Power-System Modelling for Computation

From circuit equations to optimization and software

Draft

Frederik Geth and contributors

September 6, 2026

Contents

Preface

Chapter 1

Why I wrote this book

Page status: draft author preface for Frederik Geth's review.

This book grew out of difficulties I encountered while learning to build power-system models and developing research software. I could often find a formula for assembling an admittance matrix, yet still struggle to determine exactly what its indices represented, which assumptions justified a simplification, and how to recover the equipment quantities my study needed.

Those questions cost time. An apparently small convention about terminals, orientation, grounding, or a voltage base could affect a derivation, a data structure, and a numerical result. Moving between them meant reconstructing choices that were not always explicit in the material I was learning from.

Working on software, including PowerModelsDistribution, made these questions practical. A convention left implicit in a derivation eventually becomes an implementation choice. A model can be useful for its intended study and still be easy to misuse when carried into a different one. Explaining that boundary is part of making research software useful to the next person.

I want the next researcher to spend less time reconstructing those choices and more time asking useful scientific questions. The examples here make the choices explicit and provide calculations that readers can inspect, challenge, and reproduce. Some examples show a failure; others show exactly why a familiar simplification works. Both are necessary for learning to choose a model well.

The book is for power engineers willing to work through precise assumptions, and for computer scientists and operations researchers entering power-system applications. It asks readers to follow the chain from equipment and circuit equations to representations, constraints, and computation. It does not require that every study use the most detailed model available.

This is also an unfinished research and teaching project. It has been developed with substantial assistance from language models, under my direction and review. Executable checks, derivations, and source records make parts of the work inspectable; full independent human review remains incomplete. A recorded calculation should be read with its assumptions and evidence status. Corrections that sharpen those statements are an essential contribution to the book.

Frederik Geth — draft for author review

Part 1: A plausible model gives the wrong answer

Chapter 2

The parallel-member lesson

Page status: introductory counterexample with executable scalar and multiconductor evidence; independent mathematical review remains open.

By the end of this lesson, you will be able to check a parallel-branch replacement, recover its member currents, and derive a correct aggregate rating for a fixed scalar example. You need Ohm's law and inequalities; the calculation runs with standard-library Python.

Predict before calculating

Two lines connect the same buses. Each is rated for 100 A, but one has ten times the conductance of the other. Can the pair safely carry 200 A?

Write down a prediction, including how you expect the current to divide. First examine the current split in this resistive model:

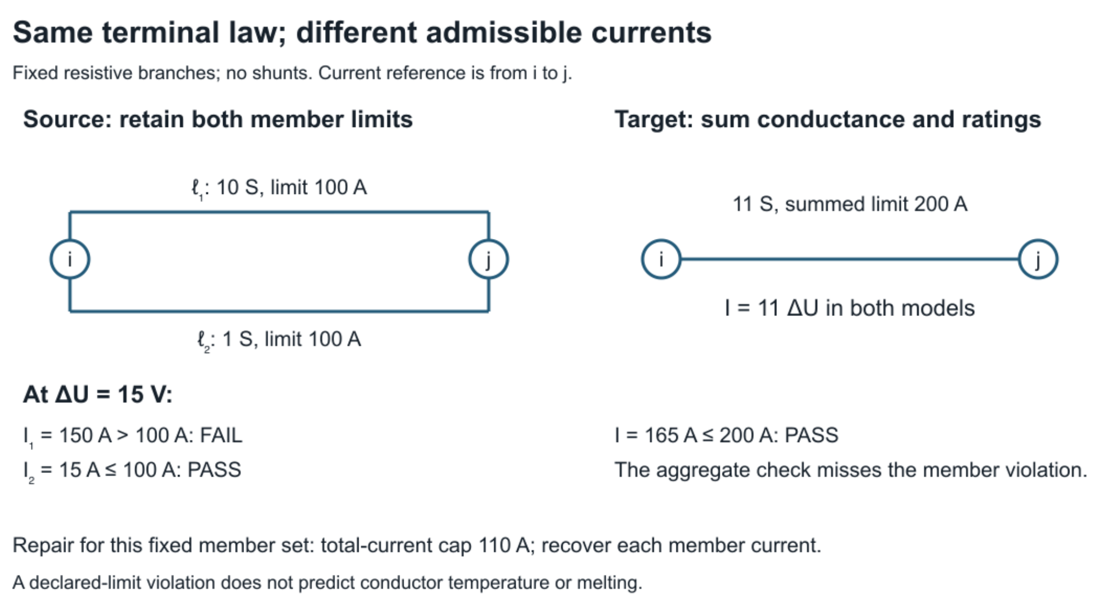

Figure 2.1: At a 15 V drop, a 165 A aggregate current passes the summed rating but its 150 A first-member current violates the 100 A limit.

Assume fixed, uncoupled, series-only resistive branches with common endpoints, no shunts, and no other constraints in this local calculation. Consider two scalar resistive branches $\ell_1 ij$ and $\ell_2 ij$ with intrinsic impedances

$$Z_{\ell_1} = 0.1\ \Omega, \qquad Z_{\ell_2} = 1.0\ \Omega,$$

and member current limits

$$I^{\max}_{\ell_1} = I^{\max}_{\ell_2} = 100\ \mathrm{A}.$$

Writing $\Delta U = U_i - U_j$ and $Y_\ell = Z_\ell^{-1}$, the terminal currents are

$$I_{\ell_k ij} = Y_{\ell_k} \Delta U, \qquad k \in \{1, 2\}.$$

Here k identifies a member, and positive current follows the stored orientation from i to j.

Replacing the pair by one branch with

$$Y_{\mathrm{eq}} = Y_{\ell_1} + Y_{\ell_2} = 11\ \mathrm{S}$$

is exact for the unconstrained total terminal-current relation. It does not follow that the replacement is exact for a decision problem with member limits.

The feasible sets differ

The source model requires both inequalities

$$|Y_{\ell_1} \Delta U| \leq I^{\max}_{\ell_1}, \qquad |Y_{\ell_2} \Delta U| \leq I^{\max}_{\ell_2}.$$

Hence its admissible voltage-drop magnitude is

$$|\Delta U| \leq \min\left\{ \frac{I^{\max}_{\ell_1}}{|Y_{\ell_1}|}, \frac{I^{\max}_{\ell_2}}{|Y_{\ell_2}|} \right\} = 10\ \mathrm{V}.$$

A tempting aggregate construction assigns the equivalent branch the summed rating, $I^{\max}_{\mathrm{eq}} = 200\ \mathrm{A}$. It admits

$$|\Delta U| \leq \frac{200}{11}\ \mathrm{V} \approx 18.18\ \mathrm{V}.$$

At the witness $\Delta U = 15\ \mathrm{V}$, the equivalent branch carries $165\ \mathrm{A}$ and satisfies its aggregate limit. Recovery gives

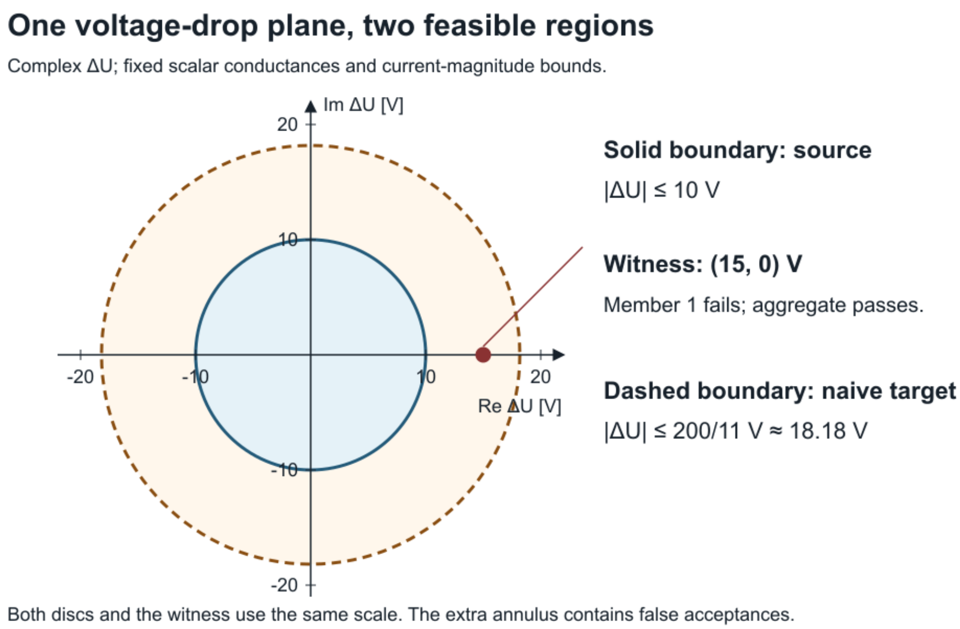

Figure 2.2: Voltage-drop discs on one scale: source radius 10 V, aggregate radius 200/11 V, and witness at 15 V.

$$I_{\ell_1 ij} = 150 \text{ A}, \qquad I_{\ell_2 ij} = 15 \text{ A},$$

so the source violates the ℓ_1 limit. The target feasible set is therefore an outer relaxation of the source feasible set.

The figure is generated by `experiments/render_parallel_feasible_set_card.py`. The retained and aggregate regions are discs in the complex ΔU plane, plotted in volts on identical horizontal and vertical scales. The point $(15, 0)$ lies in the target's extra region. No weighted or coupled constraint is represented in this scalar plot.

Repair the aggregate

The source requires $|\Delta U| \leq 10$ V. Since the total current is $11\Delta U$, the exact aggregate current-magnitude cap for this fixed scalar model is

$$I^{\max}_{\text{eq,exact}} = 11 \times 10 = 110 \text{ A}.$$

This rating reproduces the local member-constrained voltage-drop set. Both members remain recoverable from ΔU and their retained admittances. At the cap, member currents are 100 A and 10 A. The weakly conducting member cannot use its remaining capacity independently of the common voltage drop.

A correct reduction can therefore be useful. Its rating must be derived from what the study constrains. The 110 A cap belongs to this fixed member set; opening a member changes both the equivalent admittance and the cap. Retain member identities and a state-dependent construction if outages are decisions.

A decision problem exposes the gap

The same mechanism changes an optimum in a two-bus maximum-served-load model. Let the parallel members have

$$(b_{\ell_1}, b_{\ell_2}) = (1000, 100)\ \mathrm{MW/rad}, \qquad (F^{\max}_{\ell_1}, F^{\max}_{\ell_2}) = (100, 100)\ \mathrm{MW},$$

with $F_{\ell ij} = b_\ell \delta_{ij}$, nonnegative δ_{ij}, and served power equal to total flow. The source model keeps both member laws and both limits. The naïve target uses $b_{\mathrm{eq}} = 1100$ MW/rad and $F^{\max}_{\mathrm{eq}} = 200$ MW.

Formulation	Maximum served power	δ_{ij}	Active restriction
source members	110 MW	0.1 rad	$F_{\ell_1 ij} \le 100$ MW
naïve aggregate	200 MW	$2/11$ rad	summed 200 MW rating
aggregate relation with exact lifted member constraints	110 MW	0.1 rad	recovered $F_{\ell_1 ij} \le 100$ MW

The displayed exact lifted formulation retains every member law and both member limits:

$$F_{\ell ij} = b_\ell \delta_{ij}, \qquad F_{\ell ij} \le F^{\max}_\ell \quad \text{for every } \ell,$$

alongside the aggregate terminal relation. It therefore reproduces the source feasible set and optimum without pretending that a summed scalar rating is exact. This computed comparison is claim `TR-PAR-003`. JuMP and Ipopt produce the machine-readable result in `experiments/generated/parallel-opf-comparison.json`; the analytic values above also provide a solver-independent check.

Here $b_{\ell_2} = 0.1 b_{\ell_1}$ and the ratings are equal, so the ℓ_2 limit is actually implied by the ℓ_1 limit and can be certifiably pruned. Keeping both in the table isolates the lifting argument from presolve; the next case implements the pruned formulation explicitly.

The Multiconductor parallel AC decision case extends this comparison to coupled complex conductor equations, phase-to-neutral constant power, and voltage-magnitude constraints.

Run the calculation

From the repository root, with Python 3 installed:

```
python3 experiments/lessons/parallel_members.py
```

The script uses exact rational arithmetic for this scalar calculation:

```
first member current: 150 A
second member current: 15 A
aggregate current: 165 A
summed-rating check: pass
recovered member checks: fail
exact voltage-drop magnitude cap: 10 V
exact aggregate current-magnitude cap for this state: 110 A
```

No power-flow solver is involved. The calculation checks the branch laws and ratings specified above. It does not model temperature, protection operation, voltage-dependent impedance, or uncertainty in the ratings. The later AC case adds network equations and load constraints.

Run the boundary and outage checks with:

```
python3 experiments/lessons/parallel_members.py --check
python3 experiments/lessons/parallel_members.py --drop 10
python3 experiments/lessons/parallel_members.py --open-member first
```

Change one assumption

1. Increase the first member's rating to 200 A. Predict the exact aggregate cap, then run `--limit-first 200`. Which member binds?

2. Open the first member while retaining a 15 V drop. Predict the remaining current and explain why the old equivalent is invalid.

3. Reverse the voltage drop with `--drop -15`. Which feasibility conclusions change, and which depend only on magnitude?

4. Write the exact aggregate rating for arbitrary positive real admittances Y_1, Y_2 and positive member ratings. State why the same formula is not automatically a conventional multiconductor rating.

Check your reasoning

With the first rating at 200 A, the voltage-drop cap becomes 20 V and the aggregate cap 220 A; the first member still binds. With that member open, the remaining conductance is 1 S, the current at 15 V is 15 A, and the remaining rating is 100 A. Reversing the drop reverses both currents without changing the magnitude-limit checks.

For the fixed positive-real scalar model,

$$I_{\mathrm{eq,exact}}^{\max} = (Y_1 + Y_2) \min\left\{ \frac{I_1^{\max}}{Y_1}, \frac{I_2^{\max}}{Y_2} \right\}.$$

For multiple conductors, member constraints act on vector-valued currents. Their inverse images need not have the form of one conventional aggregate rating. The extension below states that set explicitly.

Preservation contract

Contract field	Value
source	two identified parallel branches with individual limits
target	one equivalent branch with summed admittance and rating
preconditions	common endpoints and voltage coordinates; linear branch laws
preserves	unconstrained total terminal current as a function of ΔU
does not preserve	the member-constrained feasible set
forgets	member identity and independent outage, maintenance, or investment state
recovery	$I_{\ell_k ij} = Y_{\ell_k} \Delta U$ if member admittances remain available
classification	outer/relaxed for this rating construction

This counterexample does not say that parallel aggregation is never useful. It says that an aggregate rating needs its own derivation relative to the intended observation or decision set. Choosing a tighter equivalent limit can reproduce this one scalar voltage-drop bound, but it still does not recreate independent member states or arbitrary member-wise constraints.

Certified redundancy is different from aggregation

Some member limits can be removed exactly without replacing the physical members. For a fixed scalar AC π-line, write the endpoint-voltage state as

$$x = \begin{bmatrix} \Re(U_i) & \Re(U_j) & \Im(U_i) & \Im(U_j) \end{bmatrix}^\mathsf{T}.$$

A current limit has a positive-semidefinite quadratic feasible set because $|I_{\ell ij}|^2 = x^\mathsf{T} Q_{\ell ij} x$ for a fixed linear terminal-current map. Normalizing by the squared current rating gives

$$\mathcal{E}_{\ell ij} = \{x : x^\mathsf{T} M_{\ell ij} x \le 1\}, \qquad M_{\ell ij} = Q_{\ell ij}/(I^{\max}_{\ell ij})^2.$$

If $\mathcal{E}_{\ell_k ij} \subseteq \mathcal{E}_{\ell_r ij}$, the retained current limit implies the candidate limit at that end.

For an apparent-power rating, $|S_{\ell ij}|^2 = |U_i|^2 |I_{\ell ij}|^2$ is generally quartic in x. Its actual feasible region is not the quadratic set above. Molzahn instead compares the normalized apparent-power flows of parallel members at the **same terminal**. The common $|U_i|$ factor cancels for nonzero terminal voltage, leaving a comparison of the normalized squared currents [1]:

$$\frac{|I_{\ell_r ij}|^2}{(S^{\max}_{\ell_r ij})^2} \le \frac{|I_{\ell_k ij}|^2}{(S^{\max}_{\ell_k ij})^2} \quad \text{for every endpoint-voltage state.}$$

This sufficient dominance condition implies the candidate apparent-power limit from the retained one. At zero terminal voltage both apparent powers vanish, so the implication also holds there. Containment of auxiliary quadratic comparison sets tests the dominance condition; those sets are not the original apparent-power feasible regions. Molzahn's eigenvalue-based test also treats singular forms, whose sublevel sets are cylinders. Removing both directional limits requires an implication at each terminal.

This is exact constraint pruning, not asset aggregation: both line laws, parameters, identities, and recovered flows remain in the model. Failure to identify redundancy is also not proof that a limit is essential; the test is a sufficient certificate. Its stated scope is fixed scalar AC transmission π-models, including fixed complex transformer ratios and shunts. It does not by itself cover arbitrary multiconductor constraint sets, switching, outages, investment states, or other state-dependent parameters. Claim `LIT-PAR-001` records that boundary.

Multiconductor form

For multiconductor branches,

$$\mathbf{I}_{\ell_k ij} = \mathbf{Y}_{\ell_k} \Delta \mathbf{U},$$

and the source feasible set is the intersection of the inverse images of every member constraint set:

$$\mathcal{D}_{\mathrm{src}} = \bigcap_k \{\Delta\mathbf{U} : \mathbf{Y}_{\ell_k}\Delta\mathbf{U} \in \mathcal{C}_{\ell_k}\}\,.$$

The summed terminal map $\mathbf{Y}_{\mathrm{eq}} = \sum_k \mathbf{Y}_{\ell_k}$ does not, by itself, encode that intersection. Mutual coupling and per-conductor limits make a single conventional rating still less likely to be an exact representation.

The generation script emits the numerical witness and its machine-readable contract as claim `TR-PAR-001`.

Part 2: From equipment to equations

Chapter 3

From source data to a canonical model

Page status: worked exact resistive assembly lesson, followed by source-conversion guidance.

By the end of this lesson, you will be able to assemble a small nodal matrix from named equipment, impose a voltage boundary, solve for the remaining voltages, and check the result against the original equipment. Only Ohm's law, KCL, and a two-equation linear solve are needed. All values below are exact for a fixed resistive model; this first construction has no phasor, transformer, load-control, or thermal dynamics.

Start with equipment and terminals

An ideal source fixes node `s` at 12 V relative to a reference terminal `0`. Two series resistors connect it to a resistive load. The equipment table is the input; the array positions will be chosen afterward.

Equipment	Ordered terminals	Constitutive data
`e1`	`s, m`	conductance 2 S
`e2`	`m, t`	conductance 1 S
`d`	`t, 0`	load conductance 1/2 S
source	`s, 0`	fixed voltage 12 V

Before calculating, predict which of `m` and `t` has the higher voltage and whether the two series currents differ. Current into the load must return through the reference connection; the source supplies it. The symbol `0` identifies that connection and its chosen voltage datum in this circuit. A voltage datum alone would not create a physical grounding path.

Use node order (s, m, t) and voltage vector $\mathbf{U} = (U_s, U_m, U_t)^\mathsf{T}$. Store branch orientation from the first terminal to the second. With the book's incidence convention (tail -1, head $+1$),

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \qquad \mathbf{I} = -\operatorname{diag}(2,1)\mathbf{A}\mathbf{U}.$$

The minus sign matters: $\mathbf{AU}$ gives head-minus-tail voltage, whereas the stored current follows tail-to-head voltage drop. Branch currents leaving nodes are $-\mathbf{A}^\mathsf{T}\mathbf{I}$.

Stamp one component at a time

A conductance g between nodes i and j contributes $+g$ to Y_{ii} and Y_{jj}, and $-g$ to Y_{ij} and Y_{ji}. The load to the fixed reference adds $1/2$ to Y_{tt}. Thus

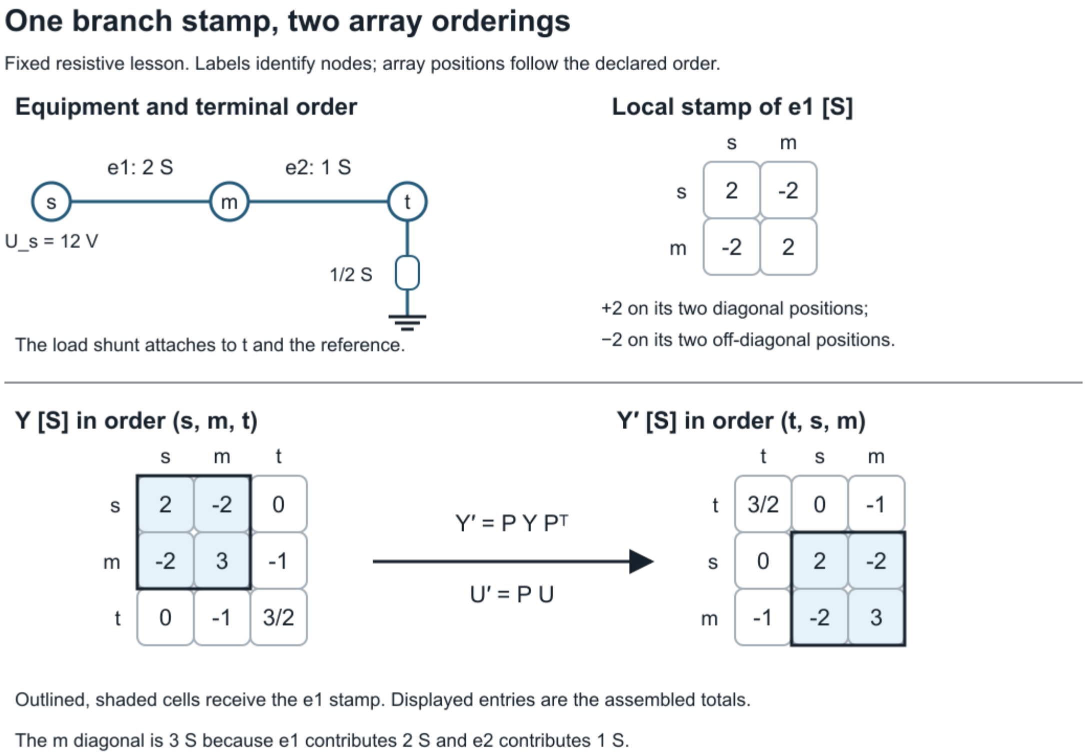

Figure 3.1: Branch e1 contributes four entries to Y; changing array order to (t,s,m) permutes those entries without changing the node identities.

$$\mathbf{Y} = \mathbf{A}^{\mathsf{T}}\,\mathrm{diag}(2,1)\mathbf{A} + \mathrm{diag}(0,0,1/2) = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 3 & -1 \\ 0 & -1 & 3/2 \end{bmatrix}\ \mathrm{S}.$$

The outlined entries receive the e1 stamp; their displayed values include all contributions. In particular, the `m` diagonal includes both branches. The lower panels use the same permutation for equation rows and voltage coordinates, so $\mathbf{Y}' = \mathbf{P}\mathbf{Y}\mathbf{P}^{\mathsf{T}}$ and $\mathbf{U}' = \mathbf{P}\mathbf{U}$. The source boundary remains at node `s`.

This matrix relates passive-device currents leaving the nodes to their voltages. The ideal source is a voltage boundary with an unknown supplied current, not a finite admittance stamped into this matrix.

At `m` and `t` there is no separate injected current. Substituting $U_s = 12$ gives the retained system

$$\begin{bmatrix} 3 & -1 \\ -1 & 3/2 \end{bmatrix} \begin{bmatrix} U_m \\ U_t \end{bmatrix} = \begin{bmatrix} 24 \\ 0 \end{bmatrix}, \qquad U_m = 72/7\ \mathrm{V}, \quad U_t = 48/7\ \mathrm{V}.$$

The source current follows from its row afterward. Deleting the source row without moving its voltage contribution to the right-hand side would solve a different problem.

Recover and check the equipment

Use the original terminal records, not the assembled matrix, to calculate

$$I_{e1} = 2(U_s - U_m) = 24/7\ \mathrm{A}, \quad I_{e2} = U_m - U_t = 24/7\ \mathrm{A}, \quad I_d = U_t/2 = 24/7\ \mathrm{A}.$$

Check KCL separately at `m` and `t`. Check power using source delivery $12I_{e1}$ and resistor absorption $I_{e1}^2/2 + I_{e2}^2 + U_t^2/2$; both equal $288/7$ W. These are complementary checks. An aggregate power check alone does not test each node's balance, and small residuals for a wrongly assembled matrix do not validate its terminal maps.

```
python3 experiments/lessons/assemble_network.py
python3 experiments/lessons/assemble_network.py --check
python3 experiments/lessons/assemble_network.py --misattach-load
```

The first command prints the matrix, rational voltages and currents, passing source-equipment KCL, and zero power mismatch. The last command deliberately stamps the load at `m` instead of `t`. It solves its own equations, giving $U_m = U_t = 48/5$ V, but fails KCL evaluated against the original equipment. The code also checks a node permutation, reversed branch orientations, an altered voltage, and a changed load. It never writes an artifact.

This verification separates stamping from equipment evaluation but shares the input conductances and laws. It can catch a mapping or solve error; it does not prove those conductances describe real equipment.

Change the construction

1. Change the load conductance to 1/4 S. Derive both voltages and currents before calling `solve(load=Fraction(1, 4))` in the lesson module.
2. Store nodes in order `(t,s,m)`. Explain how the matrix and the voltage array change while voltages attached to named nodes remain the same.
3. Remove the load. Explain the null vector of the full passive matrix and why fixing the source voltage still determines the other two voltages.
4. Replace the resistive data by fixed complex admittances. Which assembly identities survive, and which power expression requires conjugation?

Answer guidance. For the changed load, $U_t = 96/11$ V, $U_m = 120/11$ V, and both series currents are $24/11$ A. A node permutation acts on rows, columns, and the voltage vector consistently; it cannot change a named-node voltage. Without the load, all voltages equal 12 V after imposing the source boundary; the full matrix has common-offset gauge freedom. For complex admittances and real incidence maps the stamp still uses $A^{\mathsf{T}}YA$. Terminal complex power is UI^*. A complex voltage-coordinate map instead needs its conjugate-transpose power-dual current map; see Circuit coordinate transformations.

A satisfactory solution shows the stamp, boundary substitution, recovered currents, and source checks. A table of final voltages alone is insufficient.

From this example to source conversion

A utility export, OpenDSS deck, CIM profile, or solver dictionary can hide choices that the small table made explicit: units, terminal order, defaults, connection types, and state. The remainder of this chapter describes how to record those choices. Here *canonical* means the declared internal source model for this workflow; it does not claim a unique mathematical representation of every power system.

The projection contract

Let D be a source document and C a canonical network model. An adapter is a partial map

$$P : D \longrightarrow (C, F, \Pi),$$

where F is a finding ledger and Π is a provenance manifest. The map is partial because an input may be malformed, incomplete, or outside the supported factor library. A successful parse means only that P returned a candidate model; it does not mean that every study question supported by D is preserved in C.

The canonical model should make at least these objects explicit:

Source concern	Canonical object	Why it matters for graphs
stable equipment identity	asset/property record	keeps parallel and replacement assets distinct
endpoint and conductor names	ordered terminal maps	prevents phase/neutral permutations from hiding in matrices
connectivity and switching	state-resolved incidence	separates inventory topology from active topology
constitutive relation	line, shunt, transformer, load, or factor relation	prevents an edge from standing in for unknown physics
ratings and owners	typed constraint observations	keeps limits attached to the object they constrain
defaults and inferred meanings	finding with confidence/provenance	makes assumptions auditable

The book's preferred identifiers retain the BMOPFTools-style ownership rule: ℓ identifies an element, while ℓij identifies an oriented terminal attachment. A converter may reorder arrays or create virtual objects, but it must publish the map that explains the change.

Validation gates before graph construction

The following gates are deliberately ordered. A later graph cannot repair a failure that should have been caught at an earlier semantic layer.

Gate	Question	Typical failure
schema	are fields and shapes structurally recognised?	malformed matrix or unknown component block
completeness	are required fields present for this factor subtype?	transformer winding lacks a terminal map
domain	are values physically and numerically plausible?	negative rating, impossible tap, invalid angle unit
integrity	do references and dimensions agree?	line points to a missing bus or mismatched conductor count
conformance	does the object satisfy the declared model and implementation rules?	a winding connection missing a return path required by its declared subtype
readiness	are the chosen study's inputs, unknowns, equations, and constraints specified?	a queried member limit has no identified current or recovery map

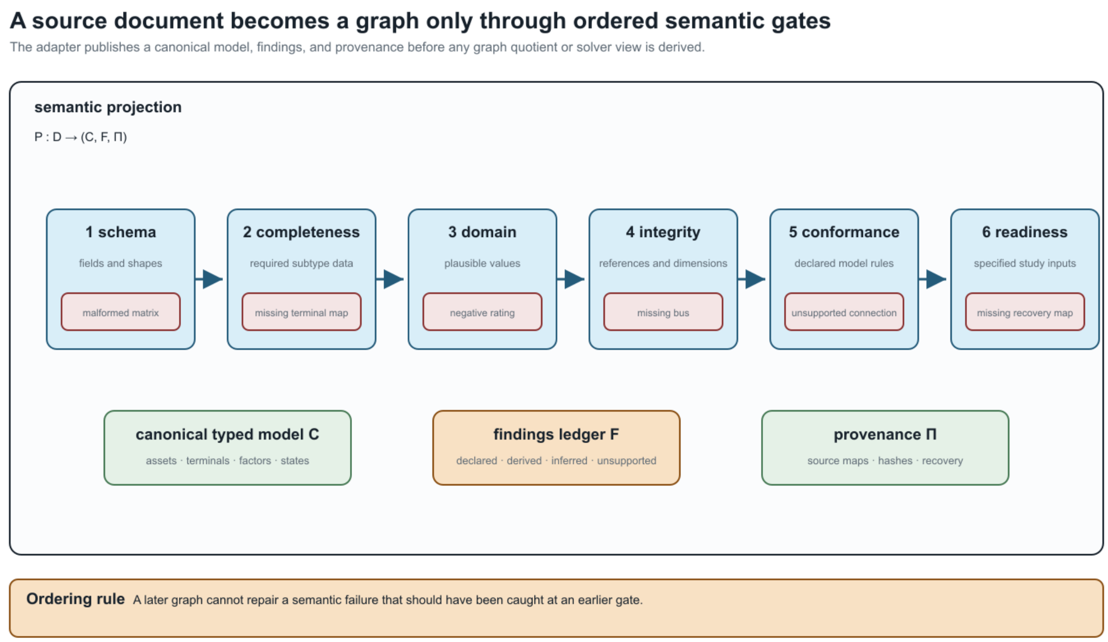

Figure 3.2: The ordered source-to-canonical validation pipeline.

An implementation can reject a valid circuit outside its supported model class. Multiple compatible voltage sources are not inherently invalid; their equations and the implementation's supported source configuration matter. Missing explicit voltage bounds or an objective involving only slack injection does not alone make a mathematical problem ill posed. Equations may determine voltages, and slack injection may be a meaningful objective. Conversely, bounds do not establish feasibility, uniqueness, or stability. Readiness checks identify specified obligations; mathematical well-posedness needs a study-specific argument.

These checks should return stable finding codes and machine-readable details, not only prose warnings. A graph quotient may be mathematically valid while the input is semantically unfit for the intended decision problem.

The ordering is semantic, not cosmetic: a later graph view cannot repair a missing terminal map, an invalid domain value, or an ownerless limit that should have been rejected earlier.

Inference is not identity

Practical formats often encode meaning positionally or through defaults. An adapter may infer that terminal 4 is a neutral, that two named objects are parallel members, or that a regulator is a transformer with a control loop. Such inferences can be useful, but recording them with provenance does not turn them into declared source facts or establish their truth. Retain the inference rule, its assumptions, and its evidence. A declared source value can also be wrong: origin and correctness are separate attributes. The canonical record should distinguish:

1. **declared** values copied from the source;
2. **derived** values computed from declared data;
3. **inferred** values introduced by an adapter rule; and

4. **unsupported** values that could not be represented.

This distinction is essential when a later transformation claims preservation. An inferred terminal permutation may be harmless for a scalar connectivity query but fatal for a conductor-current limit.

What a safe adapter publishes

For every source-to-canonical map, record:

1. source format, profile, and version;
2. stable asset and terminal identifiers;
3. units, bases, phase, neutral, earth, and grounding conventions;
4. state, scenario, and control treatment;
5. factor, rating, and objective mappings;
6. generated objects and their source parents;
7. unsupported or lossy fields;
8. validation findings and their severity; and
9. round-trip or recovery checks for the declared observations.

The practical import exercise tests these obligations on a rating sentinel. A number-preserving round trip can still change its constraint meaning; an unknown value must remain unknown.

Impedance data need one additional discipline: the canonical record should retain the full derivation context even when an adapter exports only a solver- friendly matrix. Geometry or linecode provenance, frequency, earth model, ordered conductors, series/shunt blocks, and matrix diagnostics belong to the source record; Kron, phase-to-neutral, sequence, and positive-sequence matrices are derived views with their own guards. The four-wire impedance-model ladder is the first executable example of this source-to-view contract.

Consequences for the graph views

An adapter that collapses two assets, loses a neutral, or silently grounds a terminal changes the source from which every subsequent view is built. Retain the equipment and terminal identities needed to check the derived equations; the many-graphs lesson develops the graph views.

Running-network application

The running fixture is intentionally small enough to audit. Its canonical record declares four lines, a three-winding transformer, a switch, ordered conductor sets, explicit neutral semantics, ratings, and state ownership. The generated adapter crosswalk checks those obligations against the pinned CIM/CGMES, PowerModelsDistribution, OpenDSS, and MATPOWER descriptions. That crosswalk is not an import claim: it is a checklist of what an actual adapter would have to preserve or mark unsupported.

The field-level example and this larger checklist serve different purposes: the former executes a small semantic check; the latter identifies obligations still required for a complete running-network adapter.

Chapter 4

Orientation and terminal power

Page status: foundational definitions and terminology.

Why an arrow is ambiguous

Power-system diagrams, data formats, and optimization models use arrows for several different purposes. A line drawn from i to j can mean an arbitrary stored orientation, a terminal-current sign convention, a positive reference direction for active power, an observed operating-point transfer, or a genuinely one-way admissible action. These meanings must not be inferred from one another.

Direction concept	Mathematical data	Dependence
physical incidence	unordered endpoints $\partial\ell = \{i, j\}$	equipment model
reference orientation	one selected arc $o(\ell) = \ell ij$	arbitrary coordinate choice
terminal-current sign	current defined into or out of each terminal	modelling convention
operating-point transfer	sign of $P_{\ell ij}$ at a solution	voltage, state, and controls
causal or permitted direction	asymmetric control or feasible relation	physical or study semantics

A conventional passive AC line is therefore an undirected physical connection with an orientation, not intrinsically a directed physical edge.

Power-system shorthand

An arrow on a passive branch normally fixes storage order, terminal names, or a positive reference. It does not predict the sign of current or active power at a solution.

The terminal-arc double cover

For every two-terminal element ℓ with $\partial\ell = \{i, j\}$, introduce the two terminal arcs

$$\overrightarrow{\mathcal{L}} = \{\ell ij, \ell ji : \ell \in \mathcal{L}\}.$$

The reversal map

$$\rho : \overrightarrow{\mathcal{L}} \to \overrightarrow{\mathcal{L}}, \qquad \rho(\ell ij) = \ell ji,$$

is an involution with no fixed points. A reference orientation chooses one arc from each pair; the bidirected terminal-arc view retains both. The line remains one member of $\mathcal{L}$.

With the book's incidence convention, the selected arc ℓij gives -1 at i and $+1$ at j. Choosing ℓji instead negates the column. Incidence rank, the undirected cycle space, and physical solutions are invariant under that coordinate change, while signed cycle coordinates change accordingly.

A stored orientation is not an operating direction

The stored triple ℓij determines which endpoint is written first. It does not imply any of the statements

$$P_{\ell ij} \geq 0, \qquad |P_{\ell ij}| \geq |P_{\ell ji}|, \qquad \text{or} \qquad \ell \text{ can transfer only from } i \text{ to } j.$$

Those are operating or device claims requiring separate equations. Active power can reverse between operating points, and in an unbalanced multiconductor model its sign can differ by conductor.

Reversing the stored orientation swaps end-specific records:

$$(\mathbf{N}_{\ell i}, \mathbf{N}_{\ell j}, \mathbf{Y}^{\mathrm{sh}}_{\ell ij}, \mathbf{Y}^{\mathrm{sh}}_{\ell ji}, \mathbf{I}_{\ell ij}, \mathbf{I}_{\ell ji}) \longleftrightarrow (\mathbf{N}_{\ell j}, \mathbf{N}_{\ell i}, \mathbf{Y}^{\mathrm{sh}}_{\ell ji}, \mathbf{Y}^{\mathrm{sh}}_{\ell ij}, \mathbf{I}_{\ell ji}, \mathbf{I}_{\ell ij}).$$

It does not blindly negate both terminal currents. Antisymmetry belongs to a particular internal series-current coordinate, not to every terminal quantity.

Rooted-tree orientation is a derived view

When an active network is radial, practitioners often orient every branch from the feeder source toward the leaves and call the resulting arcs *upstream* and *downstream*. This is useful, but it is not the stored orientation ℓij and it is not an intrinsic direction of a passive line.

For an active identified graph G^{σ}_{M} and a selected source root r in each component, a rooted-tree view adds a parent map

$$\mathrm{par}_{\sigma,r} : V \setminus \{r\} \longrightarrow V$$

defined by the unique root-to-node path. The resulting parent-to-child arcs, depths, ancestors and descendants are a **state- and root-dependent algorithmic view**. They are appropriate for feeder recursions, backward/forward sweeps and radial branch-flow notation, but they are not new asset attributes.

> **Graph-theory trap**
>
> A parent-to-child arc is not an operating power-flow direction. Closing a tie can create a chord, reverse power can change the sign of $P_{\ell ij}$, and a different source or spanning tree can change the parent map without changing any line asset.

If G^{σ}_{M} is meshed, choose a spanning forest only if an algorithm needs one. Tree edges then receive parent-child roles, while chords retain their cycle equations and must not be called upstream or downstream without a separate convention. A spanning-tree orientation is therefore a coordinate or algorithmic choice, not a claim that the physical network is radial.

The figure makes the sign discipline explicit: reversing ℓ_{ij} changes the coordinate convention, while changing the operating point can reverse $P_{\ell ij}$ without changing the stored asset orientation.

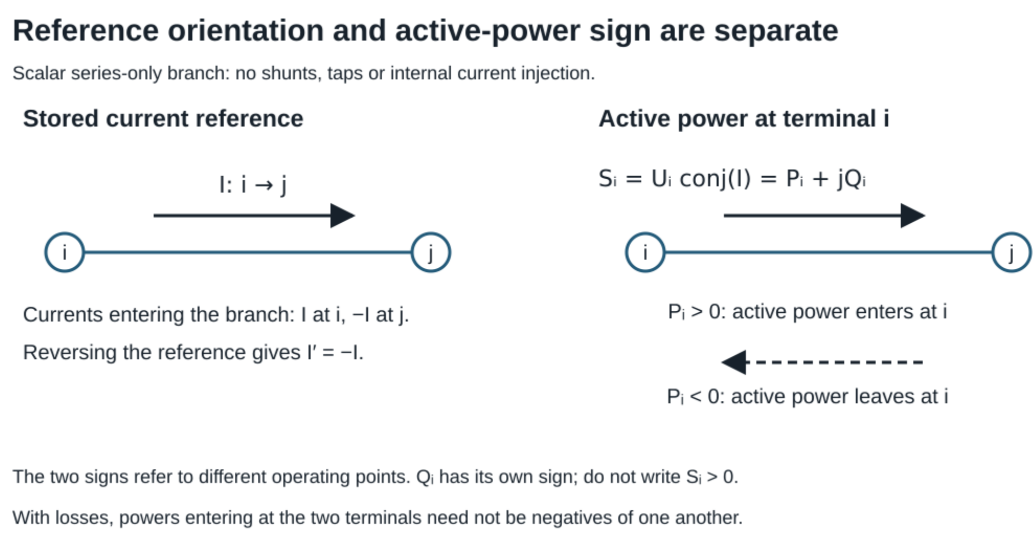

Figure 4.1: Stored orientation and operating-point power transfer are separate records.

Series current and terminal power

For a reciprocal scalar series element, use current into the element at each terminal:

$$I^{\mathrm{s}}_{\ell ij} = Y_\ell(U_i - U_j), \qquad I^{\mathrm{s}}_{\ell ji} = -I^{\mathrm{s}}_{\ell ij}.$$

The corresponding terminal complex-power injections are

$$S_{\ell ij} = U_i (I^{\mathrm{s}}_{\ell ij})^*, \qquad S_{\ell ji} = U_j (I^{\mathrm{s}}_{\ell ji})^*.$$

Although series current is antisymmetric, terminal power is not conserved:

$$S_{\ell ij} + S_{\ell ji} = (U_i - U_j)(I^{\mathrm{s}}_{\ell ij})^* = Z_\ell |I^{\mathrm{s}}_{\ell ij}|^2.$$

For $Z_\ell = R_\ell + \mathrm{j}X_\ell$ with $R_\ell \geq 0$, the active-power absorption is

$$P_{\ell ij} + P_{\ell ji} = R_\ell |I^{\mathrm{s}}_{\ell ij}|^2 \geq 0.$$

A lossy branch therefore owns two terminal powers and a loss relation, not one conserved scalar flow. Calling $S_{\ell ij}$ the *power flow from* i is a useful shorthand only after its terminal sign convention is fixed.

The familiar single-flow picture is recovered in a declared lossless approximation, where $P_{\ell ij} = -P_{\ell ji}$. It should be presented as a special collapse rather than the semantics of a general edge.

Circuit-theory trap

Opposite series currents do not imply opposite terminal powers. Voltage differs across the impedance, so the two terminal powers sum to the element's complex absorption. A nominal-π factor can additionally have terminal currents that are not negatives because it contains shunts.

Nominal-pi elements

For a scalar nominal-π element, define

$$
\begin{aligned}
I_{\ell ij} &= I^{\mathrm{s}}_{\ell ij} + Y^{\mathrm{sh}}_{\ell ij} U_i, \\
I_{\ell ji} &= -I^{\mathrm{s}}_{\ell ij} + Y^{\mathrm{sh}}_{\ell ji} U_j.
\end{aligned}
$$

Then

$$
I_{\ell ij} + I_{\ell ji} = Y^{\mathrm{sh}}_{\ell ij} U_i + Y^{\mathrm{sh}}_{\ell ji} U_j,
$$

which is generally nonzero. Current has not been destroyed. It has been diverted through shunt paths retained inside the composite line factor. A conductive shunt absorbs active power, while inductive or capacitive shunts exchange reactive power.

The terminal-power balance is

$$
\begin{aligned}
S_{\ell ij} + S_{\ell ji} = Z_\ell |I^{\mathrm{s}}_{\ell ij}|^2 \\
&+ |U_i|^2 (Y^{\mathrm{sh}}_{\ell ij})^* + |U_j|^2 (Y^{\mathrm{sh}}_{\ell ji})^*.
\end{aligned}
$$

For a multiconductor member the same distinction is expressed by the complete two-end primitive

$$
\begin{bmatrix} \mathbf{I}_{\ell ij} \\ \mathbf{I}_{\ell ji} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_\ell + \mathbf{Y}^{\mathrm{sh}}_{\ell ij} & -\mathbf{Y}_\ell \\ -\mathbf{Y}_\ell & \mathbf{Y}_\ell + \mathbf{Y}^{\mathrm{sh}}_{\ell ji} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i[\mathbf{N}_{\ell i}] \\ \mathbf{U}_j[\mathbf{N}_{\ell j}] \end{bmatrix}.
$$

Full coupling makes a story about independent power commodities travelling on individual conductor edges still less reliable. The invariant object is the declared multiport relation and its terminal power balance.

Internal versus explicit shunts

The same fixed nominal-π behaviour can be factorized in two ways:

1. one composite two-port line factor containing series and shunt terms;
2. one series factor plus two explicit shunt factors attached to the endpoint junctions.

In the second factorization the series currents are negatives, and the shunt factors separately account for current to ground or neutral. In the first, the composite line's terminal currents are not negatives. Moving between the two is an exact compilation only when terminal coordinates, grounding scope, parameters, limits, ownership, and provenance are mapped explicitly.

This is why statements such as *current is conserved on every edge* are representation dependent. KCL is conserved at the complete network level; which internal path is called an edge depends on the factorization.

When a directed graph is physical

A directed edge is appropriate when order is intrinsic to the represented relation, for example:

- a one-way communication or control dependency;
- a protection logic dependency;
- a device with a genuinely asymmetric admissible transfer set;
- a study graph of causal or optimization dependencies.

Even then, a multiport factor may be more faithful than a directed ordinary edge. A controllable converter can have oriented information flow, terminal power variables at several ports, losses, and bidirectional feasible operating regions at the same time.

Direction contract used here

An arrow in this book is interpreted through the five direction concepts above. Unqualified phrases such as *directed line*, *power on the edge*, and *reverse current* are therefore replaced by the physical incidence, stored orientation, terminal quantity, or operating-point statement actually meant.

Chapter 5

Circuit formulations and assembly

Page status: formulation guide and equivalence contract; the definitions and guards below are book conventions supported by circuit and power-system precedents. General equivalence theorems beyond the declared linear scopes remain future work.

The same power network can be compiled into several equation systems. A nodal admittance matrix is one particularly useful target, but it is not a universal representation of general *power networks*. A numerical $\mathbf{Y}$ may be available while asset identity, switch states, branch currents, grounding, multi-terminal behaviour, controls, limits, or decisions have already been discarded. Conversely, a faithful source model may require a tableau, modified nodal system, branch-current variables, or an unevaluated port–factor relation.

Start with the worked assembly in From source data to a canonical network model. This chapter then asks which equation systems can express the devices and observations a study needs, and when elimination preserves them. Further graph constructions are in From source graphs to views and graph surgery.

Formulation families

Let $\mathcal{M}$ be a declared power-network model with voltage, current, power, control, state, and parameter variables. A formulation is a choice of unknowns x, equations $F(x; u, \theta) = 0$, inequalities $g(x; u, \theta) \leq 0$, and an observation map h. The formulation is therefore more than a matrix: the variable ownership, limits, state domain, and recovery map are part of its contract.

The circuit literature distinguishes nodal analysis from modified and tableau extensions precisely because some elements do not admit a convenient admittance-only branch relation [2, 3]. Power-system work makes the same boundary operational: sparse tableau formulations retain multi-port element variables and node–breaker actions that would otherwise require changing Y_{bus} between states [4].

These formulations may be mathematically equivalent after elimination on a restricted domain. They are not interchangeable source models: eliminating a current variable can remove the direct place to attach a thermal limit, a switch state, or a protection relation.

Numerical footprint of the formulation choice

The choice of equation target also changes the symbolic problem seen by a solver. Schur elimination can add fill edges, while a Jacobian dependency graph records equation–variable relations rather than physical incidence. These are formulation-level effects, so they belong here beside the lowering boundary rather than only in the later executed numerical witnesses.

The figures are conceptual structural views. The pinned numerical export and nonlinear KKT witnesses remain in Numerical consequences of representation and reduction, where their fixture-specific counts, residuals, and solver limitations are recorded.

