P V D

The PVD Design Tool

Design, simulation and back-analysis of prefabricated vertical drain improved ground

User Guide · First edition

The App Development Team

A numerical consolidation solver for drain-improved ground.
Every answer carries its evidence.

CONTENTS

CHAPTER 1

Introduction

1.1 What this tool is

Prefabricated vertical drains shorten the drainage path through soft ground. Instead of water having to travel tens of metres vertically to a free surface, it travels a few hundred millimetres horizontally to the nearest drain. Consolidation that would otherwise take decades can then be completed within a construction programme. The engineering question is never whether drains work; it is *how far apart* to place them, *how long* to wait, and *how confident* anyone can be in the answer.

The PVD Design Tool answers those three questions numerically. The traditional answer is a design chart. Charts built on the unit-cell solutions of Barron (1948) and Hansbo (1981) share a set of limitations as a class. The design procedures of Rujikiatkamjorn and Indraratna (2009) and of Abuel-Naga, Bergado and Gniel (2015) are two examples. They assume a constant coefficient of consolidation. They assume instantaneous loading. They assume a single uniform layer with fixed boundary conditions. And they idealise the smear zone as an abrupt permeability step, with no transitional smear zone between the disturbed annulus and the intact soil. Real ground obliges none of those assumptions. The tool therefore solves the consolidation equation directly, layer by layer, with the coefficient of consolidation changing as effective stress changes, and with smear and transition zones, multi-stage loading, vacuum preloading, creep, unload–reload hysteresis and partially penetrating drains all carried explicitly.

1.2 Who this guide is for

It is written for the practising geotechnical engineer who has to sign the drawing. It assumes you know what consolidation is, what a coefficient of consolidation means and why a preconsolidation pressure matters. It does not assume you know anything about finite volumes, Latin hypercubes or Bayesian inversion — where those are needed, they are built up from the physics rather than quoted.

> **KEY IDEA**
>
> The guide has one consistent bias: **a number you cannot interrogate is a number you should not use.** Wherever the tool produces an answer, this guide tries to say what would have to be true for that answer to be wrong, and how you would find out.

1.3 The three tasks

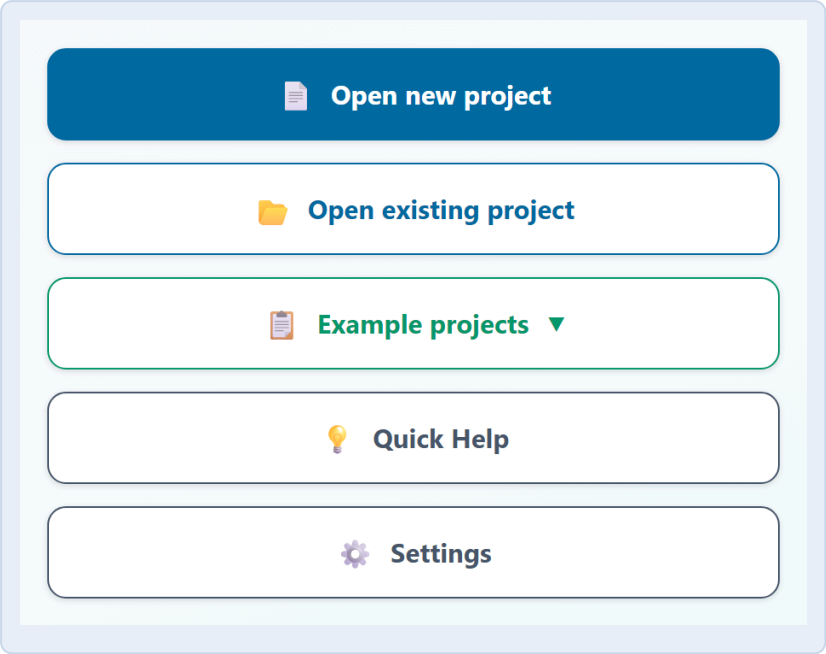


Figure 1.1. The hub, which is where the tool opens. *Example projects* is the fastest way into any of the three tasks below: it fills a complete, realistic project so the first thing you see is a working run rather than an empty form. Every worked example in this book was produced that way, so you can reproduce them by loading the same example and pressing Calculate.

The tool opens on the hub of Figure 1.1. Everything it does falls under one of three headings, and they map onto the three parts of this book, as set out in Table 1.1.

Table 1.1. The three tasks, and where each is covered.

Task	You supply	You get back	Covered in
Design	a target degree of consolidation and a date	the drain spacing that meets it	Part Two
Simulate	a drain spacing	settlement and degree of consolidation against time	Part Two
Back-calculation & Forecast	settlement plate and piezometer records	calibrated soil parameters, and a forecast with a credible band	Part Three

Design and Simulate are the same machine run in two directions: Design is simply Simulate placed inside a search. Back-calculation reverses the question entirely — instead of predicting from assumed parameters, it infers parameters from what the ground has already done.

1.4 Where this tool sits among the software you already use

This tool is built as a companion to the software already in geotechnical practice, not as a replacement for any of it. A drain project touches several kinds of package over its life, and each does its own job well. Table 1.2 groups them in general terms and states what this tool adds beside each. No package is named, deliberately: the purpose is a division of labour, not a criticism of tools that do their own work well.

Table 1.2. The software landscape for PVD work, in general terms, and the division of labour this book assumes.

Tool family	What it does well	What this tool adds beside it
Three-dimensional settlement and consolidation packages	Settlement of arbitrary load layouts at any point, staged construction, and the broader site questions. Drain zones are treated with the classical analytical radial solution and a smear factor.	The drain system itself, in finer physics: a resolved smear and transition zone, stress-dependent permeability acting inside the radial field, a smooth yield transition, and vacuum with depth attenuation. A Design mode that finds the spacing rather than checking one, and returns it with a P10–P90 band.
General-purpose finite-element suites	The peripheral geometry: differential settlement, stability, soil–structure interaction. Drains enter as line elements or explicit meshes, with advanced creep models available.	A unit-cell answer in minutes for screening and design iteration; and calibrated parameters from the monitoring record to feed the larger model, which is the integration that matters most.
One-dimensional settlement tools in national design practice	Staged embankment settlement with isotache creep and drain or dewatering schedules. Some offer an automatic fit of the computed curve to a settlement-plate record.	Inversion rather than curve scaling: physical layer parameters recovered jointly from every plate and piezometer at once, and a forecast range whose reliability was measured against known truths (Part Three) rather than assumed.
Observational-method constructions (Asaoka and hyperbolic fits, usually kept in spreadsheets)	A model-free reading of the trend of the record.	The same construction, built in as a cross-check, beside a forecast that carries the physics and an uncertainty statement.

The integration runs through parameters and files rather than plug-ins, and it works in three directions. **Calibrate here, model anywhere:** back-calculation returns the layer parameters the other packages take as input — the permeability family, the compression indices, the preconsolidation profile, the creep ratio and the smear description — so your own forward model can be re-run with values the ground has supported rather than the laboratory's first estimate. **Check a design from anywhere:** where the classical assumptions hold this solver reproduces the analytical radial solution to within one percent (Part One), so a design made in another package can be restated here, and the finer physics and the probabilistic band then show what would change. **Read the record beside any model:** observations import from CSV or Excel, results export the same way, and the whole project travels as a single file.

KEY IDEA

The claim this book makes is deliberately narrow. Outside the treated area — differential settlement across a footprint, stability, structures — the packages above remain the right tools, and §1.9 states the validity domain plainly. Inside the unit cell, and from the day the first plate reports, this tool does work that none of the general packages set out to do.

1.5 How to read this book

The parts are ordered from the foundations outwards. Part One sets out what the solver actually does — the equation it discretises, the elements it uses, and the measured evidence that its answers converge. Part Two then puts that machinery to work on the two forward questions. Part Three reverses the direction, inferring soil parameters from what the ground has already done.

That order is deliberate but not compulsory. The parts are self-contained, and an engineer with a scheme to size can begin at Part Two and refer back to Part One when a number needs defending. Reading Part One first, though, means every later result arrives with its accuracy already established rather than assumed. Table 1.3 summarises the three parts.

Table 1.3. The three parts.

Part	Covers	Read it when
One — Numerical Foundations	the governing equation, the finite-volume element, the mesh, boundary conditions, and measured convergence	you want to know what the answers rest on, before you rely on them
Two — Design and Simulation	the forward tasks, the soil model, the probabilistic spacing band, sensitivity ranking	you are sizing a scheme
Three — Back-Analysis and Forecasting	calibrating against monitoring data, the search pipeline, reading the credible band, and how far to trust it	the embankment is built and you have readings

Each part opens with a statement of what you should be able to do once you have worked through it. Within the text, four kinds of aside are used consistently:

Aside	Means
Key idea (green)	the one sentence to remember from the section
Watch out (amber)	a trap that has caught real users, or a limitation you must state in a report
Reproduce it (blue)	a result you can regenerate in the app yourself, with the settings given
Recap (violet)	the argument of a long section, compressed

1.6 On the numbers in this book

Every figure and quoted result here was produced by the tool's own solvers, on soil the app will load for you from its built-in examples, by scripts kept alongside the source. Where a result is a limitation rather than a strength, it is reported as such. A band that turned out to be over-confident, a benchmark that cannot discriminate, and a parameter that cannot be identified from settlement alone are all examples.

Claims are traceable to the script that produced them, kept in the `validation_2026/` directory beside the source — a name that predates the §1.7 taxonomy: its contents are verification and self-consistency scripts except where a script itself says otherwise. Three classes of statement are deliberately *not* script-backed, and are the only exceptions: quantities read directly from the application source (defaults, limits, the parameter lists), results attributed to a cited publication, and the schematic figures, each of which says "schematic" in its caption and carries no quantitative scale.

> **WATCH OUT**
>
> **The largest uncertainty in any consolidation prediction is the soil, not the solver.** Part One shows the discretisation error at the default mesh to be around a tenth of a percentage point of degree of consolidation. A tenfold uncertainty in horizontal permeability is entirely ordinary. Keep that ratio in mind whenever you are tempted to refine a mesh instead of commissioning another test.

1.7 Verification, validation, and what this book claims

Those two words are often used interchangeably. They are not interchangeable, and the difference decides how much any of the evidence in this book is worth. The usage here follows the ASME V&V convention, and Table 1.4 states which chapters discharge each obligation:

Table 1.4. The verification and validation hierarchy, and which chapters discharge each obligation.

Activity	Question it answers	Evidence	Where
Code verification	Does the program solve the equations correctly?	Comparison against closed-form solutions of the same equations	Chapter 10 — eight analytical benchmarks
Solution verification	Is the discretisation fine enough that the answer is the equation's, not the mesh's?	Measured order of convergence under mesh and time-step refinement	Chapter 9
Validation	Do the equations describe the real ground?	Comparison against physical observation	**§29.8 only** — one field case
Self-consistency	Does the estimator recover parameters the same model generated?	Synthetic recovery — circular by construction	Chapter 27
Cross-method comparison	Do independent methods agree?	Third-party published back-analyses	Chapter 30

WATCH OUT

Validation status. The verification evidence in this book is strong. Verification asks whether the program solves the mathematics it claims to solve. The convergence study of Chapter 9 measures second-order accuracy in both directions, and the eight closed-form benchmarks of Chapter 10 agree with the exact solutions to about one percentage point or better. Together these establish that the solver reproduces its equations faithfully. Validation is a different question. It asks whether those equations describe real ground. The validation evidence is thin. It rests on one instrumented embankment (Ballina, §29.8), plus comparisons against third-party analyses of that same site. One case is not a validation programme, and this book does not present it as one.

The residual doubt is not spread evenly across the three tasks. Design and Simulate (Part Two) solve the classical forward consolidation problem. Their physics is long established, and the closed-form benchmarks of Chapter 10 exercise the same forward path those two tasks use. Their standing is therefore close to that of any conventional consolidation analysis. Back-calculation and forecasting (Part Three) is where the validation gap is concentrated. Inferring parameters from a truncated record, and forecasting from them, can only be validated by field evidence — and the field evidence is one case.

The cases below are recommended future work, listed in rough order of the value each would add — a second instrumented PVD case first, since it would do most to move the field evidence beyond a single site. None has yet been run, and no result from them is quoted anywhere in this book.

Table 1.5. Recommended future validation cases.

Case history	Why it is in the suite	Status
Ballina, NSW	PVD-improved soft clay, densely instrumented, several published independent analyses	**Done** — §29.8
Chek Lap Kok, Hong Kong	Staged loading over layered marine clay; already in scope through Yin et al. (2022)	Not yet run
Muar, Malaysia	Classic trial embankment with long records	Not yet run
Nong Ngu Hao (Bangkok Second Airport)	Vacuum and surcharge PVD trials on soft Bangkok clay	Not yet run
Skå-Edeby, Sweden	Decades-long record; tests the creep model over a horizon no construction project reaches	Not yet run
The case histories behind published PVD design charts (e.g. Abuel-Naga et al. 2015)	Tests the tool against the records used to derive chart-based design procedures	Not yet run

Any future case should follow the pre-registration practice of the synthetic campaign (§29.2, Chapter 31): acceptance criteria — a tolerance on predicted settlement at nominated times, and on the time to a nominated degree of consolidation — stated before the run, and reported against whether or not they are met.

Prediction class

Lambe (1973) graded geotechnical predictions by two things: *when* the prediction is made relative to the event, and *whether the measured outcome was already known* to the predictor at the time. Together these decide the evidential weight of any reported agreement. A prediction made before the event (Class A) cannot have been tuned to the answer. A prediction made during the event (Class B) can use the record so far, but not the outcome. The primed classes mark the loss of blindness: B1 and C1 are predictions made during or after the event by a predictor who already knew the measured outcome. Agreement in a primed class is weak evidence, because nothing prevented the analysis from being adjusted until it matched. Every predictive result in this book is labelled with its class, and Table 1.6 states where each class appears.

Table 1.6. Lambe prediction classes, and where each appears here.

Class	When the prediction is made	Results known?	In this book
A	Before the event	No	What Design and Simulate produce for a new scheme (Part Two)
B	During the event	Not at time of prediction	Forecasting from a partial monitoring record (Part Three) — the tool's main use
B1	During the event, by a predictor who already knows the measured outcome for the period being predicted. The prediction cannot be treated as blind, so its agreement with observation is weak evidence.	Known to the predictor	None — excluded deliberately; see below
C	After the event, from data sealed before the outcome was examined	Not known to the predictor	None — requires a formal blind exercise; see below
C1	After the event	**Known** to the predictor	**§29.8 and Chapter 30**, including the comparison with Huang et al. (2025), which is itself a Class C1 study

Classes B1 and C are empty by the nature of the workflow, not by omission. A Class C result needs a completed record that was deliberately withheld from the analyst — the event over, the outcome sealed until the prediction is lodged. The tool's intended use is the opposite: forecasting while the record is still growing, which is Class B. A Class B1 result would mean this book fitted part of a record while already knowing the rest, and then presented the agreement as evidence; results of that kind are excluded because they cannot be treated as blind. Populating the two classes would take a formal blind-prediction exercise against a sealed record. None has yet been run; the declared cases of Table 1.5 are the intended vehicle.

KEY IDEA

Read the class before the number. A Class C1 agreement means the back-analysis reproduced a record whose outcome was already published. That is much weaker evidence than the same agreement obtained blind. The strongest claim this book makes on field data is Class C1, and it is made once.

1.8 Evidence ledger

The whole case, on one page: Table 1.7. Each row is a substantive claim, the kind of evidence behind it, where that evidence was measured, and what doubt remains. Every later chapter is the expansion of a row.

Table 1.7. Evidence ledger. Evidence type: *A* analytical, *N* numerical, *S* synthetic (self-consistent), *F* field, *T* third-party.

Claim	Type	Where measured	Residual doubt
The solver solves the consolidation equation correctly	**A**	Ch. 10 — 8 closed-form benchmarks, MAE ≤ ~1 pp	Benchmarks require constant coefficients; they cannot exercise the stress-dependent, creep or hysteresis physics
The answer is the equation's, not the mesh's	**N**	Ch. 9 — 2nd-order convergence measured in both directions	Demonstrated on the default geometry; a pathological profile is not covered
Discretisation error is negligible beside soil uncertainty	**N**	Ch. 9 — ~0.12 pp at the default mesh	None material
Parallel and serial runs give the same answer	**N**	§10.3 — worker/main parity, difference exactly zero	Bit-identity is by construction; floating-point order still varies across machines
The inversion recovers parameters it is given	**S**	Ch. 27 — synthetic recovery, $\chi^2_r \approx 1.07$	**Circular**: the estimator recovers a truth the same forward model generated. Confirms convergence, not accuracy on real ground
Speed parameters are identifiable from settlement	**S F**	Ch. 21, Ch. 27	Smear η is only weakly identifiable; it trades off against k_0 almost exactly
Magnitude parameters are identifiable	**S**	Ch. 21, Ch. 27	Not below U ≈ 40–50 %; the band collapses to the prior
The reported band covers the truth at its stated rate	**S**	Ch. 29 (§29.3, §29.7) — coverage measurement	The band was measured **over-confident** (40–68 % coverage for a band labelled 90 %). The shortfall is reported, not corrected — §29.7 says to design to the input-uncertainty range (77–96 % measured); coverage is a synthetic estimate
Back-analysis and forecasting work on real ground	**F T**	**§29.8 — Ballina, one case, Class C1**	**The thinnest link in the chain, and where the §1.7 validation gap is concentrated.** One site, one soil type, outcome already published. Design and Simulate are not affected to the same degree: they run the classical forward problem the Ch. 10 benchmarks verify. See Table 1.5
It agrees with independent methods	**T**	Ch. 30 — Huang et al. (2025) BUS + Hypothesis B	Comparison is on the same site, and §30.5 concedes the reproduction attempt did not match the published result. §30.6 traces the mismatch to the inputs, not to an identified defect in either implementation: the paper's priors pin the settlement magnitude but leave the rate almost free, so they do not transfer as point estimates into a different forward model. A like-for-like comparison on matched inputs remains open

1.9 Validity domain

The conditions under which the tool's answers are meaningful, collected in one place (Table 1.8) so they can be lifted into the assumptions section of a report. Each is expanded where indicated.

Table 1.8. Validity domain — the assumptions a result stands on.

Assumption	What it means, and when it bites	Detail
Unit cell, no flow across the outer wall	Every drain is assumed to have an identical neighbourhood. At a block edge, a spacing change, or next to untreated ground, water does cross that surface and consolidation is over-predicted.	§6.4
Small strain; the mesh does not deform	Drain spacing and layer thickness are held at their initial values. Large-strain sites are outside the model.	Ch. 11
Fully saturated, Darcy flow	No partial saturation, no threshold hydraulic gradient.	Ch. 11
Vertical permeability is derived from c_v and m_v	k_v is recovered per layer as $c_v \cdot m_v$ and harmonic-meaned at layer interfaces, so a low-k_v band does restrict flow. The contrast is only as good as the m_v supplied for each layer.	§7.2
No well resistance — justified, not assumed	Well resistance is negligible when the discharge factor $D = q_w/(k_h \cdot l_m^2)$ exceeds about 5 (Mesri & Lo 1991) — a requirement of only 2–80 m^3/yr for typical clays, against the 100–500 m^3/yr modern band drains deliver (Hansbo 1997; Holtz et al. 1991; Rujikiatkamjorn & Indraratna 2007; Abuel-Naga et al. 2015). The input page computes D from your k_h, drain length and specified q_w and flags $D < 5$. Watch the exceptions: very long drains, organic soils where q_w degrades with time (Miura & Chai 2000), kinked drains.	§29.9
Minimum usable record for back-analysis	Below $U \approx 40$–50 % the magnitude family is essentially unconstrained and the band collapses to the prior. $U \approx 50$ % is typical, not a floor.	§29.9
Smear is weakly identifiable	η trades off against k_0 almost exactly, so back-analysis holds it and the forward tasks range it. A fitted η should not be quoted as a measurement.	Ch. 21
Determinism	Same inputs and same solver grid reproduce the same answer. Line searches and floating-point order make the last significant figures platform-sensitive; do not expect bit-for-bit identity across machines.	§29.9
Validation basis	**One field case.** Everything else is verification or self-consistency.	§1.7

1.10 Conventions

Convention	In this book
Units	SI throughout — metres, days, kilopascals, m^2/day. PVD dimensions in millimetres, as they are supplied.
Degree of consolidation	*U* is always a degree of **primary** consolidation and cannot exceed 100 %. Creep and immediate settlement are part of the *settlement*, never of *U*.
Spelling	Australian English.
Numbering	Figures and tables are numbered by chapter — Figure 12.3 is the third figure of Chapter 12. Sections are cited as §12.3.
Equations	Displayed equations are numbered by chapter, in the same scheme as figures and tables — equation (3.2) is the second displayed equation of Chapter 3. The number is set at the right margin.
Schematic figures	A figure whose caption says *schematic* is illustrative. It carries no quantitative scale and no number should be read off it.
Code names	Identifiers such as `solveRC2D` refer to functions in the application source, given so a claim can be checked against the code.

Four kinds of aside are used. Each carries a typed label as well as a colour and a distinct rule style, so the distinction survives greyscale printing and colour-vision deficiency:

KEY IDEA

The one sentence to carry away from the section.

WATCH OUT

A trap that has caught real users, or a limitation you must state in a report.

REPRODUCE IT

A claim in this book you can independently re-derive: the dataset, the settings, and the number you should get.

RECAP

The argument of a long section, compressed.

1.11 Notation

Table 1.9 collects the symbols used throughout the book.

Table 1.9. Principal symbols. Subscript 0 denotes an initial value throughout.

Symbol	Meaning	Units
u	excess pore-water pressure	kPa
U	degree of PRIMARY consolidation (never > 100 %)	%
U_s, U_p	degree from settlement, and from pore pressure	%
r_w	equivalent drain radius	m
r_s, r_t	outer radius of the smear and transition zones	m
r_e	radius of influence (unit-cell radius)	m
η	smear ratio k_h/k_s	–
k_0, k_h, k_v	initial, horizontal and vertical permeability	m/s
C_k	permeability change index, Δe/Δlog k	–
c_h, c_v	horizontal and vertical coefficient of consolidation	m^2/day
m_v	coefficient of volume compressibility	1/kPa
C_c, C_r	compression and recompression index	–
C_α/C_c	secondary compression ratio (creep)	–
σ′, σ'_0, σ'_p	effective, initial effective and preconsolidation stress	kPa
β	yield smoothing exponent in the operative compression index	–
e_0	initial void ratio	–
E_u	undrained modulus (immediate settlement)	kPa
H	layer thickness / drainage path length	m
γ_w	unit weight of water	kN/m^3
T_v	vertical time factor	–
N_r, N_z	radial and vertical cell counts	–
Δr, Δz, Δt	cell width, cell height, time step	m, m, day
χ^2_r	reduced chi-squared misfit	–
P5, P50, P95	percentiles of a reported band (Part Three)	–
P10, P50, P90	percentiles of the design-spacing band (Part Two)	–

CHAPTER 2

A tour of the application

2.0 Opening the application for the first time

The tool is a single file. There is nothing to install, no account to create and no server to connect to: open `PVD-app.html` in a current browser and the application is running. It keeps your project in that browser tab, and a project is saved by downloading a small `.pvdproj.json` file that you can reopen later or send to a colleague. Because everything runs locally, nothing you type leaves your machine, and the application works with no network at all.

