PROBABILISTIC GRAPH MODELING



Probabilistic Modeling of Battery Energy Storage Systems

Cells, racks and the site — preview, Parts I–III

227 pages

Ricardo Marquez, Ph.D.

ricardo-marquez.com

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

Preface **ii**

0 Introduction **1**
0.1 The questions a storage operator asks . . . 1
0.2 Where the graph comes from . . . 2
0.3 One subject, not four . . . 3
0.4 Probability in the structure, not around the solver . . . 4
0.5 The boundary . . . 6
0.6 From "what is the state" to "what can I know" . . . 6
0.7 What is inherited and what is the book's . . . 7
0.8 How to read this book . . . 8
0.9 What this book does not do . . . 9
Notes and further reading . . . 9

I Physical systems as graphs **10**

1 Conservation, closure, and the boundary **11**
1.1 Three questions about any system . . . 11
1.2 Nodes conserve . . . 12
1.2.1 The balance law . . . 12
1.2.2 Node roles . . . 12
1.2.3 Incidence . . . 13
1.3 Edges close . . . 13
1.3.1 Closure relations . . . 14
1.3.2 A closure is a relation, not an assignment . . . 14
1.3.3 Parameters . . . 15
1.4 The boundary . . . 15
1.4.1 Reservoirs and imposed conditions . . . 15
1.4.2 Exactness at a cut . . . 16
1.4.3 Why the boundary matters most . . . 17
1.5 The residual system . . . 17
1.5.1 Well-posedness . . . 18
1.5.2 Sparsity . . . 18
1.6 Three worked examples . . . 19
1.7 In practice . . . 25

1.8 What is missing . . . 26
Notes and further reading . . . 27
Summary . . . 27
Exercises . . . 28
Solutions to selected exercises . . . 28

2 From equations to factor graphs 30
2.1 The rows as a graph . . . 30
2.2 Three languages . . . 31
2.2.1 Directed graphs . . . 31
2.2.2 The directionality problem . . . 32
2.2.3 Undirected graphs . . . 34
2.2.4 Factor graphs . . . 34
2.3 The physical system as a factor graph . . . 35
2.3.1 A node is not a variable . . . 36
2.3.2 Two domains on one graph . . . 36
2.4 The factor graph is not unique . . . 37
2.5 Vocabulary . . . 39
2.6 In practice . . . 40
2.7 What is missing . . . 40
Notes and further reading . . . 41
Summary . . . 41
Exercises . . . 42
Solutions to selected exercises . . . 42

II Probability on a physical graph 44

3 Where probability enters 45
3.1 A probability on the state . . . 45
3.2 Five kinds of factor . . . 46
3.2.1 Priors . . . 46
3.2.2 Closure factors . . . 47
3.2.3 Conservation factors . . . 47
3.2.4 Measurement factors . . . 48
3.2.5 Failure factors . . . 48
3.2.6 The complete list . . . 49
3.3 The joint as an objective . . . 49
3.3.1 Residuals in units of their widths . . . 50
3.3.2 Conditioning a Gaussian on one reading . . . 50
3.4 A worked example: one sensor, then two . . . 53
3.5 A second example: the ring bus with a shunt . . . 55
3.6 Reading the graph . . . 58
3.7 In practice . . . 59
3.8 What is missing . . . 60
Notes and further reading . . . 60
Summary . . . 61

Exercises . . . 61
Solutions to selected exercises . . . 62

4 Hard constraints, soft constraints, and the manifold 63
4.1 The hard joint and its manifold . . . 63
4.2 Conditioning on the manifold . . . 64
4.3 The soft joint . . . 65
4.4 Why soften . . . 66
4.5 Choosing the widths . . . 67
4.6 Worked example: the heat path with soft physics . . . 69
4.7 A second example: the ring bus with an unmodelled current . . . 71
4.8 In practice . . . 73
4.9 What is missing . . . 74
Notes and further reading . . . 74
Summary . . . 75
Exercises . . . 75
Solutions to selected exercises . . . 76

5 Factorization, separators, and treewidth 77
5.1 Factorization is not independence . . . 77
5.2 Reading independence off the graph . . . 78
5.3 Explaining away, precisely . . . 79
5.4 Ports are separators . . . 80
5.5 Elimination and fill . . . 81
5.5.1 One elimination . . . 81
5.5.2 Order and width . . . 82
5.5.3 What this means for a physical graph . . . 83
5.6 Worked examples: counting width and fill . . . 84
5.7 Why separators matter for cost . . . 86
5.8 In practice . . . 88
5.9 What is missing . . . 89
Notes and further reading . . . 89
Summary . . . 89
Exercises . . . 90
Solutions to selected exercises . . . 90