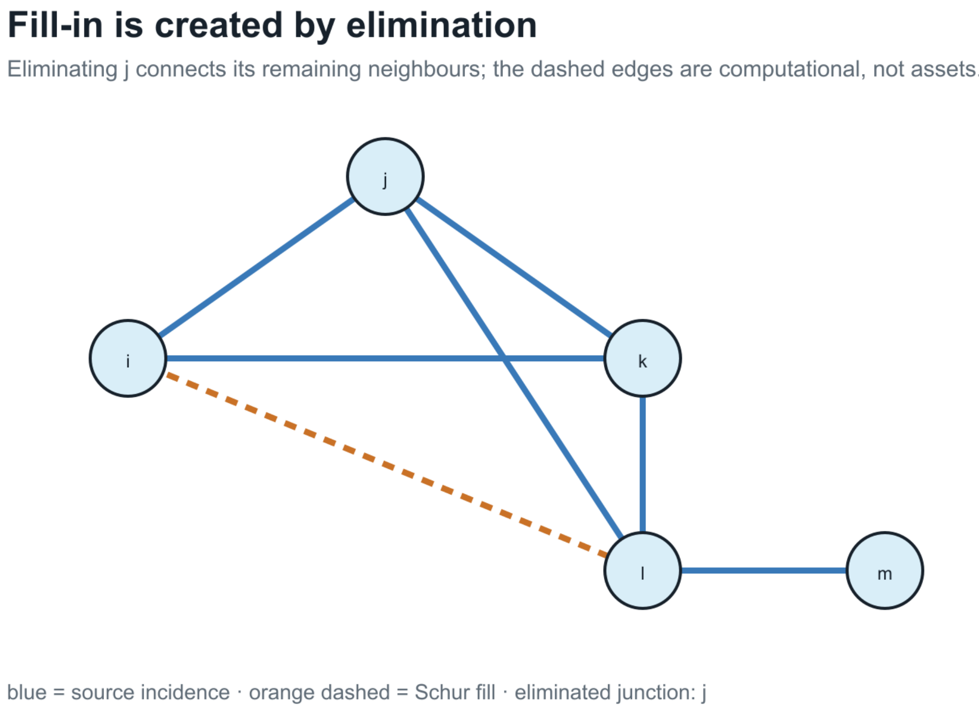

Figure 5.1: Schur elimination creates structural fill-in.

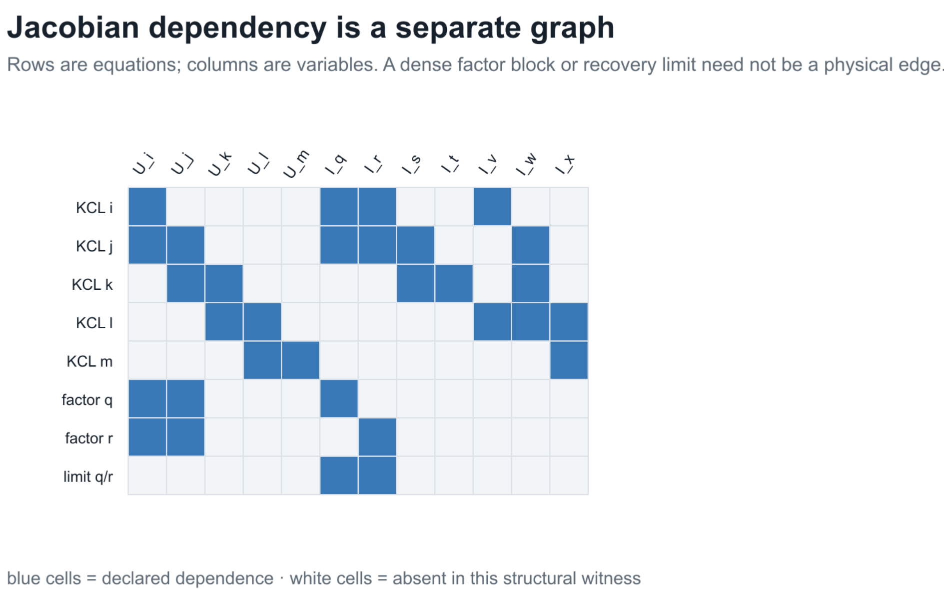

Figure 5.2: Jacobian dependency is a separate graph from physical incidence.

Formulation	Primary unknowns	Strong use	Typical loss or guard
nodal admittance	retained node voltages	fixed linear network stamping and PF linear subproblems	requires admittance-form branch relations; hides branch variables and asset identity
modified nodal analysis	node voltages plus selected branch currents	ideal voltage sources, current-controlled elements, switches, and mixed devices	larger saddle-point system; source provenance still has to be attached
sparse tableau	node, branch, port, and device variables	OPF limits, node–breaker models, multiport factors, and fixed sparsity	more variables and equations; elimination is a separate transformation
branch-current or branch-flow	voltages, currents, and often powers on identified members	explicit ratings, radial identities, and device-level constraints	equations depend on chosen branch model and orientation conventions
hybrid/port parameters	selected boundary efforts and flows	multiport composition and boundary equivalents	a representation is local to declared ports and may be singular at some frequencies or states
general port–factor relation	ordered typed ports and behavioural relations	nonlinear, controlled, dynamic, and multi-terminal equipment	not itself a single sparse matrix or ordinary graph

When an exact nodal admittance target exists

For a declared set Φ_{lin} of fixed linear factors, choose a retained voltage vector $\mathbf{U}$. If each factor has an admittance-form terminal relation

$$\mathbf{i}_\phi = \mathbf{Y}_\phi \mathbf{A}_\phi \mathbf{U}$$

and its current injection maps back through $\mathbf{A}_\phi^{\mathsf{T}}$, then assembly gives

$$\mathbf{Y}^{\mathrm{N}} = \sum_{\phi \in \Phi_{\mathrm{lin}}} \mathbf{A}_\phi^{\mathsf{T}} \mathbf{Y}_\phi \mathbf{A}_\phi, \qquad \mathbf{i}^{\mathrm{inj}} = \mathbf{Y}^{\mathrm{N}} \mathbf{U}.$$

Coupled branches and the equivalent lattice

Mutually coupled two-terminal sections are an instructive exact lowering. The source relation is not a sum of independent branch factors: the complete coupling group first supplies a joint impedance $\mathbf{Z}_\Gamma$. When it is invertible, $\mathbf{Y}_\Gamma = \mathbf{Z}_\Gamma^{-1}$ and one incidence map for all participating section drops gives

$$\mathbf{Y}_\Gamma^{\mathrm{N}} = \mathbf{A}_\Gamma^{\mathsf{T}} \mathbf{Y}_\Gamma \mathbf{A}_\Gamma.$$

For two scalar sections on four distinct terminals, this stamp admits an exact six-edge lattice realization. Four of those edges cross between the two source endpoint pairs and two can carry the negative of the mutual admittance under the stored orientation. The lattice is therefore an ordinary weighted graph, but not a physical bus–branch inventory: its cross-voltage edges are generated couplings and its source line currents require the recovery map $\mathbf{i}_\Gamma = \mathbf{Y}_\Gamma \mathbf{A}_\Gamma \mathbf{U}$.

The coupled multi-voltage corridor case derives the full sign pattern, per-unit scaling, partial-overlap sections, and state guards. If the joint primitive is singular or the target edge library cannot carry the generated weights, the compiler must retain a direct factor or tableau rather than forcing this lattice.

Here Φ_{lin} is a formulation subset, not the entire source inventory. It contains factors whose parameters and states are fixed for the declared solve. A factor carrying an unfixed tap, switching state, control law, or other decision remains in the equation/constraint operator unless the target is explicitly rebuilt pointwise over that decision domain.

This is an assembly identity, not a claim that $\mathbf{Y}^{\mathrm{N}}$ is a canonical factorization or a complete power-network model. A useful sufficient guard for exact nodal stamping is:

The important degenerate case is covered in Multigraphs for expert modelers: identifying the two terminals of a fixed linear π factor can leave an exact one-terminal shunt stamp, but only after the factor has been assembled through its terminal map. Deleting a graph self-loop before that compilation is not an equivalent operation in general.

1. the retained variables are node-voltage coordinates sufficient for the declared observation and decision queries;
2. every included factor has a well-defined linear relation in those voltage coordinates, or has already been exactly eliminated with a declared Schur complement and recovery map;
3. voltage-source, current-controlled, dynamic, and algebraic-constraint variables that cannot be eliminated are retained elsewhere in the target;
4. the topology and device state used for assembly are fixed, or the target is rebuilt for each declared state;
5. grounding, reference, terminal maps, and coordinate order are declared; and
6. every limit, control, and decision that the study queries has a surviving variable or an explicit recovery/constraint map.

Source factors versus study-specific nodal splits

The source model and the matrix used by an iterative solver are related but are not the same object. Let μ denote a study mode, σ an active equipment state, and k an operating point or iteration. A current-injection solver may use the split

$$\mathbf{Y}_{\mathrm{lin}}^{(\mu,\sigma,k)}\mathbf{V}^{k+1} = \mathbf{I}_{\mathrm{src}}^{(\mu,\sigma)} + \mathbf{I}_{\mathrm{comp}}^{(\mu,\sigma)}(\mathbf{V}^k).$$

The linear operator may contain passive delivery elements, fixed shunts, constant-admittance load or generator parts, or a declared Norton equivalent. The compensation term carries the remaining nonlinear or operating-point dependent current. A constant-impedance load can therefore appear in the diagonal of $\mathbf{Y}_{\mathrm{lin}}$ while a constant-power load at the same bus appears through $\mathbf{I}_{\mathrm{comp}}$. A generator may be a PV/PQ constraint, a Norton source, a dynamic equivalent, or a current-limited injection depending on the study.

This is not automatically a Newton–Raphson step. Newton's method forms a residual Jacobian for an increment Δx,

$$\mathbf{J}(\mathbf{x}^k)\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x}^k),$$

whereas a fixed-point current-injection method can keep $\mathbf{Y}_{\mathrm{lin}}$ fixed and update only the compensation current. Setting the compensation current to zero for a direct or initial solve is a linear starting model, not proof that the nonlinear source model is itself constant impedance.

OpenDSS documents this separation operationally: power-conversion elements use a constant primitive-admittance part plus compensation currents, and its direct or admittance solution can include load and generator equivalents in the system matrix [5, 6]. Its fault-study equations construct a mode-specific nodal model with source and generator equivalents and the portion of load current not already included in the matrix [7]. These are legitimate formulation targets, not competing definitions of the physical network graph.

The practical rule is to qualify every nodal matrix with its source class, study mode, active state, coordinate order, and linearization or reduction point. Prefer names such as $\mathbf{Y}_{\mathrm{passive}}$, $\mathbf{Y}_{\mathrm{direct}}$, $\mathbf{Y}_{\mathrm{fault}}$, or $\mathbf{Y}_{\mathrm{lin}}^{(k)}$ to an unqualified claim that the system has one uniquely meaningful Y_{bus}.

Rank and sign contract

Let $\mathbf{Y}^{\mathrm{N}}(\sigma, \gamma)$ denote the assembled operator after the declared active state σ and grounding/reference map γ have been applied, and let n_V be the retained voltage dimension. A direct solve for all retained voltages from arbitrary retained current injections requires an invertible square operator:

$$\operatorname{rank}\big(\mathbf{Y}^{\mathrm{N}}(\sigma, \gamma)\big) = n_V.$$

A reference label, a nominal ground, or a removed datum does not establish this rank by itself. An isolated component, a disconnected grounding element, a singular shunt, or a state-dependent topology can leave a nontrivial nullspace. The rank must be checked on the assembled compound operator for the declared state and coordinates. This is the scoped diagnostic supported by [8], not a universal assertion that every grounded power-system model is nonsingular. The executable witness records both a disconnected network with a declared reference and a regular grounded case.

This condition concerns that direct solve, not the existence of an exact nodal relation. A floating resistor with conductance $g > 0$ has

$$\mathbf{Y} = g \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \qquad \mathbf{Y} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0.$$

The singular matrix expresses the current relation exactly. Compatible injections determine a voltage difference; a voltage datum selects a unique representative of the common-offset family. After boundary conditions are imposed, check the operator for the remaining unknowns. A zero eigenvalue of an exported passive matrix is not automatically an error in the full model. The running-network numerical export illustrates why this distinction matters.

Keep four questions separate: can the relation be assembled, does it have gauge freedom, is the boundary-conditioned linear solve unique, and is the complete nonlinear PF/OPF problem solvable? The last question involves more than the passive matrix. Failure of an elimination or representation guard may require retained current variables or constraints; singularity alone does not imply lost physical information.

A nodal matrix is not a complete power-network graph

The support of $\mathbf{Y}^{\mathrm{N}}$ records nonzero coupling among the retained voltage coordinates. It does not, by itself, record:

- which parallel assets contributed to one block;

- whether a coupling came from a line, transformer, grounding factor, or eliminated internal node;
- which switch or breaker state produced the assembled topology;
- branch currents and member-specific ratings;
- multi-terminal factor identity or internal winding variables; or
- the feasible-set and objective semantics of an OPF decision problem.

This is why Two topology levels and the nodal projection treats nodal support as a derived computational view. A support graph can be useful and exact for one query while being insufficient as the source of another.

Modified nodal and sparse tableau targets

Modified nodal analysis augments node voltages with currents through elements whose constitutive relation is not naturally an admittance injection. A common linear schematic form is

$$\begin{bmatrix} \mathbf{Y} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{I}_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ \mathbf{e} \end{bmatrix}.$$

The blocks encode KCL, selected element currents, voltage-source or control relations, and any declared algebraic coupling. The matrix can be sparse even when eliminating $\mathbf{I}_{\mathrm{e}}$ would create dense fill-in.

The sign convention is part of the formulation contract. With $\mathbf{B}$ the signed incidence of selected element currents into the KCL rows, use

$$\mathbf{Y}\mathbf{U} + \mathbf{B}\mathbf{I}_{\mathrm{e}} = \mathbf{J}, \qquad \mathbf{C}\mathbf{U} + \mathbf{D}\mathbf{I}_{\mathrm{e}} = \mathbf{e}.$$

Thus a positive component of $\mathbf{I}_{\mathrm{e}}$ follows the stored element/terminal orientation encoded by $\mathbf{B}$; $\mathbf{J}$ is the positive KCL injection on the right-hand side, and $\mathbf{e}$ is the right-hand side of the selected voltage or device relation. Reversing an element orientation changes the corresponding incidence signs and current coordinate, not the physical factor. The witness uses this convention for its ideal voltage source.

A sparse tableau keeps this idea at the factor boundary. Let $\mathbf{z}$ collect node, terminal, branch-current, power, and device variables. A tableau target records

$$\mathbf{F}_{\mathrm{KCL}}(\mathbf{z}) = 0, \qquad \mathbf{F}_{\mathrm{KVL}}(\mathbf{z}) = 0, \qquad \mathbf{F}_{\mathrm{device}}(\mathbf{z};\theta) = 0, \qquad \mathbf{g}(\mathbf{z};\theta) \leq 0.$$

This is especially useful when the study needs branch currents, multi-port device variables, switch constraints, or member-level limits. The sparse tableau node–breaker precedent keeps breaker actions in component constraints instead of silently replacing each action with a new fixed Y_{bus} [4].

The tableau is not automatically “more physical” or “more canonical.” It is a better target only for the declared questions. If the study asks only for a fixed linear boundary voltage relation, eliminating tableau variables may be appropriate; if it asks for switching, protection, or member ratings, the uneliminated variables are part of the preservation contract.

Branch-current, branch-flow, and chain formulations

A branch-current formulation retains an oriented current for each identified member and writes KCL together with the member constitutive equations. A branch-flow or BFM formulation instead promotes selected complex powers, voltage magnitudes, and current/power products to variables; its balance equations and relaxations depend on the declared branch model and on whether parallel member identities are retained. An aggregate branch-flow equation is therefore not automatically equivalent to a member-current formulation.

For a two-port partition, a chain or ABCD representation may write a transfer relation such as

$$\begin{bmatrix} u_i \\ i_{ij} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} u_j \\ i_{ji} \end{bmatrix}.$$

This can be useful for cascading declared two-port sections, but it is a coordinate choice with its own invertibility and termination guards. It is not a universal substitute for a multi-conductor or multi-terminal factor. Hybrid effort/flow variables and scattering variables can avoid a singular chosen partition in some applications; they change the target coordinates and must carry their own recovery and observation contracts.

Structural solvability of ideal-source blocks

Ideal voltage-source loops and ideal current-source cutsets deserve a small diagnostic before numerical solution. After declaring the branch ordering and sign convention, write the affected KVL or KCL rows as an affine block $Aq = b$. The scoped check used here compares $\operatorname{rank}(A)$ with $\operatorname{rank}([A\ b])$:

rank result	interpretation
$\operatorname{rank}([A\ b]) > \operatorname{rank}(A)$	contradictory source constraints; no solution for that state
equal ranks, but $\operatorname{rank}(A) <$ number of rows	consistent redundant constraints; retain them or remove them with provenance
equal full row rank	consistent independent constraints

The distinction matters for MNA/tableau assembly. A redundant ideal-source row is not automatically an error, while a contradictory loop or cutset makes the declared state infeasible. The generated witness $experiments/generated/circuit-formulation-witness.json$ records both cases for a voltage-source loop and a current-source cutset. This is a rank- consistency diagnostic only; it is not a general statement about DAE index, dynamic regularity, or every possible dependent-source formulation.

Lowering from a source model

The formulation-aware compilation boundary is:

$$\mathcal{M} \xrightarrow{C} \mathcal{M}_{\text{port}} \xrightarrow{A} \mathcal{E} = (\mathbf{F}, \mathbf{g}, \text{obs}, \text{prov}),$$

where $\mathcal{E}$ is a declared equation/constraint operator. Optional targets are then guarded lowerings:

$$\mathcal{E} \xrightarrow{L_{\text{MNA}}} \mathcal{E}_{\text{MNA}}, \qquad \mathcal{E} \xrightarrow{L_{\text{Y}}} (\mathbf{Y}^{\text{N}}, \text{recover}) \quad \text{when the nodal guards hold.}$$

Direct factor stamping into $\mathcal{E}$ is the default. L_Y is not an automatic final pass after every factor compiler. It may fail because a factor needs extra unknowns, because an elimination block is singular, because a state-dependent switch changes the target structure, or because the resulting matrix would omit a queried constraint. The compiler must return a diagnostic and retain the source-to-target map rather than inventing a virtual admittance.

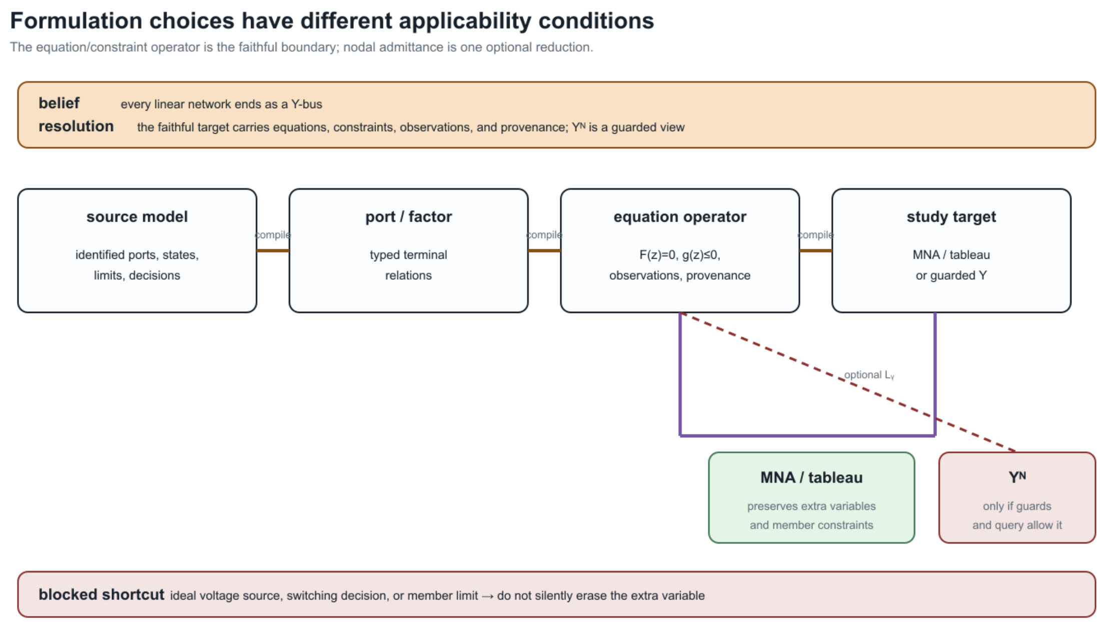

Figure 5.3: Formulation choices: the equation/constraint operator retains the declared model, while MNA/tableau and nodal admittance have different applicability conditions.

Every lowering record should state:

1. the source ports, factors, states, and coordinate order;
2. the target variables and equations;
3. the eliminated variables and the condition for elimination;
4. the preserved observations, constraints, and decisions;
5. the omitted semantics and unresolved guards; and
6. the recovery and provenance maps.

The diagram is intentionally asymmetric. The source model first lowers to an equation/constraint operator that still has somewhere to attach observations, limits, and decisions. MNA or a sparse tableau can retain those variables directly. A nodal $\mathbf{Y}$ target is a separate diagonal branch: it is available only when the declared formulation guards and query contract permit the extra variables and constraints to be eliminated. The crossed shortcut is the common but unsafe mental model in which every linear factor is silently turned into an admittance edge.

This is the formulation-specific instance of the general compiled-view contract. It also explains why a compiler pipeline can legitimately stop at a tableau or factor operator even when a smaller numerical matrix could be constructed for a narrower query.

Formulation equivalence is query-relative

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two equation/constraint operators with solution sets $\mathcal{S}_1$ and $\mathcal{S}_2$. A lowering map $T : \mathcal{S}_1 \to \mathcal{S}_2$ and a recovery map R are not sufficient to call the formulations equivalent without saying what is observed. For an observation family H (voltages, member currents, powers, states, or decisions), we call the pair *equivalent for H* when the declared domains are mapped consistently and

$$H_1(x) = H_2(Tx), \qquad \text{or, after recovery,} \qquad H_1(x) = H_2(RTx).$$

For a decision problem, this contract must also map feasible sets, objective values, and state domains. An algebraically eliminable variable may therefore be harmless for a boundary-voltage query and load-bearing for a member-current limit or switching decision. Approximate maps use the same vocabulary, but must state the error measure, domain, and which observations are only bounded or one-sided preserved.

The executable witness makes the distinction concrete. MNA and a plain nodal description agree for H_{voltage} in the ideal-source example, while they are not equivalent for $H_{\mathrm{voltage+source-current}}$ because the source current is an additional retained unknown. Likewise, summing aligned parallel admittances preserves the terminal voltage relation but not the member-current-limit observation. The rank and sign guards above are part of the domain of these equivalence statements; they are not optional annotations. The rank claim is registered as $FORMULATION - NODAL - 003$.

Scope collapses and literature alternatives

Several familiar power-flow models are valid collapses when their guards are declared:

Starting target	Collapse	What must be checked
fixed linear port factors	$\mathbf{Y}^{\mathrm{N}}$	admittance-form relations, regular elimination, fixed grounding and state
MNA/tableau	nodal admittance	all retained extra variables are eliminable and unqueried
multiconductor model	scalar or positive-sequence Y	phase symmetry, balance, grounding, limits, and observations
node–breaker tableau	bus–branch Y_{bus}	state fixed and switch/protection decisions outside the query
port–factor source	ordinary-edge graph	declared n-port realizability and retained provenance

The alternatives are therefore not a contest for one universally best graph. They are formulation choices indexed by the study's observations, constraints, and decisions. The representation taxonomy classifies their retained meanings; the transformation semantics register records the guards for moving among them.

Power-system shorthand

“The network has a Y-bus” usually means that one study formulation has assembled a nodal operator for one declared state and variable set. It does not mean that the full power-network source model is a simple graph, a two-terminal multigraph, or a lossless collection of admittance edges.

Evidence boundary

The circuit and power-system precedents establish that nodal, modified-nodal, and sparse-tableau formulations are established alternatives. The book's stronger claim—that a formulation-aware lowering contract can preserve the declared power-network decisions and provenance—is a proposed architecture. The generated artifact `experiments/generated/circuit-formulation-witness.json` supplies three minimal failure cases:

1. an ideal voltage source whose source current requires an MNA variable;
2. a floating two-node network whose nodal operator is singular without a reference or shunt; and
3. aligned parallel members whose aggregate admittance is exact for the terminal relation but cannot carry their distinct current limits.

The first case solves two right-hand sides in MNA form, showing identical source voltage but different source currents. The other cases separate numerical singularity from semantic loss. These are scoped formulation witnesses, not a universal impossibility result for every circuit or power-network model.

Part 3: Conductors, connections, and ground

Chapter 6

Load models and connections

Page status: constitutive-model reference chapter with scoped numerical load-model, connection-map, and continuation witnesses plus independent reproduction; broader nonlinear and network-level certificates remain future work.

Two networks can have identical buses, edges, terminals, and ratings while defining different decision problems because their loads obey different voltage laws. A graph representation does not identify a load model. The constitutive relation belongs to the factor or asset layer and must be carried through any preservation claim.

Vocabulary bridge

Electrical state below means the continuous voltages and currents at an operating point. Equipment status, operating scenario, state-estimator metadata, load-model parameters, and a learned hidden state are separate objects even when software stores them in one state container.

Five common voltage laws

Let $v = |\Delta U|$ be the magnitude seen by one load connection and let v^{nom} be its declared anchor. For active power, the standard families can be written as

$$\begin{aligned}
P_{\text{CP}}(v) &= P^{\text{nom}}, \\
P_{\text{CI}}(v) &= P^{\text{nom}} \frac{v}{v^{\text{nom}}}, \\
P_{\text{CZ}}(v) &= P^{\text{nom}} \left(\frac{v}{v^{\text{nom}}}\right)^2, \\
P_{\text{ZIP}}(v) &= P^{\text{nom}} \left(\alpha_Z \left(\frac{v}{v^{\text{nom}}}\right)^2 + \alpha_I \frac{v}{v^{\text{nom}}} + \alpha_P\right), \\
P_{\text{exp}}(v) &= P^{\text{nom}} \left(\frac{v}{v^{\text{nom}}}\right)^{\gamma_P}.
\end{aligned}$$

Reactive power may use its own coefficients or exponent. Constant power does not need v^{nom}; the other laws do. ZIP is a convex combination only when the declared coefficients are nonnegative and sum to one. Integer exponents $0, 1, 2$ make the exponential family coincide with a ZIP special case, but arbitrary exponents do not.

At $v = v^{\text{nom}}$ the families agree by construction. Away from that point they do not. The same topological graph can therefore produce different currents, losses, voltage margins, active constraints, and optimal decisions.

Why this belongs in a graph-model book

Consider a fixed two-bus graph with series impedance Z and a load at the receiving bus. The power-flow relation is not just the graph incidence; it is

$$U_s - U_r = ZI, \qquad S_r = U_r I^*,$$

together with a law for S_r as a function of U_r. Replacing constant power by constant impedance changes the equation graph and the feasible set without changing the bus–branch multigraph. A transformation that preserves only Y or connectivity has not preserved the decision problem unless it also preserves the load relation and the observations that depend on it.

This is a useful counterweight to a common simplification: "the network graph is unchanged, so the OPF is unchanged." The graph is one layer of the model; the load factor is another.

When a load or generator enters the nodal operator

The factor layer and the nodal operator should not be confused. A constant- impedance load is a one-terminal shunt factor, so a declared formulation may stamp its admittance into a diagonal nodal block. A ZIP or constant-power load may instead be represented by a fixed linear part plus a voltage-dependent compensation current. The same distinction applies to generators: a fixed Norton or dynamic equivalent may contribute a matrix block, while a PV/PQ control, current limit, or inverter control law remains an injection or constraint relation.

OpenDSS provides a useful engineering example. Its normal solution uses a system nodal matrix together with compensation currents from nonlinear power- conversion elements; its direct/admittance option solves with load and generator equivalents included in the matrix. The documentation also notes that loads may be switched to an admittance representation for fault studies and that the fault-study matrix has its own source, generator, and load-current composition [5–7, 9]. This is a solver and study choice, not a claim that the underlying load has become a line edge or that the source graph has changed.

The same warning applies to a two-terminal factor whose terminals become the same resolved node. A fixed linear π section can reduce exactly to a constant- admittance shunt after terminal-map assembly, but that compiled diagonal term is not a self-loop edge and does not erase the source factor, its controls, limits, or provenance. See Multigraphs for expert modelers for the derivation and refusal boundary.

The supplied application-directed distribution equivalent illustrates the same boundary at feeder scale. Its reduction first constructs a nodal equivalent and then aggregates PVs and loads at an equivalent node, with phase-shifting and shunt elements added to preserve the selected study responses. The resulting model is validated for declared power-flow and EMT observations; it is not presented as the unique graph of the original feeder [10].

For this reason, a nodal matrix should be annotated with at least:

- the source factor inventory and terminal/attachment maps;
- the study mode and active device state;
- which load, generator, shunt, or source components were stamped into the matrix;
- which nonlinear or controlled parts remain in injection and constraint maps;
- the operating point or iteration used for any linearization; and
- the observations, limits, decisions, and recovery maps that remain valid.

Decision consequences

Load-model choice becomes especially visible in three regimes:

1. **voltage regulation and CVR:** constant-power loads do not reduce demand when voltage is lowered, while impedance loads do;
2. **hosting capacity and voltage limits:** a voltage-dependent load can partially relieve a binding voltage constraint, changing the reported capacity; and
3. **weak feeders and collapse:** constant-power demand can create a nose point, so a local high-voltage solution may disappear as the load scale increases.

The relevant preservation contract must name the load family, its anchor, phase or connection map, and whether the decision varies load scale, voltage, or model parameters. A numerical result that changes after a load model swap is not necessarily a solver error; it may be the intended change in the physical problem.

Per-conductor and connection semantics

For a multiconductor load, v is not automatically a bus-voltage scalar. The factor must declare whether it uses phase-to-neutral, phase-to-phase, sequence, or another terminal combination. A delta load and a wye load can share the same bus terminals while seeing different voltage coordinates. The load model therefore belongs beside the terminal map, not as an unqualified attribute of a graph vertex.

Likewise, a three-phase load may be balanced in name but still have phase-specific P_p and Q_p values. A balanced positive-sequence view preserves it only when the load relation, grounding, limits, controls, and observations close under the same phase symmetry.

Nominal voltage belongs to the load coordinate (LOAD-BASE-001)

For a voltage-dependent load, nominal voltage is not merely descriptive nameplate metadata. It anchors the normalized terminal voltage used by the constitutive law. For example, a ZIP active-power law contains

$$P(V) = P_0 \left[\alpha_Z \left(\frac{|V|}{V_{\mathrm{nom}}} \right)^2 + \alpha_I \left(\frac{|V|}{V_{\mathrm{nom}}} \right) + \alpha_P \right].$$

The voltage V and anchor V_{nom} must use the same terminal coordinate. A WYE load uses phase-to-neutral voltage. A DELTA load uses line-to-line voltage. On a nominal three-phase system with phase-to-neutral base V_{pn}, the corresponding line-to-line base is $V_{\mathrm{ll}} = \sqrt{3} V_{\mathrm{pn}}$. Copying the same numeric value into both load records changes $|V|/V_{\mathrm{nom}}$ by a factor of $\sqrt{3}$; it is a different load law, not a harmless label change.

This rule is declaration-relative. A consistent anchor does not prove that the source voltage, transformer ratios, terminal map, coefficients, units, or solved operating point are correct. Those remain separate validation obligations.

Executable connection-map probe (LOAD-CONNECTION-001)

The generated `connection_maps` witness in `experiments/generated/load-grounding-witnesses.json` uses ordered terminals (a, b, c, n) and applies two explicit linear maps to the same balanced bus:

$$T_Y \mathbf{v} = (V_a - V_n, V_b - V_n, V_c - V_n), \qquad T_\Delta \mathbf{v} = (V_a - V_b, V_b - V_c, V_c - V_a).$$

The wye observations have unit magnitude, while the delta observations have magnitude $\sqrt{3}$ for the recorded positive-sequence voltage. The bus and graph are unchanged; only the factor's terminal map changes. This is a small structural witness, not a complete unbalanced load-flow or rating model.

Modelling checklist

Before comparing two graph views or deleting a factor, record:

- the load connection and ordered terminal map;
- the active and reactive voltage law;
- the nominal voltage anchor and units;
- phase-specific coefficients or exponents;
- whether the law changes with state, control, or time;
- the voltage and current quantities used in limits; and
- whether the study seeks feasibility, load delivery, hosting capacity, voltage regulation, or a different decision.

The book's existing parallel-line and transformer cases preserve their load equations explicitly. This chapter makes the reason general: constitutive models are part of the representation contract, even when they are invisible in a simple graph drawing.

Scoped numerical witness

The generated artifact `experiments/generated/load-grounding-witnesses.json` solves one fixed two-bus network under three load laws. With the same source, series impedance, nominal demand, voltage limit $|U_r| \geq 0.87$ and current limit $|I| \leq 1.00$, the recorded high-voltage solutions are:

Load law	$\lvert U_r \rvert$	$\lvert I \rvert$	Voltage limit	Current limit
CP	0.8592	1.0872	fail	fail
CI	0.8803	0.9341	pass	pass
CZ	0.8937	0.8348	pass	pass
ZIP (active $(0.4, 0.3, 0.3)$, reactive $(0.2, 0.3, 0.5)$)	0.8792	0.9310	pass	pass

This is a decision witness, not a universal ranking of load models. It shows that a graph-preserving change of constitutive law can change both feasibility and the active constraint set.

The left panel places the recorded operating points against the same voltage and current limits. The right panel is the finite continuation probe: CP is the first family to fail the declared iteration test, while the other three families remain traceable through scale 3.0. Neither panel is a universal voltage-collapse or load-model-ranking theorem.

The artifact `experiments/generated/load-model-independent-reproduction.json` repeats the same damped fixed-point calculation with a separate standard-library Python implementation. It reproduces all four recorded rows and their voltage/current limit decisions. This supports reproducibility of the declared scalar fixture; it does not establish global solvability, a universal ranking of load laws, or the adequacy of CP/CI/CZ/ZIP for a particular utility study. The ZIP row uses nonnegative active and reactive coefficients that each sum to one; the two coefficient triples are deliberately distinct to expose that reactive demand need not follow the active-power law.

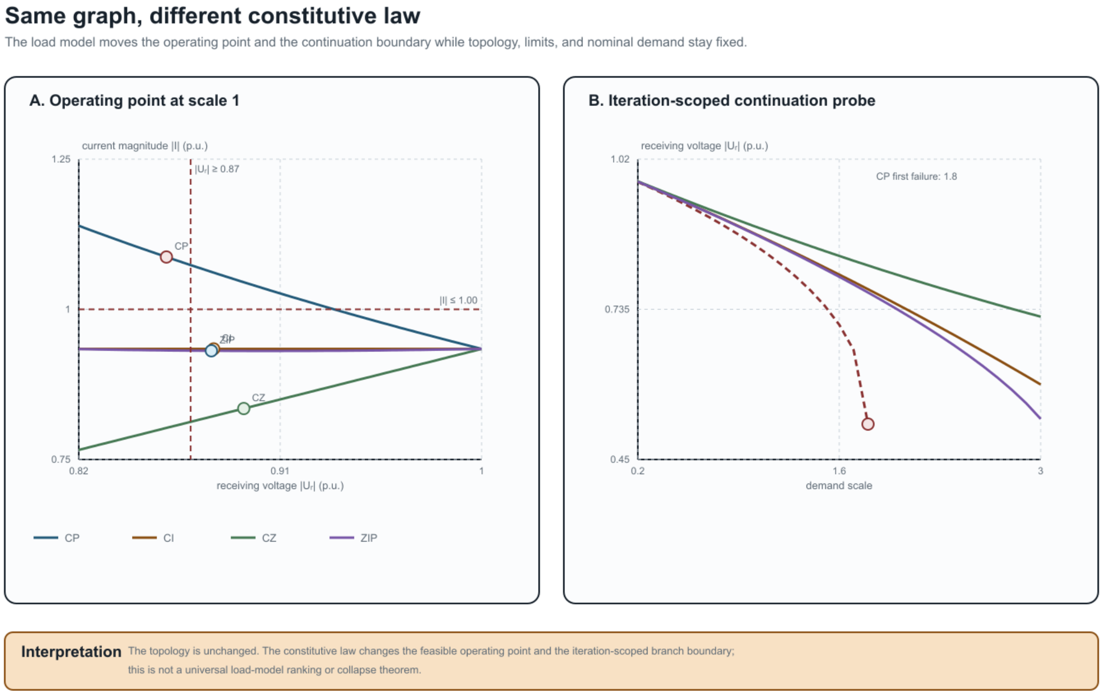

Figure 6.1: Operating-point divergence and continuation boundaries for four load laws.

Continuation probe (LOAD-CONTINUATION-001)

The same scalar fixture also includes a demand-scale continuation in `load_continuation`. It tracks the previous high-voltage iterate over scales $0.2, 0.3, \ldots, 3.0$ with damping 0.5. The recorded CP branch converges through scale 1.7 and fails the declared iteration/residual test at scale 1.8; CI, CZ, and the ZIP branch remain converged through scale 3.0. This is useful as a warning that a load-model change can move a numerical branch boundary, but the boundary is solver- and continuation-specific. It is not a global voltage- collapse theorem, and the artifact does not claim that the first failed iterate is the exact saddle-node point.

The generated artifact `experiments/generated/load-continuation-independent-reproduction.json` repeats the complete sampled path with a separate standard-library Python implementation. It matches all convergence flags, every converged voltage and residual row, and the CP failure scale. This independently reproduces the algorithm-scoped observation; it still does not convert iteration failure into an infeasibility certificate or a mathematically continued nose curve.

Chapter 7

Earth, neutral, and reference

Page status: scoped model-class taxonomy with a numerical E_2 explicit-earth/protection witness; asset-aware protection studies remain future work.

Evidence boundary

Scope: reduced-earth, neutral/grounding-factor, and explicit-earth model classes in the declared steady-state fixtures. **Evidence:** definitions, impedance constructions, and scoped numerical witnesses. **Numerical optimality:** no protection or network optimization result is claimed by the taxonomy or fixtures. **Unresolved boundary:** source-faithful earth-return data, asset-aware protection, uncertainty, and independent engineering review.

"Ground" is used for several different objects in power-system models. This book distinguishes a mathematical voltage reference, a neutral conductor, an earth-return path, and a physical grounding asset. A representation must state which of these it contains before a reduction or fault claim is interpreted.

Four distinct meanings

- **Reference:** a gauge choice such as $U_0 = 0$. It fixes coordinates but does not carry current or represent soil impedance.
- **Neutral:** an explicit conductor or terminal with its own voltage, current, impedance, and limits. It may be grounded at one or more locations.
- **Earth return:** a conductive path through soil, ground wire, shield wire, or a reduced earth-impedance model. It can carry current and couple phases.
- **Grounding asset:** an electrode, grid, bond, transformer grounding point, or protection object with identity, state, and maintenance semantics.

Conflating these meanings can make a phase-to-neutral reduction appear exact while deleting a ground-fault path or a neutral-to-earth constraint.

Model classes

The classes are not a strict accuracy ladder. E_1 may be the right model for a study whose observations are terminal voltages, while E_3 is required for a grounding-asset outage decision even if both use the same external admittance.

The ladder is generated by `experiments/render_earth_return_ladder.py`. Its arrows indicate added modelling commitments, not a universal accuracy ranking.

Class	Electrical representation	Typical use	What it cannot answer by itself
E_0 ideal reference	algebraic reference or gauge; no earth current variable	balanced transmission PF/OPF	earth-return current, grounding impedance, fault path
E_1 reduced earth return	earth effects embedded in sequence, Carson, or fitted impedance/shunt blocks	feeder PF, planning, approximate fault studies	identity and state of the physical earth path unless linked
E_2 explicit earth conductor	conductor or port with voltage/current and mutual coupling	four-wire, shield-wire, grounding, protection studies	soil detail beyond the selected conductor model
E_3 asset-aware grounding	E_1 or E_2 plus electrodes, bonds, grids, protection and maintenance relations	switching, outage, protection, asset decisions	none beyond the declared physical and study model

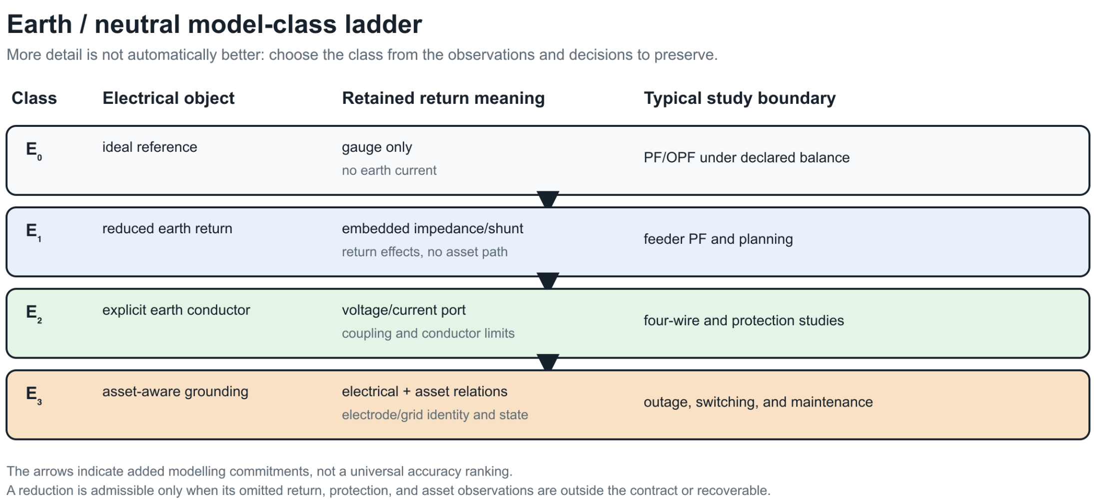

Figure 7.1: Earth and neutral model-class ladder, from an ideal reference to asset-aware grounding.

Scope contract for reductions

A transformation involving neutral or ground must record:

1. the reference convention and gauge;
2. whether neutral and earth are separate ports;
3. the earth-return class E_0–E_3;
4. the number and location of ideal or finite grounding points;
5. whether line shunts, mutual coupling, and ground-fault currents are retained;
6. which voltage, current, protection, and asset observations are preserved;
7. a recovery map for eliminated neutral or earth quantities.

For example, the phase-to-neutral map in the circuit-coordinate chapter can be exact for phase-to-neutral terminal behaviour under sparse compatible grounding, while being insufficient for E_2 neutral-to-earth voltage limits. The Geth–Heidari–Koirala reduction is useful precisely because it states grounding and radiality conditions rather than treating a three-wire picture as an automatic physical deletion [11].

Circuit-theory trap

Setting the reference potential to zero is not the same as setting a conductor voltage to zero, and neither operation removes an earth-return current. A reduced model must say which variable was fixed, eliminated, or merely not represented.

Consequences for the running network

The running fixture uses a reduced-earth E_1 model with an explicit neutral and grounding factor: h_n is a finite neutral-to-earth shunt, the four-wire line retains a neutral terminal, and the source reference is declared separately. It does not contain an explicit earth conductor or earth port, so it is not an E_2 model. The fixture does not claim a detailed soil or electrode model, so grounding-asset decisions remain outside its current numerical scope. Future explicit-earth cases should add a physical earth conductor or grounding-grid factor and test recovery of ground currents, neutral voltages, and protection observations. The separate E_2 witness below provides that minimal explicit-earth check without changing the classification of the running fixture.

A useful comparative case

The BMOPFTools grounding tutorial suggests a compact experiment that fits the book's preservation story. Hold the line matrix, load, and bus graph fixed and vary only the customer-end grounding relation:

State	Added relation	Observations that should change
floating neutral	no local neutral-to-reference path	neutral-to-earth voltage and return-current allocation
impedance grounded	finite electrode shunt	electrode current, neutral displacement, and phase-to-neutral voltage
perfectly grounded	ideal voltage constraint	neutral voltage by construction and a different current path
explicit earth return	earth conductor or factor with its own impedance	earth current, neutral current, and grounding-asset limits

This is not four different graphs in the simple-topology sense. It is one asset/connectivity view with four electrical and decision models. A study that only asks whether buses are connected sees no change; a study that asks for neutral limits, touch voltage, fault current, or electrode maintenance does. The comparison is now instantiated in the scoped artifact `experiments/generated/load-grounding-witnesses.json`. It keeps the same two-conductor bus–branch graph and varies only the customer-end relation. The recorded neutral-voltage magnitudes are approximately 0.14184 (floating), 0.10026 (finite impedance), and 0.00000 (ideal grounding); the corresponding ground-current magnitudes are 0, 0.27807, and 0.88999 per unit. These values make the decision point concrete without claiming that the three-state scalar fixture represents an explicit-earth-conductor or protection study.

The comparison is therefore evidence for the E_0-E_3 scope contract, not a reason to promote “grounded” to a single graph attribute.

Explicit earth conductor and a protection observation

The same generated artifact contains a three-node linear fixture with phase, neutral, and earth conductor voltages. A finite neutral-to-earth bond is retained as a separate relation. Three resolved states are compared:

State	Earth voltage magnitude	Earth-conductor current	Fault current	Protection observation
earth in service	0.04975	0.18540	0	no trip
earth conductor maintenance outage	0.14184	0	0	no trip
phase-to-earth fault	0.52173	1.94437	2.81114	trip
neutral-to-earth fault	0.07851	0.29260	0.25112	no trip

The outage and fault rows show why an explicit earth port cannot be replaced by an ideal reference: the earth conductor has its own current, availability, protection observation, and asset identity. The witness also records earth-node voltage as a touch-voltage observation: the declared 0.10 pu limit passes in service (0.04975 pu) and fails for both the maintenance outage (0.14184 pu) and the faults (0.52173 pu and 0.07851 pu). The earth-conductor maintenance decision is retained with its asset state and cost. Protection uses a CT ratio of 10 and a 0.20 pu secondary pickup: the phase fault measures 0.28111 pu, trips after 0.2466 s under the declared inverse-time curve, while the neutral fault measures 0.02511 pu and does not trip. A separate saturation probe caps the phase-fault secondary current at 0.18 pu and changes that trip decision. These are declared sensitivity models, not relay or CT standards. This is a deliberately small E_2/E_3 boundary witness and does not model relay curves beyond the illustrative curve, CT electromagnetic behaviour, or a complete fault-class enumeration.

Chapter 8

When a balanced model is sufficient

Page status: controlled positive-sequence specialization with factor- and network-level executable witnesses; global decision equivalence and external review remain open.

Evidence boundary

Scope: three-phase linear factors and networks with cyclic symmetry, balanced boundary data, and positive-sequence observations. **Evidence:** sequence-coordinate derivation, balanced transmission witness, and independent numerical reimplementation of that fixture. **Numerical optimality:** the witness is not a global nonlinear OPF result. **Unresolved boundary:** exactness for untransposed or unbalanced networks, phase-specific decisions, and external mathematical review.

The general multiconductor, multi-terminal model is a baseline for preserving meaning. It is not a claim that every study needs all of that structure. This chapter gives a positive derivation of the familiar balanced positive-sequence bus–branch model and lists the assumptions that make the collapse admissible.

The balanced invariant subspace

Let a three-phase two-terminal factor use the phase order (a, b, c) and the Fortescue matrix $\mathbf{F}$

This is the classical symmetrical-coordinate construction introduced by Fortescue [12]. The transform itself is always invertible; sequence decoupling is not. Decoupling requires the phase-domain operator, including connected series/shunt factors and grounding, to leave the sequence subspaces invariant.

$$\mathbf{v}_{abc} = \mathbf{F}\mathbf{v}_{012}, \qquad \mathbf{F} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix}, \qquad a = e^{\mathrm{j}2\pi/3}.$$

The positive-sequence subspace is

$$\mathcal{V}_+ = \{\mathbf{F}[0, V_1, 0]^\intercal : V_1 \in \mathbb{C}\}.$$

An exactly balanced operating point has phase voltages and currents in this subspace, with the corresponding phase shifts. Balance is a restriction on the admissible state and injections; it is not created by drawing one edge per bus pair.