One consequence is worth knowing before you start: a long calculation belongs to the tab it is running in. Leave the tab open while a back-analysis runs, and if you want it to finish promptly, keep it in front rather than minimised — a browser slows down work in tabs it thinks nobody is watching.

The rest of this book explains how the tool thinks. This chapter does something narrower and, if you have just opened the application for the first time, more urgent: it shows you where everything is. Which screen follows which, what each field on it is called, which button starts a calculation, and which of the results cards holds the number you came for.

It deliberately explains none of the underlying ideas. Every time a screen raises a question that deserves a real answer — why the soil table offers a range rather than a value, what the three rows of the forecast card mean, why one grid preset is slower than another — the caption sends you to the section that answers it properly. That division keeps the tour honest: a chapter that both showed the buttons and taught the method would fall out of step with the method the first time either changed. The final section, §2.6, is a table of every screen and card in the application against the section of this book that explains it, so nothing you can click on is left without an explanation somewhere.

> **KEY IDEA**
>
> **What you can do after this chapter.** Load a worked example, run both forward tasks and a back-analysis, and find your way to any card in the application — without being told what any of it means. Fifteen minutes, following §2.5.

> **RECAP**
>
> **These screenshots come from one build, one window width and one project.** Every screen here was captured from build **2026-09-17 15:08 (local)** at a window width of 1,500 points, running the built-in example project, so every number you can read in a screenshot is the same number the worked chapters quote. If your screen differs in a detail, the application has moved on and the figure is due for re-capture; the scripts that take these shots live beside the book and are re-run as part of a release.

2.1 The map of the application

Before the map, the one decision that decides everything else: which of the three tasks you want. They answer different questions, and the difference is not subtle.

What you want to know	The task	What you must have
How close together should the drains be, to hit a target by a deadline?	**Design**	soil properties and the loading programme
I have chosen a spacing — how far and how fast will the ground settle?	**Simulate**	the same, plus the spacing
The embankment is built and instrumented — what is the ground really doing, and what happens next?	**Back-calculation & Forecast**	the same, plus monitoring readings

Design and Simulate are the same screen asked from opposite ends, so moving between them costs nothing. Back-calculation is the one that needs field data; without readings it has nothing to work from.

The application has exactly three tasks and they share one shape. You pick a task, fill in an input page, press **Calculate**, watch a progress overlay, and land on a menu of results cards that you open one at a time. Learn that rhythm once and the only thing that changes between tasks is what the input page asks for.

Figure 2.1 draws every screen and labels every arrow with the control that causes the move, using the application's own wording. Two things in it are worth noticing before you go further. The **hub** on the left is the only screen that is not part of a task — it is where projects are opened, and where the example projects live. And every results menu has a ← **Go to input** route back, so nothing you press is a one-way door.

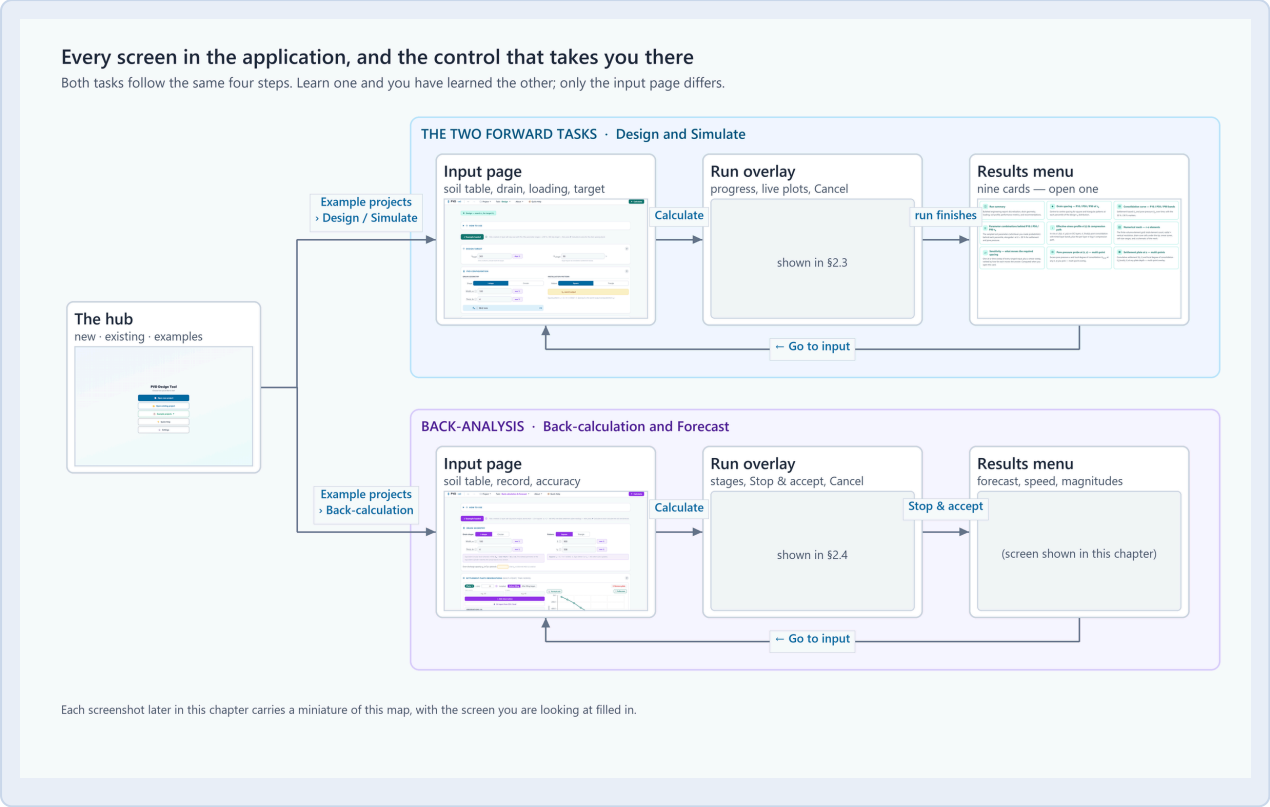
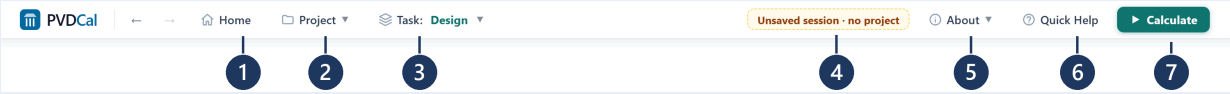


Figure 2.1. The screen-flow map: the hub, the two forward tasks, back-analysis, and the exact control that causes each transition. Every later screenshot in this chapter carries a miniature of this map with the current screen filled in, so you always know where you are. (Which task suits which question: §2.6.)

One strip of the interface appears on every screen after the hub, so it is worth naming before anything else. It carries the project, the task you are in, the help, and the button that starts a calculation.

Figure 2.2 names its seven parts. Only the last of them does anything irreversible, and even that only starts a calculation.



1. **Home** — back to the hub from anywhere. Nothing is closed or saved, and the hub offers a *Continue working* button straight back.
2. **Project** — new, open, save. Saving downloads a project file.
3. **Task** — shows the task you are in, and switches between the three. Your inputs are kept when you switch.
4. **The project state** — a green chip naming your project once you have one, or this amber *Unsaved session* chip when you do not. It is telling you that Save will ask for a project name first (§2.2).
5. **About** — version, references and the verification pages.
6. **Quick Help** — a six-step interactive walkthrough, available from anywhere.
7. **Calculate** — starts the run. Always here, on every input page. The two arrows to the left of Project step back and forward through the screens you have visited, like a browser's; the named buttons of §2.3 are usually quicker.

Figure 2.2. The top bar, present on every screen of the application. The task you are currently in is named in the middle; the button that starts a run is always in the same place on the right. On a phone the captions give way to the icons and the bar folds into two rows. (Saving and reopening projects: §2.2.)

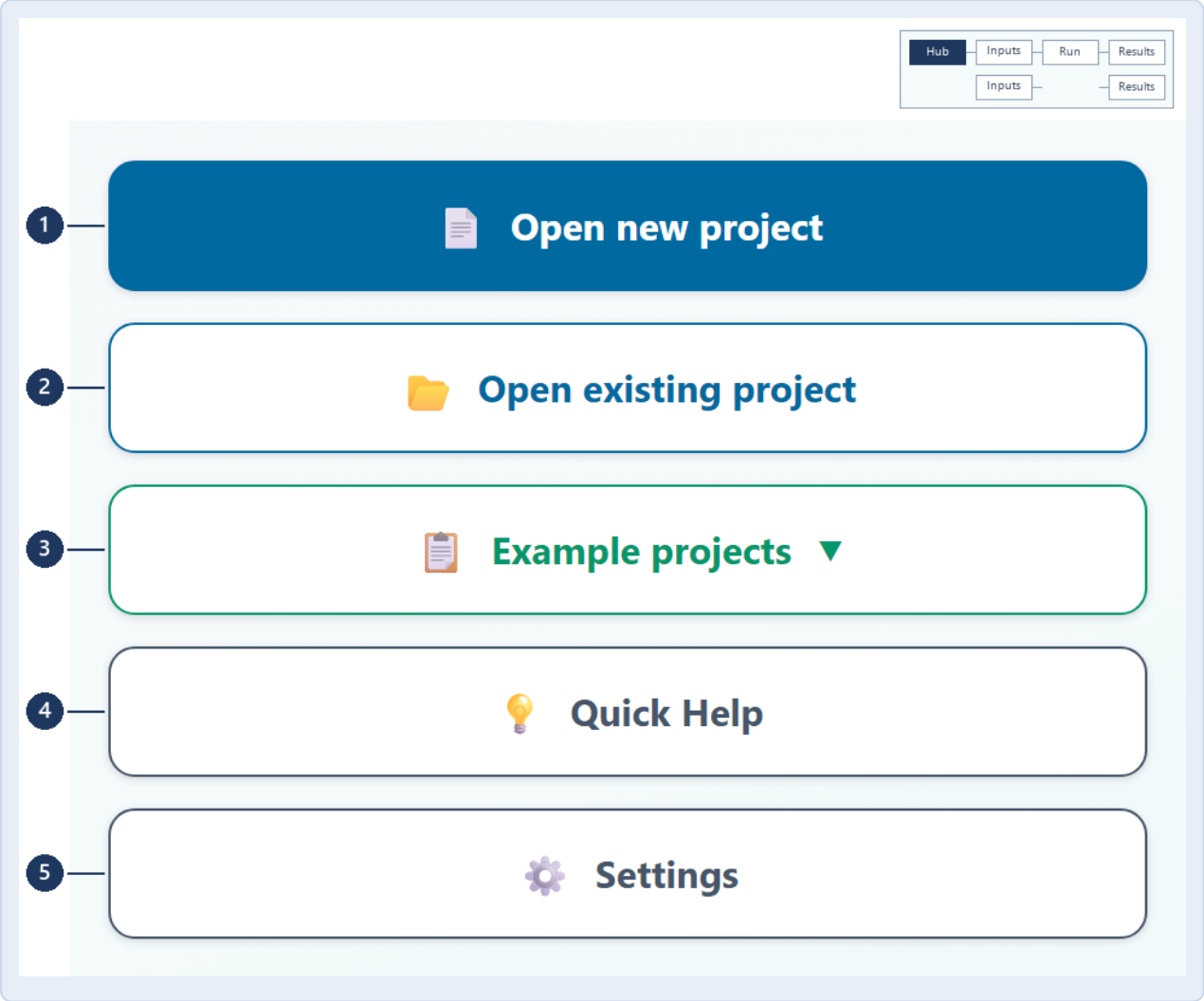
2.2 The hub, your project, and the settings

The hub is the first screen the application shows. It offers five ways in, and for a first session only one of them matters: **Example projects**. It carries three ready-made projects, one for each task, already filled in with a complete and realistic set of inputs.

Once you have opened projects before, the hub also lists the most recent of them beneath those buttons, newest first. Clicking one re-opens the file picker rather than loading silently, for the reason given in §2.2 below: a browser may not re-read a file you chose in an earlier session without asking. The list is empty on a first run, which is why it does not appear in the figure above.

This is the shortest path to a working answer, and it is also how this book was written. Load the example, press **Calculate**, and the numbers that appear are the numbers printed in the worked chapters of Parts Two and Three. Nothing in those chapters was typed by hand or copied from an older run: they are reproduced from these example projects, which is what makes the worked figures checkable. If a number on your screen disagrees with the book, that is a finding worth reporting rather than a mistake on your part.

Figure 2.3 shows the hub as it opens, with the five ways in.



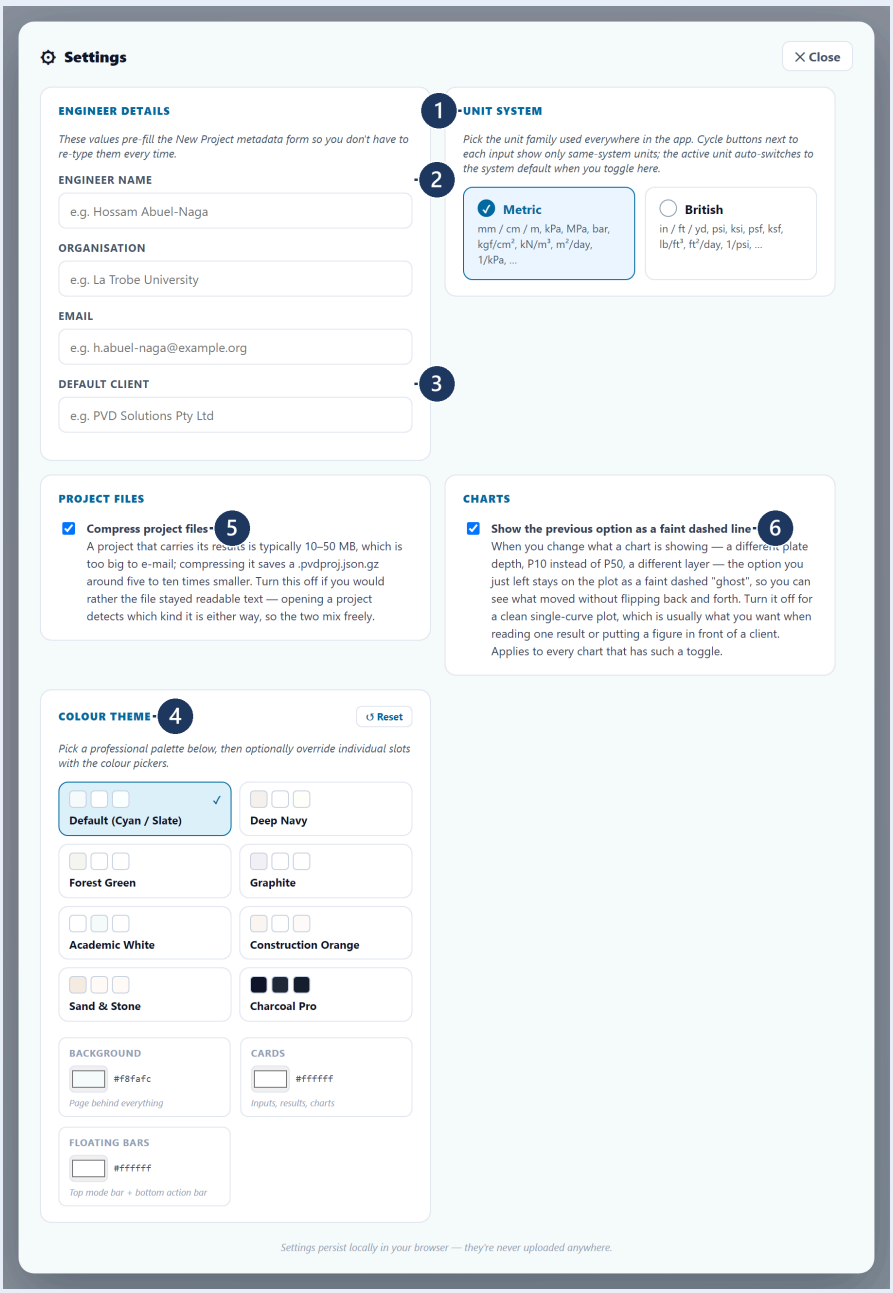
1. **Open new project** — an empty project; you choose the task next.
2. **Open existing project** — reload a project file saved earlier.
3. **Example projects** — opens a short menu: Design, Simulate, Back-calculation. Start here.
4. **Quick Help** — a one-page reminder of the workflow.
5. **Settings** — units, number formatting and appearance.

Figure 2.3. The hub. The three example projects behind callout 3 are the fastest way to a first answer, and the source of every worked number in this book. (Why the examples are reproduced rather than quoted: §1.3.)

Units, your details, and the look: the Settings dialogue

Settings is reachable two ways — the hub's fifth button and the Project menu — and it holds three things worth setting once, before you start typing.

Figure 2.4 shows the dialogue as it opens.



1. **Unit system** — Metric or British. This picks the unit family used throughout; the cycle button beside an input then offers only that family's units.
2. **Engineer name** — pre-fills the New Project form, and appears on exported reports.
3. **Default client** — pre-fills the same form, for when most of your work is for one client.
4. **Colour theme** — presets, with per-slot overrides and a reset to the default.
5. **Compress project files** — on by default; it is what keeps a project carrying its results small enough to send. §2.2 explains what it changes.
6. **Show the previous option as a faint dashed line** — on by default. When you change what a chart is showing, the option you just left stays on the plot as a faint dashed ghost. Turn it off for a clean single-curve plot.

Figure 2.4. The Settings dialogue. The unit system governs which units the cycle buttons beside each input offer; the engineer and client details pre-fill the New Project form; the theme is cosmetic. All three persist, so the application opens the way you left it. (Units on individual chart axes are separate, and are set per axis: §2.3.)

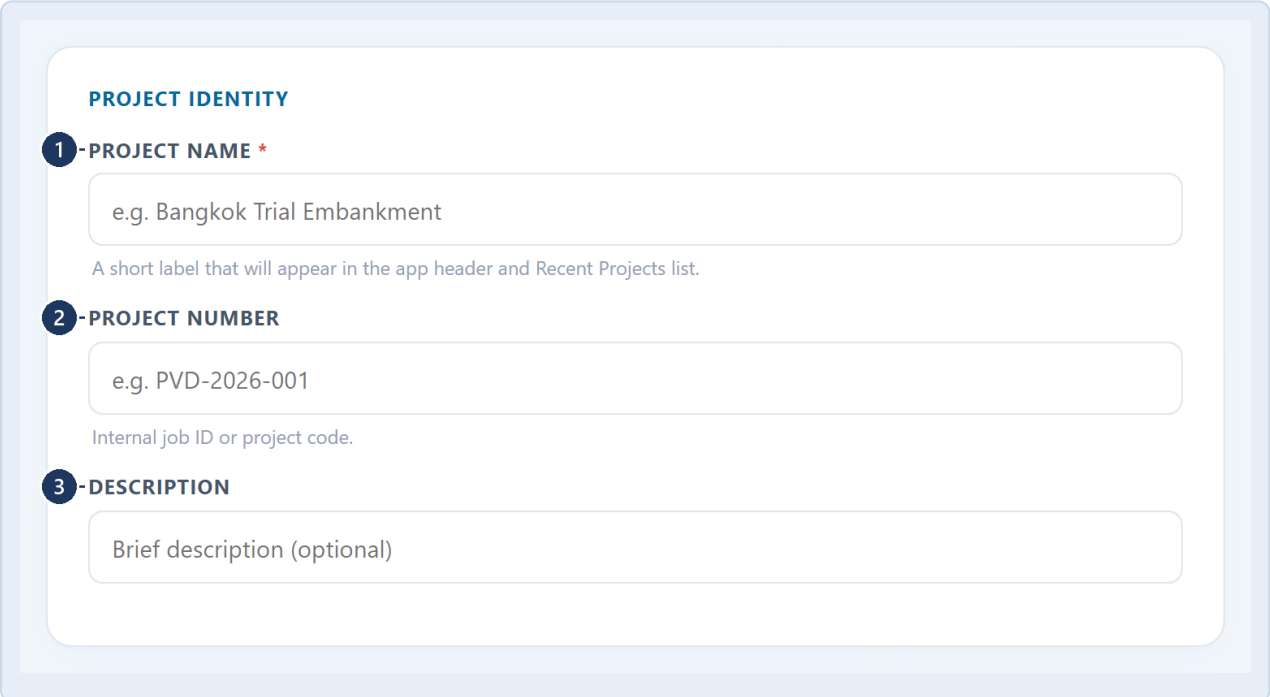
The *Charts* group, callout 6 above, holds one switch: **Show the previous option as a faint dashed line**, on unless you turn it off. Many of the charts in this application have a toggle that changes what they are showing — a different plate depth, P10 instead of P50, one layer instead of another. With this on, the option you have just left stays on the plot as a faint dashed "ghost", so you can see what your change did without flipping back and forth to compare. With it off, every chart draws only what is currently selected.

Which you want depends on what you are doing. Comparing two depths, or checking whether a percentile matters, is exactly what the ghost is for. Reading a single result, or putting a figure in front of a client, is usually cleaner without it. The switch applies everywhere at once, including the validation charts in the About pages, and it takes effect on the next change you make rather than resurrecting a comparison you left behind some time ago.

Project information, and what a project file is

Open new project asks for the project's identity before it gives you an input page: a name (the only required field), a project number, a description, and the engineer and client details that Settings can pre-fill. The name is what you will see in the application header and in the Recent Projects list, so it is worth making it specific.

Figure 2.5 shows the identity panel of that form.



1. **Project name** — required. Appears in the header and the Recent Projects list.
2. **Project number** — your internal job code, if you use one.
3. **Description** — optional free text.

Figure 2.5. The first panel of the New Project form. Only the name is required; everything else is there so the file, and any report exported from it, identifies itself six months later. (Why the examples bypass this: §2.2.)

Saving, reopening, and not losing work

Project › Save writes one file that holds everything: the project metadata, the drain and soil inputs for *every* task, and the last results each task computed. That last part is deliberate and it is why the file is large — reopening a project shows you the charts and tables again immediately, instead of asking you to re-run a back-analysis that took half an hour. **Open…** reads one back, and so does dragging a project file anywhere onto the window.

The first Save asks where the file should go. Every Save after that writes straight back to that same file, with no dialogue — the menu entry names the file it is about to overwrite, so it is never a silent surprise. **Save as…** is the one that asks again, for when you want a second copy rather than an update. Suggested filenames carry the date and the time, so two saves in one afternoon are told apart by name rather than by a browser-appended `(1)`.

Figure 2.6 shows the menu as it looks once a project has been saved and then edited.