6 The linear-Gaussian case 92
6.1 The Gaussian in information form . . . 92
6.2 The most probable state . . . 93
6.2.1 Gauss–Newton against Newton . . . 94
6.3 Elimination is Cholesky . . . 94
6.3.1 What the factorization costs . . . 95
6.4 The Laplace posterior . . . 97
6.5 Worked example: the heat path in matrix form . . . 97
6.6 Second worked example: the ring bus in matrix form . . . 101
6.7 Filters, fields, and state estimators . . . 103
6.8 In practice . . . 104

6.9 What is missing . . . 104
Notes and further reading . . . 105
Summary . . . 105
Exercises . . . 106
Solutions to selected exercises . . . 106

7 Beyond Gaussian: hybrid states and nonlinearity 108
7.1 Four ways out of the Gaussian world . . . 108
7.2 Hybrid states: the mixture over branches . . . 109
7.2.1 Worked example: a contactor that may have opened . . . 110
7.2.2 Many discrete variables . . . 112
7.3 Piecewise closures and regimes . . . 112
7.3.1 Worked example: a modulating coolant loop . . . 113
7.4 Strong nonlinearity . . . 114
7.5 Bounded parameters . . . 115
7.6 What survives . . . 116
7.7 In practice . . . 116
7.8 What is missing . . . 117
Notes and further reading . . . 117
Summary . . . 118
Exercises . . . 119
Solutions to selected exercises . . . 119

III Inference 124

8 Exact inference: elimination, junction trees, Schur complements 125
8.1 What exact means . . . 125
8.2 Messages on a tree . . . 125
8.3 Loops: the junction tree . . . 127
8.4 Regions and ports: the Schur complement as a message . . . 128
8.5 Reuse . . . 130
8.6 Worked example: two containers, one tie cable . . . 131
8.7 Second example: containers with their own references . . . 132
8.8 Discrete variables by region . . . 134
8.9 In practice . . . 135
8.10 What is missing . . . 136
Notes and further reading . . . 136
Summary . . . 137
Exercises . . . 138
Solutions to selected exercises . . . 138

9 Approximate inference 140
9.1 When exact is too expensive . . . 140
9.2 Loopy belief propagation . . . 141
9.2.1 On the book's models . . . 141
9.3 Variational methods and expectation propagation . . . 143

9.4 Simulation inside a region . . . 145
9.4.1 What a mixture message loses when collapsed . . . 146
9.4.2 Worked example: a container as a simulated region . . . 146
9.5 Pruning by evidence . . . 148
9.6 The hierarchy, and how to choose . . . 149
9.7 In practice . . . 150
9.8 What is missing . . . 151
Notes and further reading . . . 151
Summary . . . 151
Exercises . . . 152
Solutions to selected exercises . . . 153

10 Enumeration, Monte Carlo, and factorized inference 155
10.1 The question, stated honestly . . . 155
10.2 The problem . . . 156
10.3 Enumeration: the product grid . . . 157
10.4 Monte Carlo . . . 157
10.5 Attribution: where the structural difference appears . . . 159
10.6 The factorized route: cuts of the dependency graph . . . 160
10.7 Second worked example: two storage containers . . . 161
10.8 Beyond three inputs . . . 165
10.9 The claim . . . 165
10.10In practice . . . 166
10.11What is missing . . . 167
Notes and further reading . . . 167
Summary . . . 167
Exercises . . . 168
Solutions to selected exercises . . . 168

11 Sequential evidence 170
11.1 Readings arrive one at a time . . . 170
11.2 One reading, one rank-one update . . . 170
11.3 The innovation . . . 171
11.4 Many readings . . . 172
11.5 Worked example: a stream of shunt readings . . . 173
11.6 The Laplace posterior and re-linearization . . . 174
11.7 The value of a reading before it is taken . . . 175
11.8 Cost at scale . . . 177
11.9 Worked example: where to put the next thermistor . . . 178
11.10In practice . . . 179
11.11What is missing . . . 180
Notes and further reading . . . 180
Summary . . . 181
Exercises . . . 181
Solutions to selected exercises . . . 182

A Notation **184**
A.1 Conventions . . . 184
A.2 Symbols . . . 185
A.3 Letters with more than one meaning . . . 188