Factor-level assumptions

Consider a linear series or nominal-π factor with phase-domain relation $\mathbf{i} = \mathbf{Y}\mathbf{v}$. A sufficient exact-collapse contract is:

1. **compatible phase sets:** every retained terminal has the same ordered three-phase set, with no unresolved neutral or earth port;
2. **cyclic symmetry:** each series and shunt matrix commutes with the cyclic phase permutation, equivalently it is circulant in the declared phase coordinates;
3. **balanced boundary data:** sources, injections, and measurements are restricted to $\mathcal{V}_+$;
4. **sequence-compatible grounding:** zero- and negative-sequence return paths are either absent from the study or represented by fixed factors whose constraints are not queried;
5. **two-terminal closure:** each device is already a two-terminal factor, or a multi-terminal device has an explicitly verified positive-sequence compilation;
6. **decision symmetry:** controls are common to all phases, and limits, costs, and admissible states are invariant under the same phase symmetry;
7. **observation restriction:** the study does not ask for phase-specific, neutral-to-ground, negative-sequence, zero-sequence, or internal winding quantities.

Under these assumptions, $\mathbf{F}^{-1}\mathbf{Y}\mathbf{F}$ is diagonal in sequence coordinates, and the positive-sequence block is invariant:

$$\mathbf{Y}_{012} = \mathbf{F}^{-1}\mathbf{Y}_{abc}\mathbf{F} = \mathrm{diag}(Y_0, Y_1, Y_2), \qquad I_1 = Y_1 V_1.$$

Transposition alone is therefore not an exact positive-sequence reduction. It can support a balanced or approximately cyclic model, but exact restricted closure still requires the operator, grounding, devices, limits, controls, and observations to respect the declared sequence symmetry.

The federated applicability guard `PSK-000009` makes this boundary executable through BMOPFTools contract `positive_sequence_collapse_applicability`. The compact check verifies circulant series/shunt factors, phase order, and the scalar positive-sequence relation, while requiring explicit declarations for balanced boundary data, sequence-compatible grounding, two-terminal closure, phase-symmetric decisions, and positive-sequence observations. Its pass is a restricted relation check; it does not certify an unbalanced feeder or any phase-specific, neutral/earth, protection, internal-device, objective, or solver observation.

For a nominal-π factor, the same statement applies to the series and shunt blocks separately. The positive-sequence network is therefore a derived two-terminal scalar complex network with one voltage and current per bus/arc, provided the factor library and decision constraints close under the restriction.

The generated witness `experiments/generated/positive-sequence-collapse-witness.json` diagonalizes one circulant impedance matrix to numerical precision and records a non-circulant perturbation that mixes sequences. The companion `experiments/generated/balanced-transmission-witness.json` assembles a three-bus, two-arc nominal-π network with the same cyclic series and shunt factors. It solves the reduced phase-domain equations and the scalar positive-sequence equations independently, embeds the scalar voltages and branch currents, and checks their residuals. The result is a network-level positive example for the declared balanced observation family, not a global decision-equivalence theorem.

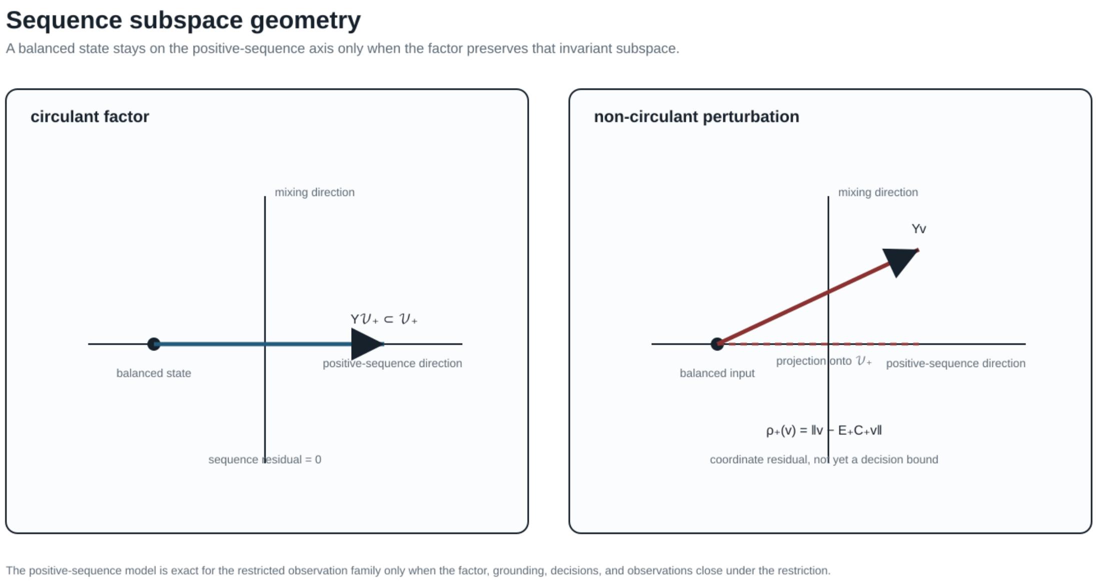

Figure 8.1: Positive-sequence invariant subspace and sequence mixing.

The generated comparison `experiments/generated/balanced-transmission-independent-reproduction.json` repeats the phase-domain and scalar solves with a separate standard-library complex Gaussian-elimination implementation. It matches the Julia voltages, residuals, and branch-current checks row-for-row. This is independent numerical reproduction of the declared fixture, not independent review of the modelling assumptions or a standards-aligned transmission validation.

The geometry is a projection aid: the circulant factor leaves the positive-sequence axis invariant, while the perturbed factor produces an off-axis residual. The residual ρ_+ is a coordinate diagnostic until propagated through constraints and decisions.

Network-level derivation

Let C_+ restrict a general model to the positive-sequence subspace and let E_+ embed a positive-sequence state into phase coordinates. Under the factor, boundary, and decision assumptions above,

$$\mathcal{F}_+ = C_+(\mathcal{F}_{abc}), \qquad E_+(\mathcal{F}_+) \subseteq \mathcal{F}_{abc},$$

and the declared observations agree on the embedded balanced states as

$$h_{abc} \circ E_+ = \widehat{h}_+.$$

This is a **restricted** factorization on $E_+(\mathcal{F}_+)$. It is not the stronger global identity $h_{abc} = \widehat{h}_+ \circ C_+$ on arbitrary phase-domain states.

Thus the positive-sequence model is exact for the restricted observation family H_+. It is not exact for the unrestricted phase-domain feasible set unless every feasible phase-domain point is balanced, which is a much stronger statement.

For an approximately balanced model, define the residual of a phase-domain state v by

$$\rho_+(\mathbf{v}) = \|\mathbf{v} - E_+ C_+ \mathbf{v}\| .$$

This is a coordinate residual, not a decision-error bound. A voltage residual must be propagated through the factor equations and constraints before it can support an engineering approximation claim.

What collapses, and what does not

General object	Positive-sequence image	Required qualification
three phase voltages/currents	one complex sequence variable	only balanced states are represented
circulant series and shunt matrices	scalar Y_1 blocks	non-circulant coupling produces sequence mixing
identical phase limits and common controls	one sequence limit/control	phase-specific constraints are outside the image
two-terminal factor	oriented bus–branch arc	multi-terminal devices need a verified compiler
neutral/earth factor	omitted or externally resolved	grounding and zero-sequence questions are forgotten
phase-specific measurements	aggregate sequence observation	not preserved by the quotient

The running fixture intentionally fails several guards: it has four-wire lines, an explicit grounded neutral, nonuniform terminal sets, a phase permutation, unbalanced loads, full coupled matrices, and a three-winding transformer. It is therefore a test of the general model, not a balanced positive-sequence witness. The transmission specialization must be a separately declared fixture or a parameterized subcase with its own residual and checks.

Power-system shorthand

"Use a positive-sequence model" is a modelling decision with assumptions, not a graph-theoretic simplification. State the balance, transposition, grounding, equipment, limit, and observation assumptions before treating the resulting bus–branch graph as exact.

Decision consequence

The positive-sequence collapse is exact for a restricted decision problem when the feasible-set inclusion, recovery embedding, and observation factorization above hold. If a contingency opens one phase, a relay observes zero sequence, a transformer has phase-dependent taps, or a conductor limit becomes active, the decision domain has left $\mathcal{F}_+$. The general port–factor model or an explicitly guarded intermediate model is then required.

Part 4: Graphs for different computations

Chapter 9

One network, many graphs

Page status: explanatory synthesis introducing the representation landscape.

Calling something *the network graph* hides the modelling decision that produced it. The same physical power network can yield several non-isomorphic structures, each correct for some questions and inadequate for others.

Power-system shorthand

Treat *the network graph* as an omitted noun phrase. Ask whether the sentence means an asset graph, active topology, terminal-connectivity graph, bus–branch multigraph, factor incidence graph, or equation/sparsity graph.

A deliberately difficult network

The running network for this book contains ordered conductor terminals, an explicit neutral and grounding impedance, heterogeneous parallel lines, switchgear, and a genuinely multiwinding transformer. It also contains limits and controls used in a decision problem. Its complete semantic specification is given in The running multiconductor network. Its first numerical realization and the six illustrated views described below are given in the Executable running network.

The representation-scoped meanings of cycles, parallelism, bridges, leaves, and radial ends are developed in Cycles, parallelism, and radial structure.

The first surprise is not a new device; it is a change of graph. At the bus level the feeder can be radial, while dense multiconductor stamps create cliques—and therefore cycles—in the scalar support graph used by a matrix algorithm. The drawing below shows only cross-bus support edges; these already contain four-cycles. Within-bus entries are omitted:

The resolving phrase is **which graph?** Bus-level radiality is a statement about equipment connectivity. Support-graph cycles are a statement about algebraic coupling. When the full stamps give the required clique structure, the support graph can be chordal and admit leaf-clique elimination. The drawing above does not show that full pattern and is not itself chordal. Neither case implies an additional physical loop.

Nothing exotic is required to create representational disagreement. Parallel lines already show the issue. If two branches ℓ_1 and ℓ_2 connect buses i and j, a multigraph retains both triples

$$\ell_1 ij,\ \ell_2 ij \in \mathcal{T}^{L\to}.$$

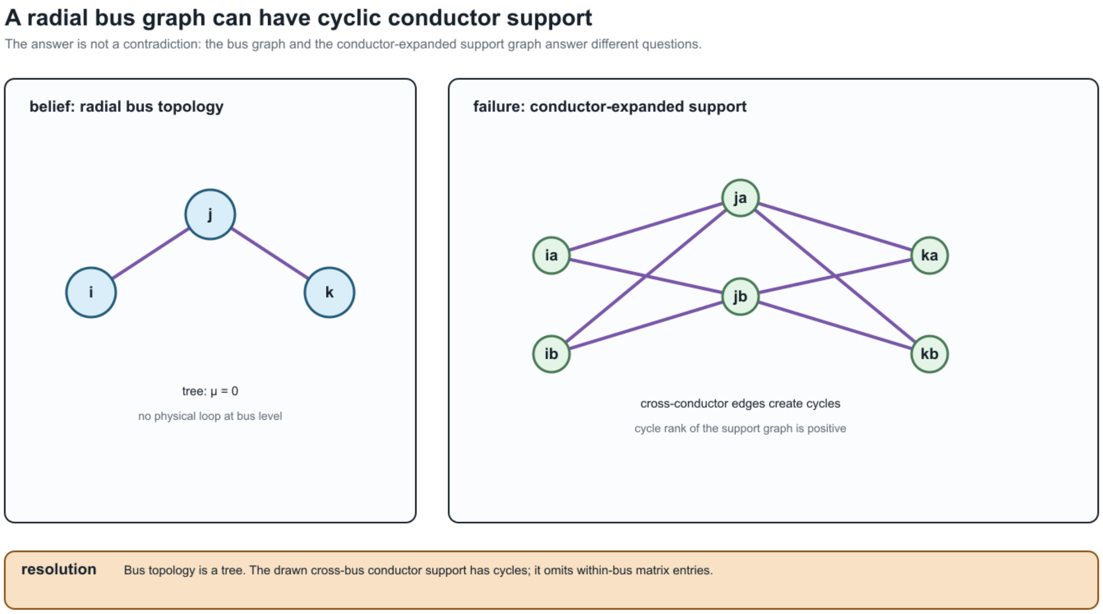

Figure 9.1: A radial bus-level feeder can have cycles in its conductor-expanded support graph.

Here i and j are bus labels, not necessarily row and column numbers in an array. If a software implementation enumerates the bus set, its stored entry is written $[\mathbf{Y}]_{\kappa(i),\kappa(j)}$; the semantic relation remains $\mathbf{Y}_{ij}$. This is the same label-before-coordinate rule used in the equation-reading bridge.

A simple graph retains only the adjacency $i \sim j$. A weighted simple graph might store

$$\mathbf{Y}_{ij}^{\mathrm{eq}} = \mathbf{Y}_{\ell_1} + \mathbf{Y}_{\ell_2},$$

but this does not by itself retain individual current limits, outages, maintenance states, ownership, or investment choices.

Six useful views

Asset view

The asset graph distinguishes each line, winding, switch, grounding device, measurement, and owner. It supports questions such as *which circuit is unavailable?* and *which construction record produced this impedance?* It need not contain the virtual buses introduced by an OPF formulation.

Terminal-connectivity view

This view records ordered bus terminals and the maps by which element conductors attach to them. It can distinguish phase a from a neutral and can represent a conductor permutation between the ends of a line. It is the natural place to resolve switchgear and grounding connectivity.

Bus–branch multigraph

Buses are vertices and identified two-terminal elements are edges. Parallel circuits remain distinct. This view is effective when the device vocabulary is genuinely two-terminal or when multi-terminal devices have been compiled into an equivalent network with explicit provenance.

Port–factor view

Ports carry terminal variables and factors impose constitutive, limit, control, or measurement relations. A multiwinding transformer can remain one factor with one port bundle per winding. The number of ports is not forced to two.

Equation or optimization view

Variables and constraints form a bipartite graph, or blocks form a computational dependency graph. This view exposes separability, coupling, and decomposition opportunities. An auxiliary variable created for numerical convenience becomes a graph vertex even though it is not a physical object.

Matrix sparsity view

The nonzero pattern of an admittance, Jacobian, KKT, or Schur-complement matrix defines another graph. Elimination may reduce the number of variables while making this graph denser. A sparsity edge means algebraic coupling, not necessarily a physical branch.

Loads, generators, and the graph boundary

Loads and generators expose why an element inventory must be declared before the word *graph* is used. In an asset or bus–branch graph they are usually devices attached to a bus, not ordinary line edges. In a port–factor graph they are explicit one-terminal or multi-terminal factors. In a nodal-support graph they appear only when a declared formulation stamps a linear or linearized part of their relation into the nodal operator. In an equation or optimization graph they also appear through injections, controls, limits, and decision variables.

The same device can therefore be absent from one graph and present in another without contradiction:

View	Device role	Typical question
asset or bus–branch graph	attached equipment or a bus-side factor; not necessarily an edge	which device is switchable, owned, or removed?
port–factor graph	one-terminal or multi-terminal constitutive factor with a terminal map	what relation, limit, or control does the device impose?
nodal-support graph	a diagonal shunt, off-diagonal block, or no direct stamp, depending on the declared model	which retained voltage coordinates are coupled by this formulation?
solver/Jacobian graph	injection, residual, derivative, control, or constraint block	which variables and equations determine the next iterate or decision?

A constant-admittance load can therefore be stamped into a diagonal nodal block without becoming a physical network edge. A Norton generator or a multi-terminal device can contribute a different block pattern. The stamping choice is a property of the declared study formulation, not a definition of the source asset graph. The circuit formulation boundary and load-model chapter make this split explicit.

Question	Required retained meaning	A useful view
Are two assets independently switchable?	member identity and switch state	asset graph or multigraph
Which conductors share a junction?	ordered terminals and terminal maps	terminal-connectivity model
What is the boundary current response?	constitutive relation at retained ports	port–factor or admittance model
Which constraints determine the OPF optimum?	feasible set, controls, objective	optimization model
Which variables should be eliminated first?	numerical nonzero structure	sparsity graph
Can a result be mapped to the source data?	provenance and recovery	linked source and generated views

Different questions select different views

The views are not arranged in a universal hierarchy. The asset graph may know more about ownership and less about electrical variables than a compiled optimization graph. Expressiveness is relative to the declared question.

The first preservation test

Suppose each parallel line obeys

$$\mathbf{I}^{\mathrm{s}}_{\ell ij} = \mathbf{Y}_\ell\big(\mathbf{U}_i[\mathbf{N}_{\ell i}] - \mathbf{U}_j[\mathbf{N}_{\ell j}]\big)$$

and has a conductor-current feasible set $\mathcal{C}_\ell$. Aggregating admittance preserves the sum of terminal series currents, but the source feasible voltage differences satisfy

$$\{\Delta\mathbf{U} : \ \mathbf{Y}_\ell \Delta\mathbf{U} \in \mathcal{C}_\ell \quad \forall \ell\}\,.$$

A single conventional edge limit need not reproduce this intersection. The transformation can be exact for one observation and wrong for a decision problem. This is why the book asks for a Preservation contracts rather than calling a smaller graph simply *equivalent*. A first failure: heterogeneous parallel branches gives an analytic witness and executable certificate.

The route through the book

Before the formal taxonomy, One network, five languages translates the recurring terms used by power engineers, software and data experts, mathematical modellers, graph theorists, and graph-machine-learning readers. The Representation taxonomy separates the major model families. Notation and modelling conventions fixes the element, arc, terminal, and winding indices. Representation implementation record records the linked architecture's executable implementation status. The transformation parts then ask which views can be derived, under what guards, and with what consequences for feasible decisions. Five buses through a multiport lowering is the compact bridge from these view names to an explicit three-winding compilation and loss ledger.

Chapter 10

Five buses: identities and cycles

Page status: generated worked graph example with hash-bound figures, BMOPFTools cross-checks, and companion conductor-terminal and multi-port lowering witnesses.

Purpose and status

This worked example separates four operations that are often drawn as if they formed one reduction chain: forgetting parallel identity, constructing an electrical aggregate, eliminating internal variables, and selecting a spanning tree. They do not have the same source, target, or preservation contract.

The general definitions of simple cycles, line-identity cycles, parallel fibres, bridges, leaves, and adjacency-versus member-radiality are given in Cycles, parallelism, and radial structure. This chapter supplies the running numerical witness for those definitions.

The graph identities below are standard finite-dimensional results. The line-indexed calculations, nodal-admittance comparison, decision witness, and BMOPFTools topology cross-check are executable. The figures are generated from that same analysis record rather than maintained as independent sketches. The scalar values are chosen to make the distinctions visible; the final section lifts the construction to the book's multiconductor and multi-terminal scope.

Source bus–branch multigraph

Let the buses and identified lines be

$$\mathcal{B} = \{i, j, k, l, m\}, \qquad \mathcal{L} = \{q, r, s, t, u, v, w\}.$$

Using the book's element–endpoint convention, declare

$$\mathcal{T}^{L\to} = \{qji,\ rij,\ sjk,\ tki,\ vlj,\ wkl,\ ulm\}.$$

Thus q and r are distinct parallel lines even though their declared orientations are opposite. Reversing either arrow changes signs in oriented coordinates but does not change the underlying undirected multigraph.

With bus order (i, j, k, l, m) and line order (q, r, s, t, v, w, u), take -1 at the declared from bus and $+1$ at the declared to bus. The incidence matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The connected graph has $\operatorname{rank}(\mathbf{A}) = 4$. Its real cycle space therefore has dimension

$$\mu = \dim \ker \mathbf{A} = |\mathcal{L}| - \operatorname{rank}(\mathbf{A}) = 7 - 4 = 3.$$

More generally, a loopless undirected multigraph with c connected components has

$$\mu = |\mathcal{L}| - |\mathcal{B}| + c.$$

The displayed integer matrix below is only one coordinate realization. Its semantic rows are indexed by buses $i \in \mathcal{B}$ and its columns by identified lines $\ell \in \mathcal{L}$. If $\kappa_{\mathcal{B}}$ and $\kappa_{\mathcal{L}}$ enumerate those sets, then the stored array satisfies

$$[\mathbf{A}]_{\kappa_{\mathcal{B}}(i), \kappa_{\mathcal{L}}(\ell)} = A_{i\ell}.$$

This distinction matters even in this small example: the line labels q and r are not row or column numbers, and the cycle labels $\mathcal{C}_q, \mathcal{C}_t, \mathcal{C}_v$ are not required to be integers. The integer ordering is useful for computation; the labelled incidence is what retains the identity of parallel members.

This count uses identified lines. Deduplicating endpoint pairs before counting cycles gives a different graph and can give a different answer.

A line-indexed fundamental cycle basis

Choose the spanning tree

$$\mathcal{T}_{\text{tree}} = \{r, s, w, u\}.$$

Its chords are q, t, and v. Adding each chord to the tree gives

$$\mathcal{C}_q = \{q, r\}, \qquad \mathcal{C}_t = \{r, s, t\}, \qquad \mathcal{C}_v = \{s, v, w\}.$$

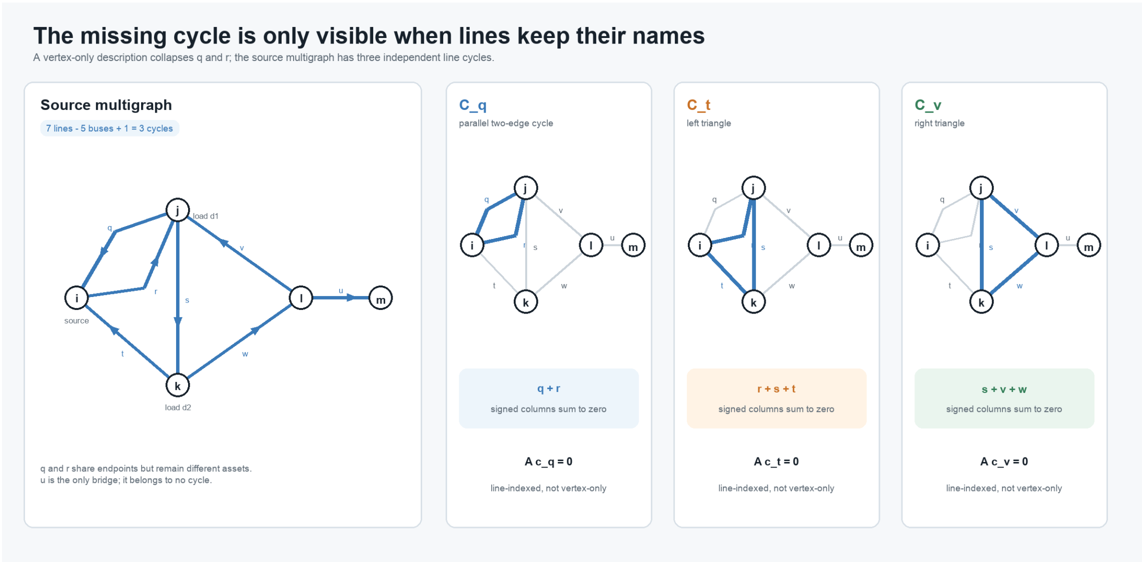

Figure 10.1: The source multigraph and the three line-indexed fundamental cycles.

The small multiples make the missing dimension visible. The cycle $\mathcal{C}_q$ has only two buses but two distinct line columns; it disappears as soon as q and r are represented by one adjacency.

The first cycle is the two-edge cycle made by the parallel pair. With the declared orientations, the signed edge–cycle matrix is

$$\mathbf{C} = \begin{array}{c|ccc} & \mathcal{C}_q & \mathcal{C}_t & \mathcal{C}_v \\ \hline q & 1 & 0 & 0 \\ r & 1 & 1 & 0 \\ s & 0 & 1 & 1 \\ t & 0 & 1 & 0 \\ v & 0 & 0 & 1 \\ w & 0 & 0 & 1 \\ u & 0 & 0 & 0 \end{array}, \qquad \mathbf{AC} = \mathbf{0}.$$

Equivalently, write the relation without choosing storage coordinates as $C_{\ell\gamma}$ for $\ell \in \mathcal{L}$ and $\gamma \in \Gamma$. The equation $A_{i\ell}C_{\ell\gamma}$ is then a finite labelled sum over the shared line index ℓ. This is the same label-before-coordinate principle used for nodal blocks and terminal maps.

Cycle bases are not unique. A *minimum* cycle basis additionally requires nonnegative edge weights and minimizes total basis weight [13]. A list of vertex tuples alone is insufficient for the source multigraph: replacing q by r produces a different line cycle, and the two-edge cycle cannot be represented without member identity.

Line u is the only **bridge**: removing it increases the number of connected components. Calling u a radial edge is less precise, and calling m a radial bus does not generalize well. A cycle-core bus can also be the attachment point of a bridge and a pendant tree. Bridges, the graph 2-core, and block–cut structure express those roles without assigning radiality to an individual vertex.

Projection is not electrical aggregation

Let π forget line identity and retain only a canonical unordered endpoint pair. Then

$$\pi(q) = \pi(r) = e_{ij}.$$

The resulting simple graph has six edges and cycle rank

$$6 - 5 + 1 = 2.$$

This is a topology projection. It loses the q–r two-cycle, member orientations, ratings, states, ownership, and provenance. Under the standard definition it also has neither parallel edges nor self-loops.

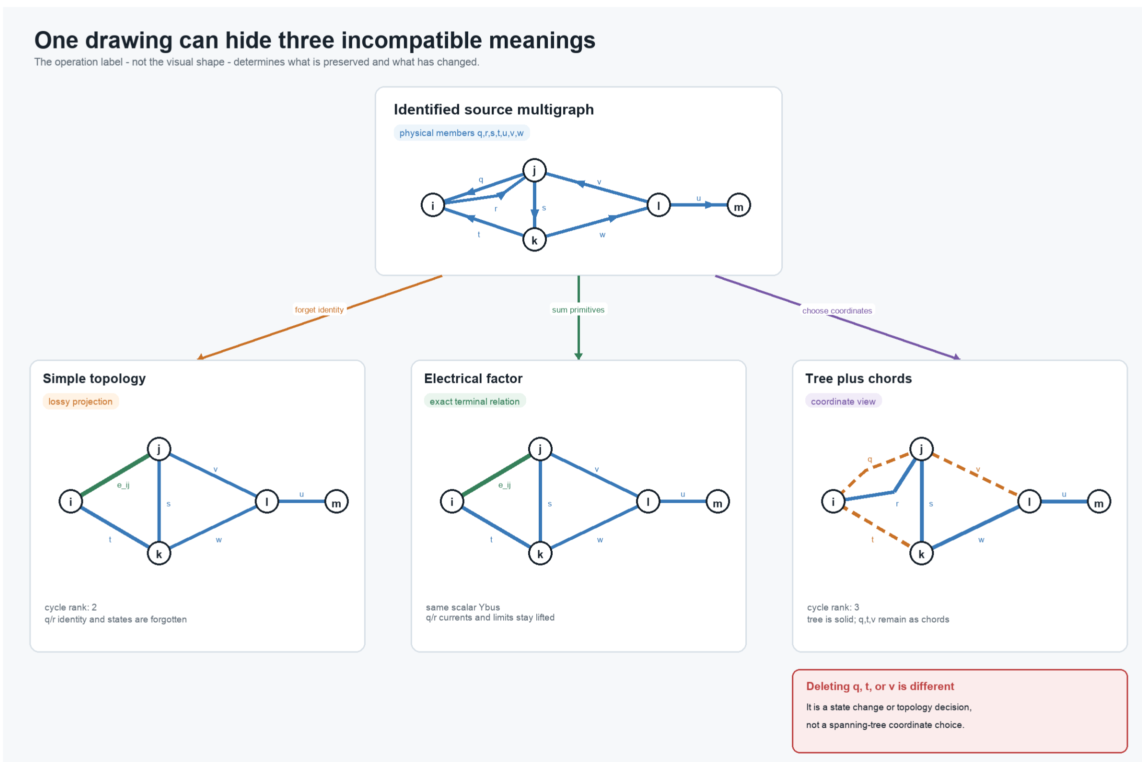

Figure 10.2: Three typed constructions that can otherwise look like one graph-simplification chain.

The first two lower panels intentionally have the same graph shape. One means only that member identity was forgotten. The other carries a summed electrical factor while the source-member recovery maps and limits remain available. The third panel has not removed anything: it merely draws the selected tree solid and the three source chords dashed.

An electrical aggregate is a separate construction. For reciprocal scalar series branches on common endpoint coordinates,

$$I_{ij}^{\text{total}} = (Y_q + Y_r)\,(U_i - U_j), \qquad Y_{ij}^{\text{eq}} = Y_q + Y_r.$$

If both impedances are invertible, this means

$$Z_{ij}^{\mathrm{eq}} = \left(Z_q^{-1} + Z_r^{-1}\right)^{-1},$$

not $Z_q + Z_r$. With the oriented incidence matrix and diagonal series admittance matrix,

$$\mathbf{Y}^{\mathrm{N}} = \mathbf{A}\,\mathrm{diag}(Y_\ell)\mathbf{A}^{\mathsf{T}}.$$

The executable example verifies that stamping the seven source members and stamping the six-edge weighted projection produce identical $\mathbf{Y}^{\mathrm{N}}$. In an explicitly expanded scalar matrix, the m, m entry is $+Y_x$; the off-diagonal l, m entries are $-Y_x$.

That equality is only a terminal-current statement. In the recorded witness, $Y_q = 10$ S, $Y_r = 1$ S, and each member has a 100 A limit. At $|U_i - U_j| = 15$ V,

$$|I_q| = 150 \text{ A}, \qquad |I_r| = 15 \text{ A}.$$

The source is infeasible, while the summed check

$$|I_q + I_r| = 165 \text{ A} \le 200 \text{ A}$$

passes.

The same source-versus-aggregate feasible-set geometry is shown in the complex-plane figure in A first failure: heterogeneous parallel branches. This five-bus chapter keeps the network topology and recovery discussion local rather than repeating the scalar card.

The weighted simple graph can still be used exactly if the member recovery maps and nonredundant source constraints remain lifted. This is the same decision-preservation mechanism developed in A first failure: heterogeneous parallel branches.

A spanning tree is a coordinate choice

The chosen tree gives a unique parent–child relation after selecting a root. It does so even though the source graph is meshed. What is nonunique is the choice of tree, not the possibility of constructing a hierarchy.

Let the vector of oriented branch-voltage drops be

$$\mathbf{u}_L = \mathbf{A}^{\mathsf{T}}\mathbf{U}.$$

Then every source cycle satisfies

$$\mathbf{C}^{\mathsf{T}}\mathbf{u}_L = \mathbf{C}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{U} = \mathbf{0}.$$

Tree paths can parameterize voltage differences, while chord equations impose the remaining cycle consistency. Similarly, the nullspace term in a branch-current description is indexed by cycle coordinates. Lines q, t, and v remain physical members with constitutive equations, limits, states, and provenance.

Deleting a chord is therefore not a graph-coordinate operation. It changes the network unless the line is already open, a decision selects it open, or a separate certificate proves the relevant relation and constraints redundant. For a radial-topology decision, the active identified lines must form a tree or forest under the declared connectivity contract. Counting only simple adjacencies would miss the two-edge cycle created when both q and r are active.

Why the apparent reductions are guarded

In the projected simple graph, bus i has degree two, but it is also the source/injection bus. The degree-two series rule must reject its elimination: topological degree alone does not establish a common series current. The exact guards are given in Degree-two series elimination.

Line u is a dangling bridge, but it cannot be removed merely because it is outside every cycle. If bus m carries a load, generator, measurement, bound, contingency, or retained voltage, pruning u changes the study. Pendant-tree removal is valid only under an observation and decision contract that either excludes that subtree or replaces it with a certified boundary model.

These distinctions classify the useful constructions as follows:

Construction	Classification	What must remain explicit
identified multigraph to unweighted simple graph	projection	forgotten member data if recovery is required
parallel primitive summation	exact terminal normalization under guards	member recovery and nonredundant constraints
guarded junction elimination	exact behavioral reduction	recovery, residual constraints, and provenance
spanning-tree selection	coordinate or algorithmic view	every chord and its source equations
chord opening	fixed state or topology decision	switch/line identity and connectivity contract

The same distinctions survive when scalar branches are expanded into conductor-terminal relations and multi-terminal factors.

Multiconductor and multi-terminal lift

For a multiconductor line, the scalar relation becomes

$$\mathbf{I}^{\mathrm{s}}_{\ell ij} = \mathbf{Y}_\ell \left(\mathbf{U}_i[\mathbf{N}_{\ell i}] - \mathbf{U}_j[\mathbf{N}_{\ell j}] \right).$$

Parallel primitives can be added only after orientations, ordered conductor coordinates, terminal maps, units, and base quantities are aligned. For a nominal-π or general fixed linear two-terminal member, the object to stamp is its complete terminal primitive, not only $\mathbf{Y}_\ell$. Component, grouped, and both-end limits remain member-indexed unless a separate implication certificate removes them.

The asset-level cycle space still records line identity, but it need not equal a conductor-resolved cycle space. Missing phases, explicit neutrals, grounding, and conductor permutations can give different connectivity by terminal. A cycle-based formulation must say whether its incidence describes buses, conductor terminals, galvanic zones, factors, or algebraic variables.

A genuinely multiwinding transformer has no canonical representation as one ordinary bus–branch edge. Its natural cycle participation belongs to the port–factor or terminal-incidence view. Any cycle basis of a compiled two-terminal realization is relative to that compilation and must retain a map back to the source transformer and windings.

The companion Five buses through a multi-port lowering keeps this chapter's line-induced graph unchanged and adds a symbolic three-port transformer extension. It compares factor incidence, a generated star, and the terminal clique produced by internal elimination, with a separate cycle rank for every declared graph.

Executable evidence

The package-independent implementation constructs $\mathbf{A}$, the fundamental basis, bridges, simple projection, and both nodal admittances. BMOPFTools independently analyzes the same seven identified line records.

The recorded checks give seven source lines on five buses, source incidence rank four, source cycle rank three, and simple-projection cycle rank two. They also give

$$\|\mathbf{A}\mathbf{C}\|_\infty = 0, \qquad \|\mathbf{Y}^{\mathrm{N}}_{\mathrm{source}} - \mathbf{Y}^{\mathrm{N}}_{\mathrm{projected}}\|_\infty = 0.$$

BMOPFTools reports three extra physical edges, and the independently computed bridge set is $\{u\}$.

Chapter 11

Equipment, terminals, and nodal assembly

Page status: literature-backed definitions with executable structural and recovery witnesses; inverse recovery is reported by explicit identifiability status and remains non-canonical without additional structure.

The missing middle between a one-line diagram and $\mathbf{Y}^{\mathrm{N}}$

Power engineers routinely move between a bus–branch diagram, a multiconductor circuit, and a nodal admittance matrix. Those three objects are related, but they do not have the same vertices, edges, or cycles. Calling all three *the network graph* hides two distinct topology levels and a many-to-one algebraic projection:

1. the **equipment/terminal topology** records identified equipment and its high-level attachments;
2. the **conductor/port–factor topology** records the electrical terminals, junctions, conductor coordinates, and constitutive factors;
3. the **nodal-operator support graph** records which retained voltage coordinates are coupled by the assembled matrix.

The first two are retained source structure in this book. The third is a derived computational view. The asset/dependency model remains an orthogonal companion to both: ownership, protection, common-mode failure, and maintenance are not entries of a nodal admittance matrix.

The same boundary applies to formulation choice. A compound $\mathbf{Y}^{\mathrm{N}}$ is an exact target for a declared class of fixed linear factors, but it is not a universal representation of a power network. General power-network studies may need modified nodal or sparse-tableau variables, branch currents, multi-terminal factor relations, switch states, controls, and limits that are not recoverable from nodal support alone. See Circuit formulations and the lowering boundary for the guarded compilation alternatives.

The figure uses parallel lines because they expose the loss immediately. Both identified factors attach to the same electrical junction coordinates, and their matrix contributions occupy the same nodal block. The assembly preserves their combined linear boundary relation but forgets how that contribution was split between assets.

The middle panel is a port–factor incidence view, not a bare bipartite graph: the small open circles are ports, the junction bars are the codomain of j, and the factor boxes are the codomain of f. The labelled j and f families are intentionally distinct. A later compilation may replace this structure by ordinary edges, but that is a new target view with a provenance obligation rather than an identity of the source object.

Parallel lines make decomposition loss visible, while a multiwinding transformer makes graph-class change visible. The companion five-bus multi-port lowering shows one three-port factor as an acyclic incidence/star object and, after internal elimination, a cyclic terminal clique. That new support cycle is fill or terminal coupling, not an additional physical transformer loop.

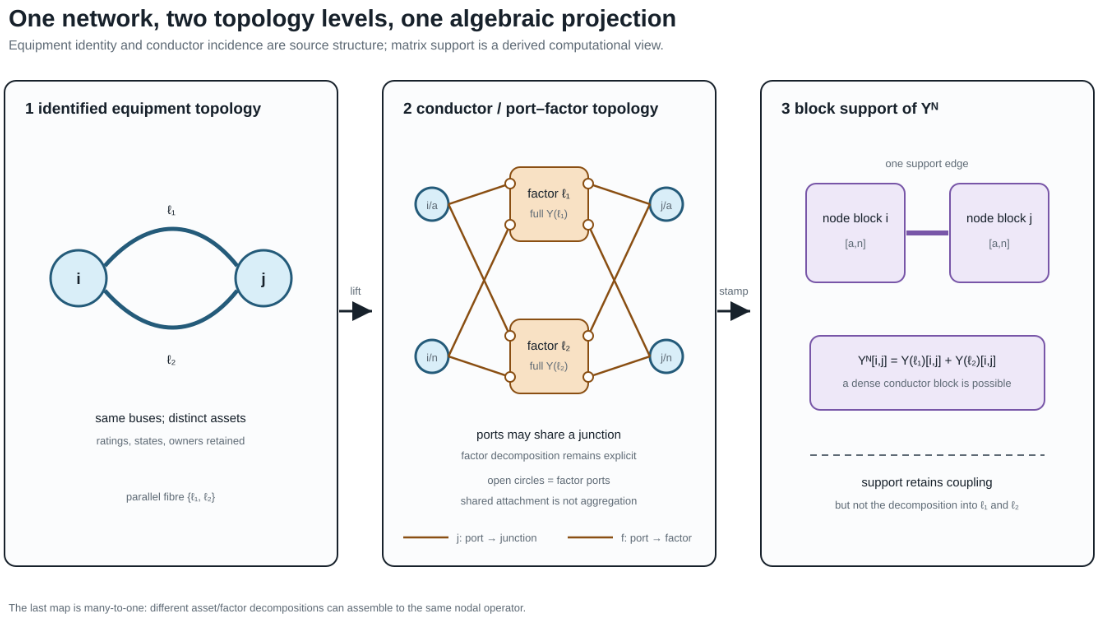

Figure 11.1: Two source topology levels and their many-to-one nodal-admittance projection.

Level 1: identified equipment and high-level attachments

For the two-terminal subset of a network, let

$$G_{\mathrm{M}} = (\mathcal{B}, \mathcal{L}, \partial), \qquad \partial(\ell) = \{i, j\}.$$

This is an identified multigraph: ℓ_1 and ℓ_2 remain distinct when $\partial(\ell_1) = \partial(\ell_2)$. A stored reference orientation is written ℓij and does not make the physical line a directed edge. The element-intrinsic impedance or primitive matrix keeps the symmetric element index $\mathbf{Z}_\ell$ or $\mathbf{Y}_\ell$; terminal observations use ℓij and ℓji.

The multigraph is only a high-level skeleton. A transformer x with winding set $\mathcal{K}_x$ is naturally multi-terminal, and a jointly coupled line group may own more than two port bundles. Such objects belong in an port–factor incidence structure, not in G_{M} unless an explicit two-terminal compilation has been selected. The orthogonal asset/dependency model links equipment records to this electrical structure through Λ; it is not this multigraph. Consequently, *radial at equipment level* must name both the selected object class and any multi-terminal compilation used to obtain an ordinary graph.

At this level, parallelism means repeated high-level attachment. It says nothing yet about terminal order, phase availability, grounding, mutual coupling, state, or whether currents may be added in a common coordinate space.

Level 2: conductor junctions and electrical factors

The canonical electrical model is the hierarchical port–factor object $\mathfrak{P}$ defined in Formal representation frameworks. For the present discussion, its important incidence maps are

$$j : \mathcal{Q} \to \mathcal{J}, \qquad f : \mathcal{Q} \to \Phi,$$

where j attaches a typed port to an electrical junction and f assigns the port to its owning factor. A junction may represent a scalar conductor coordinate such as i/a or a typed bundle such as $i/[a, b, c, n]$. The factor relation retains the full conductor coupling, terminal maps, internal variables, limits, state, and decisions.

Two distinct ports may attach to the same junction; indeed, the map j is deliberately many-to-one because that is how KCL composes devices. A different relation is needed for construction detail. Let

$$c : \mathcal{C}_{\text{phys}} \to \mathcal{Q}$$

map physical conductors or sub-conductors to their electrical ports. This map need not be injective: bundled conductors, paralleled cables landing on one terminal, and other construction-level multiplicities can share a port. The distinction is therefore precise: j composes electrical ports at a junction; c realizes a port from physical conductor objects.

Implementation note. A particular importer or line-constant tool may use a narrower source schema. The current BMOPFTools line-geometry compiler, for example, checks one geometry conductor per terminal label. That is a boundary of that adapter's present primitive, not a definition of c. A richer source must remain expanded or pass through a declared bundle-compilation rule before entering the adapter.

This level has its own notions of parallelism:

- factors can be parallel because all of their boundary port spaces and attachment maps coincide;
- physical conductors can be parallel inside one factor while sharing an electrical terminal at one or both ends;
- two factors can share high-level buses but fail to be terminal-parallel because their phase sets, neutral connections, or terminal maps differ;
- mutual coupling can place two nominal assets in one joint factor, so they are not independent edges even when the asset inventory lists them separately.

The high- and low-level decompositions are therefore related by an explicit lineage/refinement relation, not by an assumed one-edge-to-one-wire rule.

A coupling relation over line-section identities

Jointly coupled circuits expose why the identified multigraph and the factor model are both needed. Let $\mathcal{S}$ be a set of oriented electrical line sections, each with lineage to a stable line asset, and let

$$H_{\text{cpl}} = (\mathcal{S}, \mathcal{C})$$

be a typed relation graph whose edge $c \in \mathcal{C}$ identifies two spatially overlapping sections and carries their mutual blocks, coordinate maps, orientation, and overlap interval. The vertices of H_{cpl} are therefore section identities that appear as edges or refinements in the equipment view. This is a relation *over line sections*, not another bus graph.

Each connected component Γ of H_{cpl} compiles into one joint electrical factor ϕ_Γ. Its primitive is assembled before inversion or nodal stamping. This order preserves indirect coupling created by the inverse of a block matrix and avoids pretending that each line owns an independent constitutive factor. Ratings, states, and decisions remain attached to the source line or section quantities through the factor-to-asset lineage.

The coupled multi-voltage corridor case gives the smallest exact witness. Its two source line assets can belong to different voltage systems, while their joint four-port factor lowers to a generated terminal lattice with cross-voltage edges. Those edges belong to nodal support, not to galvanic connectivity or the asset inventory.

From factors to a compound nodal operator

Stack the retained junction-voltage coordinates into $\mathbf{U}$. Let Φ_{lin} be the declared subset of fixed linear unconstrained electrical factors. A load or generator does not enter this set merely because it attaches to the same network, but a declared constant-admittance part, Norton equivalent, fixed shunt, or iteration-specific linearization may enter it. The nodal operator is therefore a study- and state-specific part of the model from which some decision semantics may already have been omitted.

This distinction is important for one-terminal devices. Their attachment map belongs to the port–factor or terminal-connectivity model even when their linear contribution is stamped into $\mathbf{Y}^{\text{N}}$. Conversely, a constant-power or controlled device may have no fixed admittance stamp while still contributing an injection function, control equation, limit, or optimization decision. Inclusion in Φ_{lin} is thus a formulation choice with a declared mode, state, and preservation contract; it does not decide whether the device belongs to the source graph.

For each $\phi \in \Phi_{\text{lin}}$, let $\mathbf{A}_\phi$ be a real select/permute/sign matrix for its ordered terminal voltages, so

$$\mathbf{u}_\phi = \mathbf{A}_\phi \mathbf{U}, \qquad \mathbf{i}_\phi = \mathbf{Y}_\phi \mathbf{u}_\phi.$$

After mapping terminal currents back to the junction coordinates, linear assembly has the form

$$\mathbf{Y}^{\text{N}} = \sum_{\phi \in \Phi_{\text{lin}}} \mathbf{A}_\phi^{\mathsf{T}} \mathbf{Y}_\phi \mathbf{A}_\phi.$$

The factor primitive $\mathbf{Y}_\phi$ may already contain series, shunt, ideal-connection, transformer, or multiport structure. Incidence-matrix assembly of compound polyphase networks is developed by Kettner and Paolone [8]; nested primitive, winding, and connection maps for general multiphase transformers provide a particularly clear device-level example [14].

The transpose in this assembly is intentional. Because $\mathbf{A}_\phi$ is real, the power-dual current map is $\mathbf{A}_\phi^{\text{H}} = \mathbf{A}_\phi^{\mathsf{T}}$. If a voltage map contains complex ratios or phase actions outside $\mathbf{Y}_\phi$, its current map uses the conjugate transpose instead. This is the same distinction used for transformer maps elsewhere in the book.

The equation is an **assembly identity**, not a unique factorization of $\mathbf{Y}^{\text{N}}$. For two electrically aligned parallel factors $\ell_1 ij$ and $\ell_2 ij$, the same off-diagonal nodal block contains

$$\mathbf{Y}_{ij}^{\text{N}} = \mathbf{Y}_{\ell_1,ij} + \mathbf{Y}_{\ell_2,ij} + \sum_{\phi \notin \{\ell_1,\ell_2\}} \mathbf{Y}_{\phi,ij}.$$

Even when the last sum is empty, $\mathbf{Y}_{ij}^{\text{N}}$ does not identify its two summands. Let $\mathcal{D}$ be the linear difference space admitted by the declared aligned factor class. Every admissible Δ gives the same assembled block through

$$(\mathbf{Y}_1, \mathbf{Y}_2) \longmapsto (\mathbf{Y}_1 + \Delta, \mathbf{Y}_2 - \Delta).$$

Block structure and coordinate expansions

The phrase **vector-valued multigraph** is useful as a translation bridge for the two-terminal case: each high-level edge retains its identity and carries a vector of terminal quantities, while its constitutive data are matrices. It is not the canonical name of the general source model. In particular, an n-winding transformer is a typed port–factor relation, not an ordinary edge with a longer vector label.

Suppose each retained bus has c ordered conductor coordinates. The compound nodal operator is then naturally a block matrix,

$$\mathbf{Y}^{\mathrm{N}} = \begin{bmatrix} \mathbf{Y}_{11} & \cdots & \mathbf{Y}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{Y}_{N1} & \cdots & \mathbf{Y}_{NN} \end{bmatrix}, \qquad \mathbf{Y}_{ij} \in \mathbb{C}^{c\times c}.$$

The block labels i, j are bus identities, not necessarily integer positions. For a finite bus set $\mathcal{B}$, an enumeration $\kappa_{\mathcal{B}}$ turns the labelled family $(\mathbf{Y}^{\mathrm{N}}_{ij})_{i,j\in\mathcal{B}}$ into the stored block array

$$\left[\mathbf{Y}^{\mathrm{N}}\right]_{\kappa_{\mathcal{B}}(i),\kappa_{\mathcal{B}}(j)} = \mathbf{Y}^{\mathrm{N}}_{ij}.$$

This is the matrix analogue of the labelled edge–cycle construction in the opening example. The enumeration can change without changing the block operator, its support, or its factor provenance.