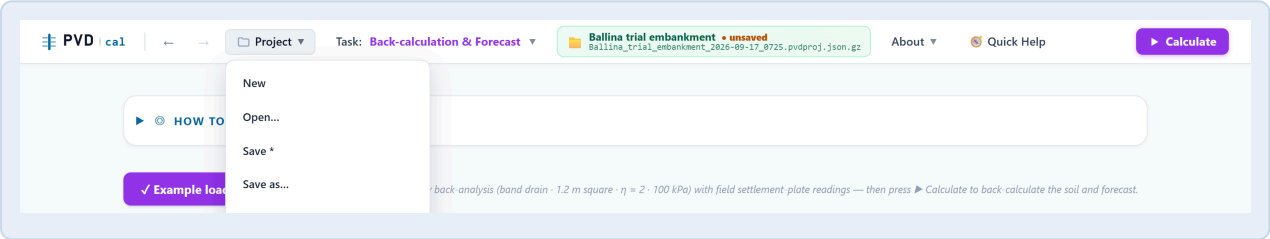


Figure 2.6. The Project menu with a project open, and the header chip beside it. Four things are visible at once: *Save* carrying an asterisk because the session has changed since it was written, *Save as…* for a second copy, the file name with its date and time, and the `.gz` that says it was compressed. The chip repeats the state in words.

Two things then look after the work you have not saved yet. The top bar marks the session **● unsaved** the moment anything changes, so the question “did I save that?” has an answer on screen. And the application keeps a **recovery copy** in the browser, refreshed every three-quarters of a minute and immediately whenever a run finishes. Close the tab with unsaved work and the browser asks you to confirm; come back later and the application offers that session back before you do anything else. Restoring it goes through the same loader a file does, so a recovered session is the same thing as a reopened project.

Project files are **compressed** by default, which takes a project carrying its results from tens of megabytes to a few — the difference between a file you can e-mail and one you cannot. The switch is in Settings, under *Project files*; turn it off if you would rather the file stayed readable text. Opening a project detects which kind it is from the file itself rather than its name, so compressed and plain projects mix freely and a renamed file still opens.

What Save actually does on your machine

The sentence above — “every Save after the first writes straight back to that same file” — is true on a computer running Chrome or Edge, and it is not true anywhere else. The difference is not the application’s choice: a browser may only write back to a file you picked if it implements the file-access

permission that makes that safe, and most do not. It is worth knowing which case you are in *before* you are standing on an embankment relying on it, so the table states all of them.

Where you are working	**What *Save* does**
Computer — Chrome or Edge	Save writes back to the file you chose, with no dialogue. The menu entry names the file it is about to overwrite. One file, updated in place, exactly as you would expect of a desktop application.
Computer — Firefox or Safari	Save hands the browser a download instead. The project keeps its name, but your downloads folder accumulates copies rather than one file being updated, and Recent Projects asks you to point at the file again.
Android — Chrome	Save downloads the project under the same name. A download never replaces an older copy of that name: Android keeps the old one and writes `name (1)` beside it. Keep the newest and delete the rest in the Files app.
iPhone or iPad — Safari	As Android: Save downloads, and Safari appends a number rather than replacing. The newest file is the one with the highest number.

Two consequences follow on a phone or tablet, and neither is obvious. Sharing a project straight into Files, Drive or a messaging app **is not possible**: phones accept only a short list of file types through the share sheet — text, spreadsheets, PDFs, images, audio and video — and a project file is none of them. And because every save leaves another numbered copy, keep **Compress project files** switched on, or a day of work becomes a folder of very large near-duplicates. Between saves the recovery copy described above is what protects the session, on every platform equally.

> **KEY IDEA**
>
> **The rule to carry.** On a computer with Chrome or Edge, Save means *update this file*. Everywhere else it means *download another copy*. The work is equally safe in both cases; only the tidying differs, and only on the second does it fall to you.

Recent Projects on the hub lists what you last worked on, and clicking an entry reopens that file: the application remembers the file itself, not just its name. The first time in a session your browser will ask permission to read it, which is the browser protecting you rather than the tool being awkward.

WATCH OUT

Three limits worth knowing before you rely on any of this.

1. **Writing back to the same file, and reopening from Recent Projects, need a Chromium browser** — Chrome or Edge. Firefox and Safari have no equivalent yet, so there Save hands the browser a download each time and Recent Projects opens the file picker for you to point at the file again. Everything else on this page works everywhere.
2. **A session started without a project cannot be saved until you name it.** Loading an example, or going straight to an input page, gives you a working session with no identity, and a file with no name, engineer or client is not worth writing. Pressing Save in that state opens the New Project form; fill it in, press Create, and the save completes on the same click with nothing lost.
3. **The recovery copy is not a backup.** It lives in this browser's storage on this machine, it holds one session — the most recent — and it is cleared the moment you save properly or press Discard. Clearing your browsing data removes it. It exists to survive a closed tab or a crash, not to replace the file you keep with the job.

KEY IDEA

Nothing leaves your machine. There is no account, no server and no cloud copy: the project file goes where you tell it, and the recovery copy stays in the browser. A project kept in a synced folder reaches your other machines because the folder syncs, not because the application sent it anywhere.

KEY IDEA

The file format is versioned. Each project file records the schema it was written with, and the application migrates older files as it opens them — a project saved by an earlier version still loads, with any block it did not contain simply starting empty. You do not need to keep an old copy of the application to read an old project.

2.3 Design and Simulate: input to output

Design and Simulate are one screen wearing two hats. Design asks "how close together must the drains be to reach my target by my deadline?" and Simulate asks "given this spacing, what actually happens, and when?" They share a soil table, a drain geometry, a loading schedule and a solver; only the last box differs, and the task menu in the top bar switches between them without retyping anything.

Figure 2.7 shows the whole input page once at a legible size. The crops that follow take it one panel at a time, and each carries the locator inset from the map so you can see where that panel sits.

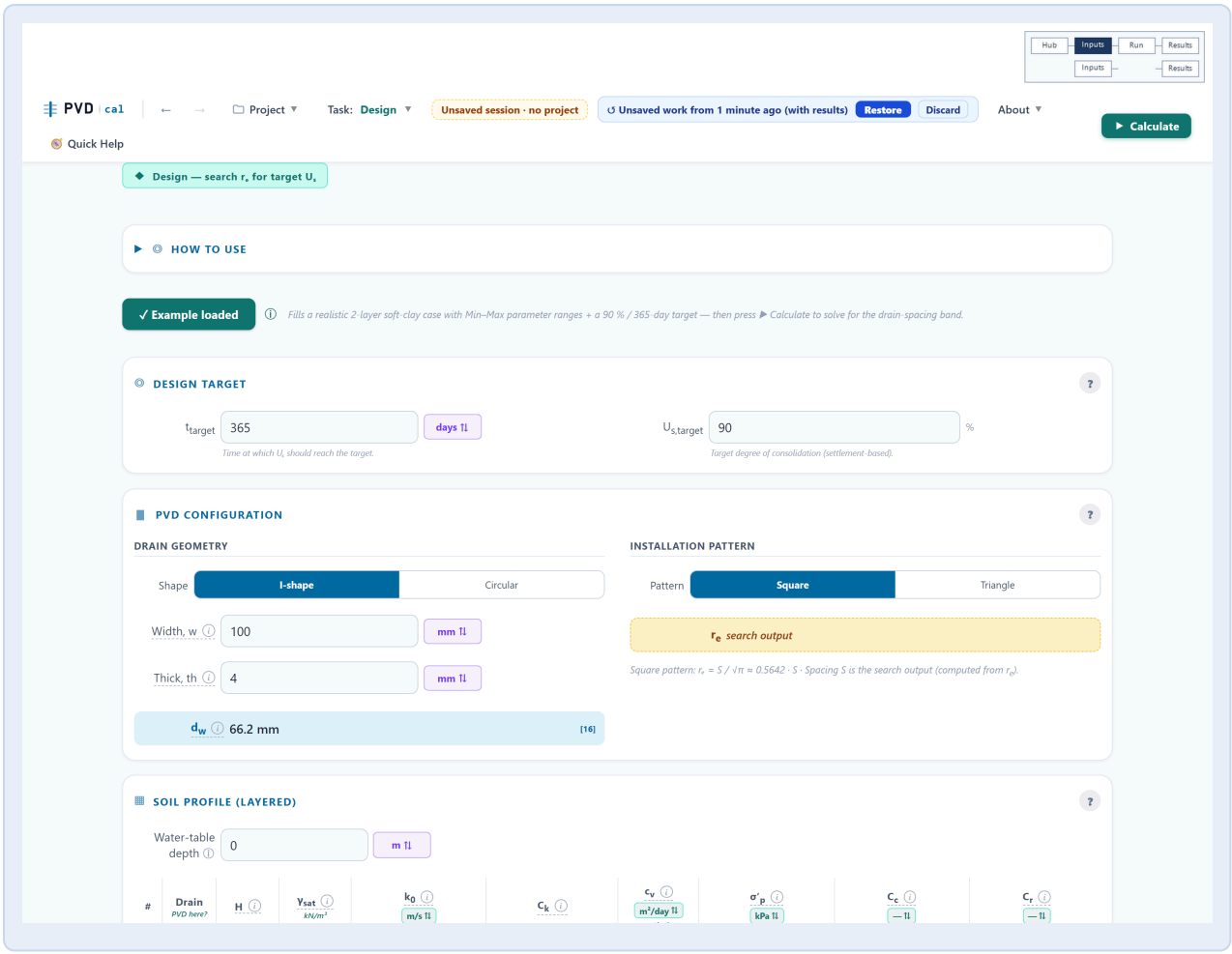
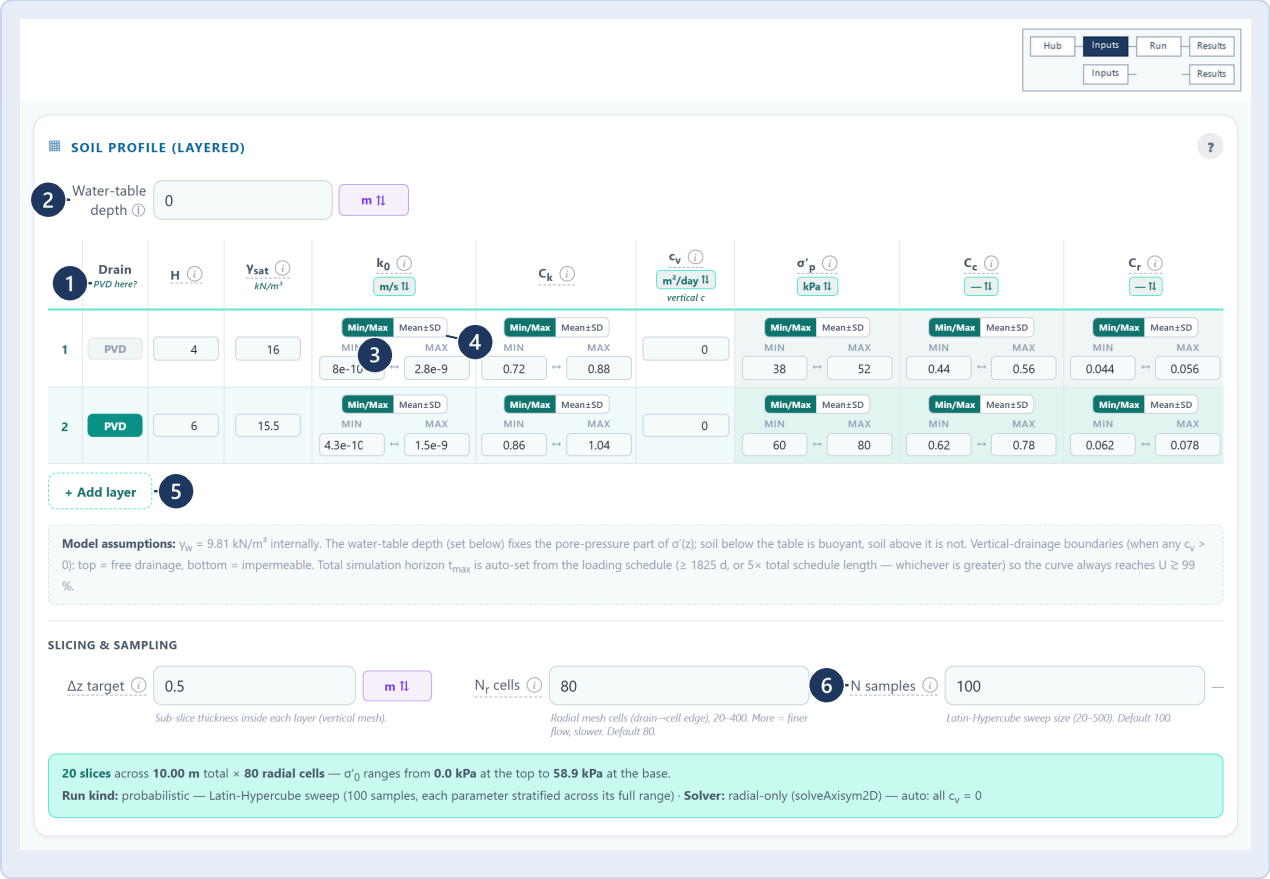


Figure 2.7. The Design input page in full, as it looks with the example project loaded. Everything below is a crop of this screen. (What the tool works out for you rather than asking: §13.3.)

The soil profile table

One row per layer, added and removed with **+ Add layer**. The first column decides whether that layer gets drains at all; the rest describe the soil. Two of the columns behave differently from the others and are worth knowing about now.

Figure 2.8 shows the table with the example's two layers in it. Two of the columns behave differently from the others and are worth knowing about now.



1. **Drain** — whether this layer has drains installed in it. A layer with no drains still consolidates, only more slowly.
2. **Water-table depth** — fixes the pore-pressure part of the initial effective stress for the whole profile.
3. **Min/Max** — the cell holds a range, and the run becomes probabilistic. This is the normal way to work.
4. **Mean±SD** — the same uncertainty stated as a mean and a standard deviation instead.
5. **+ Add layer** — appends a layer beneath the last one.
6. **N samples** — how many complete soil profiles are drawn from those ranges. The default of 100 is plenty for a first look.

Figure 2.8. The soil profile table. Each rate and compression column can hold a single value or a range, and a range is what turns a single answer into a band. (The soil model itself: §14.2. Why a range rather than a value: §17.1.)

Drain geometry, and the loading schedule

The drain panel takes the physical drain — its shape and dimensions, which the tool turns into an equivalent radius — together with the installation pattern and the smear parameters. The loading panel is a list of construction stages, each a stress increase placed over a ramp and then held.

Figure 2.9 shows the schedule for the example: a single stage, ramped and then held.

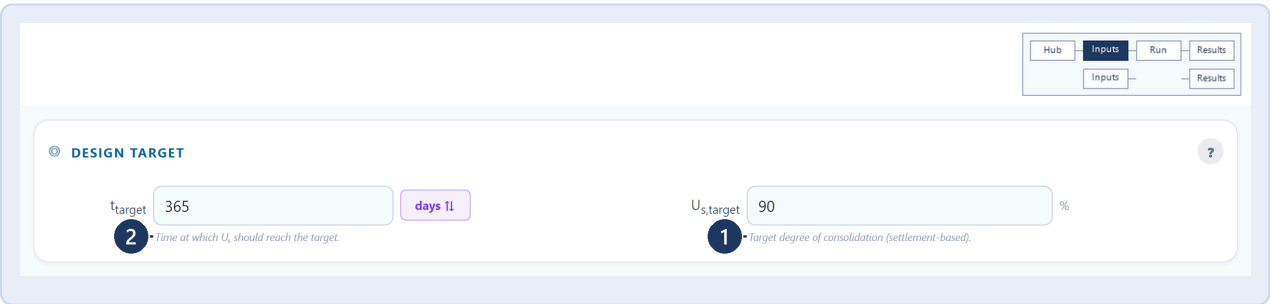
1. **Δp / cum.** — the stress this stage adds, in kPa, or the fill height and unit weight if you prefer to enter it that way; the toggle switches between per-stage and cumulative.
2. **Ramp** — how long the stage takes to place.
3. **Rest** — how long the load is then held before the next stage.
4. **Add stage** — appends a stage; remove one from its own row.

Figure 2.9. The loading schedule: one row per construction stage, ramped and then held. Stages beyond the end of any monitoring record are still applied in full. (Exactly what the solver does with a schedule: Appendix A.)

The design target, the mesh, and Calculate

This is the only panel that differs between the two tasks. In Design you state the target and the tool searches for the spacing that meets it; in Simulate you state the spacing and the tool reports what it achieves. Below it sit the numerical controls, which you can leave alone until you have a reason not to.

Figure 2.10 shows the three fields that state the design target.



1. **Target degree of consolidation** — the U the design must reach.
2. **Target time** — the deadline it must reach it by.

Figure 2.10. The design target: the degree of consolidation to reach and the date to reach it by. The installation pattern that turns the answer into a set-out spacing is set in the drain panel above (Figure 2.4). (How the search finds the spacing: §16.1. Turning a radius into a set-out spacing: §16.2.)

Figure 2.11 shows the numerical controls, which sit below the target on the same page.

1. **Δz target** — vertical cell height inside each layer.
2. **N_r cells** — radial cells from the drain to the cell edge.
3. **N samples** — the size of the probabilistic sweep.

Figure 2.11. The numerical controls. The defaults are the converged settings and need no attention for normal work. (What each knob resolves, and which one matters most: §6.6.)

Pressing **Calculate** in the top right starts the run. A progress overlay appears with live plots and a **Cancel** button; for a probabilistic Design it also reports how many samples have finished. When it completes, the application moves to the results menu on its own.

Moving between the inputs and the results, in both directions

A task has three levels: the input page, the results menu, and whichever result card you have open. You will move up and down them constantly — read a number, go back to change a value, come forward again — so it is worth knowing that there is a named button for every one of those moves, and that **none of them throws a result away**.

Where you are	Where you want to be	The control
Input page	the results	**Calculate** to compute them; or, if you have already run once, → **Go to results** beside it, which shows the results you already have without recomputing anything
Results menu	a particular card	click the card
A result card	the results menu	← **Back to results menu** in the top bar — reachable from any card, without scrolling
Anywhere in the results	the input page	← **Go to input** in the top bar
Input page, after changing a value	updated results	**Calculate** again — → **Go to results** would show you the *old* answer, because it does not recompute

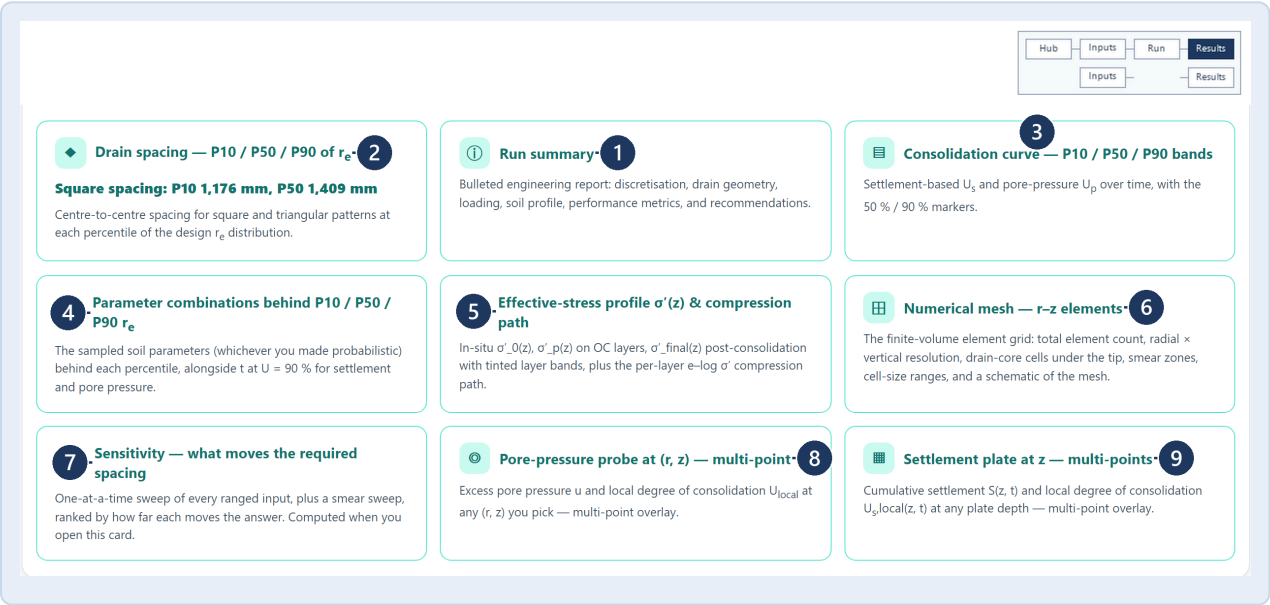
> **KEY IDEA**
>
> **Each button appears only where it applies**, which makes the top bar tell you where you are. *Go to results* is absent until a run has produced something, and disappears once you are looking at the results; *Back to results menu* appears only while a card is open. If you cannot see *Go to input*, you are already on the inputs.

> **WATCH OUT**
>
> **Go to results does not re-run anything.** That is the point of it: it is how you return to an answer you have already waited for — including the results stored inside a project file you have just reopened, which always opens on the input page. But it means that after editing an input you must press **Calculate**. Until you do, the results on offer are the ones belonging to the values you ran, not the ones now on the screen.

The results menu

Results are a **menu, not a report**. Nine cards are offered, each with a one-sentence summary of what it holds, and you open the one you want. The tiles that have a single headline number now print it — the required spacing, the time to 90 % — so the menu answers the obvious question before you click anything, and the detail behind each answer is one click away.

Figure 2.12 shows all nine cards with the application's own one-line summaries, so this book does not paraphrase them.



1. **Drain spacing** — the answer, and the first tile: centre-to-centre spacing for square and triangular patterns at each percentile. The spacing itself is printed on the tile, so a Design run can be read without opening anything. §17.4
2. **Run summary** — the whole run as an engineering report: discretisation, geometry, loading, profile and recommendations. §15.1
3. **Consolidation curve** — settlement-based U_s and pore-pressure U_p against time, with percentile bands. §15.2
4. **Parameter combinations** — which sampled soil profile sits behind each percentile. §17.4
5. **Effective-stress profile** — σ′ against depth, with the per-layer compression path. §14.2
6. **Numerical mesh** — the element grid actually used, and a schematic of it. §6.1
7. **Sensitivity** — which input moves the answer most, computed when you open it. §18.1
8. **Pore-pressure probe** — excess pore pressure at any radius and depth you choose. §15.2
9. **Settlement plate** — settlement and local degree of consolidation at any depth. §15.2

Figure 2.12. The Design results menu: nine cards, each described in a sentence, with the answer on the first tile. Several tiles carry their headline number, so the menu is a summary as well as an index. (Reading a spacing band professionally: §17.4.)

Figure 2.13 shows that card for the example project: a spacing band rather than a single number.