B Gaussian identities **190**
B.1 The two forms of a Gaussian . . . 190
B.2 Block matrices . . . 191
B.3 Marginals and conditionals . . . 192
B.4 Products and linear-Gaussian models . . . 193
B.5 Rank-one changes . . . 194
B.6 Integrals, quadratic forms and divergences . . . 194
B.7 The Laplace approximation . . . 195
B.8 Where the book uses each identity . . . 196

C From the book to the engine **197**
C.1 What TERRA is . . . 197
C.2 A model in a dozen lines . . . 198
C.3 Where each construction lives . . . 198
C.4 The case studies . . . 199

D A catalogue of closures **201**
D.1 How to read an entry . . . 201
D.2 The four patterns in the engine . . . 203
D.3 Electrical . . . 203
D.4 Storage and electrochemistry . . . 205
D.5 Thermal . . . 205
D.6 Fluid . . . 207
D.7 Moist air . . . 209
D.8 Plant equipment . . . 210

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

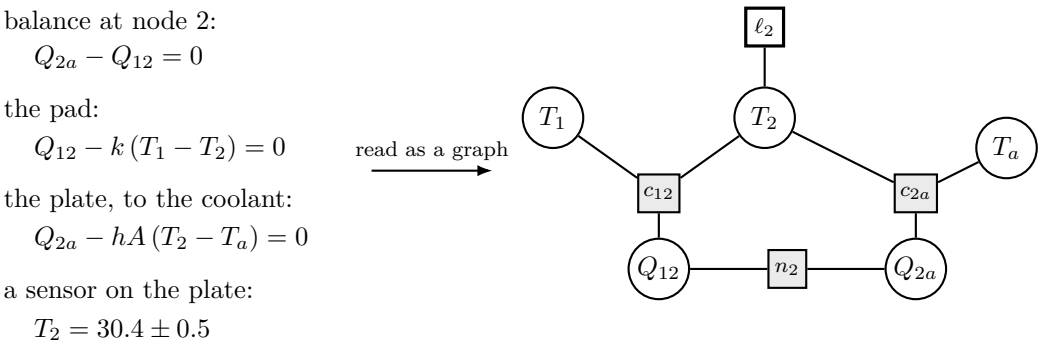


Figure 0.1: The graph comes from the equations. Left: three physics equations of the module heat path of Example 1.1 and one sensor reading, as an engineer writes them. Right: the same four statements as a factor graph. Each equation is a square joined to exactly the variables it reads; the sensor is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

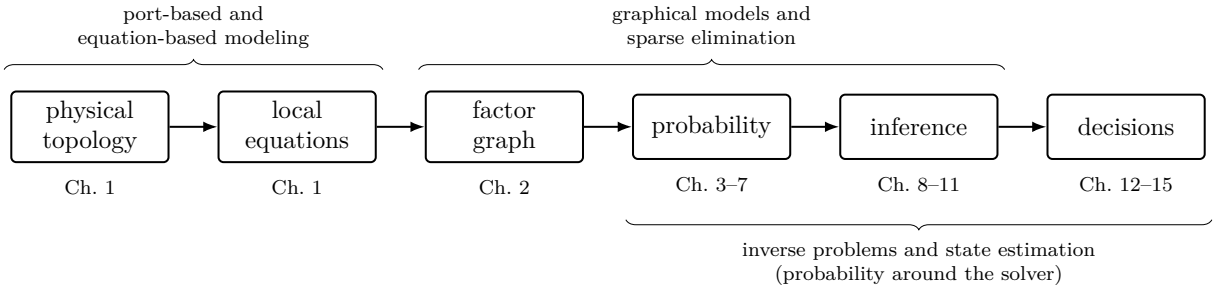
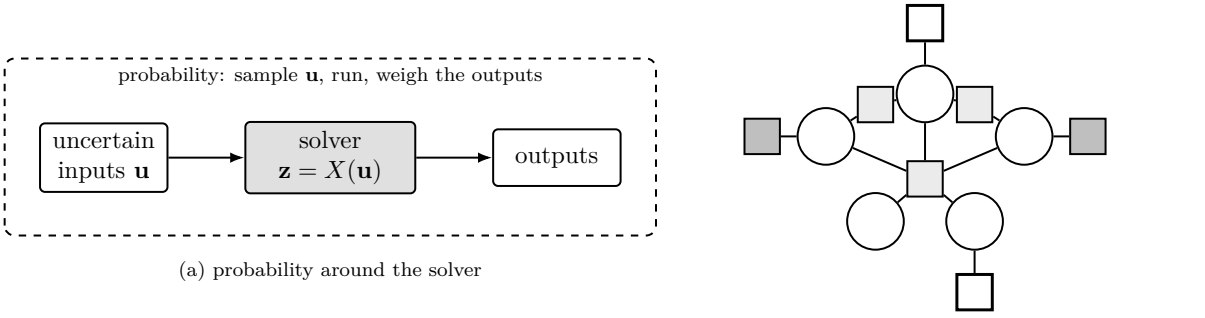