For one two-terminal series factor with a full conductor admittance $\mathbf{Y}^{\mathrm{s}}_{\ell}$, the off-diagonal blocks are dense in general, while endpoint shunts add to the diagonal blocks. A nominal-π factor is therefore one matrix-valued factor whose local stamp may be written schematically as

$$\begin{bmatrix} \mathbf{Y}^{\mathrm{sh}}_{\ell i} + \mathbf{Y}^{\mathrm{s}}_{\ell} & -\mathbf{Y}^{\mathrm{s}}_{\ell} \\ -\mathbf{Y}^{\mathrm{s}}_{\ell} & \mathbf{Y}^{\mathrm{sh}}_{\ell j} + \mathbf{Y}^{\mathrm{s}}_{\ell} \end{bmatrix}.$$

This compact block view and the expanded scalar view answer different questions:

View	Vertices or indices	What a nonzero means
factor multigraph	identified assets/factors	a declared factor attachment
block support	buses or junctions	one or more conductor coordinates are coupled
scalar support	bus–conductor coordinates	a particular coordinate pair is coupled
numerical matrix	ordered rows and columns	a coefficient is numerically nonzero at this operating/model point

Two parallel matrix-valued factors can occupy the same block support and be summed by assembly. A dense $c \times c$ factor can produce a clique in the scalar support graph even when the bus-level equipment graph is a tree. The clique is an algebraic coupling pattern, not a claim that every pair of conductors is a physical line. Eliminating an internal bus or factor can add new nonzero blocks by Schur-complement fill; it does not create new source assets.

The complex phasor representation can also be realified for software that expects real arrays. For $\mathbf{Y} = \mathbf{G} + \mathrm{j}\mathbf{B}$, one common coordinate embedding is

$$\mathcal{R}(\mathbf{Y}) = \begin{bmatrix} \mathbf{G} & -\mathbf{B} \\ \mathbf{B} & \mathbf{G} \end{bmatrix}.$$

Realification doubles coordinates, not buses, conductors, or assets. The block support relation is normally the same under this embedding, while the scalar real coordinate graph has twice as many coordinate vertices and may show a different fine-grained pattern. Say **complex-labelled graph** or **realified coordinate graph**, not "the complex graph" or "the real graph," unless the coordinate convention is explicitly part of the sentence.

One important invariant does not survive unchanged. Even when $\mathbf{Y}$ is complex-transpose-symmetric, $\mathcal{R}(\mathbf{Y})$ is generally not an ordinary symmetric real matrix unless $\mathbf{B} = 0$. Realification preserves the coordinate relation and its support, but it changes which matrix symmetry statement is appropriate; do not silently transfer reciprocity claims from the complex representation to the realified one.

The four-view bridge in How to read power-network diagrams and equations is the reader-facing summary of this distinction. Its scoped correspondence is registered as `ARCH-BLOCK-001`: the witness checks the block assembly, support projection, and realification of one declared two-bus four-wire factor, while deliberately treating factor identity as separate provenance rather than inferring it from nonzero matrix entries.

These distinctions also explain why a current-injection solver and a backward–forward sweep can use different computational views of one declared four-wire model. The former may factor a block nodal operator; the latter may traverse element-wise impedance relations on a radial view. Neither algorithmic graph replaces the source factor and terminal semantics.

For reciprocal factors, $\mathcal{D}$ may require $\Delta^{\mathsf{T}} = \Delta$. Adding passivity constraints such as $\mathrm{He}(\mathbf{Y}_k) \succeq 0$ intersects the affine family with a convex feasible set, but does not generally make it bounded: reactive or other unconstrained parameter directions can remain unbounded. Bounded ambiguity needs catalog limits, sign restrictions, measurements, or another coercive guard. Ratings, outage states, investment variables, owners, and the two-edge line-identity cycle are absent unless retained separately.

This is the central many-to-one result registered as `ARCH-NODAL-001`. The companion executable witness constructs two distinct passive reciprocal parallel splits with an identical assembled operator.

When is assembly injective?

Let Θ be the admissible family of typed factor collections and define

$$\mathcal{S}(\{\mathbf{Y}_\phi\}) = \sum_\phi \mathbf{A}_\phi^{\mathsf{T}} \mathbf{Y}_\phi \mathbf{A}_\phi.$$

The exact criterion is

$$\mathcal{S}|_\Theta \text{ is injective} \quad \Longleftrightarrow \quad \ker \mathcal{S} \cap (\Theta - \Theta) = \{0\}.$$

This criterion matters more than a blanket claim that nodal data are or are not recoverable. A practical sufficient special case is available when:

1. topology, factor types, terminal coordinates, and active state are known;
2. at most one two-terminal factor occupies each unordered junction-block pair;
3. that factor's off-diagonal block uniquely determines its complete local stamp within the declared class; and
4. after subtracting those stamps, the remaining diagonal residual has a unique decomposition among the declared shunt and grounding classes.

Then the source primitives are recoverable from $\mathbf{Y}^{\mathrm{N}}$ within that model class. Familiar line-impedance extraction is an instance of this support-separated case, not a contradiction of `ARCH-NODAL-001`.

The principal failures are now named: **multiplicity**, when parallel or overlapping stamps share support; **elimination**, when internal coordinates have already been removed; and **over-parameterization**, when different factor parameters produce the same terminal stamp. Recoverability from a boundary response can nevertheless hold for restricted classes; circular planar critical resistor networks are a major positive theory [15]. The model class and injectivity proof are therefore part of any recovery claim.

A scoped recovery vocabulary

The inverse question should return a status, not just a matrix. For an observed operator $\mathbf{Y}^{\mathrm{N}}$ and declared model class Θ, classify the preimage of the restricted assembly map as follows:

The three statuses are:

- **identifiable** — one admissible source primitive and declared residual decomposition are recovered. A typical case has known support, one two-terminal factor per block pair, and a unique local stamp map.
- **set-identifiable** — the terminal primitive is fixed, but internal parameters or construction records remain an equivalence class. This is the expected result for an over-parameterized local construction with the same terminal stamp.
- **non-identifiable** — distinct source primitives or topologies remain observationally indistinguishable. Parallel multiplicity and eliminated internal coordinates are the basic examples.

This vocabulary is deliberately relative to Θ. Tightening the class with catalog bounds, switch state, measurements, construction metadata, or grounding declarations can turn a non-identifiable class into an identifiable one; silently assuming those facts does not. Conversely, a numerically unique optimizer solution is not evidence that the source map is injective.

The recovery contract is therefore:

1. validate the coordinate order, factor class, active state, and admissible parameter domain before attempting inversion;
2. return a unique recovered primitive only for the identifiable class;
3. return a representative together with an explicit equivalence or affine ambiguity for set-identifiable and non-identifiable classes; and
4. retain the source/provenance record as the authority for asset identity.

The generated `experiments/generated/nodal-source-recovery-witness.json` exercises all four statuses. Its support-separated scalar case recovers a series primitive and declared diagonal shunts. Its parallel case exhibits two distinct member splits with the same operator; its elimination case matches a hidden three-node chain to a direct two-node boundary factor; and its over-parameterized case fixes the terminal primitive while leaving two local parameter vectors indistinguishable. These are deliberately finite witnesses, not a claim that every practical line or transformer belongs to one class.

This scoped classification is registered as `ARCH-RECOVERY-001`. It turns the warning "do not invert $\mathbf{Y}^{\mathrm{N}}$ canonically" into a usable engineering interface: declare the model class, report the recovery status, and preserve the ambiguity whenever the data do not identify the source.

Which guards actually lift the ambiguity?

Extra information should be treated as an additional observation map, not as an informal reason to choose one reconstruction. If $\mathcal{M}$ records member currents, grounding metadata, state observations, or another declared measurement, define the augmented map

$$\mathcal{F}(\theta) = \big(\mathcal{S}(\theta), \mathcal{M}(\theta)\big).$$

The same restricted-kernel test applies:

$$\mathcal{F}|_{\Theta} \text{ is injective} \quad \Longleftrightarrow \quad \ker \mathcal{F} \cap (\Theta - \Theta) = \{0\}.$$

Four small cases make the distinction operational:

- **Catalog bounds** can make a parallel-split ambiguity compact without making it unique. A bounded interval is still a set of admissible source models.
- **Member-current observations** can lift parallel ambiguity when the voltage drop is nonzero and each member current is observed in the same coordinates. The total nodal operator and the member observations then jointly determine the scalar member admittances in that restricted class.
- **Grounding declarations** can resolve a diagonal residual only when the declaration identifies the grounding contribution and its terminal. A finite grounding impedance and an unspecified local shunt otherwise remain an attribution ambiguity.
- **Transformer or switch states** must be part of the declared state space. A known tap lets a state-conditioned primitive be recovered; an unknown tap can trade off against the primitive parameter and produce multiple state–parameter pairs with the same effective terminal relation.

The generated `experiments/generated/nodal-recovery-guards-witness.json` records these four outcomes. Its statuses are deliberately more informative than a binary "recoverable" flag: `bounded-non-identifiable`, `identifiable`, `identifiable-with-declaration`, and `identifiable-with-state`. The witness is scalar and finite; it establishes the guard logic, not a general theorem for all multiconductor transformers, nonlinear grounding, or state-dependent AC models. This extension is registered as `ARCH-RECOVERY-002`.

Matrix-valued observations need excitation rank

The scalar member-current example does not generalise automatically to a multiconductor factor. For a factor ℓ with an $m \times m$ primitive, voltage snapshots $V \in \mathbb{C}^{m\times r}$, and member-current snapshots I_ℓ, the observation equation is

$$I_\ell = Y_\ell V.$$

Full primitive recovery from this relation requires $\operatorname{rank}(V) = m$ and observation of every row of I_ℓ. In the square full-rank case, $Y_\ell = I_\ell V^{-1}$; more generally the declared experiment must span the retained conductor space. If $r < m$, any nonzero admissible Δ with $\Delta V = 0$ gives the same member currents. If only selected current rows are observed, a nonzero Δ can instead satisfy $P\Delta V = 0$ for the row-selection map P. Reciprocal symmetry does not remove these directions by itself.

The generated `experiments/generated/multiconductor-recovery-witness.json` makes all three cases explicit: two independent voltage snapshots with complete current vectors recover two reciprocal two-conductor factors; one complete snapshot leaves a nonzero reciprocal ambiguity; and full-rank snapshots with only one measured current phase still leave the unmeasured phase ambiguous. This is a linear identifiability witness, not a claim about nonlinear measurement design, noise, or estimator consistency. It is registered as `ARCH-RECOVERY-003`.

Noise changes recovery into a certified uncertainty set

With noisy current snapshots

$$I_\ell = Y_\ell V + E, \qquad \|E\|_{\mathrm{F}} \le \varepsilon,$$

the full-rank pseudoinverse estimate $\widehat{Y}_\ell = I_\ell V^\dagger$ is an estimator, not an exact source recovery. For square invertible V (and, with the corresponding qualification, for a full-row-rank snapshot matrix),

$$\|\widehat{Y}_\ell - Y_\ell\|_{\mathrm{F}} \le \varepsilon\, \|V^\dagger\|_2.$$

Thus experiment conditioning is part of the recovery certificate. The same noise radius can produce a tight uncertainty set for well-conditioned voltage snapshots and a much larger set for nearly dependent snapshots. Physical symmetry or passivity constraints may intersect that set, but they should be reported as additional guards rather than used to hide the measurement uncertainty.

The generated `experiments/generated/noisy-multiconductor-recovery-witness.json` compares well-conditioned and nearly collinear two-conductor voltage snapshots under the same Frobenius noise radius. Both estimates satisfy the deterministic bound; the ill-conditioned experiment amplifies the bound by more than two orders of magnitude. This is a finite error certificate, not a statistical consistency or estimator-optimality result. It is registered as `ARCH-RECOVERY-004`.

Rank, reference, and grounding

The nodal operator is not automatically invertible. Under the connectivity, common polyphase-coordinate, and passive-component hypotheses of Kettner and Paolone, a connected shunt-free compound network has the common-mode nullspace and rank

$$\operatorname{rank}(\mathbf{Y}^{\mathrm{N}}) = (|\mathcal{B}| - 1)m.$$

Under their corresponding grounded/shunted hypotheses, an effective nonzero shunt removes that gauge freedom and the compound nodal operator is full rank [8]. For several galvanic components, absent references can contribute separate nullspaces. Ideal devices, singular terminal maps, missing phases, and more general factors require their own rank analysis.

This is why *choose a voltage reference* and *model physical grounding* must remain distinct instructions. Deleting a coordinate to fix a numerical gauge does not add a grounding asset; conversely, a finite grounding impedance is a physical factor. The running-network numerical export illustrates the conditional case: its passive 20×20 operator has reported numerical rank 18 at the declared tolerance, not because every nodal operator must be singular but because reference and grounding structure remain in that model.

Is nodal admittance a simple-graph concept?

Not in the physical sense. A nodal admittance matrix is a linear operator on a chosen ordered voltage space. From it one can derive simple support graphs at several granularities. The graph-loop, shunt, incidence, adjacency, and Laplacian conventions used to distinguish those objects are fixed in Multigraphs for expert modelers.

Definition (block support). Given bus blocks $\mathbf{Y}^{\mathrm{N}}_{ij}$, define

$$G^{\mathrm{blk}}_Y = (\mathcal{B}, E^{\mathrm{blk}}_Y), \qquad \{i,j\} \in E^{\mathrm{blk}}_Y \iff \mathbf{Y}^{\mathrm{N}}_{ij} \neq \mathbf{0}.$$

Definition (scalar support). Given retained coordinates $\mathcal{C} = \{(i,p)\}$, define

$$G^{\mathrm{sc}}_Y = (\mathcal{C}, E^{\mathrm{sc}}_Y), \qquad \{(i,p),(j,q)\} \in E^{\mathrm{sc}}_Y \iff Y^{\mathrm{N}}_{(i,p),(j,q)} \neq 0.$$

These support graphs are simple by construction: a matrix position is either zero or nonzero. But that does not make the source network a simple graph. Several factor stamps sum into one position, and their decomposition is not encoded in matrix support. If an algorithm needs the decomposition, it can use a **stamp multigraph** whose identified members are the separate $\mathbf{A}_\phi^{\mathsf{T}}\mathbf{Y}_\phi\mathbf{A}_\phi$ contributions. That multigraph is extra data; it cannot generally be recovered from the sum. This separation is registered as `ARCH-SUPPORT-001`.

The support relation also requires qualifications:

- a dense off-diagonal block can be produced by mutual conductor coupling inside one physical line;
- several contributions can occupy one block, including parallel factors and multi-terminal compilations;
- exact cancellation can make a matrix entry zero even though source factors touch both coordinates;
- a diagonal block combines incident series terms, local shunts, grounding, and possibly several compiled factors;
- changing coordinates can change scalar support without changing the underlying external relation.

Thus G^{blk}_Y is often an excellent sparsity and decomposition view, while G^{sc}_Y exposes within-block coupling. Neither is an asset register.

Why a radial network can acquire cycles

Gan and Low show that a multiphase radial network can be represented as an equivalent scalar network that is radial at the macro level but has a clique associated with each line [16]. Their companion multiphase BIM/BFM work similarly treats each bus–phase pair as a coordinate of an equivalent scalar circuit [17]. The observation is valuable here because it makes the level distinction impossible to ignore.

For an m-conductor two-terminal factor with a dense $2m \times 2m$ terminal stamp, its scalar support can contain a clique on the $2m$ endpoint coordinates. A triangle or larger cycle inside that clique is an algebraic coupling cycle. It is not evidence that operators can open an alternative physical route, that power is circulating, or that the bus-level feeder is meshed. If some primitive entries are structurally zero, the clique loses the corresponding support edges; the matrix pattern, not the word *multiconductor*, decides the scalar support.

There is also a constructive counterpart. Let the simple bus-level graph be a tree, give every bus the same m retained coordinates, and suppose each tree edge has one structurally dense two-terminal stamp whose support is the clique on its two endpoint blocks. Assume the assembled nonzeros do not cancel.

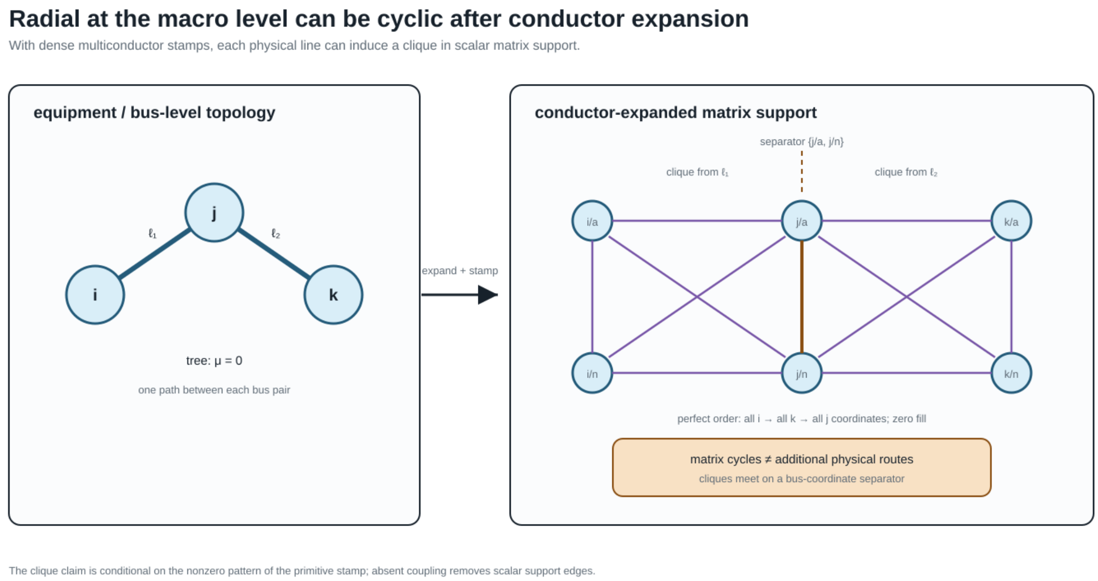

Figure 11.2: A bus-level tree and the cyclic scalar support induced by dense multiconductor line stamps.

Proposition (tree of dense stamps). The resulting scalar support graph is chordal. Eliminating all coordinates of a leaf-bus block and proceeding inward is a perfect elimination ordering and creates zero structural fill.

Proof. At a leaf bus i with parent j, every later neighbour of a coordinate (i, p) lies in $(\{i\} \times \mathcal{P}) \cup (\{j\} \times \mathcal{P})$. The dense edge stamp makes that set a clique, so (i, p) is simplicial. Eliminating the entire leaf block removes one leaf from the bus tree and leaves the same construction on a smaller tree. Induction gives a perfect elimination ordering. A perfect elimination ordering adds no fill.

The line-stamp cliques meet on bus-coordinate separators. Their clique tree is inherited from the bus tree, but is not literally isomorphic to it in general: a tree with n buses has $n - 1$ line cliques before any maximal-clique coalescence. The two-line figure, for example, has two cliques joined through the separator $\{j/a, j/n\}$.

This result, registered as `ARCH-CHORDAL-001`, explains why the cycles are benign for the declared sparse computation and why Gan and Low can exploit chordal structure in multiphase radial OPF. It is conditional: multi-terminal factors, missing coupling entries, cancellation, or a meshed bus graph can change the support and its elimination properties.

This produces several useful apparent paradoxes:

These are not contradictions. Each sentence changes the graph without saying so.

Three cycle questions, not one

The cycles and radiality chapter defines the corresponding graph objects in detail. The practical crosswalk is:

A cycle basis computed in one row is not automatically a basis for another. In particular, a parallel pair gives a two-member cycle in the identified multigraph but one edge in block support, while a dense line stamp can give many scalar-support cycles without any asset-level cycle.

Statement	Resolution
a radial feeder has cycles	equipment topology can be a tree while conductor-expanded matrix support contains cliques
two parallel lines become one edge	their stamps add in one block-support edge; asset identity has been projected away
one line becomes many edges	one dense multiconductor factor produces many scalar nonzeros
a transformer creates a triangle	a clique compilation of one multi-terminal factor creates a support cycle, not three transformer assets
a new edge appears after reduction	Kron fill-in is an equivalent boundary coefficient, not a discovered line
a physical relation has no matrix edge	cancellation, coordinate choice, or eliminated variables can hide the relation

Cycle question	Graph or incidence object	What it can support
Is there an alternative route through identified two-terminal members?	equipment/bus multigraph	switching, outages, member radiality, line-identity cycle bases
Is there repeated incidence through conductor junctions and factors?	conductor/port–factor graph or a declared compilation	terminal connectivity, factor decomposition, conductor-resolved equations
Does the assembled or reduced operator have cyclic sparsity?	block or scalar matrix-support graph	chordal decomposition, ordering, fill, sparse numerical algorithms

Executable projection and elimination witness

The generated `experiments/generated/topology-projection-witness.json` checks the two central mechanisms without relying on a power-flow solver.

- Two distinct reciprocal passive two-conductor splits assemble to a byte-identical 4×4 nodal operator. Both have zero normalized Frobenius assembly error, so the consistency certificate cannot identify the source attribution. The small negative minimum passivity eigenvalues (approximately -1.4×10^{-16} at worst) are recorded as floating- point roundoff against a 10^{-12} tolerance.
- A three-bus two-conductor bus-level tree has macro cycle rank zero and scalar structural-support cycle rank six. The declared leaf-block perfect order produces zero fill, while eliminating the separator block first produces four fill edges.

The source is `experiments/transformations/TopologyProjectionWitness.jl`; the focused test is `experiments/test/topology_projection_witness.jl`. These finite witnesses test `ARCH-NODAL-001`, `ARCH-SUPPORT-001`, and `ARCH-CHORDAL-001`; they do not claim that every factor library is passive, identifiable, or chordal.

Kron reduction adds a fourth source of apparent adjacency

Partition retained boundary coordinates B and eliminated internal coordinates I. When $\mathbf{Y}_{II}$ is invertible, Kron reduction gives

$$\widehat{\mathbf{Y}}_{BB} = \mathbf{Y}_{BB} - \mathbf{Y}_{BI}\mathbf{Y}_{II}^{-1}\mathbf{Y}_{IB}.$$

The Schur-complement term can introduce a nonzero retained block between two boundary nodes that shared no source factor. This fill edge belongs to the reduced operator support. It does not belong retrospectively to the source asset or conductor topology. Dörfler and Bullo characterize this graph effect for electrical-network Kron reduction [18], while the compound polyphase setting requires the relevant block-rank conditions [8].

Any eliminated current, voltage, or limit that still matters to a decision problem must be evaluated through a recovery map. This includes neutral- conductor current limits: the disappearance of the neutral coordinate from the retained operator does not remove the conductor's thermal constraint.

Maintain the decomposition; do not promise inversion

The canonical record should retain at least:

- stable asset and factor identities;
- ordered ports, junction attachments, and conductor/terminal maps;
- factor class and full primitive relation or a reproducible construction record;
- active-state, rating, grounding, control, and decision ownership;
- each assembly, compilation, coordinate, and reduction map;
- provenance from every matrix block or generated object back to its source factors;
- recovery maps for eliminated quantities that remain observable or constrained.

For a supplied nodal operator and claimed source decomposition, define each assembled stamp $\mathbf{S}_\phi = \mathbf{A}_\phi^\mathsf{T}\mathbf{Y}_\phi\mathbf{A}_\phi$ and the Frobenius-norm assembly backward error

$$\eta_{\mathrm{asm}} = \frac{\left\|\mathbf{Y}^{\mathrm{N}} - \sum_\phi \mathbf{S}_\phi\right\|_{\mathrm{F}}}{\|\mathbf{Y}^{\mathrm{N}}\|_{\mathrm{F}} + \sum_\phi \|\mathbf{S}_\phi\|_{\mathrm{F}}} \le \tau_{\mathrm{asm}}.$$

The denominator makes the test dimensionless and remains informative when large stamps nearly cancel; the all-zero case is handled separately. The certificate must also record the coordinate order, units, states, factor types, norm, and threshold.

This test verifies **assembly consistency**, not **source attribution**. It is invariant under every admissible regrouping in $\ker \mathcal{S}$ and is therefore structurally blind to the parallel-split ambiguity established above. Two decompositions can both achieve zero backward error while assigning different primitives, ratings, or identities to the members. Attribution requires provenance or independent identifying information.

Recovery from $\mathbf{Y}^{\mathrm{N}}$ alone is an inverse problem. It is non-identifiable whenever the restricted assembly criterion above fails. Additional catalog constraints, construction priors, measurements, switch states, or asset records can narrow the candidate set, but an estimator must report the remaining affine or constrained ambiguity rather than inventing line identity. The safe engineering objective is therefore:

> preserve the two-level source structure through compilation, and validate every derived nodal operator against it; attempt recovery only as a separately scoped inference problem.

That direction supports both rigorous proofs and practical data standardisation. Power engineers can work with familiar bus blocks and line triples, while the retained maps make clear which topology, constraints, and physical meanings survive each transformation.

Part 5: Transformations and recovery

Chapter 12

Preservation contracts

Page status: foundational definitions; certificate requirements are normative proposals.

Evidence boundary

Scope: observation-indexed feasible-set equivalence and preservation contracts for declared model classes. **Evidence:** definitions, linear recovery derivations, and scoped certificate examples. **Numerical optimality:** certificate examples are not global optimality proofs. **Unresolved boundary:** general nonlinear, mixed-integer, uncertainty-aware, or externally reviewed preservation theorems.

Observation precedes equivalence

Let $u \in \mathcal{U}$ be a fixed admissible input and let $\mathcal{F}_M(u)$ and $\mathcal{F}_{\widehat{M}}(u)$ be the source and target feasible sets at that input. Their internal coordinates may differ. Choose **joint** observation maps into the same observation space,

$$O_M(x,z,u) = (h_1(x,z,u),\dots,h_k(x,z,u)), \qquad O_{\widehat{M}}(\widehat{x},u) = (\widehat{h}_1(\widehat{x},u),\dots,\widehat{h}_k(\widehat{x},u)).$$

Definition (`PRESERVE-001`). The transformation is exact for these observations and this input domain when

$$\{O_M(x,z,u) : (x,z) \in \mathcal{F}_M(u)\} = \left\{O_{\widehat{M}}(\widehat{x},u) : \widehat{x} \in \mathcal{F}_{\widehat{M}}(u)\right\}, \qquad u \in \mathcal{U}.$$

The joint map retains relationships between observed quantities. Equality of individual observation ranges is weaker. For example, the sets $\{(0,0),(1,1)\}$ and $\{(0,1),(1,0)\}$ have identical coordinate ranges, yet maximizing the sum of their coordinates gives 2 and 1. A test of each coordinate separately would miss that difference. A declared family of maps may replace the single joint map only when it includes the joint observations needed for the claim.

For a decision problem, include the decisions whose preservation is claimed in the joint observation, and require the same objective function of those observations (or include objective value as another joint coordinate). Equal joint feasible images then give equal attainable observed decisions and objective values. Equality of optimal values does not alone imply the same decisions; existence of an optimizer and source-state recovery remain separate obligations. If u is itself optimized, rather than supplied as input, its admissible domain and source/target correspondence must also be preserved. All statements concern the declared feasible models, not a solver's ability to find their solutions.

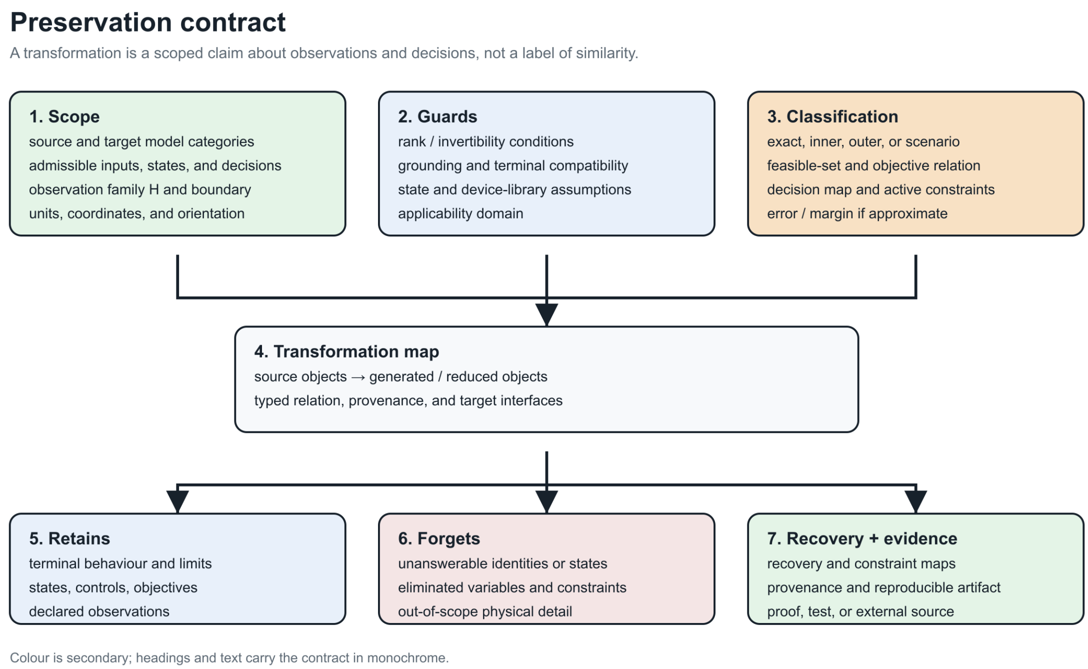

Figure 12.1: Preservation contract card: scope, guards, classification, retained and forgotten meaning, recovery, and evidence.

Decision-model consequence

Matching terminal equations or separate voltage and current ranges does not establish equality of joint constrained feasible observations. Declare the input domain, jointly observed decisions and quantities, objective correspondence, and required recovery map.

The card is generated by `experiments/render_preservation_contract_card.py`. It is a compact reading aid for the certificate fields below; it is not a replacement for the quantified definition or the machine-readable schema.

The book now separates four layers that are often collapsed under the word “preservation”: typed structure, constitutive behaviour, decision semantics, and provenance/ownership. See Transformation semantics and register for the definitions, closure rules and anti-pattern register. A certificate should be explicit when it preserves only one of these layers.

Preservation dimensions

A transformation certificate should identify which dimensions it preserves:

$$\begin{aligned}\Sigma = \{&\text{connectivity}, \text{ terminal behavior}, \ \text{phase/neutral behavior},\\ &\text{asset identity}, \text{ limits}, \ \text{switching states}, \text{ measurements},\\ &\text{protection}, \text{ spatial provenance}, \ \text{dynamics}\}.\end{aligned}$$

This is not a binary checklist. Each item needs a precise scope. For example, "terminal behavior" might mean:

- linear phasor behavior at one frequency;
- nonlinear steady-state behavior;
- an impedance function over a frequency interval;
- time-domain behavior for a specified initialization class;
- a scenario-bounded approximation to voltage magnitudes only.

Proposed certificate schema

The field table in this section is a **non-normative v1.2 proposal**. The machine-checked repository schema remains version 1.1.0 and deliberately does not yet require an $error_bound$ field or a normalized numerical-evidence object. Implementations must validate against the versioned JSON schema; the proposal below is a compatibility target for a future schema revision, not an implicit requirement on current certificates.

Every transformation record should contain:

Field	Meaning
`source_type`, `target_type`	model categories and schema versions
`rule_id`	stable identifier for the rewrite or compiler rule
`scope`	source objects affected
`preconditions`	structural, physical and state assumptions
`preserves`	formal or testable preservation claims
`forgets`	questions no longer answerable
`provenance`	source-to-target and target-to-source object maps
`recovery_map`	reconstruction of eliminated variables where possible
`constraint_map`	exact, conservative or approximate lifting of limits
`error_bound` **(v1.2 proposal)**	norm, domain and bound for approximate transformations
`evidence`	theorem, derivation, test suite, or external reference

Compact BMOPFTools decision-manifest gate

Cross-repository object PSK-`000007` connects the observation-indexed definition above to BMOPFTools contract `decision_preservation_manifest_completeness`. The package does not attempt to re-prove the general equivalence definition. It checks a narrower declaration boundary for version `0.1.0` manifests that explicitly claim exact `decision_equivalence`.

Such a manifest must name the transformation, source model, and target model, then give admissible domain, terminal behavior, observations, constraints, decision variables, objective, and recovery an explicit disposition. A `verified` disposition needs an evidence reference; `not_required` needs a justification. An omitted dimension creates an evidence gap, while `not_preserved` or `unassessed` contradicts the exact decision-equivalence claim. A terminal-only or conservative claim lies outside the gate and is not treated as a failure.

The minimized package fixture claims exact decision equivalence after citing only a terminal-relation certificate. The gate rejects that overclaim. Its evidence-complete companion passes only at the manifest layer: the package has not authenticated the references, checked the constraint or recovery maps, compared feasible sets or objectives, or established optimizer or solver equivalence. Those remain case-specific scientific and executable obligations.

The same boundary applies to the compact Kron guardrail. Cross-repository object PSK-`000008` connects the tutorial warning that Kron reduction is an assumption—not a free simplification—to BMOPFTools contract `kron_boundary_recovery_preservation`. Its four-wire-to-three-wire check requires perfect grounding of the

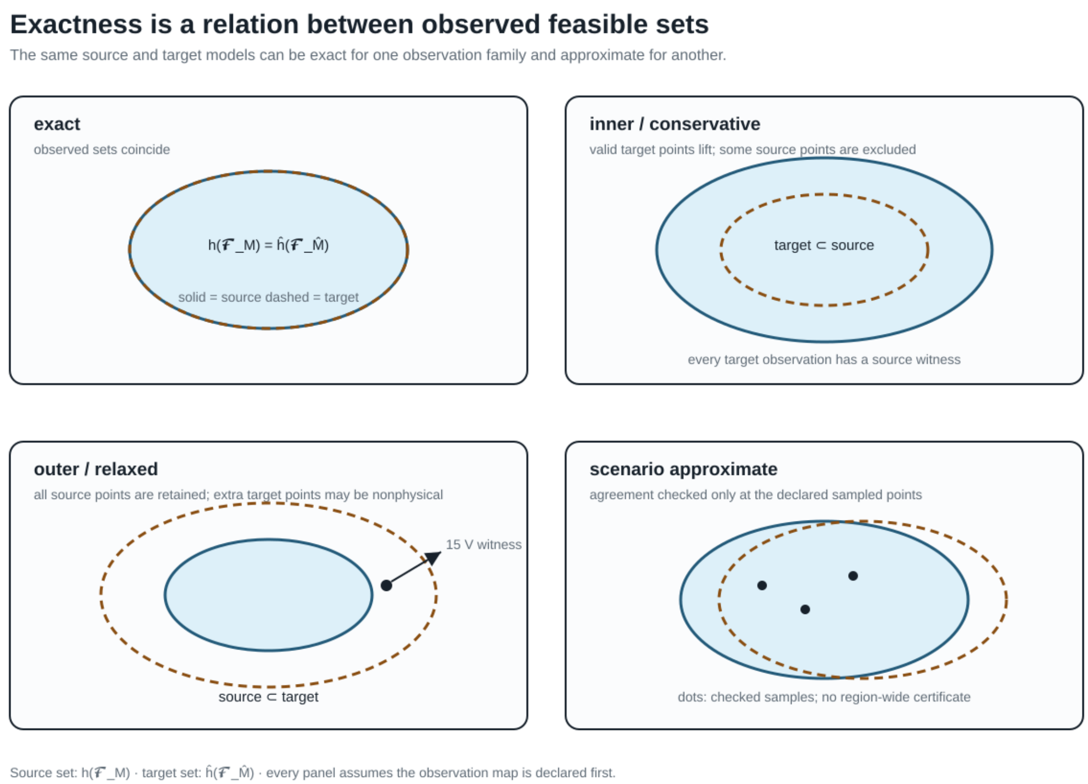

Figure 12.2: Exact, inner, outer, and scenario-approximate observed-set relations.

eliminated neutral at both endpoints, ordered phase coordinates, the neutral Schur complement, and an explicit recovery declaration. A floating or finite-grounded neutral fails the precondition even when the target matrix matches numerically; a pass says nothing about internal limits, protection, full feasible sets, objectives, or solver results.

Exact, conservative, and approximate maps

Operational constraints deserve a classification independent of the equations:

- **Exact:** reduced and original feasible observable sets coincide.
- **Inner/conservative:** every reduced feasible point lifts to an original feasible point, possibly excluding valid original points.
- **Outer/relaxed:** every original feasible observation is retained, but the reduced model may admit nonphysical points.
- **Scenario approximate:** accuracy is measured over a specified sample or uncertainty distribution.

A claim that a reduction "preserves limits" is incomplete without identifying which of these meanings applies.

This is the visual version of the classification. The sets are observed sets, not raw internal state spaces: the observation map h is part of the contract. The $15V$ point is the scalar parallel-line witness from the first failure case; it lies in the outer target's extra region and therefore cannot be called a preserved member limit.

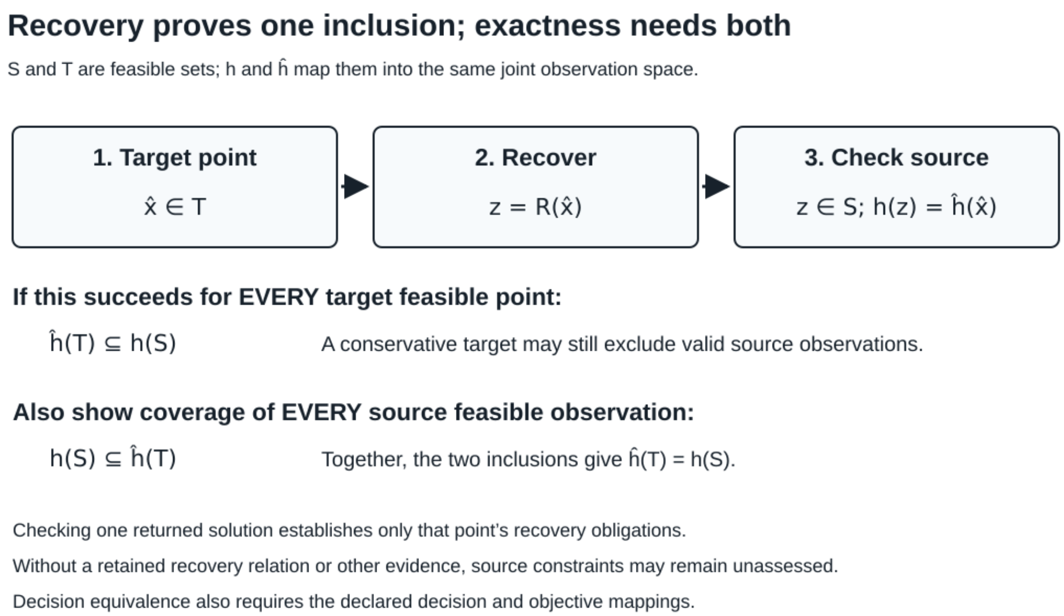

Figure 12.3: Recovery establishes target-to-source observation inclusion; exact observed-set equality also requires source coverage.

Recovery maps are first-class

If internal variables are eliminated by a Schur complement, their recovery is often available algebraically. For a partitioned linear nodal model,

$$\begin{bmatrix} i_B \\ i_I \end{bmatrix} = \begin{bmatrix} Y_{BB} & Y_{BI} \\ Y_{IB} & Y_{II} \end{bmatrix} \begin{bmatrix} v_B \\ v_I \end{bmatrix},$$

and $i_I = 0$, the internal voltage is

$$v_I = -Y_{II}^{-1} Y_{IB} v_B.$$

A recovery map that sends **every** target feasible point to a source feasible point with the same joint observation establishes $\widehat{h}(\widehat{\mathcal{F}}) \subseteq h(\mathcal{F})$. Equality also requires the reverse inclusion: every source feasible observation must be represented in the target. A conservative target can satisfy the recovery obligation while excluding valid source observations. Checking one returned solution establishes only that point's obligations, not either set-wide statement. Decision and objective preservation require their declared mappings.

The reduced admittance is

$$Y_{\text{red}} = Y_{BB} - Y_{BI} Y_{II}^{-1} Y_{IB}.$$

Kron reduction preserves the selected boundary relation, and is well characterized for loopy Laplacians [18]. It does not by itself preserve equipment identity, sparsity, protection zones, or individual branch limits. Storing the recovery operator permits evaluation of some original quantities without pretending the reduced branches are physical assets.

Compositionality requirement

A strong preservation framework should be compositional: replacing a subsystem by a certified equivalent should remain valid when that subsystem is connected to an admissible environment through its declared ports. The black-box functor for passive linear networks formalizes this principle for terminal current–potential relations [19].

For power systems, the open problem is to extend this idea to typed conductors, nonlinear devices, discrete controls, limits, uncertainty, and decision variables without losing useful computational structure.

Chapter 13

Degree-two series elimination

Page status: guarded exact series transformation with positive and negative tests; physical line-class closure remains open.

This chapter gives the first executable guarded rewrite. It deliberately separates an exact terminal-behaviour result from the stronger claim that two source assets form one longer homogeneous physical line.

Source and target

Let two multiconductor series elements be oriented as $\ell_1 ib$ and $\ell_2 bj$. The ordered conductor coordinates used by ℓ_1 at the internal junction b need not equal the order used by ℓ_2. Let $\mathbf{P}$ be the permutation satisfying

$$\mathbf{I}_{\ell_2 bj} = \mathbf{P}\mathbf{I}_{\ell_1 ib}, \qquad \mathbf{U}_b^{(2)} = \mathbf{P}\mathbf{U}_b^{(1)}.$$

The outer terminal order at j is relabelled into the coordinates of ℓ_1. The source relations are

$$\mathbf{U}_i - \mathbf{U}_b^{(1)} = \mathbf{Z}_{\ell_1}\mathbf{I}_{\ell_1 ib},$$

$$\mathbf{U}_b^{(2)} - \mathbf{U}_j^{(2)} = \mathbf{Z}_{\ell_2}\mathbf{I}_{\ell_2 bj}.$$

If Kirchhoff conservation at b gives the common series current, multiplying the second relation by $\mathbf{P}^\mathsf{T}$ and adding gives

$$\mathbf{U}_i - \mathbf{P}^\mathsf{T}\mathbf{U}_j^{(2)} = \left(\mathbf{Z}_{\ell_1} + \mathbf{P}^\mathsf{T}\mathbf{Z}_{\ell_2}\mathbf{P}\right)\mathbf{I}_{\ell_1 ib}.$$

Hence the exact behavioural composite has

$$\mathbf{Z}_{\ell_{\mathrm{eq}}} = \mathbf{Z}_{\ell_1} + \mathbf{P}^\mathsf{T}\mathbf{Z}_{\ell_2}\mathbf{P}.$$

This is claim `TR-SER-001`. It is a coordinate-aware terminal equivalence, not a license to add matrices whose rows merely happen to have the same position. The displayed formula also assumes that the two voltage-drop relations contain no mutual-impedance terms linking either source element to the other or to an external element. The coupled case is treated separately below.

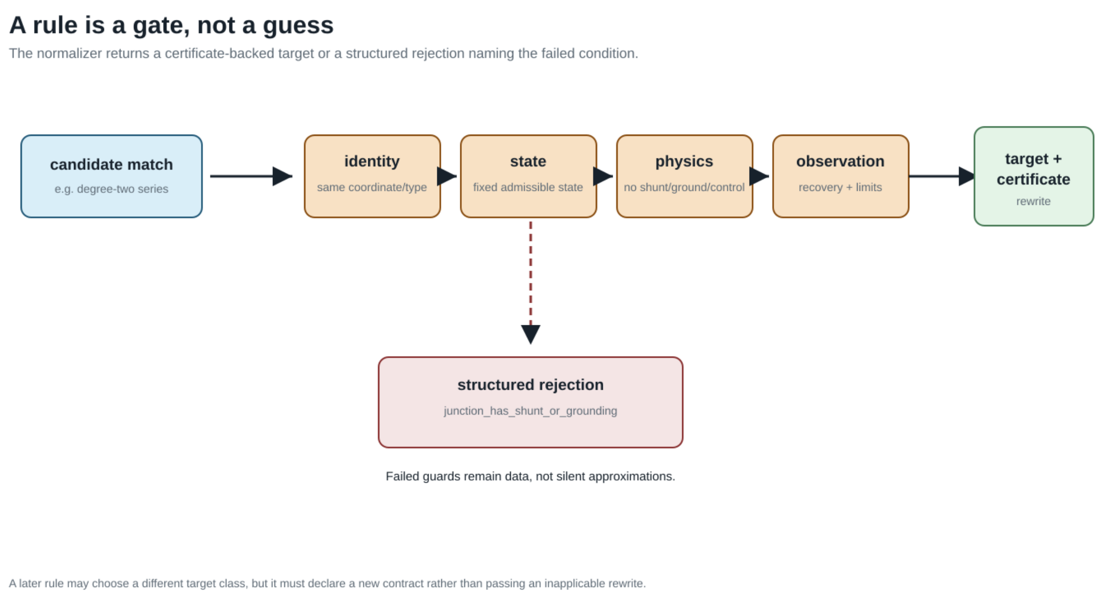

Figure 13.1: The guarded series rewrite: inspect, certify, or reject before lowering.

Guards

The gate is the operational pattern used by the executable rule: a candidate rewrite is accepted only after its structural, constitutive, decision, and provenance preconditions have been checked.

The source type supplies a precondition before the junction guards are even considered: **both factors are series-only multiconductor elements**. A nominal-π or other shunted factor is outside this rule. A shunt carried inside such a factor is not made admissible merely because no separately named shunt object is attached to b.

The implemented rule accepts the rewrite only when all of the following hold:

Guard	Reason
ℓ_1 ends and ℓ_2 begins at b	fixes the declared orientation
conductor labels at b are unique and form the same set	makes $\mathbf{P}$ well defined
no current injection at b	establishes a common series current
no shunt or grounding at b	prevents current from leaving the series path
no measurement, control, or protection boundary at b	keeps elimination within the declared observation contract
neither element is mutually coupled to the other source element	makes the displayed uncoupled impedance sum applicable
neither element is mutually coupled to any external element	prevents loss of a constitutive relation outside the pair

A failed guard returns a structured rejection with the failed condition and the source evidence. In the executable model, mutual coupling is represented by the field $SeriesElement.mutual_couplings[other_element_id]$: an element-pair cross-impedance block keyed by the other asset identity. It is deliberately not a `JunctionContext` flag, because the same pairwise constitutive relation may span a junction or extend beyond it. The guard therefore inspects the source element fields that would invalidate the formula and does not return a best-effort equivalent.

Constraint and recovery maps

For per-conductor current-feasible sets $\mathcal{C}_{\ell_1}$ and $\mathcal{C}_{\ell_2}$, the exact target constraint is

$$\mathcal{C}_{\text{eq}} = \mathcal{C}_{\ell_1} \cap \mathbf{P}^{\mathsf{T}} \mathcal{C}_{\ell_2}.$$

Independent upper current magnitudes therefore become the coordinate-aligned componentwise minimum, not their sum. Source quantities recover as

$$\mathbf{I}_{\ell_1 ib} = \mathbf{I}_{\text{eq}}, \qquad \mathbf{I}_{\ell_2 bj} = \mathbf{P}\mathbf{I}_{\text{eq}},$$

$$\mathbf{U}_b^{(1)} = \mathbf{U}_i - \mathbf{Z}_{\ell_1} \mathbf{I}_{\text{eq}}.$$

The generated target retains both member identities and the eliminated-junction identity in its provenance record.

Why this is not automatically a physical merge

Different construction codes do not invalidate the algebra above. They do invalidate the stronger rewrite into one instance of a homogeneous physical line class. Even equal codes are only a candidate condition: conductor material and geometry, frequency basis, line model, rating semantics, splices, maintenance boundaries, ownership, thermal state, outage state, and other physical facts may still differ.

Graph-theory trap

Degree two is only a structural candidate for elimination. A valid rule must also inspect terminal coordinates, injections, shunts and grounding, observations, constraints, decisions, and the intended target equipment class.

This distinction is claim `TR-SER-002`: closure under behavioural elimination and closure within an equipment class are different questions.

Anti-patterns: algebra is not a type checker

Three tempting rewrites should be shown as refusals or as explicitly typed compositions:

The safe target for the first row is usually a `CompositeSeriesBranch`; for the second and third rows it is a typed multiport retaining the transformer and ground ports. If the target library has no such factor, the transformation is ill-typed even when a matrix calculation can be performed.

This is also why a nominal-π series merge needs more than the displayed Z matrices. Shunt currents make the two segment currents different at the intermediate bus, and cascading the sections generally produces a general two-port rather than a nominal-π section with naively summed parameters.