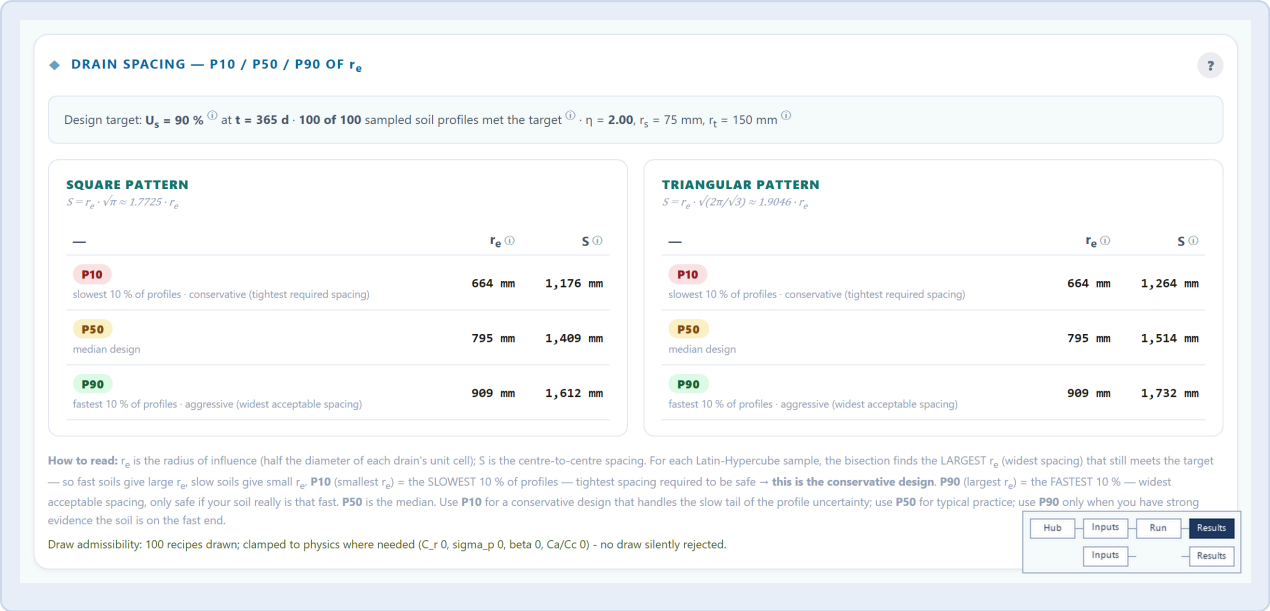


Figure 2.13. The Drain spacing card — the answer to the Design question, as a band rather than a single number. (What the band is, and what it is not: §17.6.)

Figure 2.14 shows the ranking for the example, worked out from the same machinery as a probabilistic run.

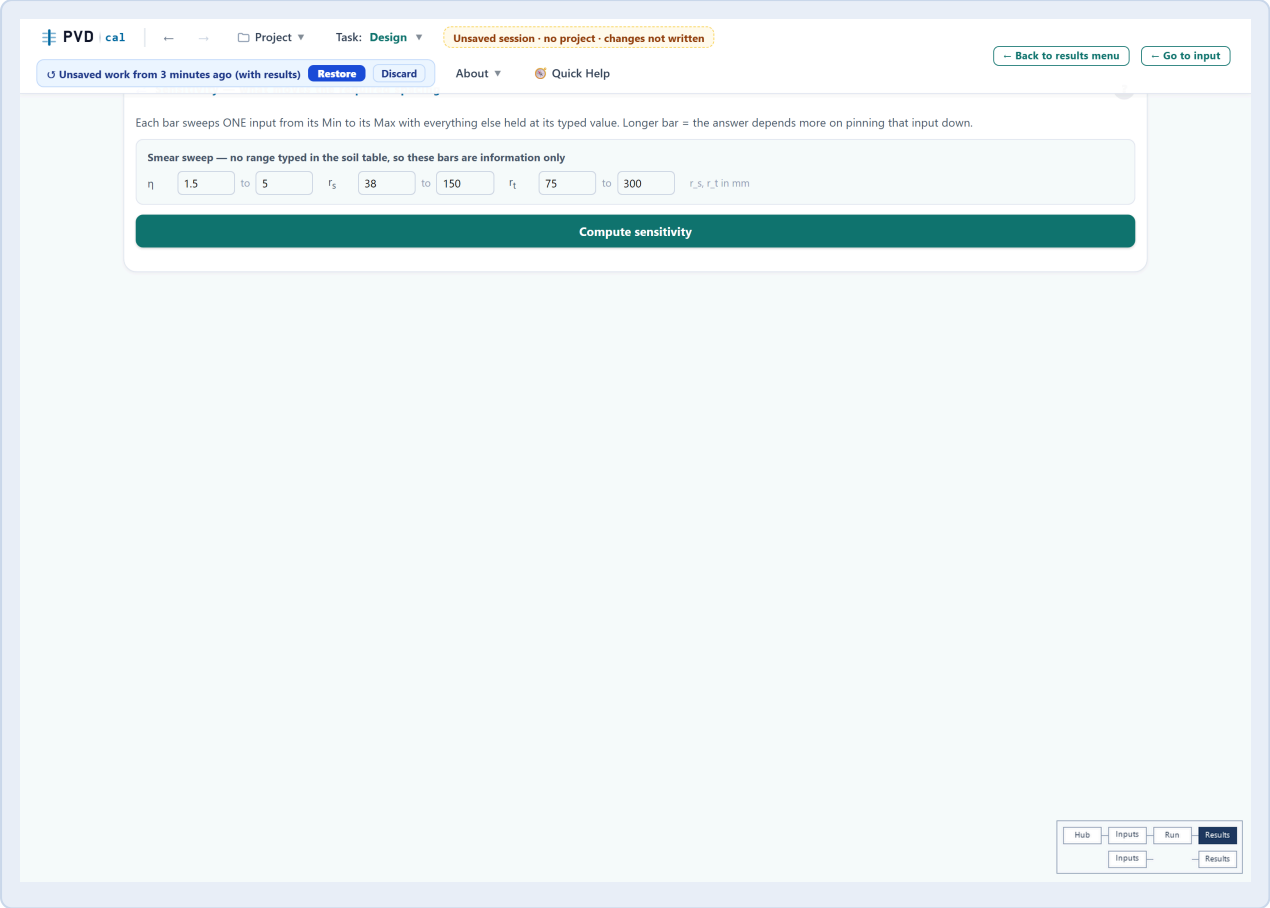


Figure 2.14. The Sensitivity card, ranking every ranged input by how far it moves the required spacing. The violet bar at the foot pushes every range to its unfavourable end at once. (How the bars are computed: §18.3. Why smear is ranged: §18.4.)

Figure 2.15 shows what the example was actually computed on.

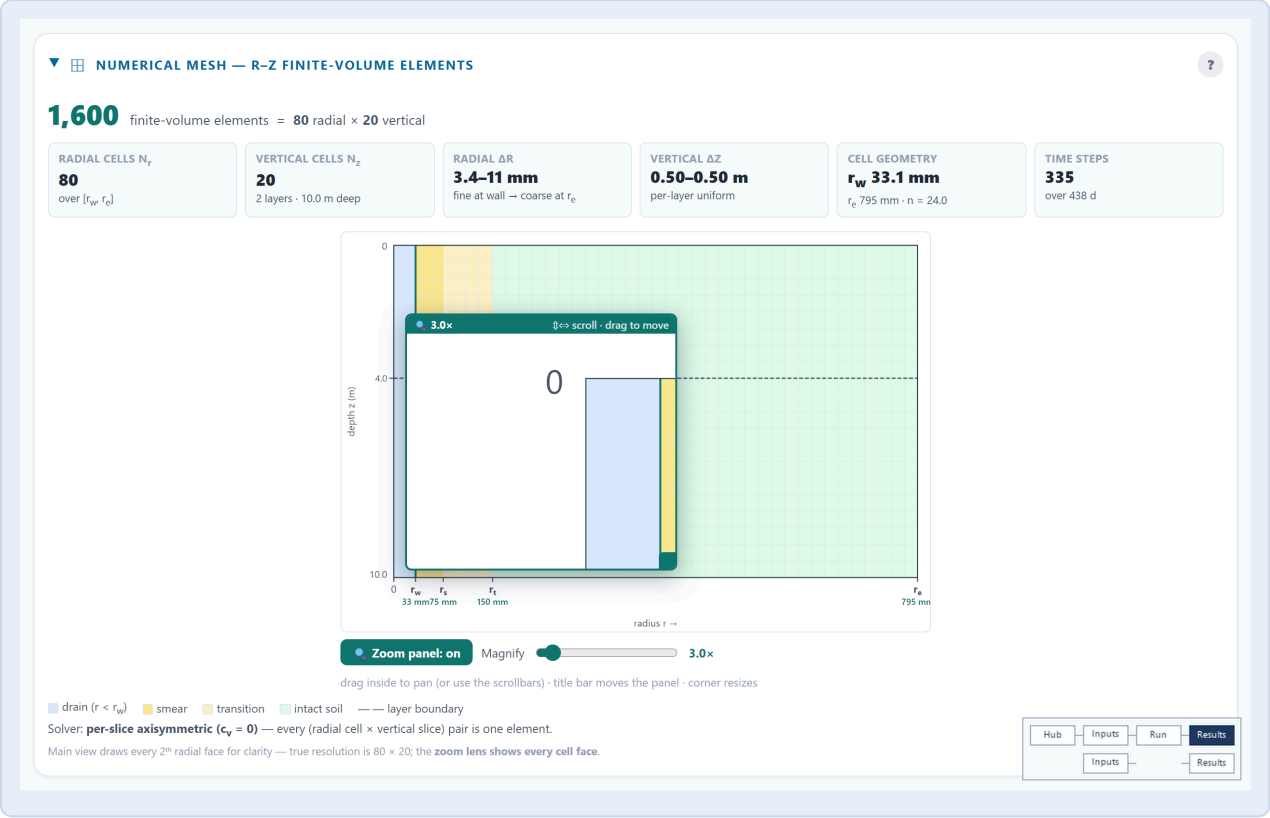


Figure 2.15. The Numerical mesh card: the element grid the run actually used. (Where the mesh puts its cells, and why: §6.1.)

Simulate, and where it differs

Switching the task menu to Simulate keeps every input you have typed and replaces the target panel with a spacing. The results menu then answers the other question — how far and how fast this spacing gets you — and its first card is the consolidation curve rather than a spacing.

Figure 2.16 shows the Simulate menu for comparison with the Design one.

Figure 2.16. The Simulate results menu. The same soil table and solver as Design, asked the opposite question. (Settlement and degree of consolidation are different questions: §15.2.)

Figure 2.17 shows one of its cards: settlement against time at the spacing you specified.

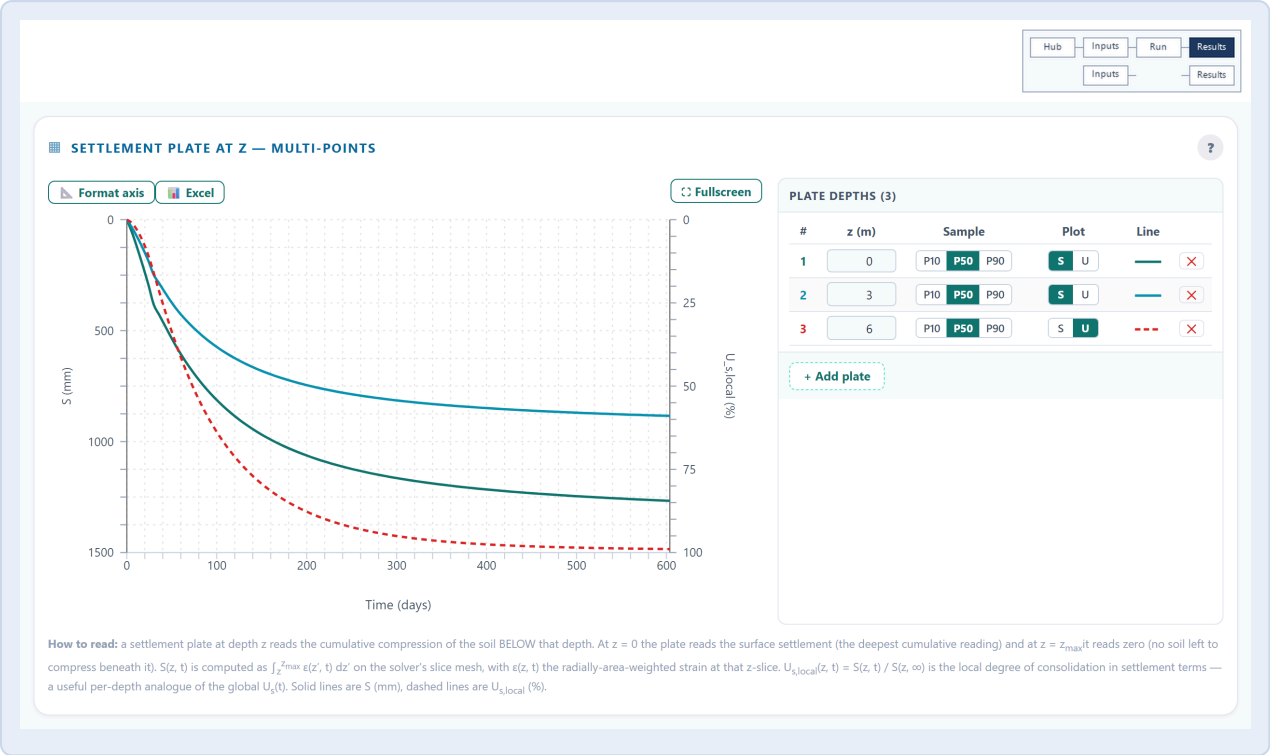


Figure 2.17. A Simulate results card: settlement against time for the spacing you specified. (Why U is a degree of primary consolidation only: §14.4.)

What every chart and every results page also gives you

Three things are available on every card in every task, and they are easy to miss because they sit quietly in a corner. They are worth knowing on your first visit rather than your tenth.

Every chart can be reformatted. The 📐 **Format axis** button on a chart's control row opens a dialogue laid out like the one in a spreadsheet: pick an axis from the tabs along the top, then set its *bounds* (minimum and maximum, blank for automatic), its *display unit*, its *major and minor step*, whether the scale is *logarithmic*, the *tick-mark* style and where the *labels* sit. Nothing is destructive — **Reset axis** at the foot puts everything back — and the display unit is the control to reach for when a reviewer wants months rather than days, or metres rather than millimetres. Beside it, **Excel** downloads that chart's own data as a spreadsheet, and the expand icon opens the plot full-screen.

Figure 2.18 shows both at once: the consolidation curve for the example, the control row beneath its title, and what Format axis opens.

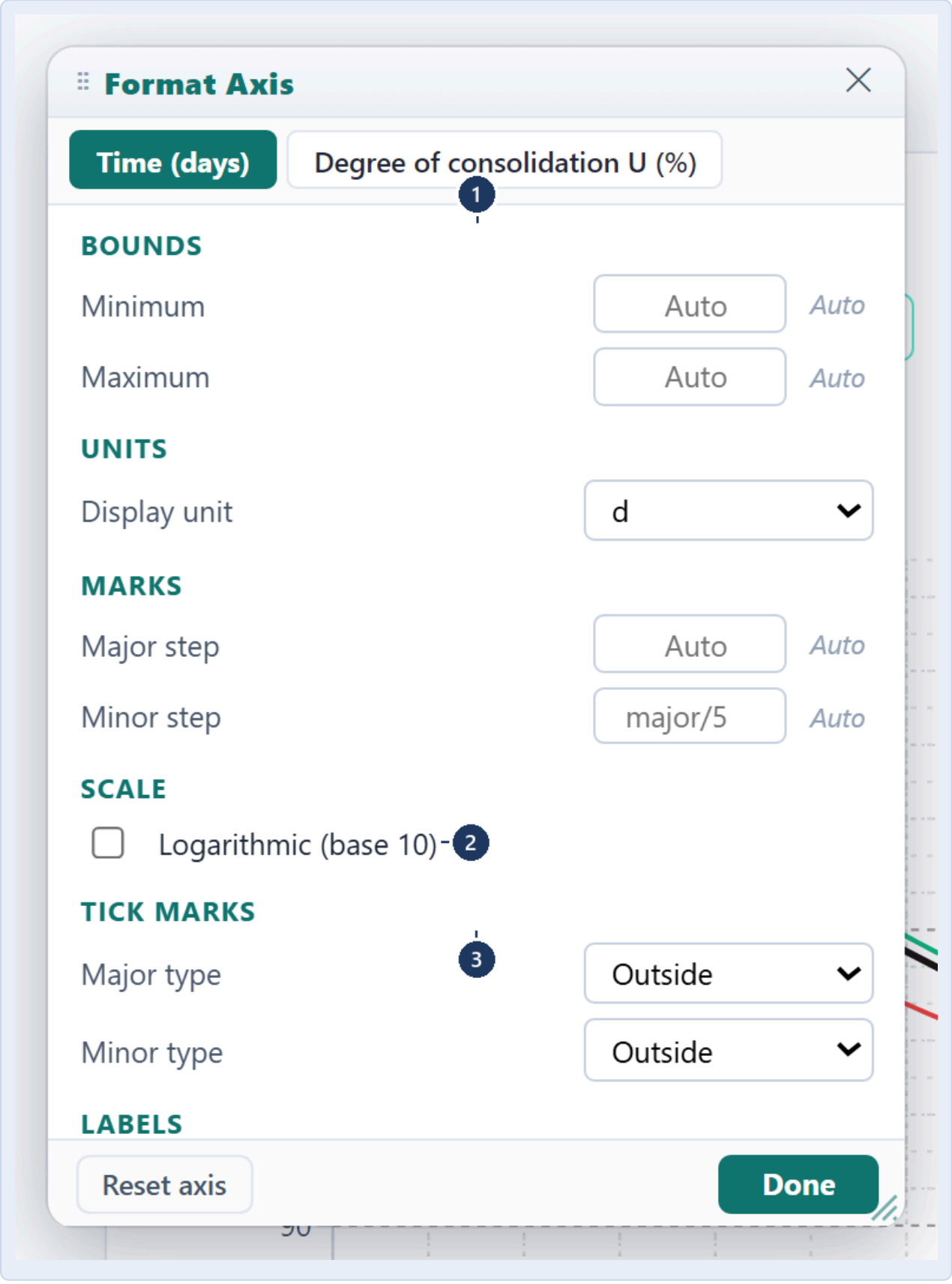


Figure 2.18. A results chart with its control row, and the Format axis dialogue open on the time axis. The same dialogue is available on every chart in all three tasks; Reset axis undoes everything in it. (What this particular chart is showing: §15.2.)

Every card explains itself. The small **?** beside a card's title is its own help text, written for that card. Where this chapter gives you a cross-reference to a section, that button gives you a paragraph on the spot.

A completed back-analysis can be exported as a report. One tile at the foot of the back-analysis results menu, **Export results — Word report**, assembles a Word document containing the project header, the soil profile, the back-calculated rate parameters, the per-plate fit charts with their band, and executive-summary, methodology and conclusions sections. It appears only once an inversion has produced a result, which is why it is absent from the forward tasks' menus. In Chrome and Edge you are offered a Save-As dialogue so you choose the folder; other browsers put the file straight into your downloads. The forward tasks have no report export of their own — from Design or Simulate the route to a document is the per-chart **Excel** download, alongside the run-summary card.

2.4 Back-calculation and Forecast: input to output

The third task runs the other way round. Instead of predicting from assumed soil properties, it takes the settlement and pore-pressure readings from an embankment that has already been built, searches for the soil properties that reproduce them, and then forecasts forward from the calibrated model. Its input page therefore asks for two things the forward tasks never need: your monitoring record, and how accurate you believe it to be.

Figure 2.19 shows the whole back-analysis input page. It is longer than the forward one, and the crops that follow work down it in order.

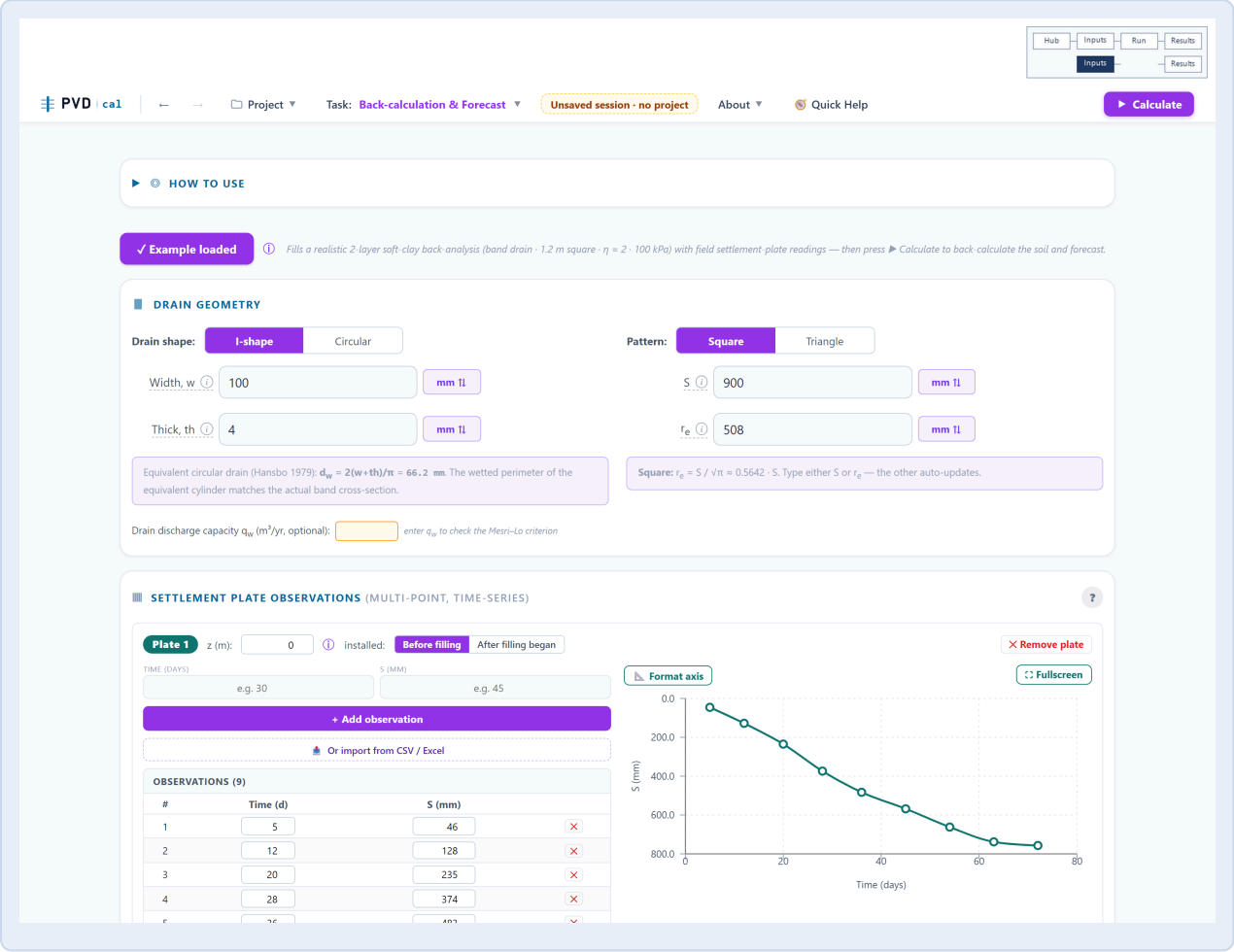
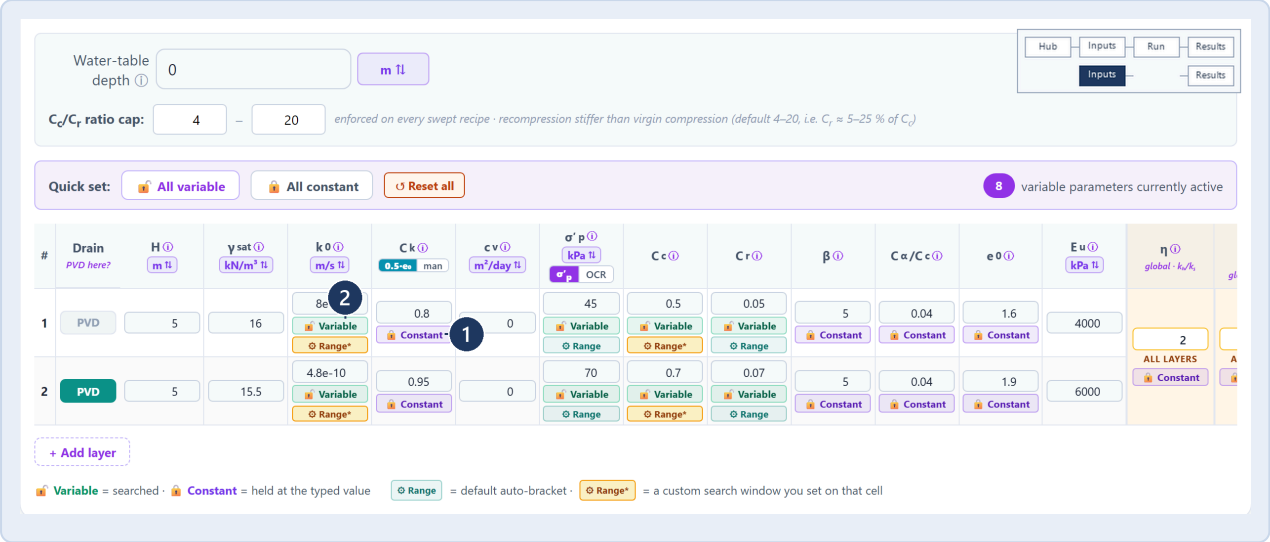


Figure 2.19. The Back-calculation input page in full. The panels below are crops of it. (What the whole method does, in one breath: §21.4.)