Figure 0.2: The chain of ideas the book follows, with the chapters that cover each link and the traditions that own parts of it. No one tradition carries the object from the topology to the decision.



(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

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

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

Once uncertainty and observations enter, the absence of direction becomes the point. A reading of one 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.

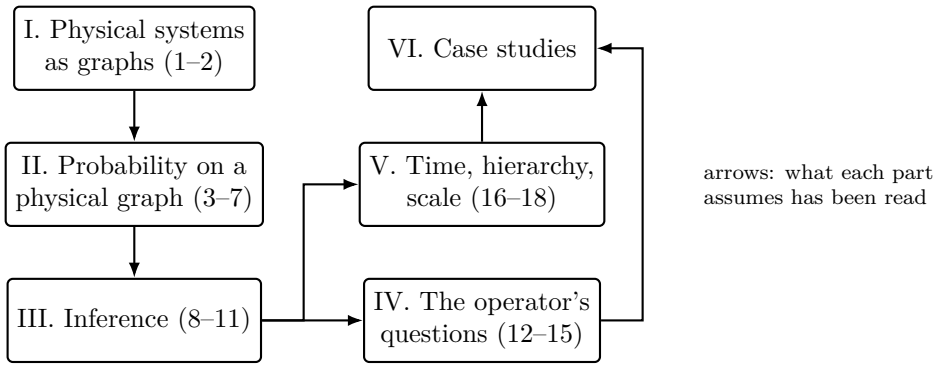


Figure 0.4: The parts of the book and their dependencies. Parts IV and V each need Part III and not each other; Part VI needs both.

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

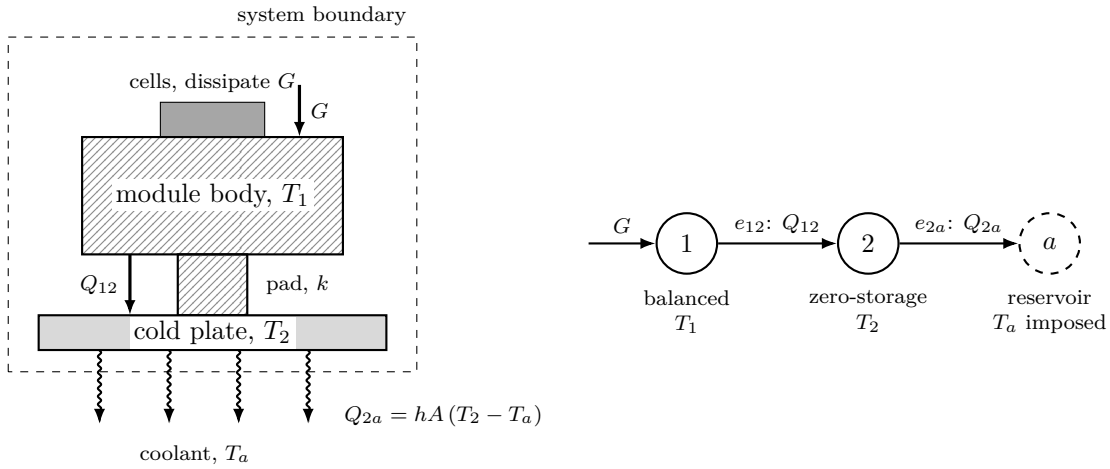


Figure 1.1: The module heat path of Example 1.1. Left: the physical arrangement. The cells dissipate G into the module body at T_1; the body conducts through the thermal pad, of conductance k, to the cold plate at T_2; the plate gives the heat up to the coolant at T_a. The dashed box is the system boundary 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

END OF THE PREVIEW

Probabilistic Modeling of Battery Energy Storage Systems

These are the opening pages. The complete book runs 227 pages.

The full PDF opens with a token. Write to **rmarquez123@gmail.com**, name the book, and I will send you one.

One token opens every book on the site.

Ricardo Marquez, Ph.D.

ricardo-marquez.com