Rewrite	What can be true	Why the physical merge is unsafe
different line constructions ℓ_1 and ℓ_2 → one line	the pure-series terminal impedance can still be $Z_1 + P^{\mathsf{T}} Z_2 P$	the target may falsely claim one construction, one owner, one thermal state or one rating basis
line + transformer → line	a fixed cascade can have a generic two-port relation	turns ratio, vector group, galvanic boundary, winding limits and controls disappear
external ground + transformer → transformer-only	a fixed nodal admittance can sometimes absorb the branch	neutral-current ownership, earth return, protection and topology dependence disappear

Power-system shorthand

Rejecting every heterogeneous series pair would be too strong. The error is silently asserting membership in a narrower physical line class, or dropping a junction constraint, shunt, grounding branch, control, rating or provenance boundary without recording it.

Mutual-coupling counterexample

Suppose the two series sections are themselves mutually coupled. In coordinates where their drop relations contain cross blocks $\mathbf{Z}_{12}$ and $\mathbf{Z}_{21}$, substituting $\mathbf{I}_2 = \mathbf{P}\mathbf{I}_1$ gives

$$\mathbf{Z}_{\mathrm{eq,coupled}} = \mathbf{Z}_1 + \mathbf{Z}_{12}\mathbf{P} + \mathbf{P}^{\mathsf{T}}\mathbf{Z}_{21} + \mathbf{P}^{\mathsf{T}}\mathbf{Z}_2\mathbf{P}.$$

The two cross terms do not disappear merely because b has zero injection. Coupling between adjacent sections of the same corridor is also not naturally described as *external coupling at the junction*: it is a constitutive relation between the two element identities. The implementation therefore stores mutual-coupling blocks against the other element identity and rejects the local uncoupled rule whenever either an internal-pair or external-pair block exists. A more general coupled-factor elimination could be exact, but it would be a different rule with the full coupled block as its source.

The executable negative witness uses two reciprocal two-conductor sections. It compares the displayed uncoupled sum with the four-term expression above and records an approximately 11.65% relative Frobenius-norm error. This is a counterexample to the insufficient guard, not a claim that mutual coupling always produces an error of that size.

The witness also records the representation boundary explicitly: a junction-only data model cannot express this case, while the pair-keyed `mutual_couplings` field can. A coupled-factor elimination would need the full pair block and a different target interface; silently dropping that field is a semantic loss.

A separate exact rule for a coupled section pair

The cross terms are not merely a reason to weaken the guard. They define a different, narrower rewrite. `TR-SER-003` applies when the two candidate elements are series-only, the junction guards above hold, both pairwise blocks $\mathbf{Z}_{12}$ and $\mathbf{Z}_{21}$ are declared, and neither element has any mutual-coupling block to a third element. No reciprocity assumption is needed by the algebra; if a physical model requires $\mathbf{Z}_{21} = \mathbf{Z}_{12}^{\mathsf{T}}$, that is an additional model-specific guard.

With $\mathbf{P}$ aligning the conductor order at b, the target is the terminal-behaviour composite

$$\mathbf{Z}_{\mathrm{eq,coupled}} = \mathbf{Z}_1 + \mathbf{Z}_{12}\mathbf{P} + \mathbf{P}^{\mathsf{T}}\mathbf{Z}_{21} + \mathbf{P}^{\mathsf{T}}\mathbf{Z}_2\mathbf{P}.$$

The recovery map remains $\mathbf{I}_1 = \mathbf{I}_{\mathrm{eq}}$ and $\mathbf{I}_2 = \mathbf{P}\mathbf{I}_{\mathrm{eq}}$; member current limits therefore map by the same intersection as in the uncoupled rule. The certificate retains the pair identities and declares that physical homogeneous-line closure is *not* asserted. In particular, this rule does not turn a corridor coupling model into two independent line objects, and it rejects any external coupling that would be left dangling after the rewrite. It is exact for the declared external terminal voltage/current relation, not a license to erase the internal coupling semantics.

The executable certificate records this rule alongside the negative witness in `experiments/generated/degree-two-series-cert` The implementation and tests are self-checked; an independent mathematical review of the new rule remains an explicit review item.

Grounding counterexample

If a grounding or shunt admittance $\mathbf{Y}_g$ is attached at b, then

$$\mathbf{I}_{\ell_1 ib} - \mathbf{P}^{\mathsf{T}}\mathbf{I}_{\ell_2 bj} = \mathbf{Y}_g\mathbf{U}_b^{(1)}.$$

The currents are no longer a single common series variable. A Schur complement may still eliminate $\mathbf{U}_b$, but the result is a more general terminal relation and must not be reported as the series rule above. The prototype therefore rejects this application with `junction_has_shunt_or_grounding`.

Executable certificate

The package-independent prototype is implemented in `experiments/transformations/SeriesElimination.jl`. It returns either a target plus preservation certificate or a structured rejection. The executable case uses a two-conductor permutation and heterogeneous construction codes:

```
julia --project=experiments experiments/run_series_elimination.jl
julia --project=experiments experiments/test/runtests.jl
```

Its machine-readable result is `experiments/generated/degree-two-series-certificate.json`. The certificate classifies the result as an exact behavioural reduction, records the permutation and constraint intersection, and explicitly refuses to call the target a homogeneous physical line. It also records the cross-coupled negative witness, its four-term exact expression, the failed element-pair guard, and the relative error made by the uncoupled expression. It conforms to the common interface in Certificate schema and composition, where the separately certified coordinate normalization is composed with this rule.

Chapter 14

Conductor-coordinate normalization

Page status: guarded exact coordinate transformation with an executable certificate; independent mathematical review remains open.

Conductor order is representation, not physics. It is nevertheless part of the meaning of every vector, matrix, and componentwise constraint. A rewrite that changes an order without changing all dependent objects is therefore a model error, not a harmless formatting operation.

Vocabulary bridge

Normalization in this chapter is an invertible rewrite into a declared conductor order. It is not per-unit conversion, schema normalization, or ML feature scaling. Those operations require their own maps, units, inverses where applicable, and preservation claims.

Rule

Consider an element ℓij with source conductor order $\gamma_\ell = (n, a)$ and a requested order $\widehat{\gamma}_\ell = (a, n)$. Let $\mathbf{P}_\ell$ be the unique permutation for which

$$\widehat{\mathbf{x}}_{\ell ij} = \mathbf{P}_\ell \mathbf{x}_{\ell ij}.$$

Intrinsic element data retain only the element index. In particular, the coordinate-normalized impedance is

$$\widehat{\mathbf{Z}}_\ell = \mathbf{P}_\ell \mathbf{Z}_\ell \mathbf{P}_\ell^\mathsf{T},$$

and a vector of componentwise limits becomes

$$\widehat{\mathbf{I}}_\ell^{\max} = \mathbf{P}_\ell \mathbf{I}_\ell^{\max}.$$

The terminal map at j is reordered by preserving each original from–to conductor pairing. Thus a coordinate position can move, but conductor identity cannot silently change.

Exactness and inverse

A permutation matrix satisfies $\mathbf{P}_\ell^{-1} = \mathbf{P}_\ell^\mathsf{T}$. Hence every normalized state and every transformed intrinsic matrix has a unique recovery:

$$\mathbf{x}_{\ell ij} = \mathbf{P}_\ell^\mathsf{T}\widehat{\mathbf{x}}_{\ell ij}, \qquad \mathbf{Z}_\ell = \mathbf{P}_\ell^\mathsf{T}\widehat{\mathbf{Z}}_\ell\mathbf{P}_\ell.$$

The rule is therefore an exact normalization (`TR-COORD-001`). It forgets no declared source semantics. This statement depends on permuting every indexed quantity consistently; permuting only $\mathbf{Z}_\ell$ or only a terminal list is not the same rule.

Guards and rejection

The executable rule requires:

- unique conductor labels in both orders;
- equal source and requested conductor sets;
- equal arity; and
- preservation of the original paired terminal map.

A missing, duplicated, or new conductor label produces a structured rejection. The implementation does not guess whether, for example, `g` and `n` are interchangeable.

Executable example

For the order reversal used by the running series example,

$$\mathbf{P}_\ell = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The rule moves limits $(80, 110)$ A in order (n, a) to $(110, 80)$ A in order (a, n) and permutes both axes of $\mathbf{Z}_\ell$. Run:

```
julia experiments/test/coordinate_normalization.jl
julia --project=experiments experiments/run_coordinate_series_composition.jl
```

The machine-readable result is `experiments/generated/coordinate-normalization-certificate.json`. The next transformer chapter applies the shared coordinate action to a winding's full terminal-to-coil relation. The series chapter then composes line-coordinate normalization with degree-two elimination.

Chapter 15

Kron, Ward, and optimized equivalents

Page status: scoped reduction definitions, audited literature synthesis, and a package-independent Kron/Ward/scenario comparison; independent mathematical review and source-faithful Opti-KRON implementation remain open.

Evidence boundary

Scope: linear and affine boundary reductions, operating-point Ward-style targets, and small scenario-approximation witnesses. **Evidence:** Schur-complement derivations, recovery checks, and package-independent numerical fixtures. **Numerical optimality:** scenario probes and local solves do not establish global nonlinear AC optimality. **Unresolved boundary:** universal physical realizability, uncertainty-robust preservation, and external review.

Three different questions

Kron reduction, Ward equivalents, and Opti-KRON are related, but they do not make the same modelling choice:

Kron:	which linear boundary relation results from elimination?
Ward:	how is the eliminated external system represented at the boundary?
Opti-KRON:	which nodes or clusters should be retained under an error objective?

All three require a declared source model, boundary, injection model, and observation contract. None is inherently an asset-preserving transformation.

Linear Kron reduction

Executable boundary guardrail

The BMOPFTools tutorial `docs/src/tutorial_grounding.md` makes the key pedagogical point explicit: eliminating a neutral is conditional, not an automatically lossless simplification. The federated object `PSK-000008` links that warning to BMOPFTools contract `kron_boundary_recovery_preservation`. The compact check accepts a series-only four-wire source and three-wire target, requires perfect neutral grounding at both connection points, compares the ordered neutral Schur complement, and requires a declared recovery map. A floating or finite-grounded neutral is a grounding-precondition failure even if the matrices match; a passing check does not establish internal limits, protection, complete decision equivalence, or solver results. The companion fixture and its grounded target are intended as an executable teaching counterexample, alongside the richer Kron witnesses below.

Partition a linear nodal relation into retained boundary variables B and eliminated internal variables I:

$$\begin{bmatrix} \mathbf{i}_B \\ \mathbf{i}_I \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{BB} & \mathbf{Y}_{BI} \\ \mathbf{Y}_{IB} & \mathbf{Y}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{v}_B \\ \mathbf{v}_I \end{bmatrix}.$$

Proposition. If $\mathbf{Y}_{II}$ is invertible and $\mathbf{i}_I$ is fixed, then eliminating $\mathbf{v}_I$ gives the exact affine boundary relation

$$\mathbf{i}_B = \mathbf{Y}_{\mathrm{K}} \mathbf{v}_B + \mathbf{K}_I \mathbf{i}_I,$$

where

$$\mathbf{Y}_{\mathrm{K}} = \mathbf{Y}_{BB} - \mathbf{Y}_{BI} \mathbf{Y}_{II}^{-1} \mathbf{Y}_{IB}, \qquad \mathbf{K}_I = \mathbf{Y}_{BI} \mathbf{Y}_{II}^{-1}.$$

The eliminated voltages are recovered by

$$\mathbf{v}_I = \mathbf{Y}_{II}^{-1} (\mathbf{i}_I - \mathbf{Y}_{IB} \mathbf{v}_B).$$

Proof. Solve the internal block equation for $\mathbf{v}_I$ and substitute it into the boundary block equation.

For zero internal injection this becomes

$$\mathbf{i}_B = \mathbf{Y}_{\mathrm{K}} \mathbf{v}_B.$$

This is exact equality of a selected linear terminal relation. Schur elimination can create dense coupling among neighbours of eliminated nodes, and the resulting entries need not correspond to physical lines. Dörfler and Bullo analyze the graph and loopy-Laplacian properties of this operation [18]. Closure inside a more restrictive dynamic or device class requires additional physical assumptions [20].

The dashed boundary coupling is a reduced coefficient. It is exact for the retained linear boundary relation, but it is not by itself a physical asset with recoverable currents, limits, or outage identity.

Kron reduction does not by itself preserve:

- eliminated asset identity or topology;
- internal branch currents and limits unless recovery is retained;
- switching, outage, maintenance, or investment decisions;
- protection and failure dependencies;
- sparsity;
- nonlinear constant-power behaviour over arbitrary operating points.

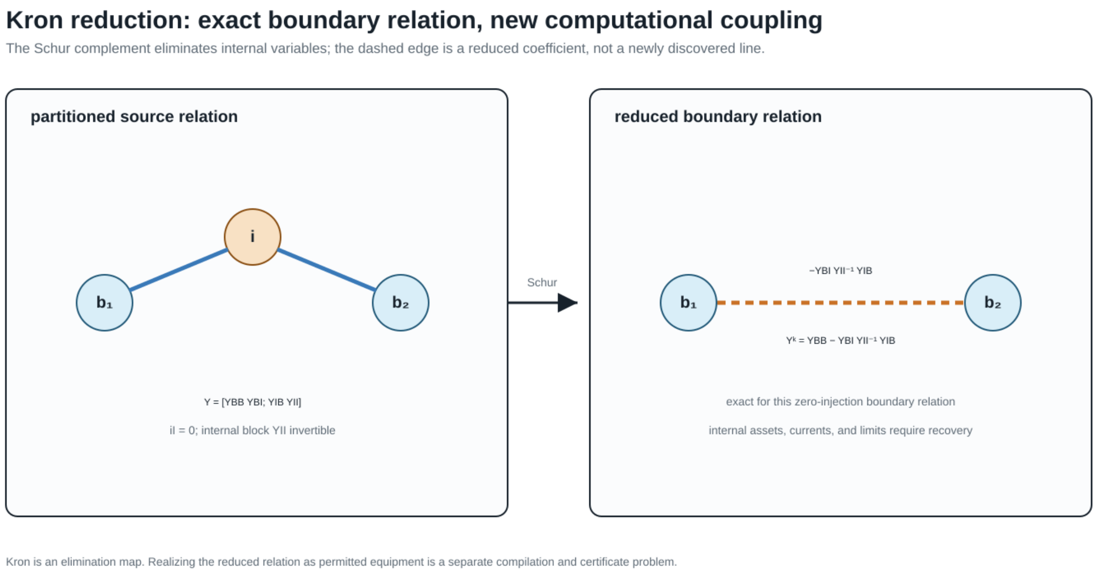

Figure 15.1: Kron reduction creates a boundary fill edge.

The Schur complement first produces a reduced multiport relation. Realizing that relation as a permitted collection of bus–branch devices is a separate synthesis or compilation problem.

Decision-model consequence

Eliminating a neutral voltage does not eliminate the neutral conductor's current rating. If the source relation gives the neutral branch current as $I_n = A_{nB}v_B + A_{nI}v_I$ and Kron recovery gives $v_I = Y_{II}^{-1}(i_I - Y_{IB}v_B)$, then the reduced decision model must retain the recovered constraint $|A_{nB}v_B + A_{nI}Y_{II}^{-1}(i_I - Y_{IB}v_B)| \leq \bar{I}_n$ (or its declared multiconductor norm). The boundary Schur complement alone is not enough for an OPF, hosting-capacity, protection, or contingency problem that observes that neutral limit. The constraint may be omitted only when the neutral rating is genuinely outside the observation set or is proved redundant by a separate certificate.

Typed multiconductor Kron reduction

In the multiconductor case, the symbols B and I denote direct sums of typed terminal-coordinate spaces, not merely lists of scalar buses. Let

$$\mathcal{V}_B = \bigoplus_{k \in B} \mathbb{C}^{n_k}, \qquad \mathcal{V}_I = \bigoplus_{k \in I} \mathbb{C}^{n_k}.$$

Each port voltage is first aligned with its junction by its terminal map. If $\mathbf{N}_q$ maps the ordered junction voltage into the ordered coordinates of port q, then

$$\mathbf{v}_q = \mathbf{N}_q \mathbf{U}_{j(q)}, \qquad \mathbf{I}_{j(q)} \mathrel{+}= \mathbf{N}_q^{\mathsf{H}} \mathbf{i}_q.$$

The conjugate transpose in the current map is the power-dual action: it makes $\mathbf{v}_q^{\mathsf{H}}\mathbf{i}_q$ agree with the corresponding junction power pairing. The assembled nodal blocks in the preceding section are formed only after these terminal maps have been applied. Consequently every product $\mathbf{Y}_{BI}\mathbf{Y}_{II}^{-1}\mathbf{Y}_{IB}$ is a typed map $\mathcal{V}_B \to \mathcal{V}_B$.

Proposition (typed coordinate-covariant Kron reduction). Suppose the assembled linear multiconductor relation has $\mathbf{Y}_{II} : \mathcal{V}_I \to \mathcal{V}_I$ invertible and the internal injection $\mathbf{i}_I$ is fixed data, independent of $\mathbf{v}_I$. Let $\mathbf{T}_B$ and $\mathbf{T}_I$ be invertible coordinate actions such that $\mathbf{T} = \mathrm{diag}(\mathbf{T}_B, \mathbf{T}_I)$ respects the retained/internal partition, with

$$\mathbf{v}_B = \mathbf{T}_B\widetilde{\mathbf{v}}_B, \qquad \mathbf{v}_I = \mathbf{T}_I\widetilde{\mathbf{v}}_I, \qquad \widetilde{\mathbf{i}}_B = \mathbf{T}_B^{\mathsf{H}}\mathbf{i}_B, \qquad \widetilde{\mathbf{i}}_I = \mathbf{T}_I^{\mathsf{H}}\mathbf{i}_I.$$

Then Kron reduction before or after the coordinate change gives the same boundary relation. Specifically,

$$\widetilde{\mathbf{Y}}_{\mathrm{K}} = \mathbf{T}_B^{\mathsf{H}}\mathbf{Y}_{\mathrm{K}}\mathbf{T}_B,$$

and the affine internal-injection term transforms as

$$\widetilde{\mathbf{K}}_I\widetilde{\mathbf{i}}_I = \mathbf{T}_B^{\mathsf{H}}\mathbf{K}_I\mathbf{i}_I.$$

The recovered internal voltage is coordinate consistent:

$$\widetilde{\mathbf{v}}_I = \mathbf{T}_I^{-1}\mathbf{Y}_{II}^{-1}(\mathbf{i}_I - \mathbf{Y}_{IB}\mathbf{v}_B).$$

Proof. The transformed blocks are

$$\widetilde{\mathbf{Y}}_{XY} = \mathbf{T}_X^{\mathsf{H}}\mathbf{Y}_{XY}\mathbf{T}_Y, \qquad X, Y \in \{B, I\}.$$

Using

$$(\mathbf{T}_I^{\mathsf{H}}\mathbf{Y}_{II}\mathbf{T}_I)^{-1} = \mathbf{T}_I^{-1}\mathbf{Y}_{II}^{-1}\mathbf{T}_I^{-\mathsf{H}}$$

in the transformed Schur complement cancels the internal coordinate maps and leaves $\mathbf{T}_B^{\mathsf{H}}\mathbf{Y}_{\mathrm{K}}\mathbf{T}_B$. The same substitution proves the affine and recovery identities.

The proposition covers phase permutations and other invertible port-coordinate actions. Per-port block diagonality within $\mathbf{T}_B$ or $\mathbf{T}_I$ is an additional modelling restriction that keeps the action local to each port; it is not needed by the covariance proof. A dense action within the retained partition and a dense action within the internal partition satisfy the same identities. What is excluded is a coordinate map that mixes retained and internal variables, because then it changes the eliminated/observed partition.

The fixed-injection qualification is load-bearing. If an internal device has a voltage-dependent law, for example $\mathbf{i}_I(\mathbf{v}_I) = \overline{\mathbf{S}_I \oslash \mathbf{v}_I}$, then $\mathbf{K}_I\mathbf{i}_I(\mathbf{v}_I)$ is not a fixed affine term and the proposition does not provide a voltage-independent boundary operator. Such a device may still be eliminated with a nonlinear or state-dependent implicit relation, but that is a different transformation contract.

A sequence-coordinate statement needs its transform convention declared: with the usual non-unitary Fortescue matrix $\mathbf{F}$, applying the same voltage and current coordinates gives the similarity action $\mathbf{F}^{-1}\mathbf{Y}\mathbf{F}$; the power-dual convention above gives the congruence action $\mathbf{F}^{\mathsf{H}}\mathbf{Y}\mathbf{F}$. These coincide only under additional normalization assumptions. Neither convention justifies dropping a conductor, identifying a neutral with earth, or aggregating phases: those maps are not invertible coordinate changes and require separate preservation claims.

Corollary (reciprocity is convention-relative). Kron reduction preserves complex symmetry of a reciprocal nodal matrix in physical coordinates. A real coordinate congruence $T^{\mathsf{T}}\mathbf{Y}T$ also preserves complex symmetry. The power-dual action $T^{\mathsf{H}}\mathbf{Y}T$ preserves it when T is real (or under other explicit compatibility conditions), but not for an arbitrary complex T. Hermitian structure and complex symmetry are distinct properties; the certificate must record which one is required and which coordinate action is being used. Thus “Kron preserves reciprocity” and “this complex coordinate representation remains symmetric” are separate claims.

Executable typed fixture

The first package-independent witness is recorded in `experiments/generated/typed-kron-witness.json` and certified by `experiments/generated/typed-kron-certificate.json`. It has three retained two-conductor ports and one eliminated two-conductor port. The fixture checks the reduced covariance residual, affine-injection covariance, internal-voltage recovery, and source-current limit evaluation after applying complex block-diagonal power-dual coordinate actions. It also checks dense actions within the retained and internal partitions, confirming that per-port block-diagonality is only a locality restriction. The reported residuals are below 2×10^{-15} for the boundary identity and below 7×10^{-17} for internal-state recovery; the fixture records the condition numbers of the internal block and coordinate actions.

The witness evaluates a deliberately different constant-power internal injection at the same recovered $\mathbf{v}_I$. Its current differs materially from the fixed datum, and the affine boundary term changes accordingly. This is a scope diagnostic: it does not disprove nonlinear elimination, but prevents the fixed-injection affine certificate from being reused for a voltage- dependent device.

The same artifact records two target-library outcomes. The general reduced multiport is reciprocal but its off-diagonal conductor blocks are not all individually symmetric, so the direct line–shunt construction is rejected for that restricted library. A separate admissible full-matrix, block-symmetric line–shunt witness is stamped exactly, with residual below 10^{-15}, while a diagonal-only line library is rejected. This is deliberately a positive and negative realizability test, not a claim that every Kron-reduced multiport is a physical bus–branch network.

The same witness now applies a deliberately narrower transformer-library test. In this selected library, each winding block is required to be diagonal in the declared conductor coordinates and the complete relation must remain reciprocal. The coupled target above is rejected because at least one winding block is dense; a companion target obtained by removing those cross-conductor entries is accepted. This is a closure test for a restricted transformer vocabulary, not an identification of turns ratios, leakage parameters, or grounding data.

Direct running-network line witness

The witness family is split by the question it answers rather than presented as one undifferentiated Kron example:

The table is the scope summary; the paragraphs below retain the construction details needed to reproduce each check.

Running-network line and explicit neutral recovery

The canonical running fixture now has a direct typed-Kron check in `experiments/generated/running-network-typed-kron-wit` Line l_1 is split into two equal four-conductor series sections, with the midpoint retained as the internal block. Eliminating that midpoint reproduces the original i_1-i_2 line primitive to below 10^{-11} and recovers the midpoint

claims	fixture/question	what varies	what changes	not claimed
TR-KRON-NEUTRAL-001	running four-conductor line midpoint	retained midpoint and neutral limit	recovered neutral current changes feasibility	physical rating generality; nonlinear loads
TR-KRON-NEUTRAL-002	explicit five-conductor earth coordinate	neutral–earth bond and earth terminal	neutral and earth KCL currents remain separate	standards-aligned grounding/protection
TR-KRON-NEUTRAL-003	two explicit grounding points	two bond locations	both bond-current observations survive	one aggregate neutral constraint
TR-KRON-NEUTRAL-004	grounding-impedance sensitivity	two declared impedance pairs	neutral current and feasibility classification	uncertainty quantification
TR-KRON-NEUTRAL-005	one state-dependent bond	endpoint state	frozen nominal map leaves residual; recomputation restores it	global nonlinear theorem
TR-KRON-NEUTRAL-006	two state-dependent bonds	endpoint state and two maps	segment currents and residual change	global continuation/protection
TR-KRON-NEUTRAL-007	finite continuation	five endpoint states	neutral-limit margin changes	adaptive/global continuation
TR-KRON-NEUTRAL-008	local derivative bound	perturbation scale	recomputed Jacobian reduces local error	global error bound
TR-KRON-FIVE-001, -002	five-bus scalar pendant/non-pendant cases	eliminated bus and retained boundary	exact recovery; non-pendant elimination creates fill	general multiconductor closure

voltage $(U_{i_1} + U_{i_2})/2$ to the same tolerance. Terminal order, bus identity, and the four-conductor boundary are retained explicitly.

The same artifact carries a deliberately tight neutral-limit witness. The four-conductor midpoint is eliminated, but the recovered neutral current is $(-0.0430252 + 0.0197760\,\mathrm{i})$ p.u. on both half-sections. A declared limit of 0.0426173 p.u. is therefore violated by this boundary point. The test records that the current recovery is exact and that dropping the neutral constraint would accept a point the source model rejects. This is claim `TR-KRON-NEUTRAL-001`.

The companion `experiments/generated/neutral-kron-independent-reproduction.json` reconstructs the four-conductor impedance and midpoint recovery with a separate standard-library complex solver. It matches both half-section neutral currents, reproduces the exact recovery identity, and retains the deliberately violated limit.

The same witness now includes a midpoint neutral-to-reference shunt probe. The shunt changes the recovered neutral current from the series-only value, adds a reference-current term to the midpoint KCL, and retains a separate neutral limit evaluation. Its KCL residual is below 10^{-11}, and the independent reproduction checks the shunted currents as well as the series case.

Explicit earth coordinate and grounding-parameter sensitivity

The next probe makes the earth-return coordinate explicit rather than treating it as an unnamed reference. In `TR-KRON-NEUTRAL-002`, a synthetic five-conductor (a, b, c, n, e) series midpoint retains an earth terminal e and stamps a midpoint neutral–earth bond. Kron recovery reports separate neutral and earth currents, verifies their KCL equations with opposite bond-current signs, and evaluates the neutral current limit on the recovered half-section. The companion `experiments/generated/explicit-earth-kron-independent-reproduction.json` uses a separate standard-library complex solver. Collapsing e into n would erase the observed bond-current relation.

The same artifact adds a two-grounding-point extension, `TR-KRON-NEUTRAL-003`. A three-segment five-conductor chain has explicit internal points m_1 and m_2, each with its own neutral–earth bond. The recovered segment currents verify separate neutral and earth KCL at both points, and both bond currents remain observable after the two internal blocks are eliminated. This is the smallest useful warning against replacing a distributed grounding structure by one aggregate neutral constraint.

Finally, `TR-KRON-NEUTRAL-004` holds the topology and terminal order fixed while sweeping four declared pairs of grounding impedances. The recovered neutral current changes across the cases, and a fixed 0.028 p.u. neutral limit changes feasibility classification: the structural Kron and KCL checks remain valid, but the decision observation does not. The generated sweep and its standard-library reproduction therefore separate structure preservation from parameter-dependent feasible-set preservation. This is a finite sensitivity probe.

State-dependent grounding bonds

The next boundary is local state dependence. In `TR-KRON-NEUTRAL-005`, the neutral–earth bond current is defined by an illustrative voltage-dependent law $i_{ne} = y_0(1+\alpha_{\mathrm{g}}|V_n - V_e|^2)(V_n - V_e)$. Here α_{g} is a grounding-law coefficient, distinct from the served-load decision α used in the parallel-line cases. After an endpoint state shift, the nominal bond map leaves a nonzero nonlinear residual and a different recovered neutral current; a local Newton solve with the bond recomputed at the shifted state restores the relation and re-evaluates the neutral limit. The independent reproduction uses a separate finite-difference Newton implementation.

The distributed version is also exercised locally in `TR-KRON-NEUTRAL-006`. The three-segment chain has two voltage-dependent neutral–earth bonds. After the same endpoint shift, freezing both nominal bond maps leaves a nonzero chain residual and changes the recovered neutral currents on the three segments; a two-point Newton solve with both maps recomputed restores the local relation. The companion standard-library reproduction checks the midpoint values and residuals.

`TR-KRON-NEUTRAL-007` records a finite endpoint-state continuation of that two-point chain at $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. Each state is solved with both nonlinear bond maps recomputed; the five nonlinear residuals remain small and the fixed 0.05 p.u. neutral limit changes classification along the path. Reusing the nominal map fails away from $\lambda = 0$. The independent reproduction checks every continuation row. This is a finite local path.

`TR-KRON-NEUTRAL-008` adds a local derivative certificate for the same illustrative voltage-dependent neutral–earth bond law. At a declared base state, the analytic real Jacobian is compared with the frozen nominal coefficient after a state shift. The frozen map leaves a nonzero residual, whereas the recomputed Jacobian gives the smaller local linearisation error; the error decreases as the declared step is reduced. The generated `experiments/generated/nonlinear-grounding-local-bound-witness.json` records the Jacobian, residuals, step scales, and interpretation. This is a local Taylor/conditioning check.

Five-bus scalar reductions

The five-bus companion `experiments/generated/five-bus-typed-kron-witness.json` covers the scalar pendant case directly. Eliminating bus m through its sole incident line u gives the same retained Y-bus as deleting that leaf line from the graph, and the recovered boundary current matches the full nodal relation to machine precision. This is claim `TR-KRON-FIVE-001`.

The same witness also eliminates the non-pendant bus ℓ while retaining i, j, k, m. Boundary-current recovery remains exact, but the Schur complement creates retained support on $j - m$ and $k - m$. These are fill edges: they are couplings in the reduced relation, not automatically new physical lines. The recovered u-branch current is exact as well, but a deliberately tight declared u-limit is violated by the recorded state. This is claim `TR-KRON-FIVE-002` and is the small scalar analogue of the ordering, fill-in, and retained-constraint consequences discussed later in the numerical chapter.

Realizability is a second theorem

The reduced matrix always defines a general linear factor on the retained ports. It need not define a conventional collection of lines. Realizability is therefore relative to a target device library.

Power-system shorthand

A nonzero off-diagonal block in a reduced admittance is a boundary coupling, not automatically a physical line. Any line–shunt realization is a second construction whose asset meaning, limits, states, and provenance must be declared separately.

For a useful baseline, suppose all retained junctions use the same ordered c-conductor coordinates and the target library permits:

1. one reciprocal full-matrix series primitive between any pair of retained junctions; and
2. one full-matrix shunt primitive at every retained junction.

Write the reduced admittance in $c \times c$ blocks $\mathbf{Y}^{\mathrm{K}}_{pq}$.

Proposition (direct line–shunt realization). If every off-diagonal block obeys

$$\mathbf{Y}^{\mathrm{K}}_{pq} = \mathbf{Y}^{\mathrm{K}}_{qp} = (\mathbf{Y}^{\mathrm{K}}_{pq})^{\mathsf{T}},$$

then the reduced relation has the exact complete-graph realization

$$\mathbf{Y}^{\mathrm{s}}_{pq} = -\mathbf{Y}^{\mathrm{K}}_{pq}, \qquad \mathbf{Y}^{\mathrm{sh}}_{p} = \mathbf{Y}^{\mathrm{K}}_{pp} - \sum_{q \neq p} \mathbf{Y}^{\mathrm{s}}_{pq}.$$

Proof. A reciprocal series primitive contributes $-\mathbf{Y}^{\mathrm{s}}_{pq}$ to both off-diagonal blocks and $\mathbf{Y}^{\mathrm{s}}_{pq}$ to both corresponding diagonal blocks. Summing all pairwise stamps reproduces every off-diagonal block. The residual shunt definition then reproduces each diagonal block. Thus the proposition is an exact stamping identity in a library that permits full $c \times c$ reciprocal blocks; it is not a claim that a smaller physical line library is closed under Kron reduction. For N retained junctions the construction uses $N(N-1)/2$ pairwise series blocks plus N shunts.

This is an algebraic realization, not yet a physical line realization. A passive line–shunt library imposes additional conditions such as

$$\mathrm{He}(\mathbf{Y}^{\mathrm{s}}_{pq}) \succeq 0, \qquad \mathrm{He}(\mathbf{Y}^{\mathrm{sh}}_{p}) \succeq 0,$$

along with its reciprocity, frequency, grounding, and parameterization rules. A restricted diagonal or sequence-decoupled library imposes stronger closure conditions. Ideal-transformer terminal maps can realize a broader class of off-diagonal blocks, but then transformer ratios, winding coordinates, grounding, and provenance become part of the target certificate. If none of these libraries closes, retaining $\mathbf{Y}_{\mathrm{K}}$ as one general multiport factor is still exact.

Internal current and limit recovery is yet another layer. If a source branch current has the affine form

$$\mathbf{I}_\ell = \mathbf{A}_{\ell B}\mathbf{v}_B + \mathbf{A}_{\ell I}\mathbf{v}_I,$$

substitution of the voltage recovery map gives an exact affine boundary map for $\mathbf{I}_\ell$. Keeping that map permits source limits to be checked; discarding it does not make limits on artificial reduced branches equivalent.

Fixed versus state-dependent equivalents

The finite continuation and nonlinear grounding witnesses make a second boundary explicit: a Ward or Kron map calibrated at one operating point is not automatically reusable when load, grounding, or another state parameter moves. Cross-repository object `PSK-000010` links this distinction to BMOPFTools contract `state_dependent_equivalent_provenance`. The compact package check requires a shared parameter, non-singleton domain, aligned base state, and an update-rule identifier; a frozen target is reported as provenance loss rather than promoted to a global equivalent. This declaration gate does not validate the update law or prove decision, protection, or solver equivalence.

Ward equivalents

The affine term

$$\mathbf{K}_I\mathbf{i}_I$$

shows what a network equivalent must do when the eliminated region has nonzero injections. A Ward-type equivalent combines the reduced boundary admittance with equivalent boundary injections, shunts, or sources intended to represent the external system in a power-flow study. Ward's original construction explicitly approximates suppressed loads and generation as constant currents, retains the tie terminals, replaces the eliminated network by a boundary mesh, and places equivalent injections at those terminals [21].

For a linear fixed-current source model, the affine relation above is exact. For a constant-power AC model, however,

$$\mathbf{i}_I(\mathbf{v}_I) = \left(\mathbf{S}_I \oslash \mathbf{v}_I\right)^*,$$

where $\oslash$ denotes componentwise division. The internal injection is then voltage dependent, so replacing it by a fixed boundary source is not a global exact reduction of the nonlinear feasible relation. Its validity must instead be tied to an operating point, linearization, iteration, or scenario domain.

The term *extended Ward* is not just another name for that base construction. Monticelli and coauthors build an external equivalent for static security from a single estimated operating state, address boundary-bus designation, and discuss treatment of external shunts and generator-outage studies [22]. A Ward–PV construction instead retains external generator buses after load-node elimination; the reduced Ward–PV model then aggregates selected coherent generator groups [23]. These targets preserve different state and control structure.

The resulting source taxonomy is:

- **classical Ward:** constant-current approximation followed by external-bus elimination and boundary-mesh/injection realization;
- **operating-state extended Ward:** a base-state-calibrated external equivalent with extra boundary, shunt, and contingency treatment;

- **Ward–PV:** retention of selected generator/PV structure before any subsequent coherent aggregation;
- **nonlinear or iterative Ward-type methods:** later constructions that update or fit the boundary source model over operating points and must state their own domain.

These are historically related, not mathematically interchangeable. In particular, the exact affine result for fixed currents does not make a base-state-calibrated AC equivalent globally exact for constant-power or voltage-controlled devices.

Opti-KRON

Opti-KRON adds a structural selection problem around a Kron-based electrical reduction. An assignment matrix maps original nodes to retained supernodes; the selection trades the degree of reduction against reproduction error for declared voltage observations and operating scenarios. The three-phase work also constrains phase availability and connectivity [24]. A related extension identifies nodes to restore so that the final reduced network recovers radiality [25].

It is useful to factor the method conceptually as

$$\text{optimized structural assignment} + \text{Kron-based electrical reduction} + \text{scenario observation metric.}$$

The Schur-complement step can be exact for its retained linear boundary relation while the supernode representation of eliminated voltages is approximate. The combined method is therefore not classified simply as *exact Kron reduction*. Its certificate must record at least:

- the retained-node and assignment decision spaces;
- phase and connectivity guards;
- the operating scenarios used to evaluate voltage error;
- the voltage observation norm and bound;
- any radiality restoration;
- the source injections and controls represented in those scenarios;
- whether source constraints and decisions can be recovered.

Low voltage error over a scenario set does not alone prove equality of AC OPF feasible sets, active limits, discrete decisions, or objective values. Those are additional observation families requiring their own evidence.

Decision-model consequence

Scenario voltage accuracy answers one observation question. It is not a surrogate theorem for feasibility, active-limit, objective, or discrete decision accuracy.

Classification in the book's transformation language

The classification depends on the complete construction, not on whether its name contains *Kron*:

- **Zero-injection linear Kron:** from a linear nodal or multiport relation to its boundary relation. This is exact behavioural reduction; physical realization and internal constraints are not automatic.
- **Fixed-current affine Kron:** from a linear relation with fixed internal injections to an affine boundary relation. This is exact behavioural reduction for that source model, not for arbitrary voltage-dependent injections.
- **Ward equivalent:** from an external-system study model to a boundary network with equivalent injections. It may be exact, local, or approximate depending on the injection model; there is no assumed universal definition across Ward variants.
- **Opti-KRON:** from scenario data and a candidate topology to a selected reduced network. This is mixed structural optimization and scenario approximation; general decision equivalence is not automatic.
- **Radiality restoration:** from a nonradial reduced topology to one with selected nodes restored. This is structural postprocessing under its stated rule, not electrical or decision equivalence by itself.

These distinctions place each method relative to a preservation contract before relating it to the book's local rewrite rules.

Relation to local transformations

The guarded degree-two series rule is a special zero-injection elimination for which the reduced relation remains inside a declared series-element family. A star–mesh transformation is another local Schur-complement realization. Parallel primitive summation is different: it combines factors sharing a boundary but eliminates no bus. Redundant-limit removal is different again: it is exact presolve on the constraints while the physical members remain.

This separation prevents every operation involving a smaller network from being called Kron reduction.

Open research boundary

Executable comparison: exact, operating-point, and scenario-selected

The comparison artifact `experiments/generated/kron-ward-scenario-comparison.json` uses four declared scenarios and a shared observation contract for three targets. It records boundary voltage and current, recovered internal voltage and current, the source-current constraint margin, and the scenario objective together with the selected structural decision. This prevents a sparse candidate from looking successful merely because its boundary-current error was reported without the state, constraint, or decision quantities that the study actually uses.

The selection is intentionally small and transparent. It demonstrates the classification boundary rather than reproducing a particular published Opti-KRON implementation: the target candidates and penalty are declared in the artifact, and the selected target is judged on the same observation family as the alternatives. The result is claim `TR-KRON-002` and does not establish global AC feasibility, objective, or control preservation.

The fixture now also exposes the boundary-support distinction directly. An **extended Ward support target** retains the same base-calibrated reduced admittance and fixed base injection, but supplies the explicit support term

$$\Delta \mathbf{i}_B^{\mathrm{support}} = \mathbf{K}_I(\mathbf{i}_I - \mathbf{i}_I^{\mathrm{base}}).$$

Target	Construction	Result in the fixture
exact Kron	recompute the affine boundary relation for each fixed internal injection	exact for every declared scenario
operating-point Ward	retain the exact reduced admittance but freeze the boundary injection at the base scenario	exact at the base point; relative current errors of about 1.5–3.3% off base
Opti-KRON-style target	select full, banded, or diagonal retained couplings using an explicit scenario error plus sparsity penalty	selects the banded target; this is scenario approximation, not decision equivalence

For the declared fixed-current linear fixture this support term makes the target exact at every scenario, and its off-base norm is nonzero. That result does not make the construction a globally exact AC Ward equivalent: it records the additional boundary quantity that must be available, and the source model under which it is valid. The generated comparison records these rows under `extended_ward_rows` alongside the operating-point rows.

Certified approximation chain

The next artifact composes the approximation vocabulary into a decision test. In the same one-state fixture, the Ward target freezes the internal injection at the base scenario. For a scenario injection mismatch $\delta i_I = i_I^{\mathrm{base}} - i_I$, the exact linear maps give

$$\delta i_I \longmapsto Y_{II}^{-1}\delta i_I \longmapsto K_I \delta i_I \longmapsto m = L - \|\widehat{i}_B\|,$$

where the middle term bounds recovered-state error, the next term bounds boundary-current error, and m is the approximate current-limit margin. A declared error bound e yields the same three-way test used in the numerical consequences chapter: $m > e$ is certified feasible, $m < -e$ is certified violated, and $|m| \le e$ is ambiguous.

The generated witness `experiments/generated/certified-approximation-witness.json` reports this chain for all four scenarios:

Scenario	Approximate margin	Error bound	Classification
base	-0.09535	0	certified violated
high-load	0.02203	0.03430	ambiguous
low-voltage	0.12026	0.03028	certified feasible
internal-outage proxy	-0.07692	0.07498	certified violated

This is claim `TR-KRON-003`. The normwise bound is exact for the declared one-state linear fixture, so it demonstrates composition of the machinery, not a general nonlinear or uncertainty-aware certification theorem. In particular, the high-load row is intentionally ambiguous even though the nominal Ward point satisfies the limit: the error interval crosses the decision boundary.

Scoped nonlinear AC probe

The generated `nonlinear-ward-witness.json` takes one deliberately small step toward the AC case. Its eliminated state has a constant-power injection, so the internal current is $\mathbf{i}_I(\mathbf{v}_I) = \overline{\mathbf{S}/\mathbf{v}_I}$ and the exact state is found with a damped Newton solve. The Ward target still freezes the base internal current. For each scenario the witness reports the nonlinear residual at the Ward state, a local inverse-Jacobian estimate, the direct boundary-current error, and the resulting local decision classification:

Scenario	Local result	What it demonstrates
base	locally certified feasible	calibration is exact at the base point
small shift	locally certified feasible	the local estimate dominates the observed boundary-current error
large shift	local-bound ambiguous	nonlinear residual and the error interval grow away from calibration

This is an exploratory numerical witness, not a theorem. The inverse-Jacobian estimate is local, depends on the chosen Newton solution, and does not certify global AC feasibility, bifurcation behaviour, parameter uncertainty, or KKT preservation. A solver-exported Jacobian/KKT comparison remains a separate roadmap item.

The current evidence leaves four implementation questions open:

1. executable realizability tests for selected line, shunt, transformer, and general-factor libraries;
2. recovery and constraint maps for internal currents, powers, and losses;
3. an executable small example comparing exact Kron, a Ward operating-point equivalent, and an Opti-KRON-style scenario approximation;
4. a decision experiment measuring feasibility, active constraints, and objective error in addition to voltage error;
5. extension of the certified-approximation chain to nonlinear AC residuals, uncertain parameters, and an independently derived error analysis beyond this scoped probe.

Part 6: Constraints and decisions

Chapter 16

Parallel AC decision case

Page status: guarded multiconductor decision case with executable redundancy certificates; general state-dependent classification remains open.

This is the multiconductor specialization of the canonical parallel-member failure in the first scalar counterexample. That chapter owns the general warning: an aggregate terminal relation does not automatically preserve member-constrained decisions. The present chapter adds coupling, complex voltages, and AC power balance without restating that warning as a separate modelling claim.

The scalar parallel example shows the feasible-set mechanism with almost no algebra. This case retains complex conductor voltages, mutual impedance, phase-to-neutral power, voltage bounds, and nonlinear AC power balance. Its purpose is to test whether the same representation failure changes an AC decision optimum.

The grid explains why the later cases are not repetitions of the scalar example. Coupling, explicit neutrals, end shunts, two-end observations, and control decisions are added deliberately; the evidence obligation grows with the model rather than being hidden behind a larger diagram.

Source model

Two buses i and j have ordered conductor set (a, n). The sending voltage is fixed at

$$\mathbf{U}_i = (1, 0)^{\mathsf{T}} \text{ p.u.}$$

and two parallel members satisfy

$$\mathbf{I}_{\ell ij} = \mathbf{Y}_\ell(\mathbf{U}_i - \mathbf{U}_j), \qquad \ell \in \{1, 2\}.$$

The full, coupled series impedances are

$$\mathbf{Z}_{\ell_1} = \begin{bmatrix} 0.04 + 0.08\mathrm{j} & 0.01 + 0.02\mathrm{j} \\ 0.01 + 0.02\mathrm{j} & 0.04 + 0.08\mathrm{j} \end{bmatrix}, \qquad \mathbf{Z}_{\ell_2} = 10\mathbf{Z}_{\ell_1}.$$

Every member and conductor has limit $|I_{\ell ij,c}| \leq 0.6$ p.u. Receiving-end conservation requires

$$\sum_\ell I_{\ell ij,a} + \sum_\ell I_{\ell ij,n} = 0.$$

The worked cases are an escalation, not repetition

Each row adds a modelling distinction or a stronger evidence obligation.

Parallel-member decision cases

case	coupling	non-proportionality	neutral	shunts	end structure	check
scalar DC	—	—	—	—	—	analytic
scalar AC π	—	—	—	yes	one-end	analytic
proportional multiconductor	yes	no	explicit	—	one-end	certificate
non-proportional four-wire	yes	yes	explicit	—	two-end	independent
four-wire nominal-π	yes	yes	explicit	yes	two-end	independent

Transformer compilation and control cases

case	electrical scope	control domain	forgotten/auxiliary detail	evidence
leakage relation	pairwise	fixed	—	—
reference compilation	all windings	fixed	recovery	round-trip
terminal assembly	WYE/DELTA	fixed	ground	certificate
factor completion	multi-port	fixed	auxiliary	cross-check
tap decisions	multi-port	discrete	controls	solver-backed

The progression makes the research programme visible: conductor coupling and decision observations grow together with the guard and reproduction burden.

yes/explicit = present; partial = scoped or one-end; — = deliberately absent in that case.

Figure 16.1: Escalation grid for the parallel-member and transformer worked cases.

The decision $\alpha \geq 0$ scales a constant-power direction across phase and neutral:

$$S_j = \alpha(1 + 0.2\mathrm{j}) = (U_{j,a} - U_{j,n}) \left(\sum_{\ell} I_{\ell ij,a} \right)^* .$$

The phase-to-neutral voltage magnitude is restricted to $[0.70, 1.05]$ p.u., and the objective maximizes α.

