] Paul Irofti, Luis Romero-Ben, Florin Stoican, and Vicenç Puig. Factor graph optimization for leak localization in water distribution networks. arXiv:2509.10982, 2025.

[38] 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.

[39] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.

[40] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[41] 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.

[42] 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.

[43] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[44] William M. Kays and Alexander L. London. *Compact Heat Exchangers*. McGraw-Hill, 3rd edition, 1984.

[45] 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.

[46] Brian W. Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, 1970.

[47] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[48] Gabriel Kron. *Tensor Analysis of Networks*. Wiley, New York, 1939.

[49] 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.

[50] 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.

[51] Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[52] 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.

[53] 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.

[54] Dennis V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[55] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[56] 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.

[57] Hans-Andrea Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

[58] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[59] 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.

[60] 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.

[61] Albert W. Marshall and Ingram Olkin. A multivariate exponential distribution. *Journal of the American Statistical Association*, 62(317):30–44, 1967.

[62] Mihajlo D. Mesarović, D. Macko, and Yasuhiko Takahara. *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York, 1970.

[63] 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.

[64] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001.

[65] Modelica Association. Modelica: A unified object-oriented language for systems modeling. language specification, version 3.6. `https://specification.modelica.org`, 2023.

[66] Alcir Monticelli. *State Estimation in Electric Power Systems: A Generalized Approach*. Kluwer Academic, Boston, 1999.

[67] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[68] George Oster, Alan Perelson, and Aharon Katchalsky. Network thermodynamics. *Nature*, 234:393–399, 1971.

[69] Art B. Owen. Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.

[70] Constantinos C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, 1988.

[71] Henry M. Paynter. *Analysis and Design of Engineering Systems*. MIT Press, Cambridge, MA, 1961.

[72] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[73] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[74] Gregory L. Plett. *Battery Management Systems, Volume I: Battery Modeling*. Artech House, 2015.

[75] 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.

[76] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[77] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

[78] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

[79] 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.

[80] Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, 2005.

[81] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, 2002.

[82] Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2nd edition, 2023.

[83] 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.

[84] N. N. Semenov. Zur theorie des verbrennungsprozesses. *Zeitschrift für Physik*, 48(7–8): 571–582, 1928.

[85] 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.

[86] 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.

[87] 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.

[88] Brian Stott, Jorge Jardim, and Ongun Alsaç. DC power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300, 2009.

[89] Andrew M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[90] Kaarthik Sundar, Carleton Coffrin, Harsha Nagarajan, and Russell Bent. Probabilistic N-k failure-identification for power systems. *Networks*, 71(3):302–321, 2018.

[91] 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.

[92] 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.

[93] 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.

[94] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods: Algorithms and Theory*. Springer, 2005.

[95] 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.

[96] 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.

[97] J. B. Ward. Equivalent circuits for power-flow studies. *Transactions of the American Institute of Electrical Engineers*, 68(1):373–382, 1949.

[98] Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[99] Frank M. White. *Fluid Mechanics*. McGraw-Hill Education, 8th edition, 2016.

[100] Allen J. Wood, Bruce F. Wollenberg, and Gerald B. Sheblé. *Power Generation, Operation, and Control*. Wiley, 3rd edition, 2013.

[101] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981.