The soil table: Constant or Variable

Here every soil parameter carries a decision that the forward tasks do not ask for: should the search be allowed to move it? A **Constant** parameter is held at what you typed. A **Variable** one is searched, between walls you set through the ⚙ **Range** popup on the same cell.

Figure 2.20 shows the table with those decisions visible on every row.



1. **Constant** — held exactly as typed; the search will not move it.
2. **Variable** — searched between the walls you give it; the walls themselves are set in the gear popup of the next figure.

Figure 2.20. The back-analysis soil table. Every row carries a Constant/Variable decision, and the walls for a Variable parameter come from the gear popup beside it. (Which parameters a record can actually resolve: §22.1. Why the walls you type matter so much: §23.1.)

Figure 2.21 shows that popup open on one cell.

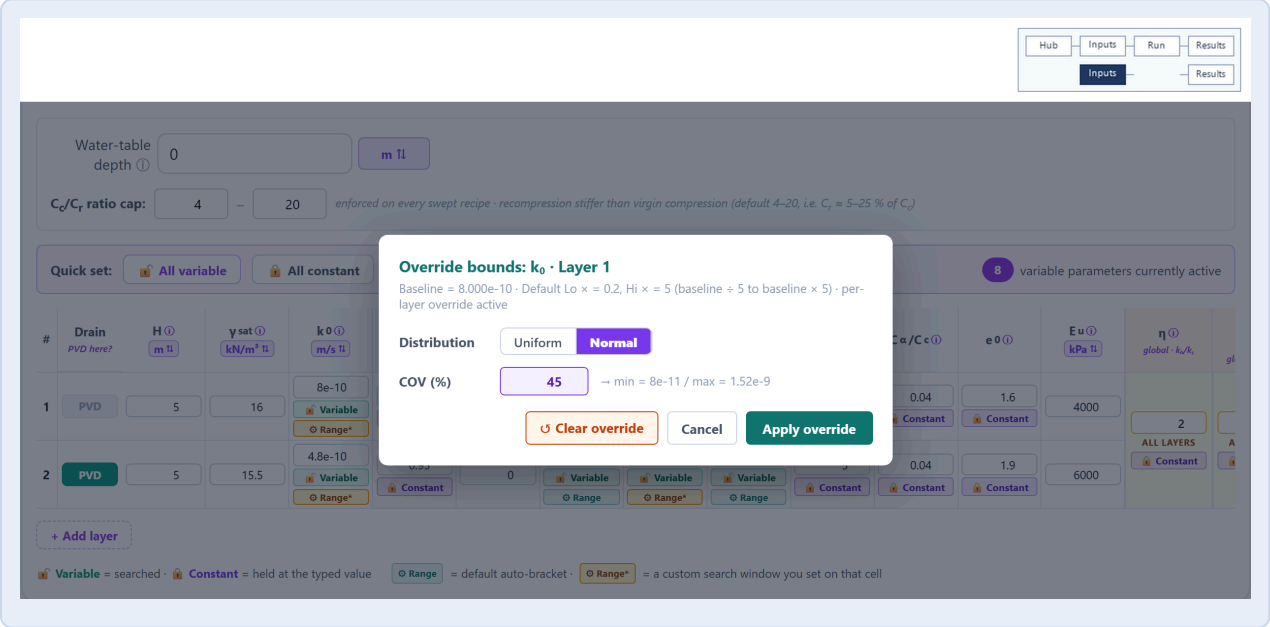


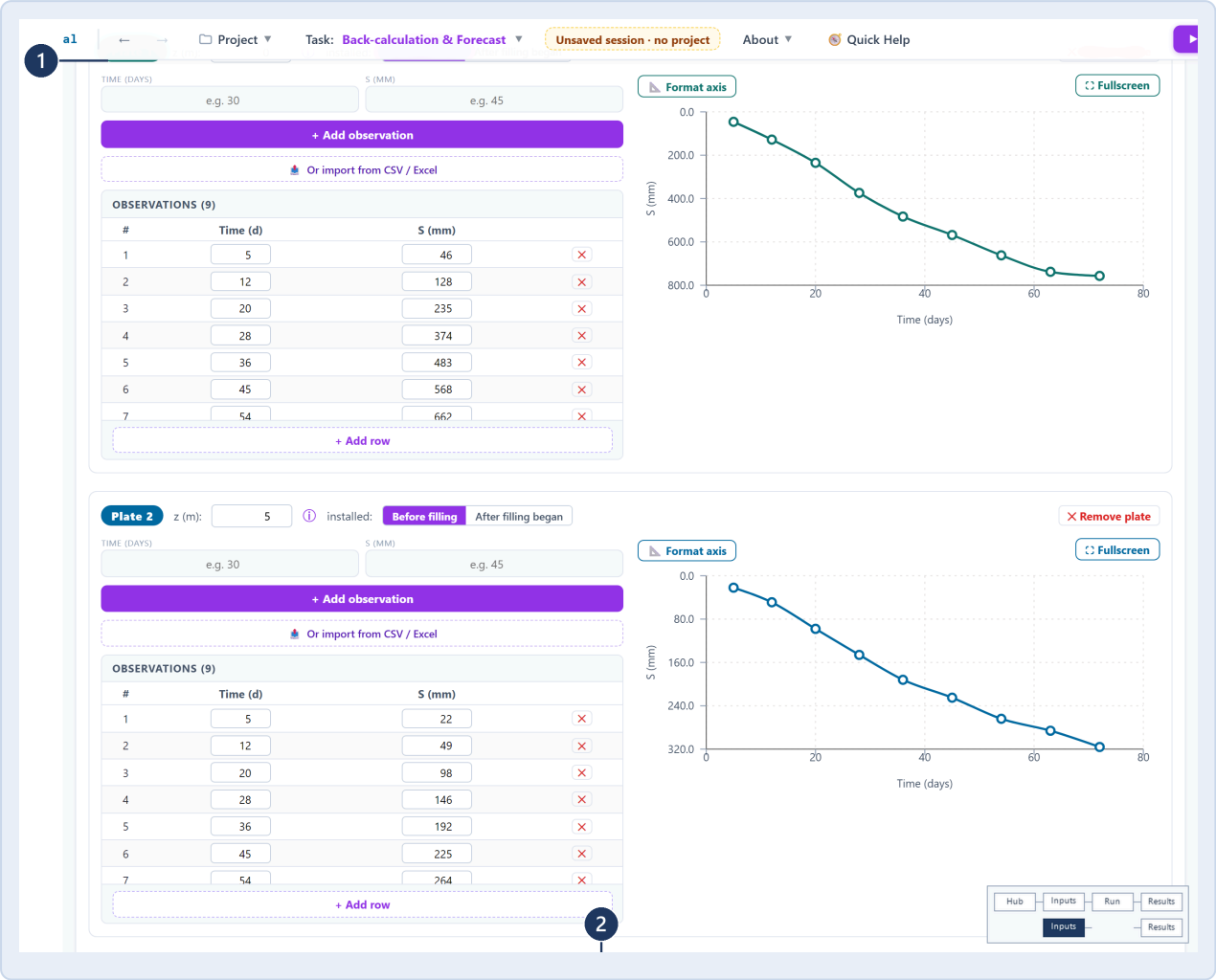
Figure 2.21. The gear popup that sets a parameter's search walls. (What the tool then does with them, including the expansion it applies: §23.2.)

The monitoring record, and two questions about the instrument

Readings are entered per instrument — settlement plates by depth, pore-pressure probes by radius and depth. Each instrument also asks two questions that are easy to answer on site and expensive to get wrong: when was it installed relative to the start of construction, and what did it read at that

moment.

Figure 2.22 shows the example's record: one settlement plate and one pore-pressure probe, with their installation answers beside them.



1. **Settlement plate observations** — one block per plate: its depth, when it was installed, and the readings themselves.
2. **+ Add plate** — a second plate, with its own depth and its own record; the fit uses every plate at once.

Figure 2.22. The monitoring record. The two installation questions per instrument — its time lag and its datum — decide what the readings are measured from. (Where zero is, and whose clock: §22.4.)

How accurate is the record, and when should the search stop

Three dialogues sit behind buttons rather than on the page, because they are set once and rarely revisited. **Observation accuracy** states how precisely each instrument can be read. **Advanced solving** chooses the numerical grid every step of the pipeline runs on. **Stopping criteria** decides when each step has done enough and hands on to the next.

Figure 2.23 shows the accuracy dialogue as the example leaves it.

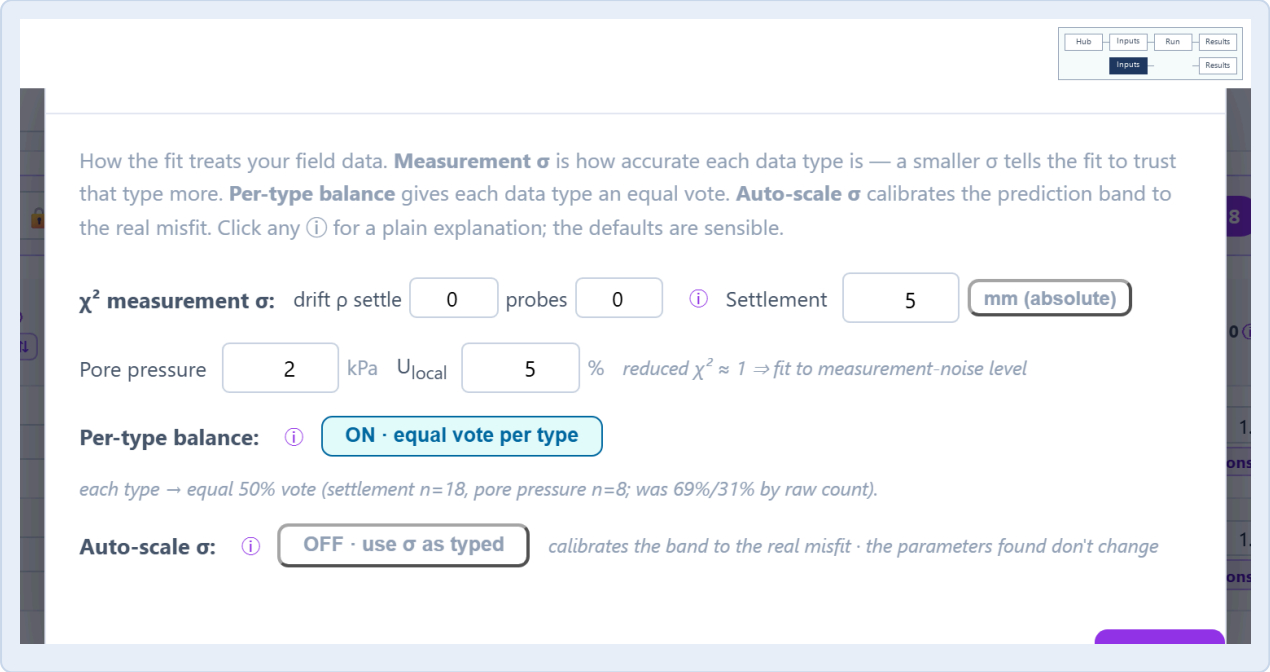


Figure 2.23. Observation accuracy: the measurement precision of each instrument family, the auto-scale switch, the per-type balance rule, and the error-drift setting. (How the score uses these, and the auto-scale caveat: §21.2.)

Figure 2.24 shows the solving dialogue, with the preset the example uses selected.

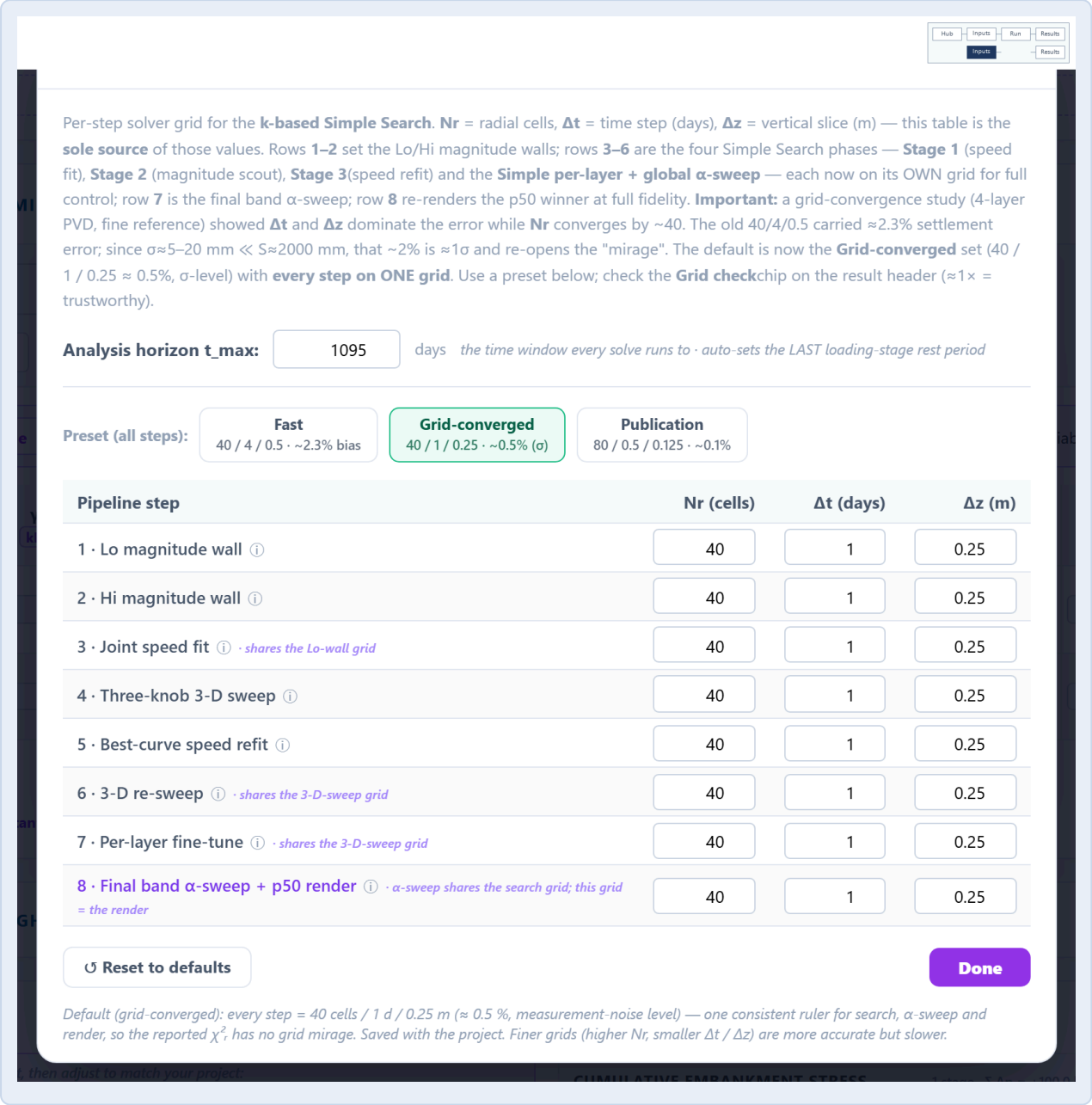


Figure 2.24. Advanced solving, with the three one-click grid presets. Every step of the pipeline is put on the same grid, deliberately. (Which knob deserves your effort: §6.6. What the presets cost and deliver: §6.7.)

Figure 2.25 shows the dialogue as it opens. You can run the whole book without touching it; it is here for the record that needs a rule loosened, and for the run you need to fit into the time you have.

COMMONLY ADJUSTED

Good-enough fit — χ^2_r target	1	χ^2_r · default 1.0
Plateau window	2	passes · default 2
Whole-search budget	0	minutes (0 = off) · default 0

PER STEP

Step 6 — re-sweep rounds	6	rounds · default 6
Step 7 — per-layer passes at most	3	passes · default 3
Step 7 — retire a layer below	0.01	% of χ^2_r · default 0.01
Step 7 — ...for this many passes	3	passes · default 3
Step 8 — effective-recipe floor	10	recipes · default 10

WHEN IS A PLATE "DONE"? (ALL FOUR MUST PASS)

R^2 at least	0.98	– · default 0.98
KGE at least	0.95	– · default 0.95
NRMSE at most	3	% · default 3
Absolute RMSE at most	20	mm · default 20

CYCLE AND PASS RULES

Per-cycle stop ⓘ · $\Delta\chi^2_r$ < 8 % for 2 cyc

Per-layer stop ⓘ · $\Delta\chi^2_r$ < 10 % after compulsory 3 passes

Dual-track ⓘ if 2nd winner within 5 % of best χ^2_r

Every value in this dialogue defaults to what the search used before these controls existed, so a run you have not touched behaves exactly as it always did. Looser criteria finish sooner and can stop short of the best fit; tighter ones keep refining for gains that may be smaller than your measurement error.

Figure 2.25. Stopping criteria. Four groups: the three worth adjusting most often, the per-step rules, the four gates that decide when a settlement plate counts as fitted, and the cycle and pass rules shared between steps. Every value shown is the default. The ⓘ beside a criterion explains what it does in plain words; hovering the label itself adds when to change it. (What ends each step, and which cases are a reason to change a rule: §24.9.)

The run, and its two exits

A back-analysis takes minutes rather than seconds, and the overlay names the stage it is in as it goes. It has two exits and they are not the same. **Stop & accept** ends the search and keeps the best answer found so far. **Cancel** abandons the run and keeps nothing.

Figure 2.26 shows the overlay part-way through the example inversion, with both exits visible.

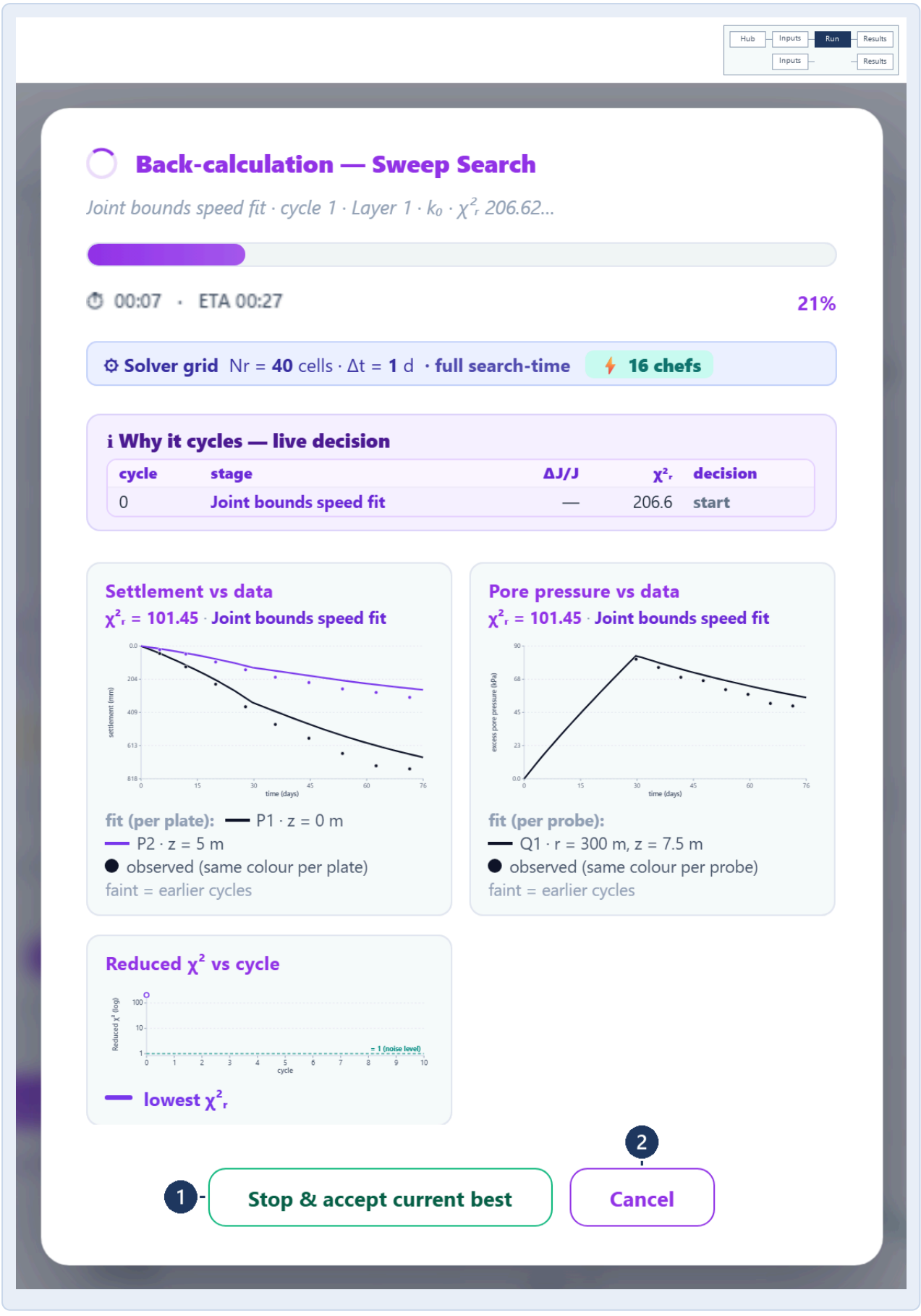


Figure 2.26. The run overlay during a back-analysis, showing the stage in progress and both exits. (What the stages are doing: §21.4.)

The results page, top to bottom

The back-analysis results menu holds seven cards, and its shape is worth stating plainly: **the numbers you came for are nearly all inside the first card**, *Forecast & back-calculated parameters*, whose tile also prints the forecast itself. The forecast and its ranges, the fitted rate parameters, the fit statistics and the observability panel are all blocks within it, not separate menu entries. The other six cards are the supporting views: the effective-stress profile, the settlement-plate and pore-pressure comparisons against your readings, the mesh, the animated contour, and the export tile.