Four formulations

The **source** formulation retains each $\mathbf{I}_{\ell ij}$ as a variable and enforces every member limit. The **naive aggregate** uses

$$\mathbf{Y}_{\mathrm{eq}} = \mathbf{Y}_{\ell_1} + \mathbf{Y}_{\ell_2}$$

and assigns each aggregate conductor the summed limit 1.2 p.u. The **exact lifted aggregate** uses the same aggregate terminal relation but recovers

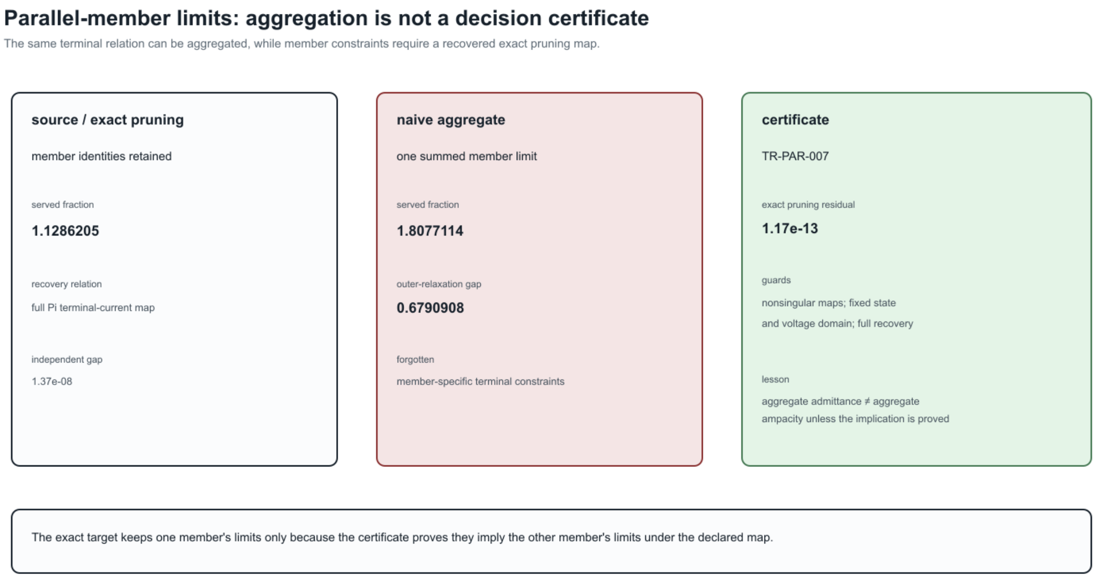

Figure 16.2: Parallel-member aggregation preserves a terminal relation, but exact decision pruning needs a recovered member-current map and a proved implication.

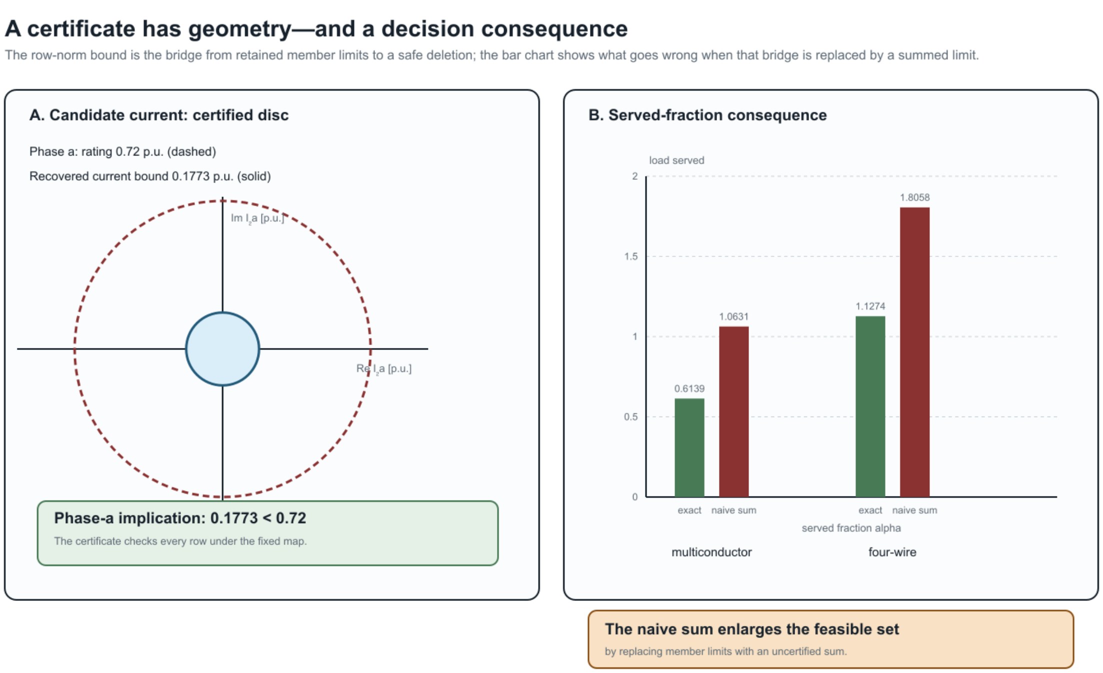

Figure 16.3: The phase-a recovered-current bound and candidate rating share one scale; artifact-derived bars compare exact-pruned and naive served fractions.

$$\mathbf{I}_{\ell ij} = \mathbf{Y}_\ell(\mathbf{U}_i - \mathbf{U}_j)$$

inside the target model and applies the original 0.6 p.u. limits.

The **exact pruned aggregate** first observes that $\mathbf{Z}_{\ell_2} = 10\mathbf{Z}_{\ell_1}$, and hence

$$\mathbf{Y}_{\ell_2} = 0.1\mathbf{Y}_{\ell_1}, \qquad \mathbf{I}_{\ell_2 ij} = 0.1\mathbf{I}_{\ell_1 ij}.$$

Because the members have equal componentwise limits, every ℓ_2 current circle is implied by the corresponding ℓ_1 circle. The formulation therefore keeps both recovery maps but enforces only the certified nonredundant ℓ_1 limits. This is the multiconductor proportional special case of the constraint-pruning idea; it does not assume that the general scalar quadratic test in [1] automatically extends to arbitrary matrix-valued conductor models.

General linear-current containment test

The proportional proof is now implemented as a special case of a reusable linear-current certificate. Let A_r map the stacked complex endpoint voltages to a retained current group, and let A_c define a candidate constraint. For any complex matrix A, define its realification

$$\mathcal{R}(A) = \begin{bmatrix} \Re(A) & -\Im(A) \\ \Im(A) & \Re(A) \end{bmatrix}$$

and the normalized quadratic form

$$Q(A, I^{\max}) = \frac{\mathcal{R}(A)^\mathsf{T}\mathcal{R}(A)}{(I^{\max})^2}.$$

The retained constraint implies the candidate constraint exactly when

$$Q(A_r, I_r^{\max}) - Q(A_c, I_c^{\max}) \succeq 0.$$

To see this, write the constraints as $x^\mathsf{T}Q_r x \leq 1$ and $x^\mathsf{T}Q_c x \leq 1$. Positive-semidefinite dominance gives the forward implication immediately. Conversely, homogeneity lets any x with $x^\mathsf{T}Q_r x > 0$ be scaled to the retained boundary; directions in the nullspace can be scaled without bound and therefore must also lie in the candidate nullspace. Thus implication requires $x^\mathsf{T}Q_c x \leq x^\mathsf{T}Q_r x$ for every x. The argument includes singular cylinders, not only bounded ellipsoids.

For componentwise multiconductor limits, the implementation applies this test to every aligned conductor at both terminal ends. A non-proportional test uses different row factors, 0.2 and 0.4, so the member admittance matrices are not scalar multiples even though every candidate current circle is certifiably implied. A second test is safe at ij and unsafe at ji and is correctly rejected. This establishes claim `TR-PAR-005`.

The test is necessary and sufficient for each individual centered Euclidean norm implication. The current member-level algorithm is only a pairwise certificate: it does not yet detect a constraint implied jointly by several other limits, nor does it cover affine offsets, non-Euclidean thermal regions, or decision-dependent line, tap, outage, and switching states.

Formulation	Served fraction	Receiving voltage magnitude	Largest recovered member current	Variables / constraints
source members	0.6138908	0.9485579	0.6000000	13 / 19
naive aggregate	1.0630833	0.9034471	1.0909091	5 / 9
exact lifted	0.6138908	0.9485579	0.6000000	5 / 11
exact pruned	0.6138908	0.9485579	0.6000000	5 / 9

Results

The naive target serves about 73% more load than the source by violating the stronger member's current limit. The exact lifted formulation reproduces the source optimum while using the aggregate current relation and eight fewer real current variables in this implementation (four fewer complex member currents). This is claim `TR-PAR-004`. The exact pruned formulation has the same variable and constraint counts as the naive target but the source optimum: model size alone therefore does not establish fidelity.

Solver-independent check

For the chosen proportional matrices, a phase-to-neutral current sees loop impedances

$$z_\ell = Z_{\ell,aa} + Z_{\ell,nn} - Z_{\ell,an} - Z_{\ell,na}.$$

The equivalent loop impedance is $z = 0.05454545 + 0.10909091\mathrm{j}$ p.u. If C is the limiting total-current magnitude, $s = 1 + 0.2\mathrm{j}$, and v is the receiving voltage magnitude, then

$$1 = v^2 + 2Cv\frac{\Re(z)\Re(s) + \Im(z)\Im(s)}{|s|} + |z|^2C^2, \qquad \alpha = \frac{Cv}{|s|}.$$

The source member limit gives $C = 0.66$ p.u.; the summed aggregate gives $C = 1.2$ p.u. The served-power derivative on the high-voltage branch remains positive at both limits, so each current cap is binding. The positive quadratic roots reproduce both Ipopt objectives to better than 10^{-7}. These are values on the traced high-voltage branch: Ipopt supplies local solves here, not a global-optimality certificate. The exact pruning conclusion also relies on the deliberately exact proportionality $\mathbf{Z}_{\ell_2} = 10\mathbf{Z}_{\ell_1}$; near-proportional data require the general quadratic-containment test.

Scope and reproducibility

This is a deliberately minimal nonlinear AC case, not a three-phase benchmark. It includes conductor coupling and an explicit return path, but uses proportional member matrices and one scalable load direction so that a closed form check remains possible. Its proportional current map supplies a complete redundancy proof for this example. The non-proportional three-phase four-wire case is the next extension: it breaks proportionality inside the solved decision problem, adds all three phases plus neutral, certifies joint componentwise implication, and cross-checks the line primitives with BMOPFTools.

Run:

```
julia --project=experiments experiments/run_multiconductor_parallel_ac.jl
julia --project=experiments experiments/test/multiconductor_parallel_ac.jl
```

The generated AC certificate contains all four solutions, the proportional cross-check, the two-end quadratic-containment certificate, recovered member currents, model sizes, residuals, and closed-form differences. An automated independent re-derivation reproduces the reported figures and the binding constraint interpretation; it is not human peer review. The generic checker is in `experiments/transformations/MulticonductorFlowLimitRedundan`

Chapter 17

Transformer tap decision case

Page status: solver-backed transformer-control case with independent local reproduction; broader control and global-optimality claims remain open.

The fixed-voltage witness in the previous chapter proves that evaluating a tap start value can remove the best decision. This case takes the next step: the tap-dependent 11-terminal transformer factor is stamped into nonlinear network voltage and power-balance equations and solved with JuMP and Ipopt.

Its result is deliberately different from the preceding fixed-boundary witness: that witness minimises a transformer-local leakage-current metric at a prescribed boundary voltage and selects 1.05. Here the tap is embedded in a network decision that maximises served load subject to AC voltage, KCL, and recovered leakage-current constraints, and the boundary voltages are variables. The network objective therefore selects 0.95. The examples are complementary objective/boundary-condition cases, not contradictory tap recommendations.

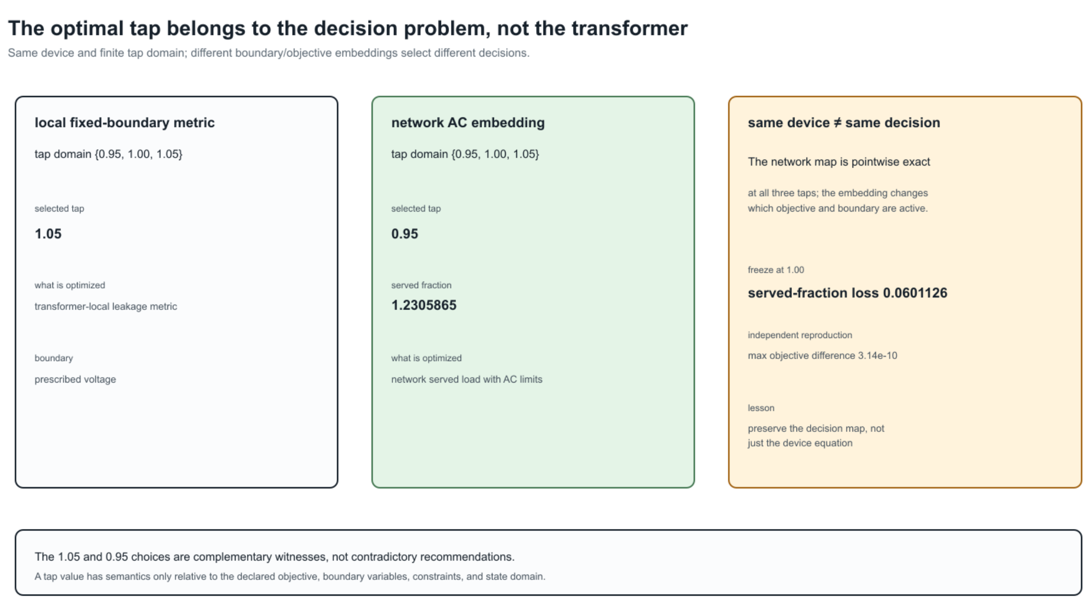

Figure 17.1: The optimal tap belongs to the decision problem, not the transformer: the same finite tap domain gives different choices under different embeddings.

The example deliberately retains the running transformer's WYE/WYE/DELTA structure. It is not a balanced positive-sequence transformer surrogate.

Network embedding

Transformer x_1 has winding set $\mathcal{K}_{x_1} = \{1, 2, 3\}$. Winding 1 is a balanced three-phase slack with an explicit neutral voltage. Winding 2 is a four-terminal WYE load bus. Its phase-to-neutral voltage is

$$V_{x_1,2,p} = U_{x_1,2,p} - U_{x_1,2,n}, \qquad p \in \{a, b, c\}.$$

The scalar decision $t_{x_1,2}$ retains the contract domain $\{0.95, 1.00, 1.05\}$. At a selected value, `TR-XFMR-005` evaluates the complete terminal and lifted leakage-current maps

$$\mathbf{I}^{\mathrm{term}}_{x_1} = \mathbf{Y}_{x_1}(t_{x_1,2})\mathbf{U}_{x_1}, \qquad \mathbf{I}^{\mathrm{w,leak}}_{x_1} = \mathbf{M}^{\mathrm{leak}}_{x_1}(t_{x_1,2})\mathbf{U}_{x_1}.$$

Currents are positive into the transformer factor. A balanced constant-power load has per-phase direction $0.50 + 0.10\mathrm{j}$ MVA and served fraction $\alpha \geq 0$. Phase power balance is therefore

$$-V_{x_1,2,p}\left(I^{\mathrm{term}}_{x_1,2,p}\right)^* = \alpha(0.50 + 0.10\mathrm{j})\ \mathrm{MVA}.$$

The winding-2 neutral is not silently grounded or eliminated. Its KCL equation is

$$\sum_{c \in \{a,b,c,n\}} I^{\mathrm{term}}_{x_1,2,c} = 0.$$

All three phase-to-neutral magnitudes remain in $[0.90, 1.05]$ p.u., and all nine original leakage-path limits are recovered and enforced:

$$\left|I^{\mathrm{w,leak}}_{x_1,k,c}\right| \leq I^{\max}_{x_1,k,c}, \qquad k \in \mathcal{K}_{x_1}.$$

Open delta tertiary and its gauge

Winding 3 is an open delta. Its three terminal currents are zero, but the terminal-to-ground voltage common mode is not physically observable by the delta incidence. The implementation imposes two independent complex KCL rows; the third follows from their sum. It then fixes

$$\sum_{p \in \{a,b,c\}} U_{x_1,3,p} = 0$$

only as a voltage-reference gauge. This does not add a grounding branch or change any line-to-line voltage. Separating the coordinate gauge from physical grounding is essential in a multiconductor model.

Exact discrete enumeration

The finite-domain problem is solved by enumeration. For each declared tap, the direct source contract and the parameterized target each produce a 15-variable, 30-constraint continuous nonlinear subproblem. The target does not relax, interpolate, round, or freeze the tap. Selection occurs only after all three tap-conditioned subproblems have been solved.

Tap $t_{x_1,2}$	Served fraction α	Served MW	Winding-2 voltage (p.u.)	Largest leakage loading
0.95	1.2305865	1.845880	1.0291923	1.0000000
1.00	1.1704739	1.755711	0.9789174	1.0000000
1.05	1.1159504	1.673926	0.9333170	1.0000000

The direct source and parameterized target agree at every tap to the reported solver tolerance and both select $t_{x_1,2} = 0.95$. The binding constraints are the three 2200 A winding-2 leakage-current limits; the voltage bounds remain slack.

Freezing the tap at its 1.00 start reduces the served fraction by 0.0601126. With three 0.50 MW phase directions, that is a served-load loss of 0.090169 MW. This is a network decision error, not merely a change in a transformer-local current metric.

What the certificate establishes

Certificate `TR-XFMR-006` has two distinct layers of evidence:

1. algebraically, direct source evaluation and parameterized factor evaluation stamp the same terminal admittance and leakage-current recovery map at each retained tap;
2. numerically, the corresponding nonlinear AC subproblems return the same states, constraint activity, objectives, and selected tap to the recorded tolerances.

The first layer establishes exact compilation of the network equations. The second is reproducible evidence for this nonconvex example; an Ipopt local termination status is not a general global-optimality proof.

Maximum recorded residuals are below 10^{-7} A for winding-2 neutral KCL and open-tertiary KCL, and below 10^{-8} MVA for phase power balance. The delta common-mode gauge residual is below 10^{-8} p.u.

Independent numerical reproduction

`TR-XFMR-007` reproduces the decision with a separate numerical engine. After receiving the same certified transformer matrices and case data, this engine uses only linear-algebra operations; it does not construct a JuMP model or call an external optimizer.

For fixed tap t and served fraction α, the seven unknown complex terminal voltages become fourteen normalized rectangular coordinates $\mathbf{x}$. The independent residual stacks

$$\rho(z) = \begin{bmatrix} \Re(z) \\ \Im(z) \end{bmatrix}, \qquad \mathbf{r}_t(\mathbf{x}, \alpha) = \begin{bmatrix} \rho(\Delta S_a) \\ \rho(\Delta S_b) \\ \rho(\Delta S_c) \\ \rho(\Delta I_n) \\ \rho(I_{x_1,3,a}) \\ \rho(I_{x_1,3,b}) \\ \rho(\Delta U_3^0) \end{bmatrix} \in \mathbb{R}^{14},$$

where ΔS_p is phase-power mismatch, ΔI_n is winding-2 neutral KCL mismatch, and ΔU_3^0 is the open-delta common-mode gauge residual. Nonzero residual blocks are scaled by fixed MVA, current, or voltage bases; the scaling is invertible and does not change their zero set.

A central finite-difference Jacobian and damped Newton iteration solve $\mathbf{r}_t = 0$. Continuation begins at the direct linear no-load solution and increases α along the high-voltage branch. The search must observe a feasible point followed by a converged infeasible point before it may bisect an upper boundary. This guard is tested explicitly: a truncated scan and a case with no feasible voltage interval both return structured rejections.

Tap	Independent α	JuMP/Ipopt α	Difference
0.95	1.2305865268	1.2305865271	-2.90×10^{-10}
1.00	1.1704738807	1.1704738810	-3.14×10^{-10}
1.05	1.1159503676	1.1159503679	-2.74×10^{-10}

Both methods select $t_{x_1,2} = 0.95$. Across all positions, the largest secondary-voltage difference is 1.94×10^{-12} p.u. and the largest leakage-current difference is 5.18×10^{-7} A. The independent boundary states have scaled equality residuals below 5.6×10^{-11}.

This reproduction changes the nonlinear algorithm and optimization machinery, but it deliberately shares the certified input matrices and case assembly. It is therefore an independent numerical check, not an independent data-model or transformer-primitive implementation. Continuation also certifies only the traced high-voltage branch under the recorded bracketing assumptions; it is not a general global-optimality method for nonconvex AC problems.

Run:

```
julia --project=experiments experiments/run_transformer_tap_ac_decision.jl
julia --project=experiments experiments/test/transformer_tap_ac_decision.jl
julia --project=experiments experiments/run_transformer_tap_ac_independent_reproduction.jl
julia --project=experiments experiments/test/transformer_tap_ac_independent_reproduction.jl
```

Boundary and next controls

This first solver-backed case uses one ganged scalar magnitude tap, a finite domain, a balanced load direction, and tap-independent leakage, excitation, and internal grounding. It does not yet cover phase-angle regulation, independent phase taps, mechanically coupled tap decisions, automatic deadbands, or tap-dependent loss parameters in this full 11-terminal network.

The scoped companion artifact `experiments/generated/transformer-control-family-witness.json` now checks the control-domain boundary without overstating solver evidence. It shows that scalar magnitude, phase-angle, independent-phase, mechanically coupled, and automatic-deadband controls can all compile pointwise when the same typed control map is retained. It also shows that a tap-dependent loss parameter must be evaluated at the retained tap: freezing it at the base tap leaves a nonzero off-tap residual. Each declared map also has a small JuMP/Ipopt feasibility probe, establishing executable solver-backed control-domain evidence. The phase-angle and tap-dependent-loss maps additionally run through a two-bus AC served-current network probe, while independent-phase and mechanically coupled maps run through a three-phase uncoupled probe. The report therefore crosses the control/network boundary without presenting these fixtures as a full neutral-coupled unbalanced network OPF. Richer multiwinding domains remain open; the four-wire probe now records mutual impedance, neutral displacement, and return-current KCL explicitly, but does not replace a full multiwinding network case.

The same certificate now includes a two-scenario switching-cost ledger. It enumerates all (3^2=9) ordered tap pairs, evaluates the two locally solved AC scenario objectives, subtracts the declared switching cost, and records the selected pair. This is branch-complete for the declared finite pair domain; it is not a global certificate for the continuous nonconvex subproblems. The certificate also sweeps five switching-cost values. For this fixture the selected pair remains ((0.95,0.95)) throughout the tested range, which is a reported stability result—not an assumption that the policy is cost-invariant in other scenarios. It also records the positive intersections of the affine branch objectives, so potential policy changes can be inspected analytically before choosing a cost sweep.

The companion `TR-XFMR-008` ledger keeps the same 11-terminal transformer but changes the second scenario by phase: its three constant-power directions are multiplied by the explicitly recorded vector $(1.08, 0.91, 1.04)$. All nine ordered tap pairs are still enumerated, and the scenario directions remain attached to their phase identities. This is the useful distinction between a phase-selective unbalanced scenario and a scalar stress factor: the former cannot be represented faithfully by silently rescaling one aggregate load. The result is finite, local solver-backed evidence; it does not claim global optimality or a general unbalanced multiwinding theorem.

The finite path extension `TR-XFMR-009` evaluates three phase-selective scenarios, using the explicitly recorded scales $(1, 1, 1)$, $(1.02, 0.98, 1.01)$, and $(0.99, 1.03, 0.98)$, and enumerates all $3^3 = 27$ ordered tap triples. Its objective is the sum of the three locally solved served fractions minus a declared cost on tap movement between consecutive scenarios. This makes the temporal/control semantics explicit without pretending that a two-scenario ledger is a general multi-period OPF. The branch ledger is complete for this finite path domain; continuous global optimality, operation-count limits, and richer topology decisions remain open. The separate finite-difference reproduction also traces the nine scenario/tap boundaries and selects the same 27-branch path, with a recorded maximum net-objective difference below 10^{-8}.

`TR-XFMR-010` adds an explicit operation policy: at most one tap movement is allowed across the two scenario transitions. The ledger still enumerates all 27 triples before filtering, leaving 15 admissible branches. This ordering matters—filtering first would hide the distinction between the full decision domain and the policy-constrained feasible set.

Chapter 18

BIM/BFM: shared coordinates and member constraints

Page status: literature-informed formulation case; the equations illustrate scope boundaries and are not a new executable certificate.

The notes by Geth and Liu provide a compact warning about expressive notation. They study BIM and BFM second-order-cone formulations for parallel lines and Π-sections with ideal transformers and shunts [26]. The case illustrates how member identities and consistency relations affect what a formulation establishes.

This is a notation-capability plate, not a new numerical certificate. It makes the scope boundary visible before the equations are interpreted as a theorem.

Shared voltage coordinates can retain member constraints

Fixed scalar series members: indexed admittances and recovery maps retain the member distinction.

question	shared W + member data	separate W + equalities
member current limit	yes: member data + bound	yes: same physical relation
independent outage/state	requires state-dependent laws	requires state-dependent laws
recovered branch current	yes: member current map	yes: member current map
member measurement	yes: observation map	yes: observation map
common voltage products	shared by construction	equality constraints required

Dropping consistency equalities can change a relaxation; adding an index alone proves no equivalence.

Squared member current: |Y_l|^2 (W_ii + W_jj - 2 Re W_ij). Source member data must remain available.

Figure 18.1: A variable-signature capability matrix shows which member-level questions shared voltage coordinates can express when member data and constraints are retained.

The branch identity is part of the variable signature

For the following displayed identities, assume fixed scalar series-only branches with no shunts or transformer taps. Consider two parallel branches ℓij and kij. Their impedances are Z_ℓ and Z_k and their series flows are $S_{\ell ij}$ and S_{kij}. A BIM representation can use one bus-pair cross-product $W_{ij} = U_i U_j^*$ and write

$$S_{\ell ij} = Y_\ell^*(W_i - W_{ij}), \qquad S_{kij} = Y_k^*(W_i - W_{ij}).$$

The shared W_{ij} expresses that both members see the same endpoint voltage product. In a BFM representation, each branch may instead retain its own lifted current L_ℓ and L_k. The two variable spaces have different coordinates and different relaxation geometry. They are not equivalent merely because both are described as "the branch-flow model for the same network."

For parallel members, the missing consistency relation can be written as

$$Z_\ell^* S_{\ell ij} = Z_k^* S_{kij}.$$

Without it, independently chosen branch flows can satisfy the balance equations while failing to arise from one common voltage drop. Adding the relation is a formulation repair, not a graph transformation.

Shared voltage coordinates do not remove member-specific constraints. Writing $W_i = |U_i|^2$ and $W_j = |U_j|^2$ in the exact lifted model gives

$$|I_{\ell ij}|^2 = |Y_\ell|^2 \left(W_i + W_j - 2\Re(W_{ij})\right).$$

Each retained member can therefore have its own current limit using the same voltage products. Member identity must remain in the parameters, constraint indexing, and recovery relations; it need not appear on every voltage variable. The same principle extends to fixed linear terminal-current maps, with the appropriate coefficients for taps and shunts.

Decision-model consequence

Sharing physical voltage coordinates is compatible with separate member limits. Discarding member data or constraints is a different operation. Introducing $W_{\ell ij}$ with equality constraints may be a redundant reformulation; allowing independent values can change a relaxation. The variable index alone does not establish equivalence or its failure.

For switching or outages, retain the state variables and conditional member laws as well. The fixed-state identity above does not establish preservation of those decision domains.

Terminal power is not series power

For a nominal-Π member, distinguish the series current from the current seen at each terminal:

$$I_{\ell ij}^{\mathrm{tot}} = \frac{I_{\ell ij}^s + I_{\ell ij}^{\mathrm{sh}}}{T_{\ell ij}^*}, \qquad I_{\ell ji}^{\mathrm{tot}} = I_{\ell ji}^s + I_{\ell ji}^{\mathrm{sh}}.$$

Consequently,

$$S_{\ell ij}^{\mathrm{tot}} = U_i (I_{\ell ij}^{\mathrm{tot}})^*, \qquad S_{\ell ij}^{s} = \frac{U_i}{T_{\ell ij}} (I_{\ell ij}^{s})^*.$$

An apparent-power limit placed at the terminal therefore constrains the total current, including shunt current and the ideal-transformer scaling. A limit on the series impedance current is a different observation. In a lossy branch, there is no single conserved scalar called “the flow on the edge.”

Implied current limits and relaxations

If a terminal apparent-power limit is $|S^{\mathrm{tot}}| \leq S^{\max}$ and $|U_i| \geq U_i^{\min} > 0$, then

$$|I_{\ell ij}^{\mathrm{tot}}| \leq \frac{S^{\max}}{U_i^{\min}}.$$

This is a valid implied current bound. In the exact AC model it cannot tighten the feasible set beyond the original apparent-power limit, but in a relaxation it can bind first and strengthen the relaxation. The distinction is important: an implied constraint is not a new nameplate rating, and a relaxation proof is not automatically a physical-network proof.

Four common overclaims

Tempting statement	What is actually established
“BIM and BFM are equivalent for this parallel network.”	Only after the variable correspondence and parallel-consistency constraints are stated; otherwise the relaxations may be incomparable.
“The branch has a current limit.”	Which current: series, sending terminal, receiving terminal, conductor total, or a recovered winding current?
“We can use one edge for the two lines.”	The aggregate terminal relation may be preserved, but member identity and member limits require a recovery map or a certified projection.
“The proof covers the power-flow model.”	It may cover only a fixed-parameter SOC relaxation, a selected projection, or a numerical test family.

The paper’s two-bus examples make these differences visible without a large network. They complement the book’s multiconductor parallel cases: the latter focus on terminal-current recovery and feasible-set preservation, while this case focuses on variable signatures, Π-section bound semantics and the boundary between physical and relaxation-level equivalence.

House notation for the book

The book will use the BMOPFTools-style convention consistently:

- ℓij is a stored oriented arc, not a claim about operating power flow;
- ℓ identifies the physical/model member;
- symmetric intrinsic parameters such as Z_ℓ carry only the member index;
- directional quantities such as $I_{\ell ij}$ and $S_{\ell ij}$ carry the full arc triple;

- total-terminal and series quantities receive distinct superscripts;
- a shared bus-pair quantity such as W_{ij} is used only when the theorem declares the sharing relation explicitly.

This is an expressive-notational rule: the symbols needed to state a constraint must be represented by its variables, indexed data, constraints, and recovery relations. A physical quantity may be derived rather than an independent variable.

Part 7: Evidence for a computation

Chapter 19

Numerical consequences

Page status: generated structural, linearized, symbolic KKT, and package-level solver-diagnostics crosswalk witnesses; solver-internal nonlinear exports remain future work.

A graph choice is also a numerical choice

Two representations can describe the same declared electrical behaviour while producing very different numerical problems. A quotient may reduce the number of variables but worsen scaling; a Kron reduction may preserve a boundary relation but introduce dense couplings; a factor model may expose physical structure while a bus–branch compiler produces a smaller, more familiar Jacobian. These are computational consequences, not afterthoughts.

The relevant comparison is therefore not only

same terminal map?

but also

same feasible observations, with what conditioning, sparsity, and recovery cost?

The statements below concern a declared operating point, coordinate system, and solver tolerance. They do not claim that one representation is uniformly better.

The federated PSK-`000011` guard makes one practical boundary executable: `reference_singularity_validation` compares mapped connected-island reference incidence and rank declarations before accepting a transformed model. A successful import or solver termination is not an independent certificate of full rank. The package pass remains declaration-level evidence; equation, Jacobian, conditioning, and solver-specific checks still belong to the study.

Vocabulary bridge

Electrical loss is power absorbed by a device model. Information loss is a failure of recovery under a representation map. An optimization or ML loss is an objective function. The shared word does not make these quantities comparable, and reducing one need not reduce either of the others.

Scaling and conditioning

Let a linearized or linear subproblem be

$$\mathbf{A}x = b.$$

Changing units or terminal coordinates is represented by nonsingular maps $x = \mathbf{D}_x\widetilde{x}$ and $b = \mathbf{D}_b\widetilde{b}$. The coefficient matrix becomes

$$\widetilde{\mathbf{A}} = \mathbf{D}_b^{-1}\mathbf{A}\mathbf{D}_x.$$

The coordinate map bijectively relates the two solution sets. For nonsingular $\mathbf{A}$ and an induced matrix norm, the condition number can change:

$$\kappa(\widetilde{\mathbf{A}}) = \|\mathbf{D}_b^{-1}\mathbf{A}\mathbf{D}_x\|\|\mathbf{D}_x^{-1}\mathbf{A}^{-1}\mathbf{D}_b\|.$$

This is why per-unit coordinates are a numerical convention as well as a reporting convention. A declared base scales voltage, current, power, and impedance together; it does not erase poor physical conditioning, singular topology, or an omitted conductor. A coordinate transformation is safe to call an exact rewrite only when its inverse and its power-dual action are retained, as in the coordinate-transformation chapter.

Circuit-theory trap

A dimensionless or per-unit matrix is not automatically well conditioned. The conditioning claim must name the norm, the bases, and the operating domain. A small residual in badly scaled coordinates can coexist with a large error in a physical voltage, current, or decision.

For a computed solution $\widehat{x}$ define the residual $r = b - \mathbf{A}\widehat{x}$. A backward-error measure is, for example,

$$\eta(\widehat{x}) = \frac{\|r\|}{\|\mathbf{A}\|\,\|\widehat{x}\| + \|b\|}.$$

For nonsingular $\mathbf{A}$, nonzero exact solution x, and the same vector norm and its induced matrix norm throughout, put $\theta = \kappa(\mathbf{A})\eta(\widehat{x})$. If $\theta < 1$, an explicit bound is

$$\frac{\|\widehat{x} - x\|}{\|x\|} \leq \frac{2\theta}{1-\theta}.$$

To see the factor of two, use $\widehat{x} - x = -\mathbf{A}^{-1}r$ and $\|b\| \leq \|\mathbf{A}\|\|x\|$. The relative error δ then satisfies $\delta \leq \theta(2+\delta)$. This bound concerns the stated linear system; it does not certify a nonlinear solution or the physical input data. The LAPACK error-analysis discussion also distinguishes normwise and componentwise backward error. Record both residual/backward-error information and conditioning, with their norms and scaling, rather than relying only on solver termination.

Choose a model against a constraint margin

The parallel-member lesson derives an exact aggregate for fixed data. Now suppose the total nonnegative current J is prescribed, both members are rated 100 A, the second conductance is 1 S, and the first is known only to lie in $[8, 12]$ S. The nominal first conductance is 10 S. These are declared interval assumptions for a resistive teaching case, not measured equipment uncertainty or a probability model.

For conductance g expressed numerically in siemens, current division is

$$I_1(J, g) = \frac{g}{g+1} J, \qquad I_2(J, g) = \frac{1}{g+1} J.$$

Compare three choices: solve the two nominal member relations; use their exact nominal aggregate cap of 110 A with recovery; or require both member limits for every conductance in the interval. The first two describe the same nominal feasible set. Neither settles the uncertain-data question.

Because I_1 increases with g and I_2 decreases, their worst values occur at opposite interval endpoints. Thus the exact robust cap in this local model is

$$J \leq \min\{100(13/12), 100(9)\} = 325/3 \text{ A}.$$

At 99 A, the nominal first-member current is 90 A; the worst member current is $1188/13$ A, leaving $112/13$ A of margin. At 109 A, the nominal model still accepts, but the worst member current is $1308/13$ A and exceeds its rating by $8/13$ A. The same uncertainty that was harmless for the first decision changes the second decision's feasibility classification.

A precise nominal computation cannot remove an uncertain input. Compare the possible observation error with the source constraint margin. Here the nominal aggregate is defensible for a stated nominal question; the interval requirement needs the robust cap. This is a proof over the full declared interval using monotonicity, not a claim based only on sampled scenarios. Other coupled or nonlinear models need their own worst-case argument.

```
python3 experiments/lessons/model_choice.py
python3 experiments/lessons/model_choice.py --check
python3 experiments/lessons/model_choice.py --benchmark
```

The first two commands use rational arithmetic. The optional local timing comparison uses floating-point arithmetic for nominal explicit, nominal aggregate, and interval-robust evaluations. Each measured call includes its cap/current calculation, limit checks, and current recovery; imports and I/O are excluded. It reports the median of seven batches of 10,000 calls. This is a tiny algebraic microbenchmark with no PF solver: use it to learn to count recovery work, not to rank network formulations. Timings are printed for the actual machine and are not stored as universal performance evidence.

Transfer exercise. Narrow the interval to $[9, 11]$ S. Derive the new robust cap and decide whether 109 A is admissible. The cap is $1200/11$ A, so it is admissible under this interval. State which changed assumption made that conclusion possible. Do not report it as evidence for the original $[8, 12]$ S interval.

Jacobians: physical topology is not matrix sparsity

For a nonlinear decision model $F(y, u) = 0$, Newton or interior-point methods use a Jacobian such as

$$\mathbf{J} = \frac{\partial F}{\partial y}.$$

The physical graph predicts some nonzeros, but the Jacobian graph is a graph of variable–equation dependence. A single multi-terminal factor can create a dense block between all of its ports. A nominal-π element can couple endpoint voltages through series terms and endpoint currents through shunts. Constraint rows for limits, controls, and recovery variables add edges that have no corresponding physical line. Conversely, a physical asset may be absent from a particular Jacobian block when its state is fixed or its equation has been eliminated.

Write G_phys for a chosen physical incidence graph and G_J for the bipartite Jacobian graph. In general there is no equality $G_phys = G_J$. The useful statement is a declared dependency map

$$\delta : (\text{factor}, \text{terminal}, \text{constraint}) \longmapsto (\text{equation rows}, \text{variables}),$$

from which G_J is generated. This map is the numerical analogue of source provenance: it explains why a fill-in or a dense block exists.

The five-bus structural witness makes the distinction concrete. Its source has seven line identities but only six edges after simple projection because q and r are parallel. Eliminating j then adds the i–l fill edge to the projected pattern. The fill argument and the dependency argument are shown separately so that a reader does not confuse a Schur-complement edge with a numerical derivative. The conceptual pair now appears in the formulation and lowering chapter; this chapter retains the executed witnesses and their fixture-specific counts.

The data and checks are recorded in `experiments/generated/numerical-structure-witness.json`; the renderer is `experiments/generate_numerical_structure_views.py`. This separation is intentional: both structural views can be checked without pretending that either is a solver-exported Jacobian.

Fill is not inevitable. The two-level topology chapter proves a positive structural case: dense two-terminal conductor stamps over a bus-level tree form a chordal scalar-support graph, and eliminating leaf-bus coordinate blocks inward is a perfect elimination order with zero symbolic fill. The warning here concerns arbitrary orderings, meshes, reductions, and factor libraries; it should not hide structure that a declared model makes exploitable.

The numerical witness now carries a crosswalk to `experiments/generated/five-bus-typed-kron-witness.json`. That crosswalk uses the non-pendant ℓ elimination, whose fill edges are $j-m$ and $k-m$, and records the small boundary residual together with the recovered u-branch limit observation. This ties the Schur-complement example to the numerical-structure discussion without conflating fill edges with physical assets or solver-private factorization output.

A pinned numerical export

The running fixture provides a second, numerical witness through BMOPFTools' passive and constant-Z linearized Y-bus exporters. In the exporter's node order, the passive matrix has 20 rows and 166 nonzeros. The constant-Z linearized matrix agrees with that passive matrix for this fixture. Realifying the complex current relation gives a 40-by-40 real matrix with 664 nonzeros. This is a coordinate embedding, not a promise that every complex invariant is preserved: the complex Ybus is transpose-symmetric to numerical precision, whereas the realified current-Jacobian embedding has a large transpose residual. That residual is expected for a nonzero imaginary part: the block matrix $\mathcal{R}(Y)$ with diagonal blocks $\Re Y$ and off-diagonal blocks $-\Im Y$ and $\Im Y$ is generally not symmetric. This is not a round-off failure. Realification preserves the coordinate support

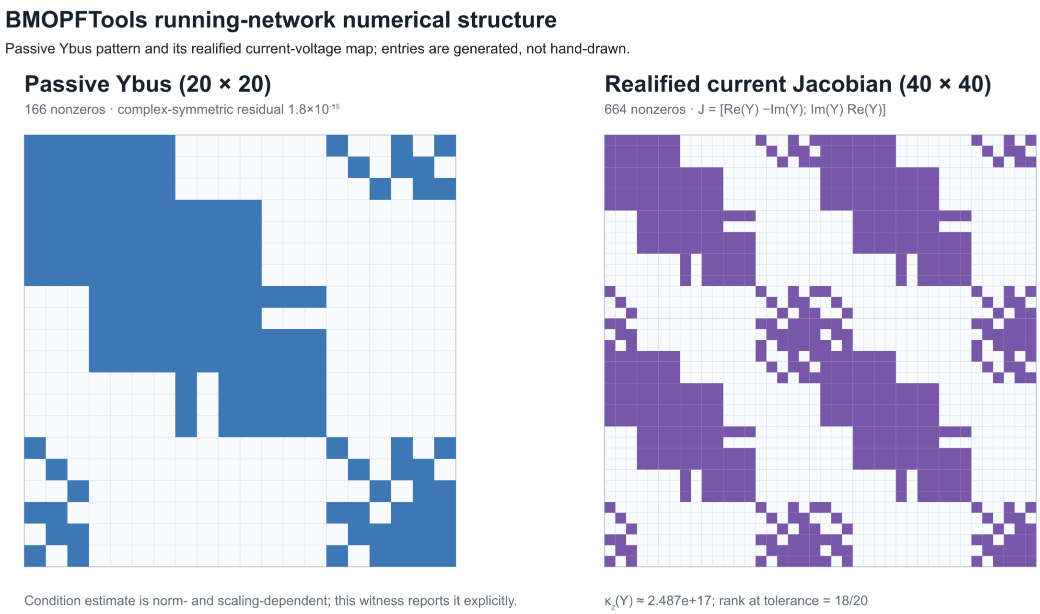

Figure 19.1: Generated passive Ybus and realified current-Jacobian patterns.

and the current relation, but symmetry claims must be stated in the representation in which they are checked (the witness residual is approximately 448.79):

$$\begin{bmatrix} \Re I \\ \Im I \end{bmatrix} = \begin{bmatrix} \Re Y & -\Im Y \\ \Im Y & \Re Y \end{bmatrix} \begin{bmatrix} \Re V \\ \Im V \end{bmatrix}.$$

The ordinary 2-norm condition estimates are approximately 2.49×10^{17} (unscaled) and 2.36×10^{16} (after simple row/column equilibration). Because both matrices are numerically rank deficient (rank 18 of 20 at the witness tolerance), those finite values should not be read as meaningful condition numbers: they are dominated by singular values below the modelling tolerance. The witness therefore also reports the rank-aware effective estimates $\kappa_{\text{eff}} = \sigma_1/\sigma_{18}$ and its equilibrated counterpart. These are diagnostic, not universal constants: they depend on units, scaling, norm, rank tolerance, and the chosen fixture. Pure row and column permutations preserve singular values, so they preserve the exact 2-norm condition number and a ratio of the same selected singular values. Ordering can change sparse-factorization fill, pivoting behavior, runtime, and floating-point estimation details; these are algorithmic effects, not a change in the exact spectral condition number. For this export the effective estimates are approximately 6.50×10^8 and 1.13×10^7 after equilibration. A solver must account for reference and grounding structure rather than treating any of these numbers as a graph invariant.

The complete export, including node order, nonzero entries, checks, and the linearization convention, is recorded in `experiments/generated/ybus-jacobian-witness.json`. Its generator and renderer are:

```
experiments/run_ybus_jacobian_witness.jl
experiments/render_ybus_jacobian_view.py
```

This is a pinned linear current/Jacobian witness; it is not yet a nonlinear OPF KKT Jacobian or an independent-solver comparison.

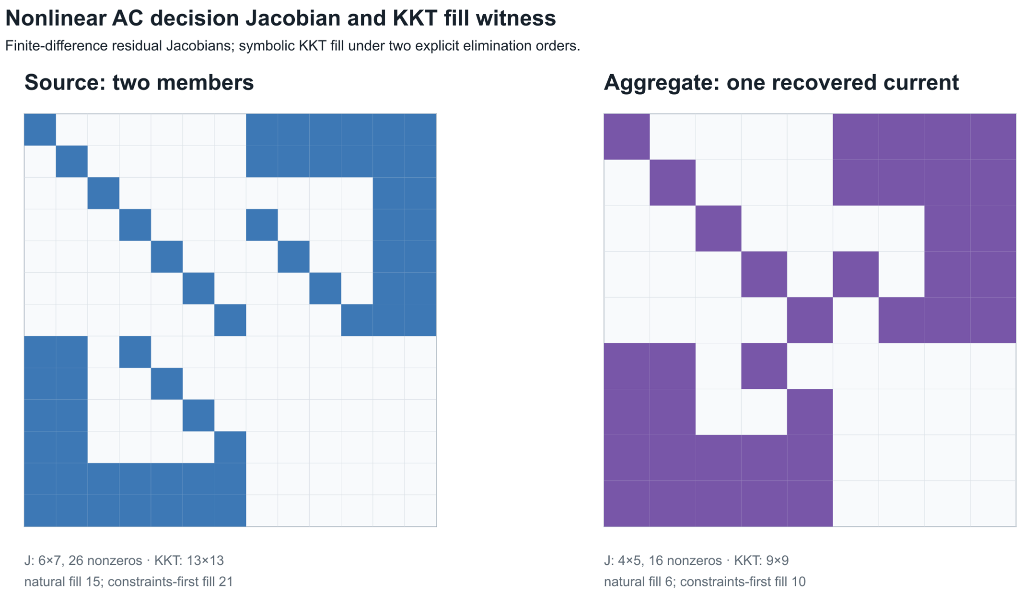

Figure 19.2: Nonlinear source and aggregate KKT sparsity and fill witness.

Nonlinear decision Jacobian and KKT fill

The next witness adds the nonlinear term that makes the parallel-line warning decision relevant. For a fixed complex endpoint voltage U and member currents I_1, I_2, the residual includes

$$S(U, I_1, I_2) - \alpha S_0 = U(I_1 + I_2)^* - \alpha S_0.$$

The source formulation retains two explicit current laws and two current vectors. The aggregate formulation replaces them by one summed current law. At the recorded operating point both residuals are exactly zero, but their finite-difference Jacobians and KKT patterns differ:

For avoidance of a common counting error, the source witness has two complex member-current coordinates, represented as four real variables, in addition to the three retained real coordinates. It therefore has seven real source variables; the aggregate witness has five real variables. The reduction removes two real variables (one complex current), not eight. The source KKT system has 13 rows and columns because it adds six residual/constraint multipliers; that KKT dimension is a different count from the number of primal variables.

formulation	residual Jacobian	KKT dimension	natural fill	constraints-first fill
two explicit members	6×7 (26 nonzeros)	13	15	21
summed member current	4×5 (16 nonzeros)	9	6	10

The fill counts are symbolic graph counts, not floating-point factorization times. They show why an ordering is part of a numerical certificate: the same KKT graph has different fill under different elimination orders. The source and aggregate models also have different graphs, so a smaller KKT system does not by itself establish decision equivalence.

The complete finite-difference and symbolic-fill artifact is `experiments/generated/nonlinear-kkt-witness.json`, regenerated by `experiments/run_nonlinear_kkt_witness.jl` and `experiments/render_nonlinear_kkt_view.py`.

It is deliberately labelled a nonlinear decision witness rather than a solver-internal Ipopt KKT export. A package-level crosswalk now binds it to the BMOPFTools Ybus witness, while actual solver-internal factorization diagnostics remain a separate boundary.

Package-level diagnostics crosswalk

The generated `solver-diagnostics-crosswalk.json` composes the two witnesses without pretending that either is a production solver export. It retains the physical node/terminal order, the BMOPFTools passive and constant-Z Ybus, the realified current Jacobian, the finite-difference nonlinear residual Jacobian, and the symbolic KKT graph under two declared orders. In the source formulation, for example, the natural and constraints-first orders produce different fill counts, so the crosswalk records ordering as part of the diagnostic identity.

The crosswalk also exercises the public `BMOPFTools.opf_checked_kkt_factorization` callback on a staged OPF context. The probe accepts a regular matrix and rejects a near-singular one. This is a checked callback boundary. A minimal parameterized OPF now passes that callback through DiffOpt: the forward sensitivity agrees with a central finite difference to the recorded tolerance, and the KKT diagnostic is accepted. The adapter also captures the matrix passed to the callback, recording its dimension and nonzero count, alongside the JuMP variable and constraint order used to account for its rows and columns. Four additional callback rows remain outside that declared JuMP block, so they are retained as an explicit DiffOpt/solver-internal boundary rather than silently assigned a physical label. The crosswalk still records `solver_internal_kkt_export = false` because solver-provided row labels, scaling, linear-solver choice, pivot or inertia data, and factorization statistics are not yet exported as a source/target comparison.

The same witness records BMOPFTools' differentiability report: inequality counts, active, near-active, weakly-active, and violated labels, minimum inactive slack, and qualifications. In this deliberately unconstrained fixture the inequality count is zero. The report's `ready` flag is state provenance for the selected local solve, not a proof of LICQ, strict complementarity, second-order sufficiency, global optimality, or branch stability.

The crosswalk now also builds a regular JuMP mirror and records its native affine/quadratic variable-support count alongside a separate `JuMP.NLPEvaluator` view. This gives an executable constraint-row/nonzero summary for the model-level derivative graph (23 supported variable entries in the current 19-row fixture), while leaving `native_nlp_export_is_solver_internal = false`: it is not Ipopt's private KKT ordering, pivoting, inertia, or factorization export.

For a deliberately narrow parallel-line fixture, the crosswalk also compares a two-member source against a single 0.25 Ω equivalent member. The tested voltage sensitivity is preserved, while the captured KKT matrix grows for the explicit source. The native JuMP support count changes as well, so the distinction is visible before any solver-private factorization. This is evidence that an exact electrical scalar equivalent can still change solver structure; it is not a general AC or multiconductor reduction certificate.

Numerical evidence boundary

The three numerical artifacts answer different questions and should not be collapsed into one solver claim:

An actual solver diagnostic would need to bind the exported derivative rows and columns to the source/target variable order, record scaling and tolerances, identify the linear solver and ordering, and retain the factorization or inertia report. Until that export exists, the chapter's KKT language is a symbolic comparison, not an implementation claim about a production solver.

Elimination and fill-in

Partition a sparse matrix by retained variables B and eliminated variables I. The Schur complement is

$$\mathbf{S} = \mathbf{A}_{BB} - \mathbf{A}_{BI}\mathbf{A}_{II}^{-1}\mathbf{A}_{IB}.$$

Artifact	What is measured	What it does not establish
structural witness	incidence, dependency, and symbolic fill edges	numerical derivative values or solver performance
Ybus/Jacobian witness	pinned passive and constant-Z matrix patterns, rank, and conditioning diagnostics	nonlinear OPF sensitivities, active-set stability, or factorization timings
nonlinear KKT witness	finite-difference residual structure and symbolic fill under two declared orders	Ipopt's internal derivative graph, linear-solver pivoting, or global optimality
solver-diagnostics crosswalk	shared node/order provenance, a checked-KKT callback probe, a DiffOpt sensitivity cross-check, and JuMP ordering metadata	an exported source/target solver KKT comparison, solver-native row labels, or ordering/factorization metadata

Even when $\mathbf{A}$ is sparse, $\mathbf{S}$ can be dense on the neighbours of I. In the symbolic graph, eliminating a vertex connects its remaining neighbours into a clique. The extra nonzeros are **fill-in**. The same mechanism appears in Kron reduction: preserving a boundary relation does not preserve sparsity, asset identity, or the cost of a subsequent solve.

The order of elimination matters. For a fixed matrix, two orders can have the same exact Schur complement but different intermediate fill, memory use, and factorization time. A certificate that reports a reduction should therefore record at least:

numerical field	required meaning
`ordering`	symbolic elimination or factorization order
`nnz_input`, `nnz_factor`, `nnz_fill`	structural nonzero counts under that order
`rank_guard`	rank/invertibility test for eliminated blocks
`conditioning`	estimate for the retained and eliminated solves
`recovery_cost`	work/storage needed to reconstruct omitted variables

These fields are properties of a compiled numerical problem, not of a simple graph in isolation.

Solver behaviour and decision margins

For a constrained decision problem, solver behaviour is part of the evidence but not the preservation theorem. A reduced model may converge faster because it has fewer variables, or slower because it has denser and more ill-conditioned blocks. A feasible point can also be numerically ambiguous when a constraint margin is comparable with the estimated forward error.

A termination label such as `LOCALLY_SOLVED` or `OPTIMAL` records what the algorithm reports under a particular solver interface and tolerance. It is not an independent recomputation of the returned primal point. A scientific use of that result must separately name and check the relevant evidence: finite numeric values, model equations, declared bounds, device and network residuals, recovery obligations, and the level of optimality actually supported. Missing checks remain missing even when the status string sounds successful.

This separation also works in the other direction. A status that does not claim feasibility is not evidence that a candidate primal point passed an independent validator, and a narrowly validated primal point does not upgrade a local solver result into a global optimum. Status, primal validation, and optimality evidence are distinct fields of a solution certificate.

For a scalar inequality $g_k(y,u) \le 0$, define the signed margin

$$m_k = -g_k(\widehat{y}, \widehat{u}).$$

If e_k is a declared bound or estimate for the error in that constraint, classify the result as:

$$\begin{array}{ll} \text{certified feasible} & m_k > e_k, \\ \text{numerically ambiguous} & |m_k| \le e_k, \\ \text{certified violated} & m_k < -e_k. \end{array}$$

This is deliberately stronger than reporting $g_k \le 0$ at printed precision. The same classification applies to decision comparisons: if two objectives or tap choices differ by less than the propagated numerical error, the certificate should not claim a unique optimum.

Decision-model consequence

A reduction that preserves a boundary voltage to tolerance may still change the active member limit, the selected switch state, or the winning discrete decision. Record the decision margin and the recovery error for every claimed decision-preservation result.

What a transformation certificate should expose

The numerical fields extend the general preservation contract:

$$\mathcal{N} = (\text{coordinates}, \text{scaling}, \kappa, \eta, \text{dependency graph}, \text{ordering}, \text{fill}, \text{recovery cost}, \text{decision margins}).$$

An exact electrical map may therefore be computationally unattractive, while an approximate map may be useful if its error and decision domain are explicit. The running network's six generated views should eventually carry this record alongside their source maps; the current fixture records the source maps and solver outcomes but does not yet claim a complete cross-solver conditioning study.

Scope and open work

This chapter gives definitions and reporting requirements, not a universal solver benchmark. The next executable tranche should compare the running multiconductor network and its five-bus quotient under a pinned ordering and coordinate convention, report Y/Jacobian sparsity and fill-in, and test whether recovered current and rating margins remain decisive after reduction.

Chapter 20

Australian construction reproduction

Page status: source-backed Carson/OpenDSSDirect reproduction; overhead and underground reference matrices remain independently compared outputs, and the CS1035 construction mapping is explicitly unresolved.

This case is a provenance check, not a transcription of the matrices already stored in the source repository. We lift the construction data that is available in the `ImpedanceModels.jl` line-library history, regenerate the multiconductor primitive with BMOPFTools' Carson implementation, and then solve a small four-wire OpenDSS circuit with OpenDSSDirect.jl.

What is lifted

The reproducible input record is `experiments/data/australian_source_inputs.toml`. It records the source commit, the 50 Hz modified-Carson setting, and the construction fields:

- the Pluto overhead wire diameter, AC resistance, four conductor positions, and 0.2 km length;
- the available UGHV 185 Al PILC four-wire construction fixture, its wire diameter and AC resistance, its four positions, and 0.1 km length.

The generated line code is assembled in memory from those fields. No `rmatrix`, `xmatrix`, or `cmatrix` from an existing `.dss` case is used to construct the model. Each OpenDSS load explicitly sets `vminpu=0` and `vmaxpu=2`, so the reproduction does not inherit a hidden load-model voltage clamp.

The companion register `experiments/data/australian_source_audit.toml` keeps provenance judgements separate from those inputs. Each case records whether a field is lifted, derived, inferred, or unresolved, together with machine-readable finding codes. In particular, the overhead 60 Hz/order alignment is an inference from the probe, not a statement found in the source file; the underground `CS1035` mapping remains unresolved.

The generated artifact also carries the versioned `power-network-impedance` contract from `experiments/data/impedance_contr`
It requires ordered terminals, series and shunt blocks, units, ampacity limits, grounding assumptions, source lineage, named derived views, and findings. The validator is deliberately small: it checks that a record is structurally safe to exchange, while the audit register carries the domain-specific judgements that no generic schema can infer.

What is compared, but not used

The files under `data/opendss/Australian_overhead` and `data/opendss/Australian_underground` contain derived reference matrices. We parse those matrices after the new solve and report the maximum complex-series error in the generated artifact `experiments/generated/australian-carson-reproduction.json`. The artifact marks these matrices with `matrix_used_as_input = false`.

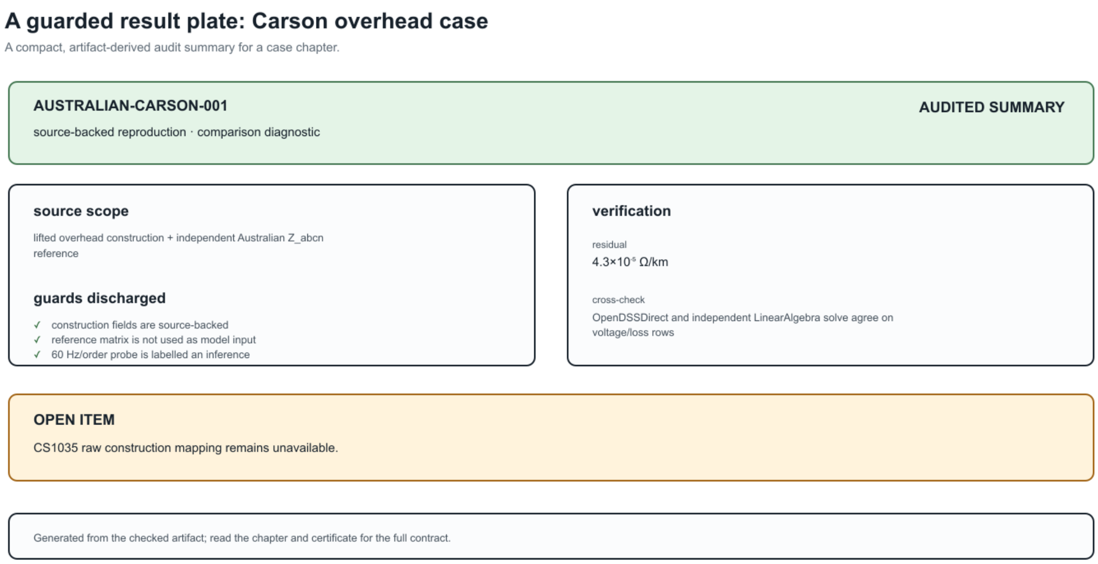

Figure 20.1: A generated guarded-result plate makes the Carson claim, discharged guards, residual, cross-check, and open item visible without replacing the certificate.

The overhead source geometry in the line-library history regenerates a series matrix with approximately $R_{ll} = 0.21735\,\Omega/\mathrm{km}$ and $X_{ll} = 0.73853\,\Omega/\mathrm{km}$. The Australian `Z_{abcn}` reference file instead has approximately $0.22722 + j0.87933\,\Omega/\mathrm{km}$ on its diagonal. A frequency/order probe resolves this apparent mismatch: at 60 Hz, and with the source conductor order $[4, 1, 2, 3]$ rather than the lifted primitive order $[1, 2, 3, 4]$, the maximum complex-series error falls to about $4.3 \times 10^{-5}\,\Omega/\mathrm{km}$. The generated artifact retains the source-backed 50 Hz result as its primary comparison and records this 60 Hz/order diagnosis separately; the reference file itself does not declare that frequency in the available source data.

The underground `CS1035` file is handled more cautiously. Its active 4-by-4 matrix is available for comparison, but the source repository does not provide the raw cable construction mapping for that case. Changing the Carson probe from 50 Hz to 60 Hz does not reduce the mismatch (the maximum error increases slightly), so frequency is not a sufficient explanation. We therefore label the UGHV construction as an explicit fixture, keep the CS1035 matrix independent, and leave the mapping as an open data task.

The source `StandardLinecodes.dss` also warns that the underground geometry's negative physical heights cannot be passed directly to OpenDSS; the reproduction uses the documented positive reference-plane convention and records that choice in the construction code rather than silently treating it as a measured cable depth.

Reproduce

From the repository root:

```
julia --project=experiments experiments/run_australian_carson_reproduction.jl
```

The JSON records the construction inputs, generated series and capacitance matrices, OpenDSS commands, convergence diagnostics, bus voltages, losses, the independent-reference comparison, and a status for each cross-check. A second LinearAlgebra-only constant-power solve agrees with OpenDSS on all rows when voltage

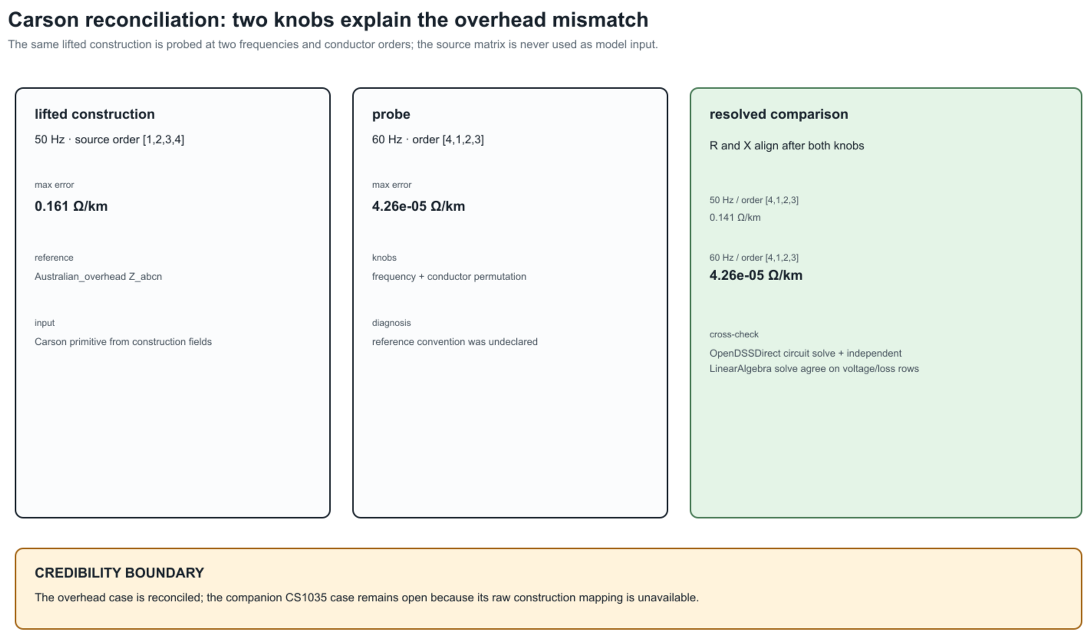

Figure 20.2: The overhead mismatch is reconciled by frequency and conductor-order probes, while the independent CS1035 matrix remains outside the reproduced model.

and line losses are compared (the voltage error is below 10^{-6} V and the line-loss error below 10^{-3} VA in the current fixture). OpenDSS `Circuit.Losses()` reports the line-element loss but not the separately modelled grounding-reactor loss; the artifact therefore keeps both line-loss and total-loss comparisons. The embedded OpenDSS engine is left at its stable 60 Hz default; the Carson primitive follows the 50 Hz source-library setting. This distinction is explicit in the artifact rather than hidden behind a text-level `set frequency` command, which is unstable in the current OpenDSSDirect release.

The underground fixture also includes low and high grounding-impedance rows. They keep the construction and load row fixed while changing only the grounding factor, making the neutral-voltage and grounding-loss sensitivity observable rather than an implicit assumption. Both rows remain within the same independent line-loss cross-check contract.

The result is therefore useful in two ways: it verifies that the available construction data can be regenerated end to end, and it identifies exactly which published cases still need their original construction provenance before they can be claimed as faithful reproductions.

Reconciliation and the unresolved companion case

The two knobs are shown together because either one alone leaves a substantial residual. They are diagnostic probes, not retroactive source metadata: the artifact records the 60 Hz/order alignment as an inference.

This is a deliberate credibility boundary. The available fixture fields and the independent matrix are useful, but without the raw conductor/screen geometry, earth-return convention, frequency, and ordering, the construction map cannot be reconstructed faithfully.

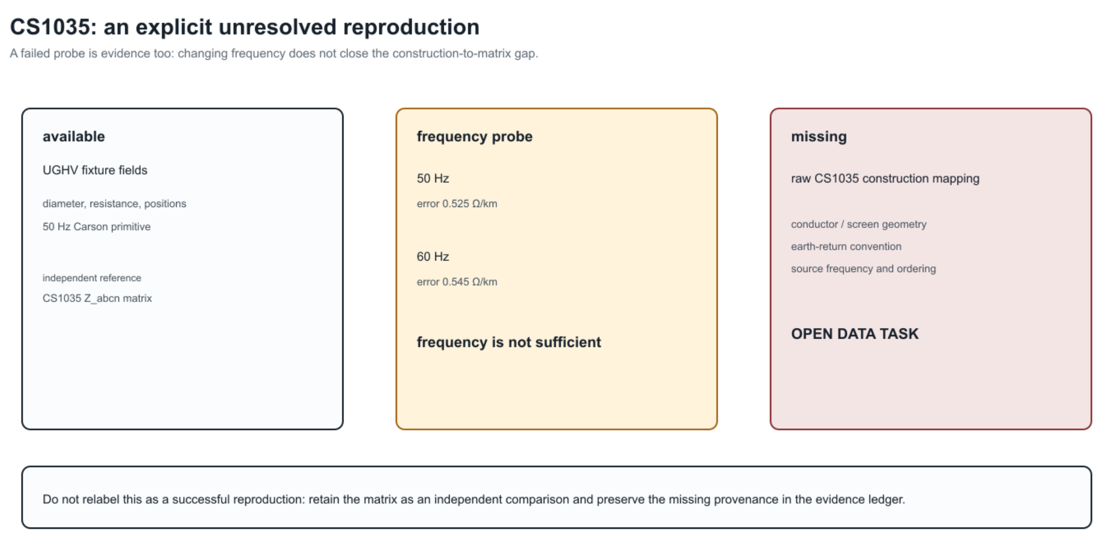

Figure 20.3: CS1035 is retained as an explicit unresolved data task rather than being labelled a successful reproduction.

Chapter 21

Computational case guide

Page status: teaching guide to existing evidence and the scalar opening lesson.

Choose a case by the question you want to answer. Commands below run from the repository root. Start with the scalar lesson; it needs only Python 3 and prints its result without creating files. The Julia studies use the pinned `experiments` environment and may regenerate their recorded artifacts.

First calculation: parallel members

```
python3 experiments/lessons/parallel_members.py
python3 experiments/lessons/parallel_members.py --check
```

Expect a passing 165 A aggregate check, a failing 150 A first-member check, and a derived 110 A aggregate cap. The opening lesson derives every number. Change the voltage drop, first-member rating, or open-member state and predict the result before running it. This is exact arithmetic for a declared resistive circuit.

Assemble a model, then choose its scope

```
python3 experiments/lessons/assemble_network.py --check
python3 experiments/lessons/assemble_network.py --misattach-load
python3 experiments/lessons/model_choice.py --check
python3 experiments/lessons/model_choice.py --benchmark
```

The assembly lesson derives the matrix and checks recovered equipment currents. The deliberate mapping error solves its own equations but fails the original circuit checks. The model-choice exercise compares nominal exactness with an interval-robust current limit and measures local evaluation cost with recovery. These commands print results without changing the evidence collection.

Import, edit, and interpret a model

The practical modelling chapter works through three failures: a rating sentinel misread as a physical zero, a star reduction updated by deleting the wrong derived edges, and scaled multipliers reported as physical prices. Derive the expected behavior, run the deliberately faulty alternatives, then use the source-semantic checks to distinguish them:

```
python3 experiments/lessons/practical_model_checks.py
python3 experiments/lessons/practical_model_checks.py --check
```

The 15 test methods include unknown and unrepresentable fields, stale reductions, linear basis checks, objective/constraint scaling, and nonunique multipliers. The field adapter is not a complete MATPOWER importer; the dispatch example is an analytically solved scalar LP, not an OPF solve. No files are written.

Five further investigations

The coordinate and Kron commands run checks; the case generators update their artifacts. Read the associated source and target assumptions before interpreting an accepted transformation. An inapplicable result can correctly identify that a requested rule does not cover the input.

Eliminate two series sections

Read series elimination, then predict which assumptions permit elimination and which quantities need recovery.

```
julia --project=experiments experiments/run_series_elimination.jl
```

Inspect `experiments/generated/degree-two-series-certificate.json` for the guards, recovered quantities, and coupled near-miss.

Relabel conductors

Read conductor normalization. Predict which arrays must change together under a conductor permutation.

```
julia --project=experiments experiments/test/coordinate_normalization.jl
```

Inspect the test output and assertions for accepted coordinate actions and rejected inconsistencies. A relabelling should not change the declared physics.

Recover a hidden current violation

Read Kron reduction. Decide whether an exact boundary relation alone settles an internal neutral-current constraint.

```
julia --project=experiments experiments/test/running_network_typed_kron.jl
```

Inspect the boundary residual assertions and recovered neutral-limit witness.

Put the parallel error in an AC network

Read the parallel AC case. Predict whether retaining a terminal equation also retains member feasibility.

```
julia --project=experiments experiments/run_multiconductor_parallel_ac.jl
```

Compare the source, naive, lifted, and pruned results in this artifact:

```
experiments/generated/multiconductor-parallel-ac-certificate.json
```

Change the transformer tap domain

Read the transformer tap case. Predict what a fixed tap snapshot can establish about an adjustable study.

```
julia --project=experiments experiments/run_transformer_tap_ac_decision.jl
```

Inspect the tap domain, local solve, recovery, and objective in this artifact:

```
experiments/generated/transformer-tap-ac-decision-certificate.json
```

Prepare the larger studies

The recorded review-case workflow creates an isolated environment and a fresh output directory:

```
bash scripts/reproduce_clean_fixture.sh --check
bash scripts/reproduce_clean_fixture.sh
```

It pins the recorded Julia/package combination and runs the scoped returned- solution checks. It preserves the earlier historical evidence. For other maintainer case generators above, instantiate the `experiments` environment with the sibling `BMOPFTools.jl` checkout and check their paired identity:

```
julia --project=experiments -e 'using Pkg; Pkg.instantiate()'
python3 scripts/check_federated_knowledge.py --check --bmopf-root ../BMOPFTools.jl
```

These direct generators may update tracked artifacts; use a separate checkout when investigating them. A passing pair check binds the current knowledge exports, not every historical experiment. The specialized construction study has its own Australian Carson reproduction boundaries.

Record the result you can defend

For each run, record the input revision, model assumptions, command, numerical method and tolerances, source/target quantities, and source checks after recovery. State whether the evidence is an identity, a numerical witness, or a local solver result. Explain the next question it leaves unanswered.

For example, a passing coordinate permutation checks consistency of a model under relabelling. It does not establish that the original conductor data were correct. Agreement between two solution methods strengthens evidence for the specified equations; shared construction assumptions still need scrutiny.

The study workbook turns these records into a complete exercise. The knowledge-base index provides the full artifact and claim inventory when you need more detail.

Read independence at the level of the calculation

The scalar assembly uses a separate equipment evaluator, but shares the conductances and circuit laws with its stamp builder. The transformer tap comparison changes the numerical algorithm while sharing primitive matrices and case assembly. The balanced transmission comparison uses a separate Gaussian-elimination implementation on the declared common fixture. Each check challenges a different possible error; none authenticates the inputs merely by agreeing numerically.

The running-case verification adds post-solve line-current recovery through package-owned primitives. Its complete-feasibility gate remains indeterminate. The Australian construction study separately records unresolved provenance of an external reference matrix. Inspect those boundaries before treating another successful run as independent support for a physical-model claim. Independent human review is a further evidence dimension, with no promoted claims here.

Reproduce the teaching diagrams

The numerical diagrams read the standard-library lessons; the redundancy plate reads the maintained certificate files. From the repository root:

```
python3 experiments/render_teaching_figures.py
python3 experiments/render_parallel_certificate_geometry.py
python3 scripts/test_teaching_figures.py
```

Rendering requires `rsvg-convert`; the checks use only Python's standard library. Four regression tests check current SVG generation, the parallel witness and circle scales, the certificate circle scale, and served-fraction bar ratios. They do not replace scientific or visual review. The SVGs and PNG companions are maintained together; colour is supplemented by labels, outlines and line styles.

Part 8: An end-to-end modelling study

Chapter 22

Building and changing a model you can check

Page status: three exact executable teaching witnesses with deliberate failures and transfer exercises; not a production importer, incremental network compiler, or OPF price validator.

A colleague sends a network file, asks for one outage, and wants the marginal cost of serving more demand. Each request looks routine. Each crosses a boundary where software can change the mathematical problem without producing an error. This chapter works through those boundaries using three small models, so every expected result can be derived before running code.

The assembly lesson supplies the method for constructing and checking equations. Here the task is to keep their meaning intact during import, modification, and interpretation. Run all three examples from the repository root; they require only standard-library Python and write no files:

```
python3 experiments/lessons/practical_model_checks.py
python3 experiments/lessons/practical_model_checks.py --check
```

Import meaning, not just numbers

Suppose a source record contains a branch rating of zero. Predict the result of testing a 10 MVA transfer: should it pass, fail, or remain undecided?

There is no answer until the field convention is known. In MATPOWER's version-2 case format, `RATE_A=0` specifies an unlimited rating. `TAP=0` indicates a line, whose ratio is mathematically unity, rather than a transformer of zero ratio. These are documented encodings, not defects [27].

The lesson's small internal rating type separates four states:

Internal state	Meaning in this example	Result at 10 MVA
finite, 0 MVA	impose a zero magnitude bound	failed
unbounded	impose no bound from this rating field	passed
unknown	no rating value or explicit unbounded declaration available	indeterminate
not applicable	this particular rating check does not apply	not applicable

These are outcomes of one rating check, not a feasibility verdict on the network. The source format still requires its fields; representing an absent field as unknown is a way to retain incomplete input for diagnosis, not to declare it a valid MATPOWER case.

The faulty adapter copies `RATE_A=0` into an internal finite zero bound. It accepts the file, retains the number and units, and rejects the transfer. The correct field adapter decodes the sentinel and passes this particular check. Neither result depends on solving power flow.

A numerical round trip does not expose the faulty adapter: a mistaken reader and writer can reproduce the same zero while imposing the wrong constraint in between. Test the decoded *behavior* against the source convention. Claim `PRACTICE-IMPORT-001` records this narrow counterexample.

The reverse conversion also has limits. A true finite zero bound cannot be written as `RATE_A=0` without changing its meaning. The lesson refuses to encode that state, an unknown rating, or a not-applicable state into this single field. A production exporter needs an explicitly supported alternative representation or a clear refusal. It must not silently replace them with unlimited capacity.

For TAP, preserve both the effective ratio and the source kind: a line encoded by zero and a unity-ratio transformer encoded by one can share a ratio without having identical equipment semantics. Other transformer fields and conventions are outside this field-level exercise.

Change the input. Set `RATE_A=10`, then test transfers of 10 and 11 MVA. Remove the rating field rather than setting it to zero. Explain the three results before executing the checks. The finite rating passes at its bound, fails above it, and the missing field remains unknown. No inference has supplied an equipment rating.

Rebuild after an edit before attempting reuse

Consider three resistive arms connecting boundary terminals a, b, c to an internal node n. All three conductances are 1 S. There is no current injection or shunt at n. Currents are positive from each boundary terminal toward the center.

For nonnegative arm conductances g_a, g_b, g_c with positive sum s, KCL gives

$$U_n = \frac{g_a U_a + g_b U_b + g_c U_c}{s}, \qquad I_k = g_k (U_k - U_n), \qquad s = g_a + g_b + g_c.$$

Eliminating the center therefore gives the exact boundary operator

$$\mathbf{K} = \operatorname{diag}(g_a, g_b, g_c) - \frac{1}{s}\mathbf{g}\mathbf{g}^{\mathsf{T}}, \qquad \mathbf{g} = (g_a, g_b, g_c)^{\mathsf{T}}.$$

For the three equal arms this is a triangle of conductance $1/3$ S on each edge. Predict the new boundary operator when the *source arm* $n - c$ is opened.

Opening that arm sets $g_c = 0$. It leaves a series path from a to b with conductance $1/2$ S, and an isolated terminal c:

$$\mathbf{K}_{\text{open}} = \begin{bmatrix} 1/2 & -1/2 & 0 \\ -1/2 & 1/2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ S}.$$

A tempting update deletes the two triangle edges incident to c. That leaves conductance $1/3$ S between a and b. The updated graph has the expected isolated terminal, a symmetric operator, and zero row sums, yet it is wrong. At boundary voltages $(1, 0, 0)$ V, the correct injection at a is $1/2$ A; the faulty update gives $1/3$ A. Both boundary-current vectors sum to zero, so an aggregate current-balance check cannot discriminate them.

Recover $U_n = 1/2$ V from the edited source, then evaluate the original arm laws to expose the mismatch. Claim `PRACTICE-UPDATE-001` concerns this specific failed update. It does not assert that reductions can never be updated incrementally: an update using sufficient source information can be correct.

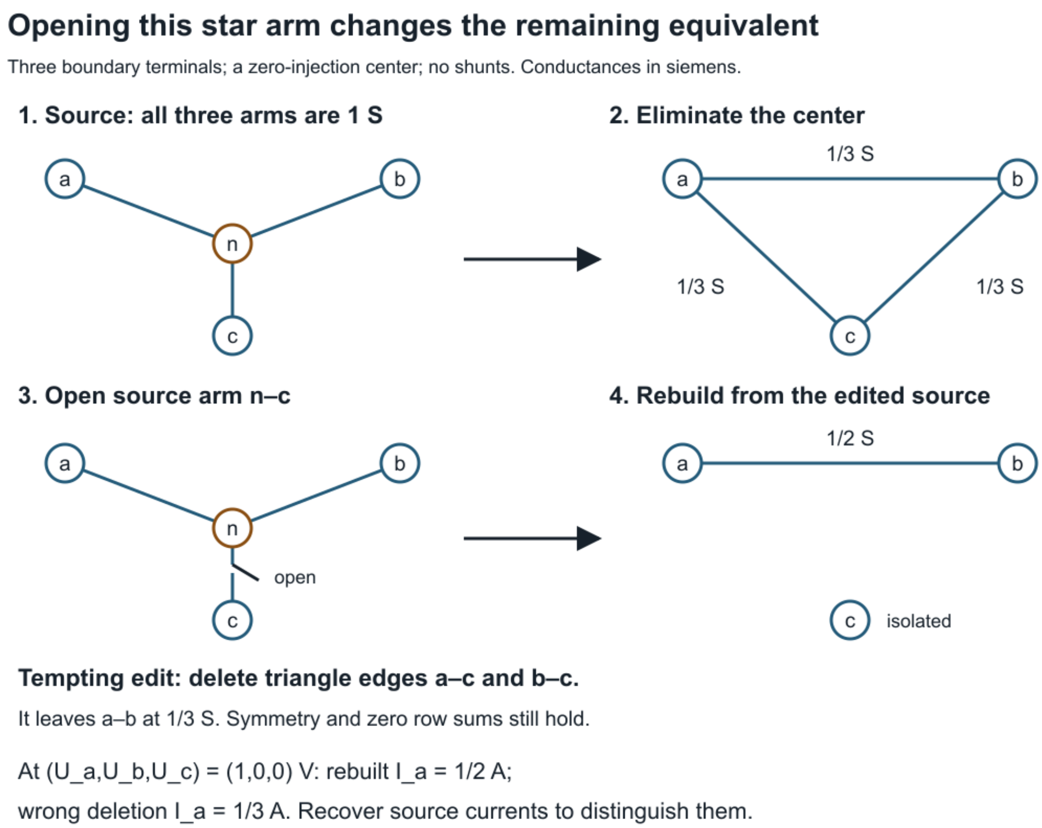

Figure 22.1: Opening star arm n–c changes the correct a–b equivalent to 1/2 S; deleting triangle edges instead leaves the wrong 1/3 S.

The code keeps the source conductances with each compiled operator. Using the old operator with changed conductances raises a stale-reduction error. Rebuilding from the changed source restores agreement. The deliberately faulty update even attaches the new source identity to the wrong matrix: correct metadata alone does not establish that the equations were updated correctly.

For each tested source, the checker compares the matrix with a separate arm-law evaluation on all three boundary basis vectors. Because both maps are linear, equality on that basis establishes equality throughout their boundary-voltage space in exact arithmetic. This conclusion relies on the declared resistive model, linearity and recovered center equation; it does not authenticate the conductance data or establish a general nonlinear update method.

Change the input. After opening $n - c$, change g_b to 2 S. Derive the remaining series conductance before rebuilding. It is $2/3$ S. Then open all three arms. The displayed inverse formula is inapplicable because $s = 0$; the center voltage is undetermined, although the boundary current map can still be represented as zero. The lesson rejects that input rather than dividing by zero or claiming that the circuit cannot be represented at all.

Recover the meaning of a multiplier

Use a separate scalar dispatch model to isolate the interpretation issue. Let p be supplied power in MW, $d > 0$ a demand requirement, and $c > 0$ a constant cost in currency/MWh. There is no upper capacity limit in this example:

$$\min_{p\in\mathbb{R}} cp \quad \text{subject to} \quad d - p \le 0.$$

The unique optimum is $p^* = d$ and the cost rate is cd in currency/hour. Using the displayed inequality convention, the Lagrangian is $cp + \lambda(d - p)$. Stationarity gives $\lambda = c$.

Now multiply the constraint by $\alpha > 0$ and the objective by $\beta > 0$. These are explicit positive numerical scaling factors:

$$L' = \beta cp + \lambda'\alpha(d - p), \qquad \lambda' = \frac{\beta c}{\alpha}.$$

Dispatch is unchanged, but the raw multiplier is not. The derivative of the *unscaled physical cost rate* $C(d) = cd$ with respect to physical demand is

$$C'(d) = c = \frac{\alpha}{\beta}\lambda'.$$

For $c = 50$ currency/MWh and $d = 10$ MW, changing α from 1 to 100 while retaining $\beta = 1$ changes the raw multiplier from 50 to $1/2$. Both cases supply 10 MW at 500 currency/hour; both have marginal physical cost 50 currency/MWh. Reporting the raw $1/2$ as that price would be an interpretation error.

This is a direct perturbation calculation, consistent with the duality and sensitivity treatment in [28]. It fixes the inequality sign and objective convention explicitly; a solver interface using another convention requires its own mapping. It is not an AC-OPF locational-price certificate. Claim `PRACTICE-DUAL-001` records the scalar scaling and nonuniqueness example.

Duplicate the original unscaled constraint. Stationarity now requires $\lambda_1 + \lambda_2 = c$ with both multipliers nonnegative. The allocations $(50, 0)$, $(0, 50)$ and $(20, 30)$ all satisfy the KKT conditions at the same optimum. Individual constraint multipliers are not unique. If demand changes in both copies together, its cost sensitivity is their sum. Changing only one copy is a different perturbation question.

The left panel plots the analytical multiplier against α on a logarithmic horizontal axis with $\beta = 1$. The right panel concerns the duplicated, unscaled model. Its entire segment satisfies KKT at $p = d$; the three marked points are examples, not three unique solver outputs.

Change the scaling. Set $\alpha = 100$ and $\beta = 1/1000$. The raw multiplier becomes $1/2000$; mapping back still gives 50 currency/MWh. Check this by increasing demand by $1/10$ MW: the physical cost rate increases by 5 currency/hour. A negative constraint scale would reverse the inequality unless the formulation were changed; it is outside the stated rule.

Organize the implementation around these checks

Keep source records separate from generated equations and results. A source record identifies the equipment and its declared state. A compiled object identifies the assumptions and source values used to construct it. A result identifies the compiled object that was actually solved. A change to one does not automatically update the others.

Start with a rebuild path whose output can be checked. Add reuse only after specifying which changes preserve the cached object and which invalidate it. For a larger network, an invalidation key may need terminal ordering, topology, device parameters, operating point, formulation options and implementation version. The three conductances are sufficient only for this fixed star API.

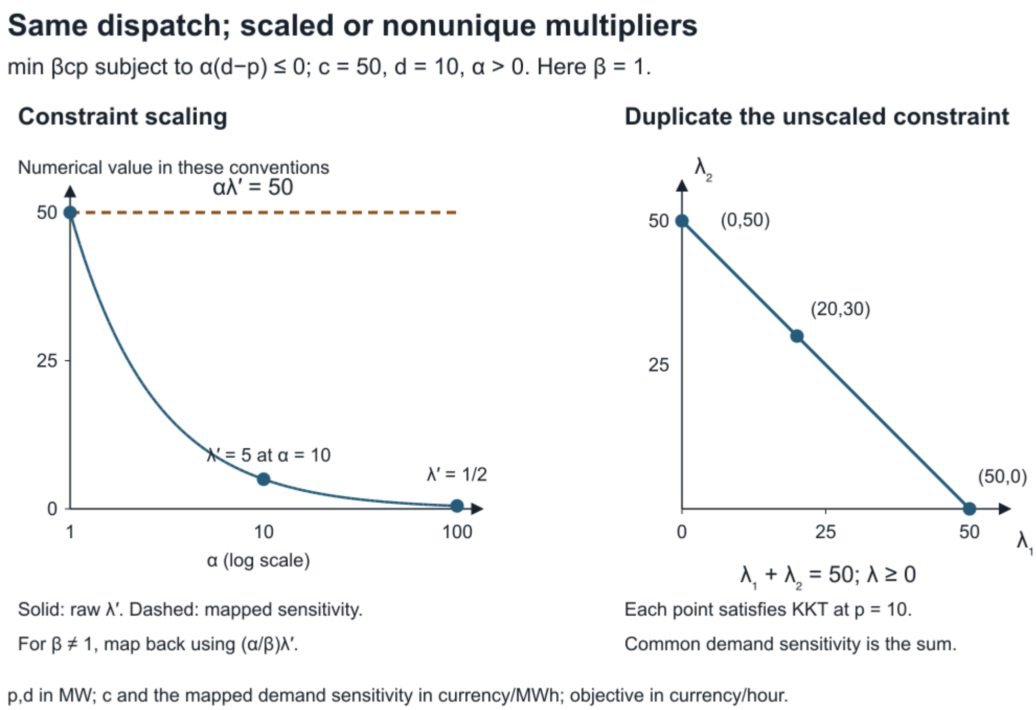

Figure 22.2: Positive constraint scaling changes raw multipliers but not mapped physical sensitivity; duplicated constraints admit a line segment of multiplier allocations.

This is the question of *from-scratch consistency* in incremental computation: does reusing work after an input change agree with recomputing the specified computation [29]? For numerical solvers, compare mapped observations and declared tolerances rather than demanding identical iterates.

Likewise, translating a source network into equations resembles a compiler pass. Translation validation checks a particular source/target pair, with its own checker and soundness obligations [30]. The book's finite witnesses use that idea; they are not formally verified compilers. Sharing a primitive formula or mistaken input between builder and checker can leave a common error undetected.

Use the following sequence when a colleague reports a disagreement:

1. **Identify the question.** Is the disagreement about a field, equation, feasible state, recovered observation, objective, or sensitivity?

2. **Keep the source.** Record the exact input and edit; do not repair the only copy of the failing case in place.

3. **Rebuild and recover.** Compare fresh and reused computations, then evaluate the original quantities needed by the study.

4. **Change a diagnostic input.** Use an independent boundary excitation, a limit-crossing value, or a specified perturbation. A test that both wrong and right models pass does not resolve their disagreement.

5. **Minimize the failure.** Remove unnecessary equipment while retaining the assumptions and failure criterion; keep that case as a regression test.

Submit a small, reviewable correction

For one example, submit the original input, the proposed edit, your predicted result, the faulty result, and the check that distinguishes them. State a plausible test that would *not* detect the problem. Finally, state what your correction does not establish.

A satisfactory import correction distinguishes the explicit sentinel from missing data and refuses an unrepresentable export. A satisfactory update correction recovers the edited source currents, including the asymmetric transfer case. A satisfactory dual correction names the perturbation, scaling and sign convention and handles the duplicated constraints without inventing unique individual prices. These are separate scientific obligations even when all three implementations return numbers successfully.

Chapter 23

Study workbook

Page status: draft capstone workbook over the existing running fixture; no new solved application is claimed.

Your task is to explain and defend a computational model, including one transformation and its recovery checks. Use the existing multiconductor running fixture. Its difficulty is deliberate: it combines conductor connections, grounding, parallel equipment, and a multiwinding transformer. You are now using these features together after studying smaller examples.

Before extending this case, complete one correction from Building and changing a model you can check. Its import, outage-update and multiplier examples show how to choose a discriminating check and state what remains unverified.

1. Declare the study

Read The running multiconductor network and Executable running network. State the operating state, the quantities the study observes, and the decision domain actually solved by the fixture. Distinguish fixed data from optimized variables. List at least one question the fixture does not answer.

Use the existing PF/OPF study as your starting task. Introducing a new contingency, estimator, or protection calculation would require its own model and evidence; those are possible extensions after this exercise.

2. Trace equipment into equations

Choose one line, one grounding relation, and one transformer winding. For each, record its stable identity, ordered terminals, units, and the equations and constraints that use its quantities. Locate its contribution in a generated view. Explain any virtual node or arc encountered along that path.

Draw a small part of the equipment topology and the corresponding matrix support. Identify an adjacency whose meaning changes between those views. Use the source maps supplied with the fixture rather than guessing equipment identity from a matrix entry.

3. Predict a transformation

Choose a transformation already supported by a listed case in the computational guide. State its source, target, applicability guards, joint observations, and recovery map. Predict which constraints need evaluation on recovered quantities.

Do not assume that a transformation tested on a separate small fixture is applicable to the complete running network. Either establish the guards for its chosen subsystem, or keep the small fixture as the transformation study and explicitly record that boundary. A justified rejection is a valid result.

4. Reproduce and recover

Follow the execution command and environment in the associated case chapter. Record the actual software revisions, inputs, solver status where applicable, residuals, and tolerances. Compare source and target observations. Recover the eliminated quantities and evaluate the source constraints that depend on them.

Run the returned-solution verification exercise. Record the four line-current residuals, the package profile's findings, and the reaction to the altered `i2.a` voltage. Explain which source quantities were recomputed and which data or primitive construction were shared. The full feasibility-evidence gate should return `indeterminate`; explain the missing all-device equation and nodal KCL evidence instead of converting that refusal into a passing result.

If the case is locally solved, describe it as a local numerical comparison. Use a derivation or a separate bound for any stronger feasible-set or optimality statement. Explain which construction data and algorithms are shared by any independent comparison.

5. Challenge the conclusion

Propose one nearby input change that violates a guard: an internal shunt, a finite grounding impedance, a changed conductor order, or a changed control domain. Predict whether it should cause rejection or require a different rule. Run the corresponding existing negative test where available and record its actual result. Keep a proposed extension distinct from one you have executed.

Submit a short scientific account

Your account should contain the study question; source assumptions and input identity; one diagram; the transformation and recovery equations; a compact comparison of predicted and observed quantities; the failed near-miss; and a precise conclusion. Include the commands needed to reproduce the work.

A satisfactory account allows another reader to identify what was checked, where the original constraints enter, and why the conclusion stops at its stated boundary. A small residual alone is insufficient. If data, applicability, or recovery remain unresolved, explain the missing evidence and the next calculation that would resolve it.

Worked assessment of the verification step

A defensible account reports that the unaltered case satisfies the lesson's line-current and power-balance tolerances, while the altered voltage violates a voltage bound and disagrees with currents recovered from the line primitives. It distinguishes initialization and active-bound warnings from violated constraints. It does not infer global optimality from `LOCALLY_SOLVED` or call the whole model independently verified. The missing full residual bundle is an explicit conclusion, not a failed exercise.

For a smaller complete decision argument, reproduce the interval model-choice exercise: explain why a 109 A nominally accepted transfer is not robust over the declared conductance interval. Submit the monotonicity argument and recovered member currents. Keep its scalar scope separate from the multiconductor running fixture.

Chapter 24

The running network

Page status: semantic specification; numerical realization is versioned separately.

The book uses one synthetic reference network to compare representations and transformations. The network is deliberately small enough to draw and solve repeatedly, but it contains the features that disappear in a conventional balanced bus–branch diagram.

This page is the **semantic specification**. Its first numerical realization is the executable running network, version 0.1.0. The semantic contract remains authoritative where that fixture explicitly lists an unimplemented feature.

The layout below is the reference drawing for the rest of the book. Later figures should be read as views or overlays of this same source object set, not as unrelated replacement networks.

Design requirements

The running network must contain:

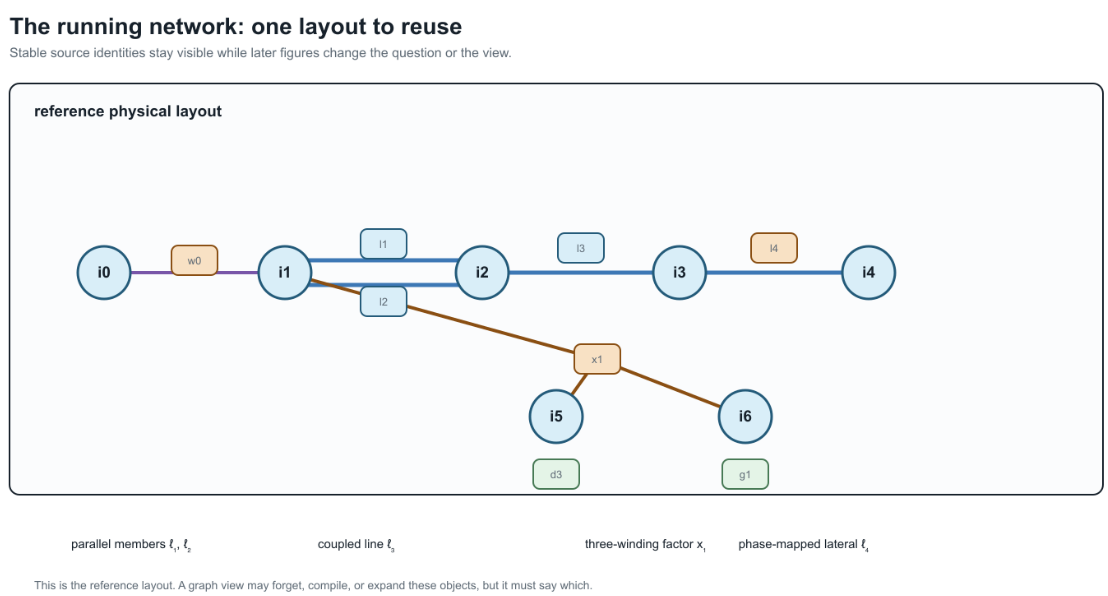

Figure 24.1: Reusable physical layout of the running multiconductor network.

- ordered, nonuniform bus terminal sets;
- a four-conductor line with full mutual coupling;
- an explicit neutral and impedance grounding;
- a conductor permutation or phase discontinuity;
- heterogeneous parallel branches with distinct limits and states;
- an ideal switch represented as an object rather than merged buses;
- a genuinely multiwinding transformer;
- at least one continuous control and one discrete decision;
- measurements and operational limits;
- stable source identities and provenance for every generated object.

Source objects

The provisional bus set is

$$\mathcal{B}^{\mathrm{R}} = \{i_0, i_1, i_2, i_3, i_4, i_5, i_6\}.$$

The superscript R denotes the running case, not an electrical quantity. Bus terminal vectors are:

Bus	Ordered terminals	Role
i_0	$[a, b, c, n]$	source-side substation bus
i_1	$[a, b, c, n]$	switched feeder head and transformer primary
i_2	$[a, b, c, n]$	grounded junction and parallel-corridor receiving bus
i_3	$[a, b, c, n]$	downstream four-wire bus
i_4	$[a, c, n]$	phase-discontinuous lateral bus
i_5	$[a, b, c, n]$	secondary winding bus
i_6	$[a, b, c]$	delta tertiary winding bus

These labels are stable source identifiers. A compiled view may create virtual buses but must not reuse or renumber these identities without a reversible map.

Two-terminal topology

The declared forward topology includes

$$w_0 i_0 i_1 \in \mathcal{T}^{W\to},$$

where w_0 is an ideal switch, and

$$\ell_1 i_1 i_2,\ \ell_2 i_1 i_2,\ \ell_3 i_2 i_3,\ \ell_4 i_3 i_4 \in \mathcal{T}^{L\to}.$$

Lines ℓ_1 and ℓ_2 are a heterogeneous parallel pair. They have distinct full impedance matrices, conductor ratings, construction records, outage states, and candidate investment decisions. An aggregated admittance may be a valid terminal-behaviour view while being invalid for the decision problem.