Read that one card from the top down, because it is ordered the way the argument runs: how well the model fitted your readings first, what it forecasts second. A forecast from a poor fit is not worth having, and the card puts the evidence before the answer.

Figure 2.27 shows the whole menu; the order below is the order to read it in.

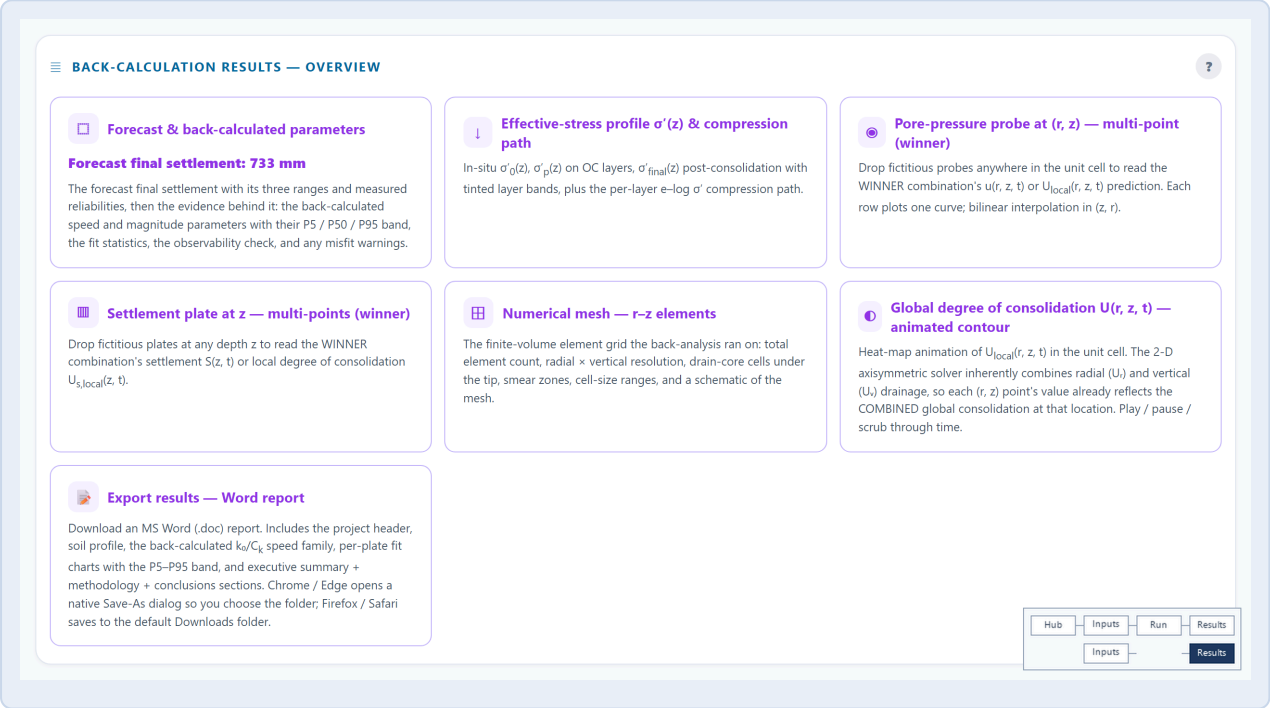
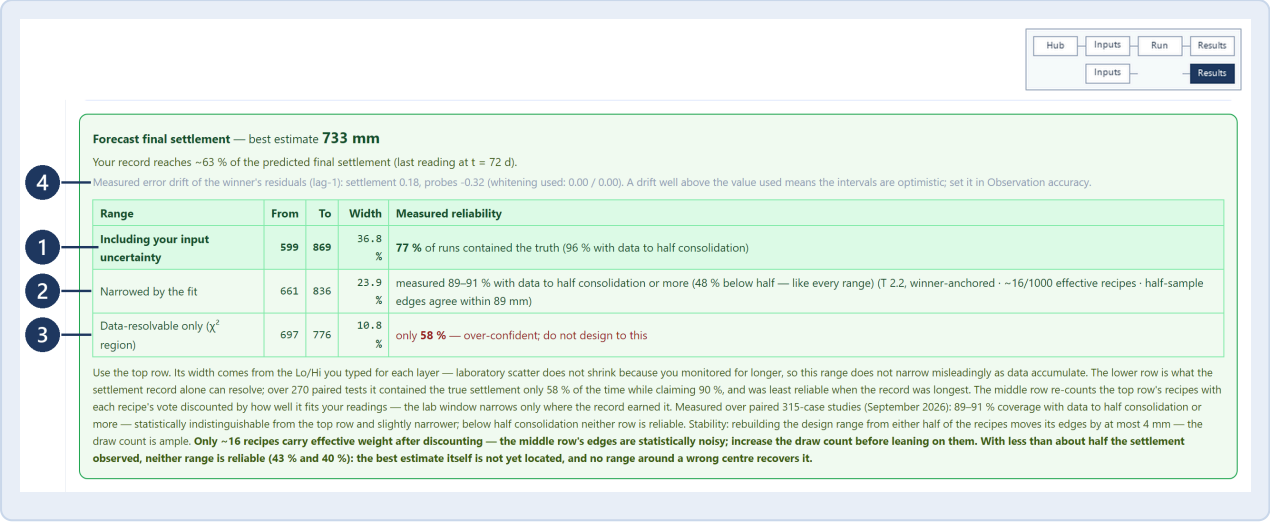


Figure 2.27. The back-analysis results menu. Fit first, forecast second. (Reading the result page professionally: §25.3.)

Figure 2.28 shows the forecast card for the example project.



1. **Including your input uncertainty** — the range to design to. Its width comes from the Lo/Hi you typed, so it does not narrow misleadingly as data accumulate.
2. **Narrowed by the fit** — the same recipes, each vote discounted by how well it fits your readings.
3. **Data-resolvable only** — what the record alone pins down. Printed because it is informative, labelled because it is over-confident.
4. **Measured error drift** — the drift found in this winner's residuals, beside the drift you told the score to assume.

Figure 2.28. The forecast block: one best estimate, three ranges, and a measured reliability printed beside each. Design to the top row. (What each row means and which to trust: §25.6. How the reliabilities were measured: §30.7. The error-drift line: §21.2.)

Below the forecast sits the observability check. Figure 2.29 shows it for the example, layer by layer, with the search range each layer was actually given.

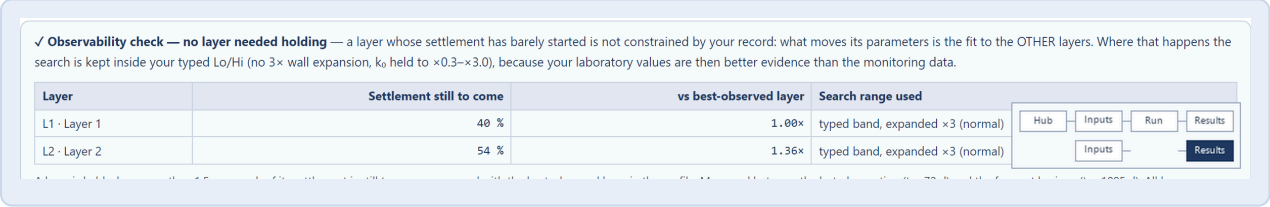


Figure 2.29. The observability check, reporting per layer how much settlement is still to come — that is, how much of the profile your record can actually see. On the example nothing needed holding, and the panel says so rather than staying silent. (Why this is per layer and not per profile: §25.8.)

Further down the same card, Figure 2.30 gives the back-calculated parameters themselves, one row per parameter per layer, beside the value you supplied.

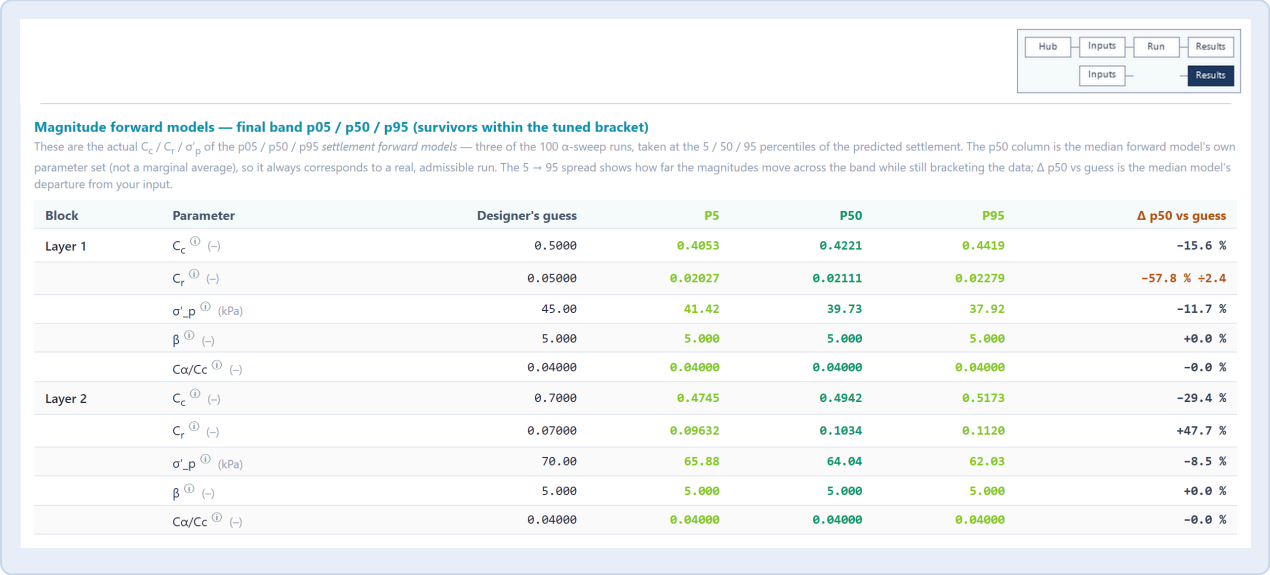


Figure 2.30. The magnitude table: what the search made of each compression parameter, against what you typed. The right-hand column is the one to read first — it is how far the calibration moved your own value. (What these parameters are and which the record can resolve: §22.1. What to write down from here: §25.5.)

Directly below the magnitude table, and read the same way, is the speed table. Figure 2.31 shows it for the example: the rate parameters the calibration settled on, against the values that were typed.

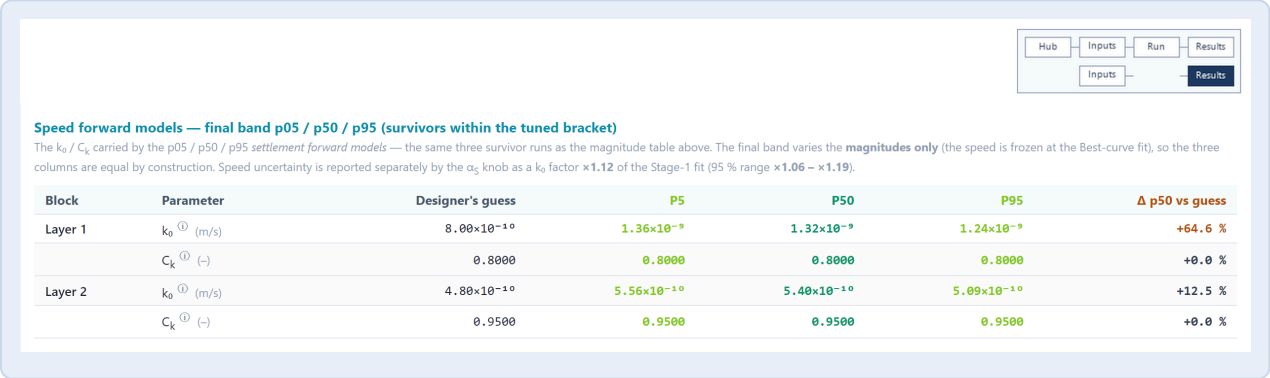


Figure 2.31. The speed table, laid out exactly like the magnitude table above it: the k_0 and C_k the calibration settled on. The three percentile columns are equal here by construction — the final band varies the magnitudes only — and speed uncertainty is reported separately, as a factor on k_0. (What the speed family is, and how much of it a record can pin down: §25.4. Why smear rides with it: §22.2.)

The **Export results** tile at the foot of this menu is described in §2.3, along with the chart controls that apply here too.

The three banners, and what each is telling you

Coloured banners appear on this page when the run has something to declare. They are easy to scroll past and they are the most important text on the page, so they are collected here: meeting one for the first time in the middle of a real job is the wrong moment to wonder what it means. Which of them you see depends on the run — a clean inversion raises none.

Figure 2.32 shows all three together so that none of them is a surprise.

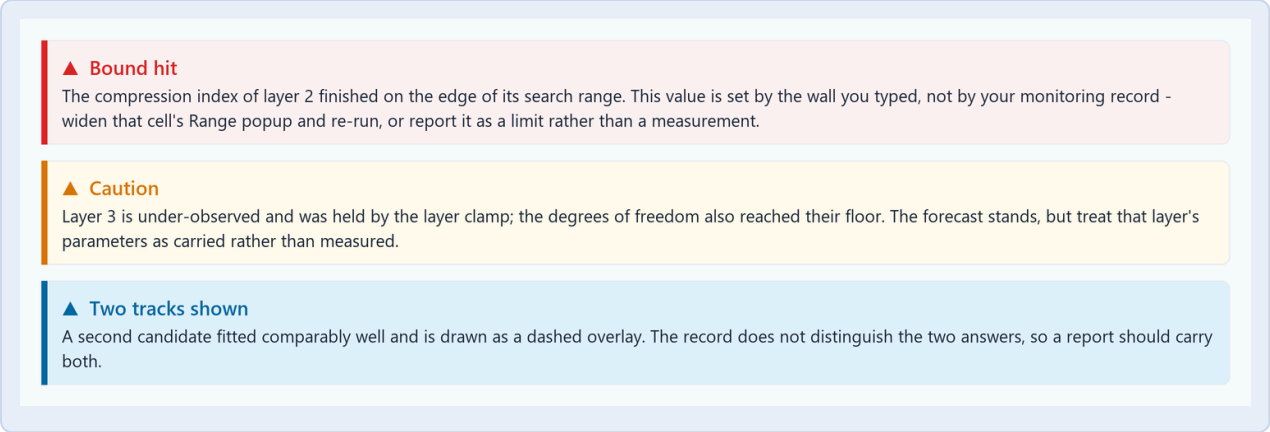


Figure 2.32. The three kinds of banner, drawn together for reference: red withdraws a number's status as a measurement, amber reports an intervention and leaves the numbers usable, and the dual-track notice says the record could not choose between two answers. *Composed, not a screenshot* — the banners are conditional and no single run raises all three, so these are drawn from the application's own colours and wording. (Bound-hit in full: §25.7. What the search does when a layer is under-observed: §25.8.)

The example project raises one of them, so Figure 2.33 shows a bound hit exactly as it appears in place, above the parameter table it applies to.

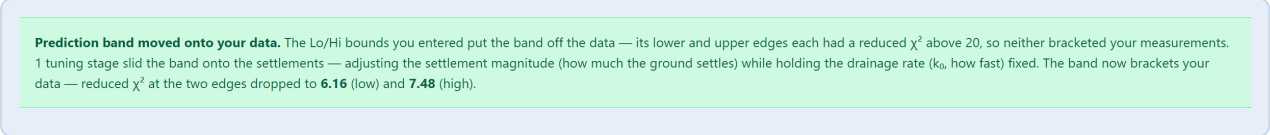


Figure 2.33. A real bound-hit banner, from the example project's own run: one fitted parameter finished against a wall that was typed rather than measured. (What to do about it: §25.7.)

2.5 Your first fifteen minutes

This is the shortest route from opening the application to understanding what it does. Follow it in order; every number you see on the way should match the ones printed here, because they come from the same example project.

1. **Load an example.** From the hub, **Example projects › Simulate**. The input page fills in with a complete two-layer profile (Figure 2.3).
2. **Press Calculate, and read the curve.** The run takes a few seconds. Open the consolidation-curve card and find the time at which U_s reaches 90 %. That single number is what the whole forward task exists to produce.
3. **Switch the task to Design and press Calculate again.** Nothing needs retyping. This time the tool searches for a spacing, and the Drain spacing card reports it as a band: the example gives a P10–P90 spacing of about 650–920 mm, with P10 the conservative choice.

4. **Open the Sensitivity card.** It computes when opened, then ranks every ranged input by how much it moves that spacing. Hydraulic conductivity dominates; smear comes second, ahead of every compression parameter.
5. **Open the Numerical mesh card.** Confirm for yourself what the answer was computed on: the element count, the radial and vertical resolution, and where the cells are concentrated.
6. **Then choose where to go next.** If you are sizing a scheme, read Part Two. If an embankment is already built and instrumented, read Part Three. If you want to know what the numbers rest on before you rely on them, read Part One.

> **KEY IDEA**
>
> **If a number differs from the book.** The examples are the source of the worked figures, not a copy of them, so a disagreement means either the application has changed since this chapter was captured or something is genuinely wrong. Either is worth reporting; neither is your error.

2.6 Where everything is explained

This table is the chapter's contract. Every screen and every results card in the shipped application appears in the left column, against the section that explains it in the right. If you can click on it, it is listed here.

Screen, panel or card	Explained in
The hub	
Open new project (the project-identity form)	§2.2
Open existing project (.pvdproj.json)	§2.2
Recent Projects list	§2.2
Project menu: New, Open…, Save, Save and Close, Settings	§2.2
→ Go to results / ← Go to input / ← Back to results menu	§2.3
The back and forward history arrows	§2.3
Settings: unit system, engineer details, colour theme	§2.2
Settings › Project files: compress project files	§2.2
Settings › Charts: show the previous option as a faint dashed line	§2.2
What Save does on a computer, on Android and on iOS	§2.2
Example projects (Design, Simulate, Back-calculation)	§1.3
Quick Help	this chapter
Design and Simulate — input	
Soil profile table; single value, Min/Max, Mean±SD	§13.3, §17.1
Drain geometry, shape and equivalent radius	§13.2
Smear parameters (η, r_s, r_t)	§18.4
Loading schedule (stage, ramp, rest)	§15.1, Appendix A

Screen, panel or card	Explained in
Vacuum preloading	§7.3
Design target (U, time, pattern)	§16.1, §16.2
Mesh controls (Δz, N_r) and N samples	§6.6, §17.2
Design and Simulate — results cards	
Run summary	§15.1
Drain spacing — P10 / P50 / P90	§17.4
Consolidation curve — U_s and U_p	§15.2
Parameter combinations behind the percentiles	§17.4
Effective-stress profile and compression path	§14.2, §14.5
Numerical mesh — r–z elements	§6.1, §6.4
Sensitivity — what moves the answer	§18.1, §18.3
Pore-pressure probe at (r, z)	§15.2
Settlement plate at z	§15.2
Back-calculation — input	
Soil table, Constant / Variable	§22.1, §23.1
⚙ Range popup (search walls)	§23.1, §23.2
Settlement plate and pore-pressure probe records	§22.4
Instrument time lag and datum	§22.4
Observation accuracy (σ, auto-scale, balance, error drift)	§21.2
Stopping criteria	§21.4
Advanced solving and the three grid presets	§6.6, §6.7
Run overlay: Stop & accept, Cancel	§21.4
Back-calculation — results cards	
Score vs cycle chart	§25.1
Sweep heat-map (shown on the run overlay, not a results card)	§25.2
Fit statistics (RMSE, NRMSE, R^2)	§25.3
Speed card (rates and smear)	§25.4
Forecast & back-calculated parameters (the card holding most results)	§25.5, §25.6
Forecast card and its three ranges	§25.6, §30.7
Bound-hit banner	§25.7
Observability panel	§25.8
Multi-point plate and probe cards	§25.3

Screen, panel or card	Explained in
Everywhere	
The “?” card-help button on every card	this chapter, §2.3
📐 Format axis — bounds, display unit, steps, log scale, ticks, labels	§2.3
Reset axis	§2.3
Excel — download a chart’s data	§2.3
Expand chart to fullscreen	§2.3
Export results — Word report (back-analysis only)	§2.3, §25.5

RECAP

Keeping this chapter true. The screenshots are generated, not taken by hand: `capture_tour_d11.js` drives the application, `annotate_tour_d11.py` draws the callouts from measured element positions, and this chapter is assembled by `gen_tour_body.py`, which refuses to build if a legend and its figure disagree about how many callouts there are. When a screen changes, the release re-runs those three scripts rather than editing an image.

PART ONE

Numerical Foundations

What the solver actually does: the equation, the elements, the mesh, the boundary conditions, and the measured evidence that the answers converge.

What this part is for

A design tool asks you to trust numbers that decide how much surcharge to place and how long to wait. That trust should be earned rather than assumed. This document opens the solver up: what equation is discretised, what an element is, where the mesh puts its cells and why, exactly which boundary conditions are imposed, and how layer and smear interfaces are handled. Most importantly, it reports the measured error, and how that error shrinks when you refine.

1. State the partial differential equation the tool solves, and what each term represents.
2. Describe the element: a cell-centred finite volume on an annular control volume.
3. Explain where the mesh puts its cells, why the zone boundaries are always cell faces, and how to change it.
4. List every boundary condition, including the vacuum and partial-penetration cases.
5. Say how conductivity contrasts at smear and layer interfaces are handled conservatively.
6. Show the measured convergence rate and the accuracy of the default mesh.
7. Read the closed-form verification results, and know what they do and do not prove.
8. Check the numerical quality of your own run.

CHAPTER 3

What is being solved

3.1 Two solvers, one kernel

The tool carries two production solvers. They share the same spatial discretisation: the same elements, the same mesh logic, the same treatment of interfaces. They differ only in whether the vertical direction is present, and therefore in how they march forward in time. Table 3.1 compares them.

Table 3.1. The two production solvers.

	`solveRC1D`	`solveRC2D`
Geometry	radial only, *r*	coupled radial & vertical, *r–z*
Discretisation	cell-centred finite volume, harmonic-mean face conductivity	
Time integration	Crank–Nicolson	Peaceman–Rachford ADI
Linear solve	Thomas (tridiagonal)	Thomas, twice per step
Unknowns	N_r	$N_z \times N_r$
Carries	smear, vacuum, creep, hysteresis (the $c_v = 0$ stacker adds the E_u immediate-settlement pass on top)	all of those, plus vertical drainage, partial penetration and immediate settlement

A third routine, `solveAxisym2D`, is not an independent solver: it is a *stacker*. It applies `solveRC1D` independently to each depth slice, then composites the slice results into the whole-profile response. The steps are worth stating exactly, because nothing about them is approximate:

1. When $c_v = 0$ there is no vertical flux, so the depth slices are mathematically uncoupled. Nothing that happens in one slice can influence another.
2. Each slice is therefore solved on its own by `solveRC1D`, at its own in-situ effective stress and its own layer properties.
3. The slice results are composited. Settlement is the thickness-weighted sum: $S(t) = \Sigma\, \varepsilon_{slice}(t) \cdot h_{slice}$ over all slices. The degree of consolidation uses the same weighting as a ratio: $U_s(t)$ is the composited *primary* settlement at time t divided by the composited ultimate primary settlement.