The generated line-identity witness makes the graph consequence explicit. In the scalar line projection of this multiconductor fixture, the four lines have one cycle-space dimension: the chord is ℓ_2 against the declared tree $\{\ell_1, \ell_3, \ell_4\}$, so the cycle is the two-member parallel fibre. Collapsing unordered bus pairs to a simple graph gives cycle rank zero. The switch and the three-winding transformer are deliberately excluded from this line-only calculation; their terminal and factor incidences belong to the port–factor view rather than being silently treated as scalar edges. The executable record is `experiments/generated/running-network-cycle-space-witness.json`.

Line ℓ_3 is a four-conductor section with nonzero mutual impedance and shunt admittance. Line ℓ_4 is a three-conductor lateral. Its terminal maps make the phase selection explicit:

$$\mathbf{N}_{\ell_4 i_3} = [a, c, n], \qquad \mathbf{N}_{\ell_4 i_4} = [c, a, n].$$

The deliberate permutation tests whether an implementation compares conductor coordinates before composing matrices or merging sections.

Grounding

The neutral terminal at i_2 is connected to earth through grounding element h_n with finite admittance y_{h_n}. It is neither a perfect voltage reference nor an annotation on the bus:

$$I_{h_n} = y_{h_n} U_{i_2,n}.$$

The source establishes the network voltage reference through a separately declared grounding condition at i_0. The fixture also contains a small reference shunt h_{ref} on tertiary terminal a at i_6; it is retained as an explicit object rather than folded into the transformer. These distinctions make the neutral-grounding counterexample observable: eliminating i_2 as a zero-injection degree-two junction is invalid if the grounding relation is forgotten.

Multiwinding transformer

Transformer x_1 has three winding ports

$$\mathcal{K}_{x_1} = \{1, 2, 3\}, \qquad \beta_{x_1}(1) = i_1, \quad \beta_{x_1}(2) = i_5, \quad \beta_{x_1}(3) = i_6.$$

The provisional connections are grounded wye, grounded wye, and delta. Each winding has its own ordered terminal map, reference voltage, current limit, and winding identity. Leakage data are specified as pairwise short-circuit impedances referred to a declared winding base. No derivation may silently replace the transformer by three independent two-winding devices.

A compiled loss network may introduce an internal virtual bus and two-terminal branches. Its certificate must map every generated object to x_1 and the relevant winding, preserve the terminal relation, state how winding limits are enforced, and provide a round-trip test. The first exact compilation of the fixture's pairwise leakage data is developed in Multiwinding leakage reference compilation. Its connection to all eleven external winding terminals is developed in Multiwinding terminal leakage assembly. The earlier five-bus multi-port lowering reuses this transformer's checked interface as a structural teaching extension; it does not replace or relocate the numerical running fixture. An explicitly marked illustrative extension then exercises excitation, transformer-internal grounding, and fixed transfer serialization in Fixed-linear transformer factor completion without changing the canonical BMOPFTools fixture. The associated parameterized transformer tap decision case then adds an explicitly illustrative discrete winding control and tests decision recovery without altering the fixture. The transformer tap AC decision case then embeds that factor in explicit network voltage, neutral-KCL, power-balance, and recovered-current constraints.

Nodal elements and controls

The case includes:

- a voltage source s_0 at i_0;
- an unbalanced controllable generator g_1 at i_3;
- an unbalanced four-wire demand d_1 at i_3;
- phase-discontinuous demands d_{2a} and d_{2c} at i_4;
- a secondary demand d_3 at i_5;
- a delta demand d_4 at i_6;
- voltage and current measurements at selected terminals;
- conductor-current limits on every line and winding.

Element connection configurations and terminal maps are explicit. A nodal attachment does not imply balanced three-phase or phase-to-neutral connection.

Decision problems

Continuous base problem

The first executable problem is a fixed-topology multiconductor AC OPF:

$$\min_{z \in \mathcal{F}_{M^{\mathrm{R}}}} c_0(P_{s_0}) + c_1(\mathbf{P}_{g_1}) + c_{\mathrm{loss}} P_{loss} + c_{shed} P_{shed}.$$

The feasible set includes terminal KCL, full conductor-coupled branch relations, transformer winding relations, grounding, voltage limits, per-conductor current limits, device capabilities, and any fixed equipment state.

Discrete extension

A later extension makes selected states decisions:

- open or close w_0;
- remove one member of the parallel pair as a contingency;
- choose a reinforcement or switching action;
- select a discrete transformer or regulator state where the device model supports it.

The continuous problem should be established first so that failures caused by representation are not confused with failures of a mixed-integer solver.

Required views

The case will be published in at least these forms:

View	Mandatory retained facts
Asset/property	stable element and winding identity, provenance, construction, state ownership
Terminal connectivity	ordered terminals, grounding, phase discontinuity, switch state
Port–factor	device relations, arbitrary winding arity, limits, controls
Bus–branch multigraph	parallel identity and compiled virtual-object provenance
Simple graph	explicitly documented forgotten distinctions
OPF model	variables, equations, feasible set, objective, source map
Sparsity graph	variable/constraint blocks with no unsupported physical interpretation

Counterexample variants

Small variants isolate one failure at a time:

1. **R-PAR:** retain only ℓ_1 and ℓ_2 and their endpoint buses; compare aggregate admittance with member current limits and outage decisions.
2. **R-GND:** retain the chain through i_2 and its grounding element; test degree-two elimination with and without the shunt relation.
3. **R-PERM:** retain ℓ_4 and its endpoint terminal maps; test conductor-coordinate normalization.
4. **R-XFMR:** retain x_1 and boundary injections; test multiwinding compilation, winding-limit recovery, and round trips.
5. **R-SW:** retain w_0 and neighbouring connectivity nodes; compare a fixed-state quotient with a model that preserves switching as a decision.

Each variant must be small enough for an independent analytic or exhaustive check.

Preservation questions

Every transformation applied to the running case must answer:

- Are terminal voltages and currents exact over the declared operating domain?
- Are per-conductor and per-winding limits preserved or lifted?
- Are switch, outage, tap, and investment choices retained?
- Is the target feasible set exact, inner, outer, or scenario approximate?
- Is the objective identical after lifting?
- Do optimal target decisions map to admissible source decisions?
- Can eliminated currents, voltages, losses, and active constraints be recovered?
- Are every generated object and parameter traceable to source identities?

BMOPFTools realization

BMOPFTools is the preferred first implementation platform where its component and decision models fit the case. The book-level source specification remains implementation independent. Any BMOPFTools fixture must record:

- the exact repository commit;
- input data and augmentation or conversion steps;
- terminal ordering and units;
- solver and environment;
- the source-to-implementation object map;
- known differences between the semantic specification and implemented model.

Chapter 25

Executable running network

Page status: executable fixture, local solver evidence, and a derived state-conditioned radiality witness; claims are versioned and verification-scoped.

Status and purpose

Empirical artifact, version 0.1.0. The semantic specification now has a numerical realization in `data/running-network/v0.1.0.json`. The fixture is synthetic. Its values are chosen to make representation choices observable, not to recommend equipment ratings or a particular feeder design.

BMOPFTools is the first implementation platform because its model retains ordered conductor terminals, full impedance matrices, explicit grounding, parallel branches, switches, and an independent `n_winding` transformer model. The book-level specification remains implementation independent.

What is realized

Object class	Source identities	Numerical feature under test
buses	$i_0, \ldots, i_6$	nonuniform ordered terminal sets
switch	w_0	fixed closed state and per-conductor current limits
parallel lines	ℓ_1, ℓ_2	distinct full impedances and distinct limits
coupled line	ℓ_3	full four-conductor series matrix
mapped line	ℓ_4	$[a, c, n] \mapsto [c, a, n]$ conductor permutation
grounding	h_n	finite neutral-to-earth admittance at i_2
transformer	x_1	WYE–WYE–DELTA three-winding model with winding limits
loads	d_1, d_2, d_3, d_4	unbalanced wye, single-phase, and delta demand
generator	g_1	bounded continuous active and reactive decisions

BMOPFTools requires each single-phase load to be a two-terminal component. The source load d_2 is therefore compiled into `d2a` and `d2c`. That realization is recorded explicitly rather than silently redefining a partial-wye object.

Six views of the same source

The panels are not levels in a single hierarchy. The provenance artifact also contains a non-visual simple-topology quotient used by the cycle and radiality definitions:

1. the asset/property view retains stable physical identity;

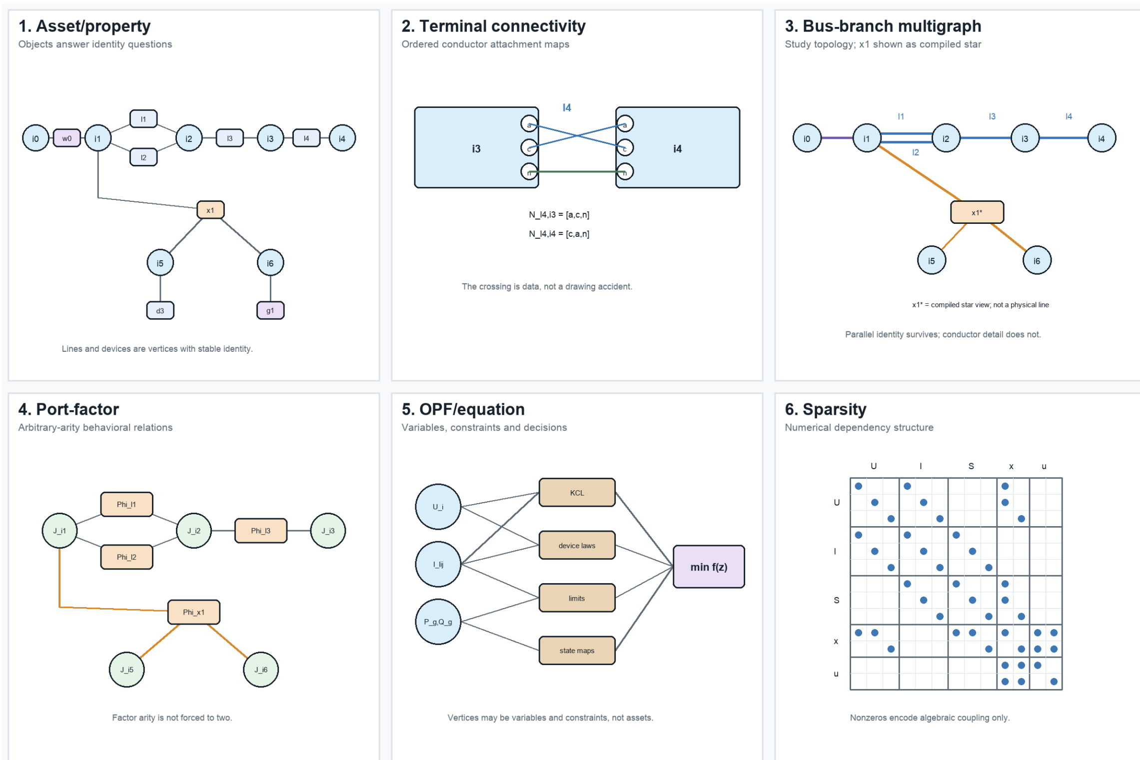

Figure 25.1: Six non-isomorphic views of the running network.

2. the terminal view exposes ordered conductor mappings and grounding;

3. the bus–branch multigraph supports conventional network algorithms while retaining ℓ_1 and ℓ_2 separately. In the figure, x_1^* is an explicitly labelled compiled star used to draw the multi-terminal device; it is not a claim that the transformer is physically three two-terminal lines;

4. the port–factor view represents x_1 as one factor of arity three;

5. the OPF graph makes variables, constraints, limits, and decisions explicit;

6. the sparsity graph records numerical coupling but does not claim that a nonzero is a physical branch.

Every generated view must retain a source map. A label such as `Phi_x1` is a generated factor identity with `source = x1`; it is not a replacement asset ID. The complete maps for the six illustrated views, together with the derived simple-topology quotient, are generated in `experiments/generated/view-source-maps.json`. Automated checks require every generated object to identify an existing fixture source and bind the map to the exact fixture and figure hashes.

The lineage is the operational companion to the view figure: virtual compiled objects are allowed, but source identity, map identity, and recovery remain available for limits, outages, maintenance, and decisions.

Executed checks

The generation script parses the exported BMOPF JSON, runs JSON-schema and model-conformance checks, solves a determined power flow, and solves an OPF with g_1 dispatchable. The current artifact reports:

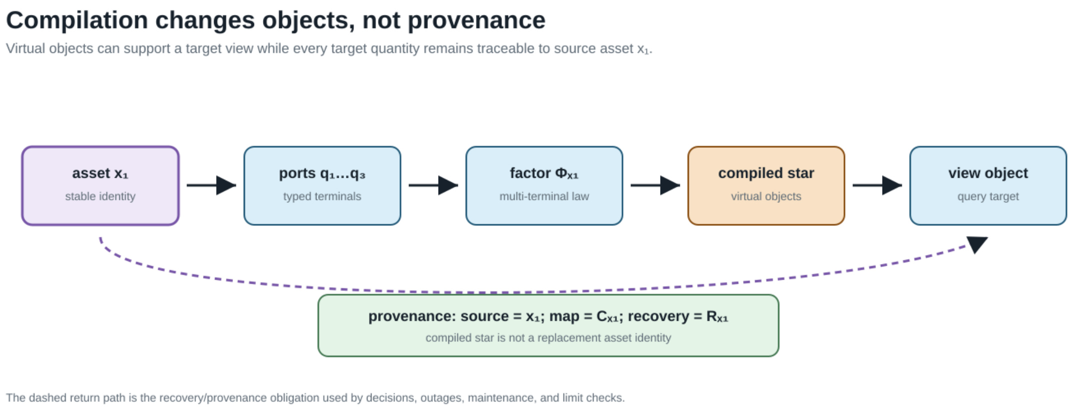

Figure 25.2: Compilation changes objects, not provenance.

Check	Result
schema valid	yes
validation errors	0
validation warnings	0
expected map diagnostic	`I.TMAP.CROSS_PHASE_LINE`
expected map diagnostic	`I.TMAP.PERMUTED_ORDER`
power-flow termination	`LOCALLY_SOLVED`
power-flow active loss	554.317 W
OPF termination	`LOCALLY_SOLVED`
OPF objective	-13.2000

The negative generator cost is a test device that drives g_1 to its active power bounds, making the continuous decision observable. The objective has no economic interpretation. Both the cost and this interpretation must change before the case is used for an economic study.

Reproduce the recorded review case

The September review adds an isolated run with a recorded Julia environment, case-source hashes, and an explicit BMOPFTools commit. From the repository root:

```
bash scripts/reproduce_clean_fixture.sh --check
bash scripts/reproduce_clean_fixture.sh
```

The first command validates recorded input hashes. The second requires the Julia version recorded in the profile, clones the pinned dependency into a fresh run directory, installs the locked environment, regenerates the fixture, compares selected numerical outputs with declared tolerances, and runs the verification lesson. The recorded export uses a different schema URI from the August fixture; the workflow checks equality of all engineering fields and records that metadata difference. Pinned replay additionally checks the exact recorded export hash. It prints the run directory and retains its inputs, resolved environment, results, and execution log. It never replaces the maintained evidence. Add `--offline` when the pinned Julia packages are cached. Use `--output /path/to/a/new/run` to choose a new directory outside the repository.

The profile is in `experiments/reproduction/review-2026-09-06/`. It binds the specific case sources by hash because this is an author-review working-tree snapshot, not a newly published book commit. It is a new recorded run; it does not repair missing metadata in an earlier experiment. Profile validation fails if a bound input changes. For a fresh development check instead, use:

```
bash scripts/reproduce_clean_fixture.sh --mode current
```

Current mode uses the dependency repository's HEAD and resolves dependencies afresh. It tests the fixture and scoped verification but does not claim the pinned numerical comparison. Readers should use these isolated commands when comparing evidence; direct `run_*.jl` generators are maintainer commands that can replace artifacts in the working tree.

Historical August record

The preserved August fixture record at BMOPFTools commit `b7aa9a1bb48bcc8b790d3bcf5417d6a32036352a` remains under `experiments/generated/clean-reproduction`. Its complete resolved environment was not recorded. Its original `generated_at` date was emitted as a constant by the old generator and must not be treated as authenticated execution time. The current generator separates fixture definition date from the actual UTC run timestamp. New run metadata is separate from deterministic fixture identity.

An explicitly requested historical reconstruction can use the old dependency revision with freshly resolved dependencies:

```
bash scripts/reproduce_clean_fixture.sh --mode historical-reconstruction
```

That operation does not reconstruct the missing historical environment, does not run newer verification APIs unavailable at the old revision, and makes no full historical numerical-replay claim. It writes to a fresh directory.

Check the returned solution

The current verification exercise can also run directly, without writing files:

```
julia --project=experiments experiments/lessons/verify_running_network.jl
```

It calls BMOPFTools `profile_solution` for supported voltage, device-limit, load-law, and network power-balance checks. It also calls the package's `line_yprim` for every line, evaluates each terminal-current map at the returned complex voltages, and compares the recovered currents with the reported currents. The comparison includes the `l4` conductor permutation. The declared absolute tolerances are 10^{-5} A for line-current consistency and 10^{-3} W/var for the reported network power balances. They are numerical acceptance tolerances for this fixture, not measurement accuracy or a global optimality guarantee.

A second result adds 1000 V to the real part of terminal `i2.a`, updates its magnitude and angle consistently, and leaves the reported currents untouched. It must trigger a voltage-limit finding and fail line-current recovery. The original result remains unchanged. Active-bound and initialization findings are retained in the report rather than treated as execution failures.

Evidence boundary. Post-solve evaluation is separate from the JuMP constraints, but the inputs and package primitive construction are shared. The profile's network power balance uses reported device powers/losses; the additional line recovery independently checks consistency with line primitives. This does not independently recalculate every transformer/device equation or nodal KCL from the original physical specification.

Accordingly, calling `check_solved_network_feasibility` on this result returns `indeterminate`: its required full independent residual bundle is absent. The lesson asserts that refusal. Supplying zeros for missing residuals would be an invalid way to obtain a pass. Full all-device/KCL verification remains a package-owned extension, and physical model adequacy requires other evidence.

Version 0.1.0 realization boundary

The versioned fixture is the current numerical contract, not a silent completion of every object in the semantic specification. It includes the declared d_4 delta demand and h_{ref} tertiary reference shunt, and it retains w_0 as a fixed closed switch. It does not add an undeclared g_2 generator or any other convenience object merely to make a continuous solve look more complete. Future fixture versions may promote semantic-only controls, but must change the version, source hashes, and provenance together.

Derived running-network topology states

The generated `experiments/generated/running-network-radiality-witness.json` lifts the topology predicates to this same fixture without changing version 0.1.0's continuous PF/OPF contract. It retains the four line identities, the w_0 switch, the two compiled transformer winding connections, and every ordered terminal map. Four derived states are compared: base switch closed, switch open, l_2 outage, and the combined switch-open/l_2-outage state.

The base and switch-open states remain member-nonradial because l_1 and l_2 are parallel members, even though their simple adjacency projection is radial. Removing l_2 removes that member cycle. The witness therefore reports both predicates and preserves transformer-factor provenance rather than treating the bus quotient as the source topology.

What remains semantic-only

The fixture currently treats w_0 as fixed closed. Discrete switching, contingency selection, investment choices, measurements, and protection zones remain in the semantic specification but are not claimed as executable in version 0.1.0. This boundary prevents a successful continuous OPF from being mistaken for validation of the full decision model.

Reference

Chapter 26

Notation and modelling conventions

Page status: maintained modelling convention; not a standalone empirical claim.

This page fixes the index discipline and physical conventions used throughout the book. The starting point is the notation of the BMOPFTools model specification, generalized only where the book must discuss representations that BMOPFTools does not directly implement.

Symbols describe semantic ownership

An index appears on a quantity when it identifies the object or oriented attachment that owns that quantity. Indices are not added merely because an equation is evaluated at an endpoint.

Symbol	Represents	Typical index
$\mathcal{B}$	buses or connectivity objects	i, j
$\mathcal{L}$	lines	ℓ
$\mathcal{X}$	transformers	x
$\mathcal{W}$	switches	w
$\mathcal{D}$	demands	d
$\mathcal{G}$	generators	g
$\mathcal{H}$	shunts and grounding impedances	h
$\mathcal{F}$	edge ends or flags in the normative multigraph object	f
$\mathcal{P}$	phases	p, q
$\mathcal{N}$	terminal names	p, q
$\mathbf{A}$	oriented incidence matrix	declared bus/edge coordinates
$\mathbf{A}_\ell$	full two-end primitive of element ℓ	nominal-π terminal-current map
A_r, A_c	retained and candidate current row maps	redundancy certificates
$\mathbf{A}_\phi, \mathbf{A}_x$	factor-incidence maps	conductor/factor assembly
$\mathbf{F}$	Fortescue phase-to-sequence transform	$abc \leftrightarrow 012$
C_+	restriction to the positive-sequence subspace	phase state $\to$ positive sequence
E_+	embedding of a positive-sequence state	positive sequence $\to$ phase state

Symbol	Represents	Typical index
σ	declared equipment/topology state; use σ_e for an element state and G^σ for its resolved graph	active/open/closed state, not a nominal-π subscript
π	projection or quotient map when decorated (for example π_σ); nominal-π is a model-family label	π_σ for connectivity quotient; π section for a circuit model
α	scalar decision or model coefficient; qualify the role, such as served-load α or ZIP α_Z	α in an optimization case; $\alpha_Z, \alpha_I, \alpha_P$ in a load law
Λ	asset-to-electrical relation, generally many-to-many rather than a function	$(a, e) \in \Lambda$
ρ	a declared realification map, sequence-mixing residual, or fitted proportionality scalar; qualify each use	$\rho(z)$, ρ_+ or $\rho^\star$
κ	enumeration map when subscripted by a semantic set, or a condition-number diagnostic	$\kappa_\mathcal{B}$ versus $\kappa(\mathbf{A})$
Θ	admissible family of typed factors or parameters; Θ_ϕ denotes a factor family	model-class or factor-parameter set
η	backward-error, residual, or scenario-validity diagnostic; qualify the source	$\eta(\widehat{x})$ or η_{asm}

Identifiers need not be consecutive integers. A data model may use stable strings while the mathematical model uses the corresponding symbols. When an implementation needs an ordinary array, introduce an explicit enumeration map, such as

$$\kappa_\mathcal{B} : \mathcal{B} \xrightarrow{\sim} \{1, \ldots, |\mathcal{B}|\}.$$

The semantic object is the label-indexed family $(Y_{ij})_{i,j\in\mathcal{B}}$; the stored array is $[Y]_{\kappa_\mathcal{B}(i),\kappa_\mathcal{B}(j)}$. This book names the semantic indices before displaying their integer coordinate realization. Integer positions are not asset identities, and changing the enumeration must not change a claim about the network.

For multiconductor models, bold $\mathbf{Y}^{\mathrm{N}}_{ij}$ denotes a block map between terminal spaces. It becomes a scalar matrix entry only after the terminal coordinates and bus enumeration have both been declared. By contrast, $\mathbf{Y}_\ell$ is intrinsic data owned by element ℓ and does not acquire endpoint indices merely because it appears in a nodal assembly.

Graph objects and graph-derived counts

The normative finite undirected multigraph is

$$G = (V, E, \mathcal{F}, s, \mathrm{ed}),$$

with two flags in each edge fibre $\mathrm{ed}^{-1}(e)$. The complete definition, including graph loops and matrix conventions, is maintained in Multigraphs for expert modelers. Power-system chapters may specialize V to buses $\mathcal{B}$ and E to identified lines $\mathcal{L}$, but they must not infer that specialization from an unqualified word such as *network* or *graph*.

For the loopless bus–branch specialization, $\partial(\ell)$ denotes the derived unordered endpoint pair and

$$\mathrm{simp} : \mathcal{L}^{\circ} \to E_{\mathrm{s}}$$

denotes the simple endpoint projection on non-loop members. The symbols p and q remain available for phase and terminal indices; graph maps use descriptive operators. For arbitrary-arity incidence structures, $\mathrm{rel} : \mathcal{F} \to R$ maps a flag to its owning relation. The state-conditioned connectivity-node quotient remains π_σ.

The following names are not interchangeable:

- d_{inc} is incidence degree and counts the two flags of a graph loop twice;
- d_{nbr} is distinct-neighbour degree in a loopless simple projection;
- incident-member count is an engineering count on identified equipment and equals d_{inc} only in the declared loopless specialization;
- terminal count belongs to a port model; and
- row or block nonzero count belongs to a compiled matrix-support graph.

Likewise, B denotes a generic signed graph-incidence matrix in the expert reference chapter, while $\mathbf{A}$ remains the preferred power-system incidence symbol in the engineering chapters. Both use -1 at the selected tail and $+1$ at the head unless a reproduced source declares another convention.

Oriented element triples

A two-terminal branch is identified independently of its orientation. Its physical incidence is the unordered endpoint pair $\partial\ell = \{i, j\}$. For a line ℓ whose stored reference orientation is from bus i to bus j, write

$$\ell ij \in \mathcal{T}^{L\to} \subseteq \mathcal{L} \times \mathcal{B} \times \mathcal{B}.$$

The opposite terminal arc and the bidirected terminal-arc set are

$$\mathcal{T}^{L\leftarrow} = \{\, \ell ji \mid \ell ij \in \mathcal{T}^{L\to} \,\}, \qquad \mathcal{T}^{L} = \mathcal{T}^{L\to} \cup \mathcal{T}^{L\leftarrow}.$$

The triple retains the identity of parallel branches. The arrow is a coordinate and terminal-order convention; it does not assert the operating direction of current or active power. Transformer and switch topology sets $\mathcal{T}^X$ and $\mathcal{T}^W$ use the same pattern when the device is genuinely two-terminal.

Quantities belonging to an oriented terminal or arc use the triple index:

$$\mathbf{I}_{\ell ij}, \qquad \mathbf{S}_{\ell ij}, \qquad \mathbf{Y}^{\mathrm{sh}}_{\ell ij}.$$

The two terminal quantities need not be negatives: terminal currents can include local shunts, and the two shunt half-sections may be asymmetric. A stored-orientation reversal swaps endpoint records; only a declared internal series-current coordinate has the simple antisymmetry relation. The full distinction is developed in Orientation, terminal quantities, and power transfer.

Element-intrinsic quantities

A physical parameter that does not change when the branch orientation is reversed carries only the element index. For a multiconductor line,

$$\mathbf{Z}^{\mathrm{s}}_{\ell} = \mathbf{R}_{\ell} + \mathrm{j}\mathbf{X}_{\ell}$$

is the series impedance of line ℓ. It is not written $\mathbf{Z}_{\ell ij}$ unless the model actually assigns different directional constitutive data. Likewise, length is L_{ℓ} and an element-level construction or provenance record belongs to ℓ.

This distinction is central to the book:

- ℓ identifies the physical or model element;
- ℓij identifies one oriented attachment of that element;
- ℓji identifies the opposite attachment;
- terminal maps determine how the element's ordered conductor coordinates meet the endpoint buses.

Bus terminals and terminal maps

Each bus i declares an ordered terminal vector

$$\mathbf{N}_i = [N_{i,1}, \ldots, N_{i,n_i}],$$

for example $[a, b, c, n]$. Its complex terminal-to-ground voltages form

$$\mathbf{U}_i \in \mathbb{C}^{n_i}.$$

An element does not assume that two buses have identical terminal order or even identical terminal sets. The ordered terminal maps $\mathbf{N}_{\ell i}$ and $\mathbf{N}_{\ell j}$ select the bus terminals to which the ordered conductors of line ℓ attach. Conductor permutations and phase discontinuities are therefore explicit mappings, not hidden matrix conventions.

For the declared forward orientation, a nominal series relation has the form

$$\mathbf{U}_j[\mathbf{N}_{\ell j}] = \mathbf{U}_i[\mathbf{N}_{\ell i}] - \mathbf{Z}^{\mathrm{s}}_{\ell}\mathbf{I}^{\mathrm{s}}_{\ell ij}.$$

The matrix $\mathbf{Z}^{\mathrm{s}}_{\ell}$ is full unless a more restrictive model is declared.

Current and power signs

Terminal currents are defined relative to the terminal at which they enter an element. In the bus balance, the corresponding contribution is given the sign required by the declared KCL convention. Every chapter must state that convention before using an unqualified current direction.

For this book, the preferred physical statements use complex terminal vectors. Rectangular real and imaginary parts are an implementation realization. Per-conductor complex power at an oriented terminal is

$$\mathbf{S}_{\ell ij} = \mathbf{U}_i[\mathbf{N}_{\ell i}] \circ \mathbf{I}^*_{\ell ij},$$

where $\circ$ is the Hadamard product and $(\cdot)^*$ is element-wise conjugation.

The pair $(\mathbf{S}_{\ell ij}, \mathbf{S}_{\ell ji})$ consists of terminal complex-power injections into the element. It is not generally one conserved edge flow: a series impedance absorbs power even when its end currents are opposite, and internal nominal-π shunts also make the terminal currents non-antisymmetric. An unqualified phrase such as *power from i to j* must therefore identify both the terminal sign convention and the operating quantity being reported.

Multi-terminal and multiwinding devices

A genuinely multi-terminal device is not assigned an artificial pair of endpoint buses. A transformer x has an ordered winding or port set $\mathcal{K}_x$. Its attachment map is

$$\beta_x : \mathcal{K}_x \to \mathcal{B}, \qquad \beta_x(k) = i,$$

and winding k has its own terminal map $\mathbf{N}_{xk}$, connection, reference voltage, current variables, limits, and provenance. Pairwise short-circuit data may be indexed by winding pairs, while a reconstructed winding-star or primitive admittance remains owned by x.

Only an explicit compilation may replace x by two-terminal elements and virtual internal buses. The generated arcs then receive their own identities and triples, together with a map back to (x, k) and to any generated internal object.

In this book, *winding* counts a transformer port or voltage-level connection unless a passage explicitly refers to physical coils.

Nodal attachments

A nodal element uses an element–bus pair, such as

$$di \in \mathcal{C}^D, \qquad gi \in \mathcal{C}^G, \qquad hi \in \mathcal{C}^H.$$

Its ordered terminal or connection map is recorded separately. A load attached at bus i is not therefore assumed to be three-phase, wye-connected, or phase-to-neutral.

Compound nodal operator

The scalar-to-matrix transition is a change in coordinate space, not a change of graph vocabulary. A scalar branch law uses $i_{\ell ij} = y_\ell(v_i - v_j)$; the multiconductor analogue uses ordered vectors and a matrix primitive.

The translation bridge and its diagram-reading checklist are given in How to read power-network diagrams and equations.

The assembled linear current–voltage operator on a declared ordered set of retained junction coordinates is

$$\mathbf{I} = \mathbf{Y}^{\mathrm{N}}\mathbf{U}.$$

The superscript N means *nodal* and does not imply a scalar, positive-sequence, or simple-graph model. Conventional $\mathbf{Y}_{\mathrm{bus}}$ and `Ybus` are retained as aliases when reproducing software interfaces or established scalar bus-matrix language, but every use must state the coordinate order and model class. A reduced image is distinguished as $\widehat{\mathbf{Y}}$ or $\mathbf{Y}_{\mathrm{K}}$; it is not silently overwritten onto the source operator.

Ground, neutral, and reference

The following are distinct:

- a voltage reference used to remove gauge freedom;
- the physical earth or an earth-return model;
- a neutral conductor;
- a perfectly grounded terminal;
- a grounding impedance or admittance;
- the elimination of a neutral variable by a declared reduction.

No chapter may use *ground* as an unqualified synonym for all of them. Perfectly grounded terminals have fixed voltage. Impedance grounding is an explicit shunt or factor. A neutral conductor remains a network conductor unless an admissible transformation eliminates it.

Parameters, variables, and sets

The mathematical prose distinguishes:

- ordinary italic lower-case symbols for scalars;
- bold symbols for vectors and matrices;
- calligraphic symbols for sets;
- roman subscripts or superscripts for descriptive roles such as sh or ref;
- a superscript $*$ for complex conjugation and H for conjugate transpose.

The HTML may use colour to distinguish real and complex quantities or parameters and variables, following BMOPFTools. No definition may depend on colour: typography, prose, and context must remain complete in monochrome PDF output.

The same rule applies to generated figures. Titles, labels, line styles, panel placement, and captions carry the interpretation; colour only provides a secondary visual cue. In matrix-pattern figures, filled cells mean nonzeros and the panel title identifies the matrix or formulation. In the running network multiview, x_1^* is labelled as a compiled star so its orange styling is not the sole indication of compilation.

Units and coordinate systems

Foundational physical equations use SI units. A per-unit system, sequence transform, conductor permutation, or real rectangular realization is a declared coordinate transformation. Its bases, ordering, orientation, and inverse or recovery map belong in the transformation record.

Equality of raw arrays is not evidence of physical equality when their terminal order, units, reference direction, or base differ.

Decision notation

A model M defines both equations and an admissible set $\mathcal{F}_M$. A decision problem is written abstractly as

$$\min_{z \in \mathcal{F}_M} \; f_M(z),$$

where z may include network states, continuous controls, and discrete decisions. A transformation is not decision preserving merely because it reproduces selected voltages. Its contract must state the relationship between source and target feasible sets, objectives, active constraints, and optimal decisions.

Local deviations

A chapter may introduce specialized notation when that notation makes a derivation substantially clearer. It must provide a local symbol table and an explicit map to this contract. Silent changes of current direction, terminal order, impedance base, or winding meaning are not permitted.

Chapter 27

References

Page status: bibliography and source register.

The bibliography below is generated from `references.bib`. Inclusion means that a source is relevant to the knowledge base; it does not imply endorsement of every claim or modeling assumption in that source.

[1] D. K. Molzahn. *Identifying Redundant Flow Limits on Parallel Lines*. IEEE Transactions on Power Systems **33**, 3210–3212 (2018). Author-hosted copy.

[2] C.-W. Ho, A. E. Ruehli and P. A. Brennan. *The Modified Nodal Approach to Network Analysis*. IEEE Transactions on Circuits and Systems **22**, 504–509 (1975).

[3] G. D. Hachtel, R. K. Brayton and F. G. Gustavson. *The Sparse Tableau Approach to Network Analysis and Design*. IEEE Transactions on Circuit Theory **18**, 101–113 (1971).

[4] B. Park, J. T. Holzer and C. DeMarco. *A Sparse Tableau Formulation for Node-Breaker Representations in Security-Constrained Optimal Power Flow*. IEEE Transactions on Power Systems **34**, 637–647 (2019).

[5] Electric Power Research Institute. *OpenDSS Solution Techniques* (n.d.). Current-injection fixed-point solution and direct initialization; accessed 2026-08-17.

[6] Electric Power Research Institute. *Power Conversion Elements* (n.d.). Norton primitive-admittance and compensation-current formulation; accessed 2026-08-17.

[7] Electric Power Research Institute. *Fault Study Mode Equations* (n.d.). Mode-specific nodal and short-circuit equations; accessed 2026-08-17.

[8] A. M. Kettner and M. Paolone. *On the Properties of the Compound Nodal Admittance Matrix of Polyphase Power Systems*. IEEE Transactions on Power Systems **34**, 444–453 (2019), arXiv:1712.08764.

[9] Electric Power Research Institute. *Load* (n.d.). Load-model and admittance-mode documentation; accessed 2026-08-17.

[10] G. V. Iswaran, S. Thakar, S. Nekkalapu, V. Vittal and M. Khorsand. *Development of an Application-Directed Distribution Network Equivalent for Transmission Planning Studies*. IEEE Transactions on Power Systems (2026). Accepted author version; application-directed feeder equivalent with PV/load aggregation and power-flow/EMT validation; accessed 2026-08-17.

[11] F. Geth, R. Heidari and A. Koirala. *Computational Analysis of Impedance Transformations for Four-Wire Power Networks with Sparse Neutral Grounding*. In: *Proceedings of the ACM International Conference on Future Energy Systems (e-Energy '22)* (2022); pp. 105–113. Phase-to-neutral impedance transformation and neutral-voltage recovery algorithm.

[12] C. L. Fortescue. *Method of Symmetrical Co-Ordinates Applied to the Solution of Polyphase Networks*. Transactions of the American Institute of Electrical Engineers **37**, 1027–1140 (1918).

[13] T. Kavitha, K. Mehlhorn, D. Michail and K. E. Paluch. *An $\widetilde{O}(m^2n)$ Algorithm for Minimum Cycle Basis of Graphs*. Algorithmica **52**, 333–349 (2008).

[14] M. Coppo, F. Bignucolo and R. Turri. *Generalised Transformer Modelling for Power Flow Calculation in Multi-Phase Unbalanced Networks*. IET Generation, Transmission & Distribution **11**, 3843–3852 (2017).

[15] E. B. Curtis and J. A. Morrow. *Inverse Problems for Electrical Networks* (World Scientific, 2000).

[16] L. Gan and S. H. Low. *Chordal Relaxation of OPF for Multiphase Radial Networks*. In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* (2014); pp. 1812–1815.

[17] L. Gan and S. H. Low. *Convex Relaxations and Linear Approximation for Optimal Power Flow in Multiphase Radial Networks*. In: *2014 Power Systems Computation Conference* (2014); pp. 1–9, arXiv:1406.3054.

[18] F. Dörfler and F. Bullo. *Kron Reduction of Graphs with Applications to Electrical Networks*. IEEE Transactions on Circuits and Systems I: Regular Papers **60**, 150–163 (2013), arXiv:1102.2950.

[19] J. C. Baez and B. Fong. *A Compositional Framework for Passive Linear Networks*. Theory and Applications of Categories **33**, 1158–1222 (2018), arXiv:1504.05625.

[20] S. Y. Caliskan and P. Tabuada. *Towards Kron Reduction of Generalized Electrical Networks*. Automatica **50**, 2586–2590 (2014), arXiv:1207.0563.

[21] J. B. Ward. *Equivalent Circuits for Power-Flow Studies*. Transactions of the American Institute of Electrical Engineers **68**, 373–382 (1949).

[22] A. Monticelli, S. Deckmann, A. Garcia and B. Stott. *Real-Time External Equivalents for Static Security Analysis*. IEEE Transactions on Power Apparatus and Systems **PAS-98**, 498–508 (1979).

[23] J. Machowski, A. Cichy, F. Gubina and P. Omahen. *External Subsystem Equivalent Model for Steady-State and Dynamic Security Assessment*. IEEE Transactions on Power Systems **3**, 1456–1463 (1988).

[24] O. Mokhtari, S. Chevalier and M. Almassalkhi. *Optimal Kron-based Reduction of Networks (Opti-KRON) for Three-Phase Distribution Feeders*. Electric Power Systems Research **263**, 113615 (2027), arXiv:2510.19608.

[25] O. Mokhtari, S. Chevalier and M. Almassalkhi. *Structure-Preserving Optimal Kron-Based Reduction of Radial Distribution Networks* (2025), arXiv:2508.15006.

[26] F. Geth and B. Liu. *Notes on BIM and BFM Optimal Power Flow With Parallel Lines and Total Current Limits*. In: *2022 IEEE Power & Energy Society General Meeting (PESGM)* (2022). Parallel-line consistency, total-terminal-current limits, and BIM/BFM SOC formulations.

[27] R. D. Zimmerman, C. E. Murillo-Sánchez and R. J. Thomas. *MATPOWER Case File Format* (n.d.). MATPOWER 8.1 documentation of version-2 case fields, including zero rating and tap conventions; accessed 2026-09-06.

[28] S. Boyd and L. Vandenberghe. *Convex Optimization* (Cambridge University Press, 2004). Chapter 5: duality and sensitivity analysis.

[29] M. A. Hammer, J. Dunfield, K. Headley, N. Labich, J. S. Foster, M. Hicks and D. Van Horn. *Incremental Computation with Names*. In: *Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications* (2015). Author manuscript revised 2021; from-scratch consistency is a property of its specified calculus.

[30] J.-B. Tristan and X. Leroy. *Formal Verification of Translation Validators: A Case Study on Instruction Scheduling Optimizations*. In: *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (2008).

[31] H. Ehrig, K. Ehrig, U. Prange and G. Taentzer. *Fundamentals of Algebraic Graph Transformation* (Springer, 2006).

[32] International Electrotechnical Commission. *IEC 60617 Database: Graphical Symbols for Diagrams* (2026). International graphical-symbol database; accessed 2026-08-17.

[33] International Electrotechnical Commission. *IEC 61082-1: Preparation of Documents Used in Electrotechnology – Part 1: Rules* (2014). Rules for presentation of electrotechnical documents and diagrams; accessed 2026-08-17.

[34] International Electrotechnical Commission. *IEC 61970-453: Energy Management System Application Program Interface – Part 453: Diagram Layout Profile* (2014). CIM-linked diagram layout exchange; accessed 2026-08-17.

[35] International Electrotechnical Commission. *IEC 61850-6: Configuration Description Language for Communication in Electrical Substations Related to IEDs* (2024). Substation configuration and switchyard structure exchange; accessed 2026-08-17.

[36] IEEE Power and Energy Society. *IEEE C57.12.70-2020: Standard for Standard Terminal Markings and Connections for Distribution and Power Transformers* (2020). Transformer terminal, polarity, neutral, grounding, and connection conventions; accessed 2026-08-17.

[37] Zepben. *TopologicalNode, CIM100 Datamodel* (n.d.). Generated CIM100 documentation; accessed 2026-08-13.

[38] PowSyBl contributors. *How to Manage Topological Views?* (n.d.). Accessed 2026-08-13.

[39] D. Sarkar, O. Waddell and R. K. Dybvig. *A Nanopass Framework for Compiler Education*. Journal of Functional Programming **15**, 653–667 (2005).

[40] A. J. van der Schaft and B. M. Maschke. *Port-Hamiltonian Systems on Graphs*. SIAM Journal on Control and Optimization **51**, 906–937 (2013).

[41] ENTSO-E. *Common Grid Model Exchange Standard (CGMES) Library* (n.d.). Official CGMES and CIM exchange-profile library; accessed 2026-08-13.

[42] PowerModelsDistribution contributors. *Engineering Data Model* (n.d.). Accessed 2026-08-13.

[43] PowerModelsDistribution contributors. *Conversion to Mathematical Model* (2020). Documentation generated 2020-06-30; accessed 2026-08-13.

[44] F. Geth, S. Claeys and R. Heidari. *On the Implementation of the Fixed Point Iteration Current Injection Method to Solve Four-Wire Unbalanced Power Flow in PowerModelsDistribution.jl* (2023), arXiv:2305.04405. Technology description of matrix-valued four-wire component models and nodal current-injection iteration.

[45] C. S. Cheng and D. Shirmohammadi. *A Three-Phase Power Flow Method for Real-Time Distribution System Analysis*. IEEE Transactions on Power Systems **10**, 671–679 (1995).

[46] R. D. Zimmerman and H.-D. Chiang. *Fast Decoupled Power Flow for Unbalanced Radial Distribution Systems*. IEEE Transactions on Power Systems **10**, 2045–2052 (1995).

[47] M. A. Wortman, D. L. Allen and L. L. Grigsby. *Techniques for the Steady State Representation of Unbalanced Power Systems: Part I. A Systematic Building Block Approach to Network Modeling*. IEEE Transactions on Power Apparatus and Systems **PAS-104**, 2805–2814 (1985).

[48] J. J. Grainger and W. D. Stevenson. *Power System Analysis* (McGraw-Hill, 1994). Sections 7.2–7.3 give the primitive-admittance building blocks and equivalent lattice for mutually coupled branches.

[49] W. H. Kersting. *The Modeling and Analysis of Parallel Distribution Lines*. IEEE Transactions on Industry Applications **42**, 1126–1132 (2006).

[50] R. Yan and T. K. Saha. *Analysis of Unbalanced Distribution Lines with Mutual Coupling across Different Voltage Levels and the Corresponding Impact on Network Voltage*. IET Generation, Transmission & Distribution **9**, 1727–1737 (2015).

[51] A. Dziendziel, H. Kocot and P. Kubek. *Construction and Modeling of Multi-Circuit Multi-Voltage HVAC Transmission Lines*. Energies **14**, 421 (2021).

[52] D. A. Tziouvaras, H. J. Altuve and F. Calero. *Protecting Mutually Coupled Transmission Lines: Challenges and Solutions*. In: *2014 67th Annual Conference for Protective Relay Engineers* (2014); pp. 30–49.

[53] International Electrotechnical Commission. *IEC Common Information Model: MutualCoupling* (2022). Generated documentation for the IEC 61970-301 MutualCoupling class; accessed 2026-08-25.

[54] PowSyBl contributors. *Grid Model Extensions: Line Couplings* (2026). Version 7.3.0-RC1 documentation; accessed 2026-08-25.

[55] PowerWorld Corporation. *Mutual Impedance Records* (n.d.). Zero-sequence mutual impedance records with branch identities, dot convention, and coupled-section fractions; accessed 2026-08-25.

[56] J. R. Carson. *Wave Propagation in Overhead Wires with Ground Return*. Bell System Technical Journal **5**, 539–554 (1926).

[57] C. Grudzien, D. Deka, M. Chertkov and S. N. Backhaus. *Structure- and Physics-Preserving Reductions of Power Grid Models*. Multiscale Modeling & Simulation **16**, 1916–1947 (2018), arXiv:1707.03672.

[58] J. Sistermanns, M. Hotz, W. Utschick, D. Hewes and R. Witzmann. *Feature- and Structure-Preserving Network Reduction for Large-Scale Transmission Grids*. In: *2019 IEEE Milan PowerTech* (2019); pp. 1–6, arXiv:1903.11590.

[59] W. Jang, S. Mohapatra, T. J. Overbye and H. Zhu. *Line-Limit-Preserving Power System Equivalent*. In: *2013 IEEE Power and Energy Conference at Illinois (PECI)* (2013); pp. 206–212. Open-access author copy.

[60] Z. K. Pecenak, V. R. Disfani, M. J. Reno and J. Kleissl. *Multiphase Distribution Feeder Reduction*. IEEE Transactions on Power Systems **33**, 1320–1328 (2018).

[61] Electric Power Research Institute. *Circuit Reduction for OpenDSS Version 8.5* (n.d.). Accessed 2026-08-13.

[62] M. Farrokhabadi and L. Vanfretti. *An Efficient Automated Topology Processor for State Estimation of Power Transmission Networks*. Electric Power Systems Research **106**, 188–202 (2014).

[63] L. Moreau and P. Missier. *PROV-DM: The PROV Data Model*. W3C Recommendation (2013). Provenance entities, activities, agents, derivations, and constraints; accessed 2026-08-17.

[64] MATPOWER contributors. *Data Model Object* (n.d.). Developer data-model documentation; accessed 2026-08-13.

[65] R. Diestel. *Graph Theory*. 6 Edition, Vol. 173 of *Graduate Texts in Mathematics* (Springer, 2025).

[66] J. L. Gross, J. Yellen and M. Anderson. *Graph Theory and Its Applications*. 3 Edition (CRC Press, 2019).

[67] R. B. Bapat. *Graphs and Matrices*. 2 Edition, *Universitext* (Springer, 2014).

[68] B. Bollobás. *Modern Graph Theory*. Vol. 184 of *Graduate Texts in Mathematics* (Springer, 1998).

[69] S. Seshu and M. B. Reed. *Linear Graphs and Electrical Networks* (Addison-Wesley, 1961).

[70] Y. Song, D. J. Hill and T. Liu. *Local Stability of DC Microgrids: A Perspective of Graph Laplacians with Self-Loops*. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (IEEE, 2017); pp. 2629–2634.

[71] Electric Power Research Institute. *Line* (n.d.). Multiphase two-port nominal-pi line model; accessed 2026-08-17.

[72] J. Oxley. *Matroid Theory*. 2 Edition (Oxford University Press, 2011).

[73] J. H. van Lint and R. M. Wilson. *A Course in Combinatorics*. 2 Edition (Cambridge University Press, 2001).