Because the slices really are uncoupled at $c_v = 0$, this stacking is **exact for that case**, not an approximation — and it is much faster than carrying a two-dimensional grid whose vertical coupling is identically zero. Figure 3.1 shows the construction.

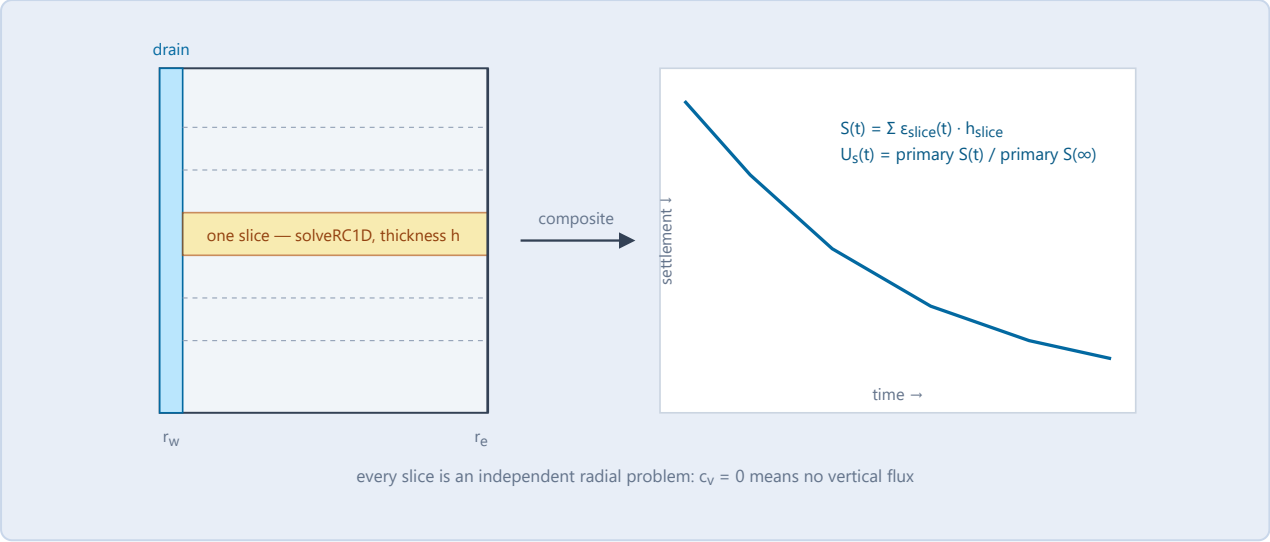


Figure 3.1. Schematic. `solveAxisym2D` for $c_v = 0$: each slice an independent radial problem at its own effective stress, the slice strains thickness-weighted into settlement — exact, because the slices are truly uncoupled.

3.2 Which solver runs, and when

Table 3.2. Solver routing. The deciding inputs are c_v and partial penetration.

Task	$c_v = 0$	$c_v > 0$
Design	`solveAxisym2D` → `solveRC1D` per slice	`solveRC2D`
Simulate	`solveAxisym2D` → `solveRC1D` per slice	`solveRC2D`
Back-calculation	`solveAxisym2D` → `solveRC1D` per slice	`solveRC2D` — the standard path

"The standard path" means the route taken by the overwhelming majority of back-calculation runs: a field profile normally carries $c_v > 0$, so the coupled solver runs. The $c_v = 0$ column is the special case, chosen deliberately when vertical drainage is to contribute nothing. Table 3.2 gives the full routing. The deciding input is c_v, with one exception: a layer whose drain is switched OFF (partial penetration) forces `solveRC2D` even at $c_v = 0$, because the decoupled stacker has no vertical flow and a drain-off layer there would never consolidate.

> **WATCH OUT**
>
> **Setting $c_v = 0$ is not a shortcut that costs accuracy.** It is a modelling statement: *vertical drainage contributes nothing*. For a deep deposit under a wide embankment with closely spaced drains that is very nearly true, because the vertical drainage path is tens of metres while the radial one is under a metre. For a thin layer, or a wide drain spacing, vertical flow matters and $c_v = 0$ is conservative — it will predict slower consolidation than reality.

3.3 The unit cell

Both solvers work on the classical **unit cell**: one drain at the centre of the soil cylinder it dewaters. A triangular or square grid of drains is replaced by an equivalent cylinder of radius r_e carrying the same area per drain. Because every drain in the field has an identical neighbourhood, no water crosses the cylinder wall — which is precisely the zero-flux outer boundary condition used in §7.4. Figure 3.2 shows the cell and both solvers' place in it.

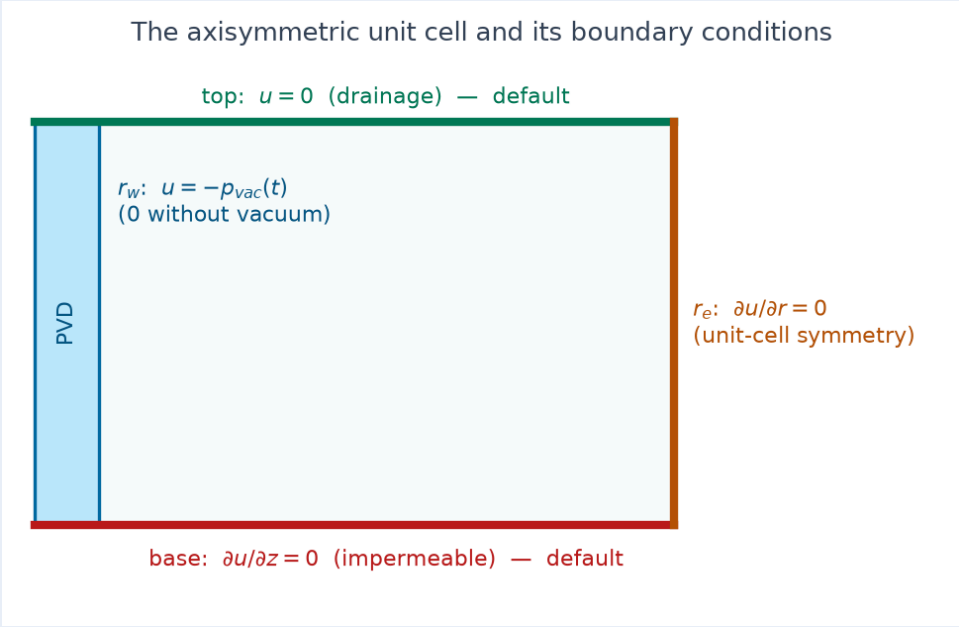


Figure 3.2. The axisymmetric unit cell solved by `solveRC2D`. `solveRC1D` solves a single horizontal line through it. The drain occupies $r < r_w$; the soil annulus runs out to the no-flow cylinder wall at r_e.

CHAPTER 4

The governing equation

4.1 The equation in its conservative form

The excess pore-water pressure $u(r, z, t)$ obeys

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[r\, c_h(r,z,t)\, \kappa(r)\, \frac{\partial u}{\partial r}\right] + \frac{\partial}{\partial z}\left[c_v(z)\, \frac{\partial u}{\partial z}\right] + \frac{dp}{dt} + \dot{s}_{\text{creep}} \tag{4.1}$$

on $r \in [r_w, r_e]$ and $z \in [0, H]$. This equation carries both an r-derivative and a z-derivative, so it is the coupled radial–vertical problem, and the routine that solves it is `solveRC2D`. `solveRC1D` solves the same equation with the z term absent, which is the purely radial problem. `solveAxisym2D` solves nothing itself: it stacks `solveRC1D` slices as described in §3.1. Those three definitions are used consistently everywhere in this book.

It is written deliberately in **divergence (conservative) form** — the derivative sits outside the coefficient, not inside it. This matters because the coefficients are *not* smooth: permeability drops by a factor of several across the smear boundary. The non-conservative form $c \cdot \partial^2 u/\partial r^2$ is simply not valid at such a jump. The divergence form remains valid, and its finite-volume discretisation enforces the physically correct condition automatically: **the flux is continuous across the interface even though the gradient is not**.

4.2 What each term is

Table 4.1 lists every term of the governing equation and where it comes from.

Table 4.1. The terms of the governing equation.

Term	Meaning	Where it comes from
$c_h(r,z,t)$	horizontal coefficient of consolidation	derived from k_0, C_k, C_c, C_r, σ'_p, β and e_0 — it is *not* a constant you type
$\kappa(r)$	smear profile, $k(r)/k_h \le 1$	$1/\eta$ inside r_s; a linear ramp from $1/\eta$ to 1 across the transition zone; 1 beyond r_t
$c_v(z)$	vertical coefficient of consolidation	typed per layer; constant within a layer
dp/dt	rate of change of applied load	the loading schedule — ramps generate this source, instantaneous stages are applied as a uniform jump in u
$\dot{s}_{\text{creep}}$	viscous (secondary) compression source	the EVP model, from C_α/C_c; rate-limited per step

The smear profile deserves a note. Rather than a step from 1/η straight to 1 at r_s, the tool supports a **transition zone** out to r_t across which the permeability ramps linearly. This reflects what is actually measured around a mandrel-installed drain: disturbance fades with distance rather than stopping at a line. Setting $r_t = r_s$ recovers the classical sharp-boundary Hansbo smear zone.

4.3 Why the coefficients move as the solution moves

In the classical theory c_h is a constant. In this tool it is not, and this is the single most consequential difference from a textbook solver. As consolidation proceeds, pore pressure dissipates, effective stress rises, the soil densifies, and its permeability falls. The tool follows that chain explicitly in every cell at every step:

$$\begin{aligned} \text{effective stress} \quad \sigma' &= \sigma'_0 + (p - u) \\ \text{permeability} \quad k_h &= k_0 \, 10^{-\frac{\Delta e}{C_k}} \\ \text{compressibility} \quad m_v &= \frac{0.434 \, C_{\text{op}}}{(1 + e_0)\, \sigma'} \\ \text{consolidation} \quad c_h &= \frac{k_h}{m_v \, \gamma_w} \end{aligned} \tag{4.2}$$

The operative compression index C_{op} blends the recompression and virgin indices through a smooth yield weight w, so the soil passes through its preconsolidation pressure gradually rather than at a kink:

$$\begin{aligned} w &= \frac{1}{1 + \left(\frac{\sigma'_p}{\sigma'}\right)^{\beta}} \\ C_{\text{op}} &= (1 - w)\, C_r + w \, C_c \end{aligned} \tag{4.3}$$

> **KEY IDEA**
>
> **The consequence for the numerics is that the problem is non-linear.** Two competing effects act on c_h as stress rises: falling permeability pushes it down, while falling compressibility pushes it up. So c_h is generally *non-monotonic* in time, dipping as the soil yields past σ'_p and recovering afterwards. A solver that assumed a fixed coefficient could not reproduce this. How the tool handles the non-linearity is §9.5.

4.4 Creep — the equivalent-time (isotache) law

Soft clay keeps compressing after the excess pore pressure has gone. That continuing movement is **secondary compression**, or creep, and on soft deposits it is not a rounding error: over a design life it can add a substantial fraction of the primary settlement. The tool models it with an **elastic–visco-plastic (EVP)** law of the Bjerrum equivalent-time, or *isotache*, family. This section states exactly what the solver does, because the formulation matters more than the name.

The governing idea is that the preconsolidation pressure is not a fixed property. It *grows with time* as the soil ages under load. A cell that has sat at constant stress for a year is stiffer than one that reached the same stress yesterday, and the isotache law expresses that by letting creep harden σ'_p.

The key to doing this without double-counting is that every cell carries **two** hardening states rather than one:

- $\boldsymbol{\sigma'_{p,load}}$ — the preconsolidation pressure from *loading alone*: simply the largest effective stress the cell has ever seen.
- $\boldsymbol{\sigma'_p}$ — the preconsolidation pressure from loading *and* creep, which can rise above the first.

The strain in a cell is then

$$\varepsilon = \varepsilon(\sigma') + [\, \varepsilon(\sigma'_p) - \varepsilon(\sigma'_{p,load}) \,] \tag{4.4}$$

where $\varepsilon(\cdot)$ is the compression curve of §4.3. The first term is the ordinary primary compression; the bracket is the creep. Written this way the creep term is a *difference* between two points on the same curve, so virgin compression can never be counted twice — which is exactly the defect of the older decoupled add-on this law replaced, where creep was stacked on top of a strain that already contained it.

Per cell and per time step the solver executes five moves:

1. **Harden by loading.** If the current σ' exceeds either stored state, raise that state to it. Reaching a new stress also resets the creep age.
2. **Read the equivalent time** from the current overconsolidation:

$$t_e = t_{ref}\,(\sigma'_p/\sigma')^{(C_c - C_r)/C_\alpha} \tag{4.5}$$

This is the standard isotache form — Bjerrum's (1967) time lines put in this algebraic shape by Garlanger (1972) and Leroueil et al. (1985). Here t_{ref} = 1 day is the oedometer convention the creep index is measured against, and C_α is the e–log t slope recovered from the typed ratio as $C_\alpha = (C_\alpha/C_c)\cdot C_c$, the same quantity that produces $C_{\alpha\varepsilon}$ in the next step.

Read what the exponent does. It is large — for a typical soft clay (C_c = 0.5, C_r = 0.05, C_α/C_c = 0.04) it is about 22 — so t_e is extremely sensitive to overconsolidation. That sensitivity is the physics, not a numerical accident: a cell only 20 % overconsolidated is already some sixty reference times old and has nearly stopped creeping, while a cell sitting exactly at yield has $t_e = t_{ref}$ and creeps fastest. The limit is reached by construction rather than imposed, which is why this form needs no floor and cannot go singular. A cell at or beyond the virgin line ($\sigma' \geq \sigma'_p$) is on the reference isotache, so it too takes $t_e = t_{ref}$.

3. **Take the log-time strain increment** over the step, $\Delta\varepsilon = C_{\alpha\varepsilon}\log_{10}[\,(t_e + \Delta t)\,/\,t_e\,]$, where $C_{\alpha\varepsilon} = (C_\alpha/C_c)\cdot C_c/(1 + e_0)$ is the same creep index expressed as a strain. Because $t_e \geq t_{ref} > 0$ for every

admissible state, the argument is always finite and the increment is exact for the step.

4. **Let that strain harden σ'_p.** Inverting the $\log_{10}$ compression law gives the multiplier $10^{\Delta\varepsilon(1+e_0)/(C_c-C_r)}$, applied to the crept state only.
5. **Add the creep strain** as the bracket above — zero whenever the two states are equal.

> **KEY IDEA**
>
> **Three properties worth knowing.** Creep runs *during* primary consolidation as well as after it (Hypothesis B), which is why the clock starts at yield rather than at the end of primary. Setting $C_\alpha/C_c = 0$ makes the two hardening states identical, the bracket exactly zero, and the answer bit-for-bit the elastoplastic one — so switching creep off costs nothing and changes nothing. And creep enters *settlement* only, never the degree of consolidation: U is a degree of *primary* consolidation, so a creeping soil can report 100 % consolidation while still settling.

Creep is always active in the solver; there is no on/off switch, because on soft clay creep is not an optional refinement. It is turned off *per layer* instead, by leaving that layer's C_α/C_c blank or zero — which, by the property above, returns exactly the elastoplastic answer for that layer. The single ratio C_α/C_c (Mesri & Godlewski 1977) carries the whole law; its default of 0.04 is the usual value for inorganic clay, against roughly 0.05 for organic soil and 0.06 for peat. The tool carries the ratio rather than C_α itself because C_α co-varies strongly with C_c, so the ratio is the more transferable number and the more identifiable one when it is back-analysed.

CHAPTER 5

The element

5.1 Cell-centred finite volume

The domain is divided into non-overlapping **control volumes**. One unknown is stored at the centre of each: the excess pore pressure. The governing equation is then integrated over each control volume and the divergence theorem applied, which converts the volume integral of the flux divergence into a sum of fluxes over the cell's faces. In words: *the rate of change of pressure stored in a cell equals the net flux through its faces, plus whatever the sources add.* Figure 5.1 draws one such cell.

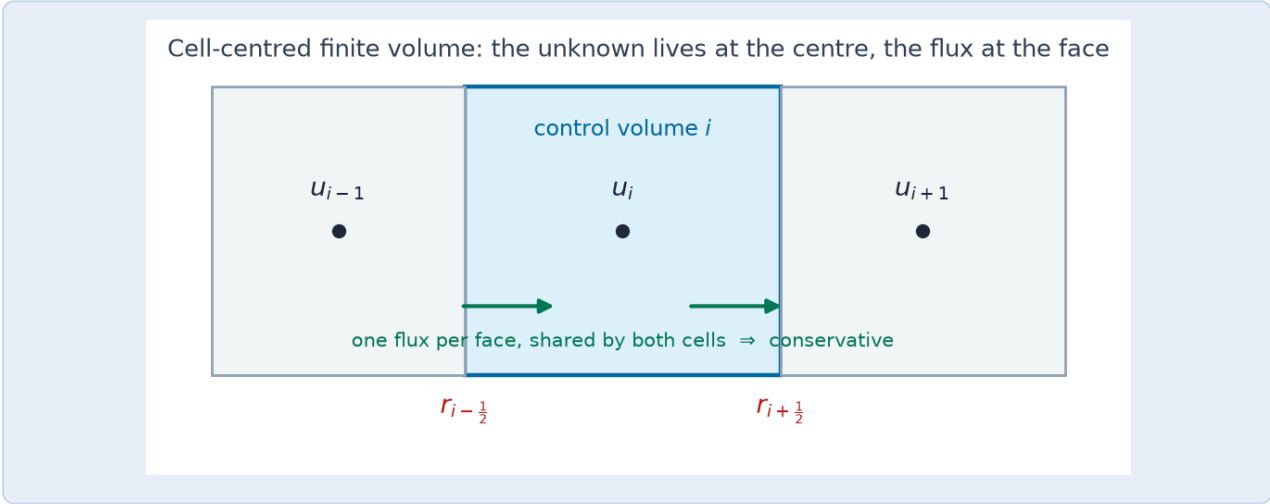


Figure 5.1. The element. The unknown u_i lives at the cell centre; fluxes are evaluated at the faces $r_{i-½}$ and $r_{i+½}$. Each face flux is computed once and used by both neighbouring cells, with opposite sign.

This is not a finite-*element* method and not a finite-*difference* method. The distinction is worth making because it is exactly what makes the method suitable here; Table 5.1 makes the comparison:

Table 5.1. Why finite volume, for this problem.

Method	Unknown at	Behaviour at a sharp permeability jump
Finite difference	grid points	needs the coefficient to be differentiable; a jump must be smoothed, which smears the smear zone
Finite element	nodes	handles jumps well if element edges align with them; conservation is global rather than element-by-element unless a mixed formulation is used
Finite volume	cell centres	**conservative by construction**; a jump is placed on a face and the shared face flux enforces flux continuity exactly

5.2 The annular control volume

In axial symmetry a cell is not a box but a **ring** — an annulus of unit height between radii $r_{i-½}$ and $r_{i+½}$. Its volume is proportional to $r_{i+½}^2 - r_{i-½}^2$, while the area of a face at radius r is proportional to r itself. Both carry a factor π which cancels, and the ratio of face area to cell volume gives the coefficient that appears throughout the code:

$$a_i = \frac{2\, c_{\mathrm{h,face}}\, r_{\mathrm{face}}\, \kappa_{\mathrm{face}}}{\Delta r_{\mathrm{centre-centre}}\, A_i} \qquad A_i = r_{i+½}^2 - r_{i-½}^2 \tag{5.1}$$

The factor 2 is exact geometry, not a tuning constant: $2\pi r / (\pi A) = 2r/A$. Because the true annular volume is used rather than a Cartesian width, the discretisation carries the cylindrical geometry exactly at every radius — including the very first cell at the drain wall, where the curvature is strongest and where the classical solutions concentrate their gradient.

5.3 One flux per face — why this matters

Cell i's outward flux and cell i+1's inward flux are the *same stored number*, used with opposite signs. Nothing is created or destroyed at a face. Summed over the whole mesh, every internal flux cancels in pairs and only the boundary fluxes and the sources survive. The scheme therefore satisfies a discrete water balance to machine precision, on any mesh, at any refinement — even a deliberately bad one.

> **KEY IDEA**
>
> This is the property that lets the tool put a factor-of-five permeability jump in the middle of the domain and still trust the answer. Conservation is not something the mesh has to be fine enough to achieve; it holds exactly, and refinement only improves *accuracy*.

5.4 The harmonic mean at the face

A face sits between two cells that may have different permeabilities. Which value should the flux use? The answer is fixed by physics, not preference. Steady flow across two layers in series sees the sum of their resistances, and resistance is thickness over conductivity. Adding resistances and inverting gives the **harmonic** mean:

$$k_{\mathrm{face}} = \frac{d_L + d_R}{\frac{d_L}{k_L} + \frac{d_R}{k_R}} \tag{5.2}$$

Here d_L and d_R are the distances from each cell centre to the face, so the denominator is the resistance actually in series between the two centres. On a mesh whose cells all have the same width this reduces exactly to the familiar $2k_Lk_R / (k_L + k_R)$, which is why that simpler expression is what most textbooks print. The meshes here are graded, so the distances generally differ and both solvers use

the weighted form above. Note what the weighting can touch: a face needs cells of different width *and* different conductivity before the two expressions part company at all. Where the two conductivities are equal the weighted mean returns that shared value for any distances, so a uniform-property profile — every closed-form case of Chapter 11 among them — is unaffected exactly, not merely nearly.

Until September 2026 both solvers used the unweighted expression, and this section asserted that the error it left behind was small. That assertion was correct, but it was an argument rather than a measurement, so it has been replaced by one. Table 5.2 collects the result; each of its rows compares the two expressions against a reference that knows nothing about either.

Table 5.2. What the distance weighting changed, measured. Every row uses a reference independent of both expressions: an analytical solution, or a mesh at which the two provably coincide. Measured by `validation_2026/s8_face_weighting.js`, `s8_diagnose.js` and `s8_design_impact.js`.

What was measured	Unweighted	Distance-weighted
Smeared drain, U at t = 365 d, against the 640-cell solution (N_r = 40, the default)	0.043 pp	**0.028 pp**
the same at N_r = 80	0.026 pp	**0.021 pp**
Worst of fifteen comparisons against Hansbo's closed form (three smear geometries × five mesh resolutions)	unchanged within 0.5 % in eleven; largest movement 0.06 pp of U	
Layer interface, tenfold k_v contrast, cell heights 2.3 m against 7.7 m: t_{50}	351.5 d either way (0.01–0.02 % change)	
Three-layer profile, whole settlement curve, creep off	the two agree to 0.0001 % of the settlement	
Required drain spacing, three profiles, the full U_s = 90 % bisection	1,276 / 1,375 / 1,759 mm, shifting by 0.1–0.2 mm	

Read Table 5.2 as a whole and it says something more useful than any single row. The weighting is the exact expression and costs nothing, so it is what ships; but the error it removes was genuinely small, and the honest headline is the last row — on a real layered profile the two expressions give the same settlement curve to one part in a million, and a full design bisection lands on the same spacing to a fraction of a millimetre. The clean improvement is the one place the geometry guarantees a steep grading and a sharp contrast at the same face: the smear boundary at a working mesh, where the error against a converged solution falls by about a third. The Hansbo rows show why that improvement cannot be seen from a closed-form comparison — there the error plateaus near 0.2 pp as the mesh is refined, because what remains is the gap between the finite-volume solution and Hansbo's equal-strain idealisation, not a discretisation error at all. A change of 0.02 pp cannot be detected through a 0.2 pp window.

WATCH OUT

Two lessons from measuring this, both worth carrying to your own checks. First, a refinement study cannot referee a change to the discretisation itself. The obvious experiment — run both expressions against a fine-grid solution — gave a confident and wrong answer, because at any affordable reference mesh the two expressions still differed there by more than the quantity being measured, and whichever one happened to share the reference's bias looked better. The references in the table above are an analytical solution and a 640-cell mesh at which the two provably coincide. Second, creep amplifies everything. With the creep law off, the two expressions agree to 0.0001 % of the settlement; with 1,100 days of creep on, the same pair differ by 0.2 %. Nothing is wrong in either run — a path-dependent law carries forward whatever small difference it is given — but it means a creep run is the wrong place to look for a small numerical effect, and it is worth remembering when you compare two long creep forecasts and find them a few millimetres apart.

An arithmetic mean would be wrong in a specific and dangerous direction: it is dominated by the *larger* of the two values, so it would let water through a low-permeability smear zone far too easily and systematically over-predict the rate of consolidation. The harmonic mean is dominated by the *smaller* — the restriction controls the flow, which is what actually happens. Both the permeability ratio κ and the coefficient c_h are distance-weighted harmonic-meaned at every face in both solvers, in the radial direction and the vertical one alike.

A number makes the difference concrete. Take a face between undisturbed soil ($k = 1$, in units of the intact permeability) and a smear-zone cell at $\eta = 10$ ($k = 0.1$). The harmonic mean gives $k_{face} = 2 \times 1 \times 0.1 / 1.1 = 0.18$. The arithmetic mean gives 0.55 — three times larger. A solver using the arithmetic mean would push water through the smear zone three times too easily at that face, and would over-predict the rate of consolidation accordingly. The error grows with the contrast: at $\eta = 100$ the arithmetic mean is twenty-five times the harmonic.

CHAPTER 6

The mesh

6.1 The radial mesh is zone-aware

The mesh is not laid down and then asked to cope with whatever geometry it finds. It is *built around* the geometry. The radius is first split into zones at the physical boundaries —

$[r_w,\ r_s]$	$[r_s,\ r_t]$	$[r_t,\ r_e]$
smear	**transition**	**undisturbed**
$k / k_h = 1/\eta$	linear ramp to 1	$k / k_h = 1$

— and each zone is then meshed separately. Cells are allotted in proportion to zone width, subject to a floor of **max(8, 5 % of N_r)** cells per zone so that a thin smear annulus can never be left with one or two cells, and any remainder goes to the widest zone. Absent a smear zone the whole radius is a single zone.

6.2 Faces land exactly on the zone boundaries

Because each zone is meshed to its own width, **no cell ever straddles r_s or r_t**. The permeability jump always falls on a face, never inside a cell. This is the condition that lets the harmonic mean of §5.4 do its job — a cell containing the jump would have to be given a single averaged permeability, smearing the interface over a cell width and degrading the smear-zone accuracy.

This is easy to assert and easy to get wrong, so it was measured. Cell centres were taken from the solver itself and the faces reconstructed exactly by inverting the cell-centre definition ($c_i = \tfrac{1}{2}(f_i + f_{i+1})$, so $f_{i+1} = 2c_i - f_i$ walking out from r_w). The distance from the nearest face to each zone boundary is then a number that can be checked rather than trusted. Table 6.1 reports that distance at four resolutions.

Table 6.1. Distance from the nearest cell face to the zone boundaries, at several mesh resolutions. Geometry: r_w = 33.1 mm, r_s = 75 mm, r_t = 150 mm, r_e = 677 mm. Measured by `validation_2026/num_guide_data.js`.

N_r requested	cells built	|face − r_s|	|face − r_t|	widest / narrowest cell
20	20	0 m	3e-17 m	110.34
40	40	0 m	3e-17 m	11.36
80	80	0 m	3e-17 m	2.72
160	160	0 m	3e-17 m	1.57
320	320	1e-16 m	1e-16 m	1.03

KEY IDEA

The boundaries are hit **exactly** — zero, or a few units in the last bit of double precision (10^{-17} m is about a millionth of an atomic diameter). This is a structural property of how the mesh is built, not a coincidence of these particular numbers, and it holds at every resolution. Figure 6.1 shows the built faces against the zone boundaries.

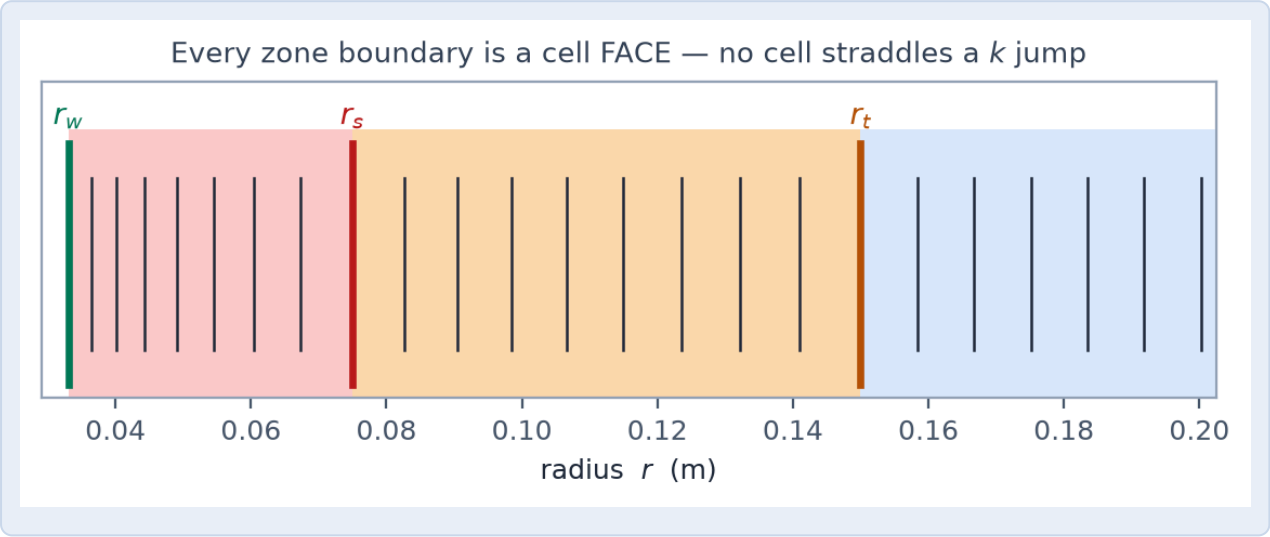


Figure 6.1. Cell faces near the drain, drawn to scale. Thin lines are ordinary faces; the heavy coloured lines are r_w, r_s and r_t. Each zone boundary coincides with a face.

6.3 Grading — fine at the drain, coarse in the far field

Within each zone the cell widths form a **geometric series**. The first cell at the drain wall is sized at about a tenth of the drain radius (bounded to between 2 and 20 mm), and the growth ratio is then solved by bisection so that the series sums *exactly* to the zone width. The last cell of one zone seeds the first cell of the next, so cell size varies smoothly across a zone boundary instead of jumping.

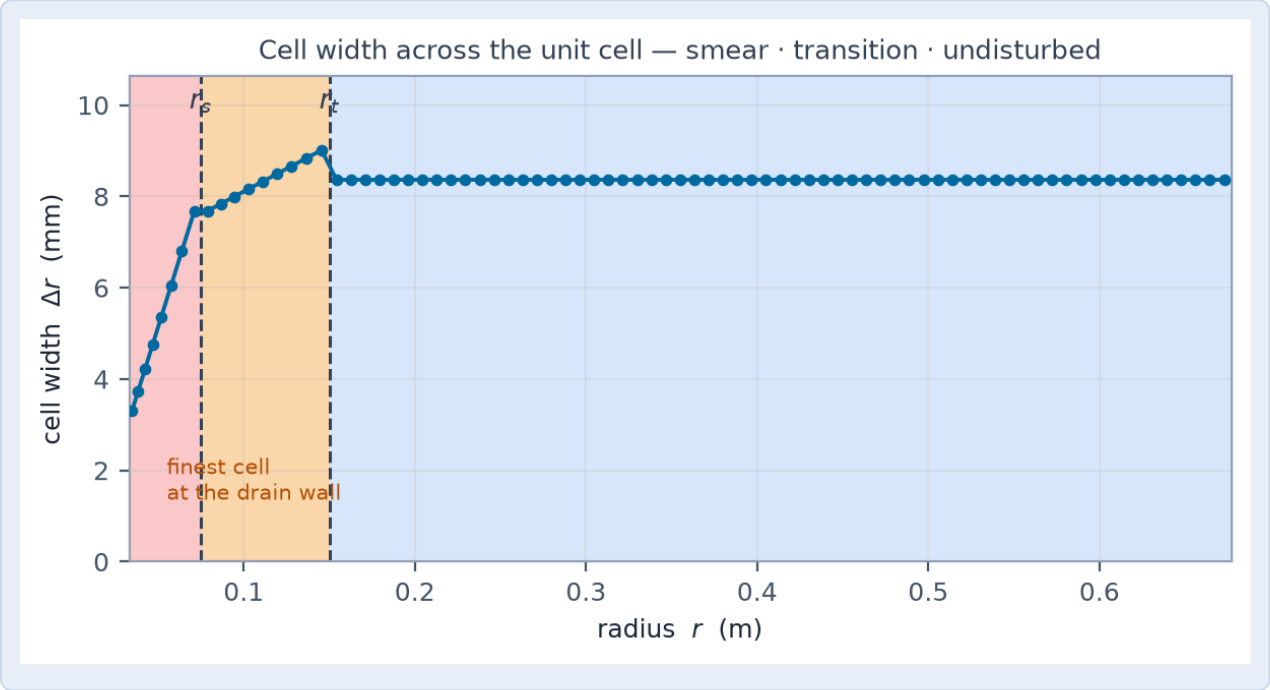


Figure 6.2. Measured cell width across the unit cell (N_r = 80 shown so individual cells are legible). Cells are finest at the drain wall, grow through the smear and transition zones, and become uniform in the far field where the solution is smooth.

The pattern in Figure 6.2 is exactly what the physics asks for. The pore-pressure gradient is steepest at the drain wall, where all the water is leaving; it is nearly flat near the outer boundary, where the zero-flux condition holds. Resolution is spent where the solution actually varies. In the measured mesh the narrowest cell is 3.31 mm at the wall and the widest 9.01 mm in the far field.

> **WATCH OUT**
>
> **Why the sizes are matched across zones.** An abrupt change in cell size at r_s (small cells inside, large cells outside) makes the discrete operator stiff at that one location. Both Crank–Nicolson and ADI respond to a locally stiff row with *ringing*: a non-physical oscillation in u(r) that sits at the interface and does not decay. Matching the sizes removes it. This is why the mesh grades rather than simply using a uniform spacing inside each zone.

6.4 The vertical mesh

The vertical mesh follows the same principle, more simply. Each soil layer is divided into a whole number of equal cells, chosen so the cell height is at most the target *dz* (default **0.5 m**). Because the division is done layer by layer, **cell faces land exactly on every layer interface** — no cell straddles a change in c_v, m_v, or over-consolidation state, for the same reason no radial cell straddles the smear boundary.

6.5 The drain core for partial penetration

When every layer is drained, the mesh starts at the drain wall r_w — there is no need to mesh the drain itself, because the drain is a boundary condition. When a drain stops short of the base of the profile, that is no longer true: below the tip, the region $r < r_w$ is soil, not drain, and water must be able to flow up through the drain above it. Figure 6.3 shows the geometry and the mesh treatment.

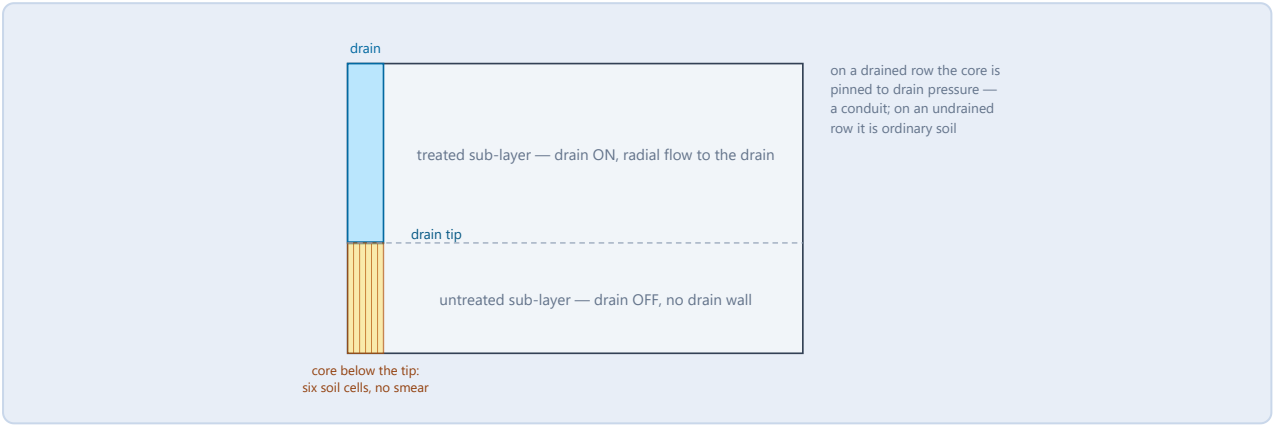


Figure 6.3. Schematic. Partial penetration: above the drain tip the drain wall is a boundary; below it the region $r < r_w$ is soil, meshed with six uniform cells (their dual, row-by-row behaviour is described below).

The solver therefore extends the mesh inward over $[0, r_w]$ with six uniform cells whenever any layer has its drain switched off. Those cells are dual-natured, row by row: on a drained row they are pinned to the drain pressure (zero, or the vacuum suction) and act as a conduit; on an undrained row they are ordinary soil. Smear is excluded from the core, because a soil element never touched by the mandrel is undisturbed by definition. On undrained rows the whole row reverts to the intact permeability profile.

6.6 Changing the mesh — the knobs you have

The mesh is not fixed. Every task exposes it, and the controls differ because the tasks make different demands: Design and Simulate run one forward solve (or a few dozen), while Back-calculation runs many thousands and therefore has to buy its accuracy more carefully. Table 6.2 lists the controls by task.

Table 6.2. Mesh controls, by task.

Task	Control	Default	Range
Design & Simulate	N_r — radial cells (count)	80	20 – 400
	dz — target cell height (m)	0.5 m	free
Back-calculation	Solver grid — N_r (count) / Δt (days) / Δz (m), **per pipeline step**	40 / 1 d / 0.25 m	three presets, or edit any step

"Per pipeline step" anticipates Part Three. A back-calculation run proceeds through a fixed sequence of stages, called the pipeline, and the solver grid can be set for each stage independently. The pipeline itself is described step by step in Part Three; nothing there changes what the grid controls mean here.

Both input fields sit empty until you fill them and show an amber cue, so the fallback is visible rather than silent. Figure 6.4 shows the two controls as they appear in the app. The app also reports the resulting grid in a **Numerical mesh** results card (Figure 6.5) — total $N_r \times N_z$ with a schematic. Read it: N_r is a **target, not a guarantee**, because the per-zone floor and the remainder rule can push the built count above what you asked for.

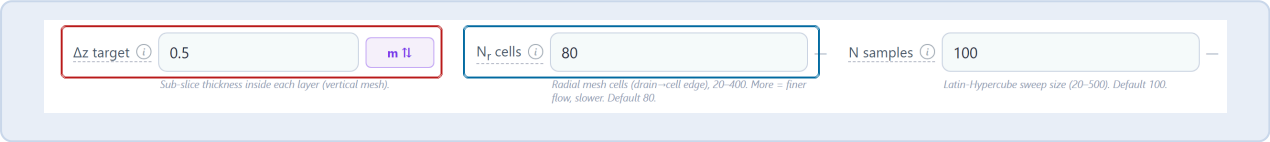


Figure 6.4. The mesh controls in the app, on the Design/Simulate input page. The red box is the vertical target Δz (m); the blue box is N_r, the radial cell count. The third field, N samples, belongs to the probabilistic run and does not affect the mesh.

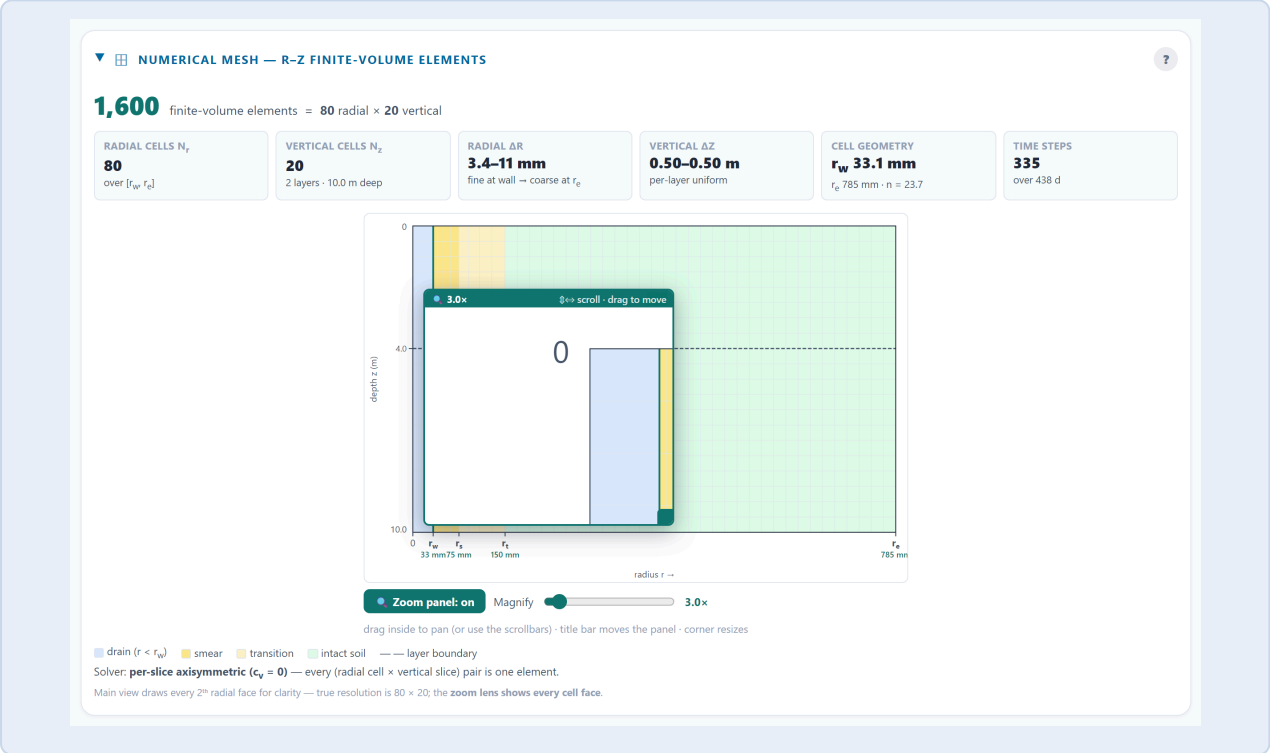


Figure 6.5. The Numerical mesh results card, reporting the grid a run actually used: the built $N_r \times N_z$ count, the cell-size ranges, the time steps taken, and an r–z schematic with the drain, smear and transition zones coloured. The zoom lens (shown open) magnifies the near-drain cells that the main view is too coarse to draw.

6.7 Choosing a grid — the tool's own convergence study

Which knob deserves your effort is not obvious, and guessing wrongly is expensive. The tool therefore ships a grid-convergence study of its own, run on a representative four-layer case against a very fine reference (N_r 360 / Δz 0.10 m / Δt 0.25 d). Each knob was swept *alone*, so the numbers below are that knob's individual contribution to settlement error.

Table 6.3. Settlement RMS error against the fine reference, each knob swept alone. From the study recorded in the source beside `BC_ADV_GRID_PRESETS`.

Knob	value → RMS settlement error				Verdict
N_r	16 → 0.99 %	24 → 0.19 %	40 → 0.06 %	80 → 0.04 %	converged by 40 — radial flow is smooth, and cells here are cheap
Δz	1.0 → 2.96 %	0.5 → 1.22 %	0.25 → 0.38 %	0.125 → 0.04 %	a **slow** knob — it sets layer-strain resolution
Δt	8 → 4.45 %	4 → 1.45 %	2 → 0.68 %	1 → 0.32 %	the **dominant** knob — early-time consolidation

These figures were re-measured after the face-conductivity change of §5.4, and the ranking and the magnitudes both stand. That is expected rather than lucky: the knob errors here are set by how finely the early transient and the layer strains are resolved, while the face expression contributes less than the smallest entry in the table — on a creep-free comparison the two expressions give the same settlement curve to one part in a million. Where a creep law is active, the amplification described in §5.4 applies to any small numerical change, this one included, which is the reason the comparison above is made without creep.

> KEY IDEA
>
> **Time step first, then vertical cell size, then radial cells.** This ranking surprises people who expect the mesh to matter most. It follows from what each knob resolves. Δt governs the steep early transient *and* how far the stress-dependent coefficients lag (§9.5). Δz governs how finely layer strain is resolved. The radial direction is smooth and converges early. Spending on N_r while leaving Δt coarse buys almost nothing. Table 6.4 turns the ranking into the three one-click presets.

Table 6.4. The three one-click presets in Back-calculation. Each sets every pipeline step to the same grid.

Preset	N_r / Δt / Δz	Settlement bias	Use when
Fast	40 / 4 d / 0.5 m	~2.3 %	exploring — *not* grid-free, see the warning below
Grid-converged (default)	40 / 1 d / 0.25 m	~0.5 %	normal work — at the level of the measurement noise
Publication	80 / 0.5 d / 0.125 m	~0.1 %	final reported results

> **WATCH OUT**
>
> **Why a coarse grid is more dangerous in Back-calculation than in Simulate.** A forward run with a 2 % bias is simply 2 % off. But inversion compares predictions against measurements whose scatter σ is perhaps 5–20 mm, while total settlement is on the order of 2000 mm — so a 1 % settlement bias is already about *one standard deviation*. It shifts χ^2 and can make a wrong parameter set look like the best fit. Two rules follow, and the tool enforces the first by construction in its presets: **(1)** use the same grid for every pipeline step, so the search and the final sweep are judged by one ruler; **(2)** keep it at least at the Grid-converged preset.

One more dependence deserves its mechanism, not just its rule. Faster c_h, or tighter drain spacing, pushes Δt finer still. The reason is the characteristic drainage time of the unit cell, which scales as r_e^2/c_h. Both a higher c_h and a smaller r_e shorten it. The early transient, which is the part of the history Δt exists to resolve, is then compressed into a shorter interval. A time step adequate for a slow, widely drained profile will under-resolve a fast, tightly drained one. The operative rule follows from the scaling: **Δt should move in proportion to r_e^2/c_h**. If c_h doubles, halve Δt; if the drain spacing halves, r_e^2 falls fourfold, so quarter it. When in doubt, refine Δt once more and confirm the answer stops moving — the five-minute check of §12.2.

CHAPTER 7

Boundary conditions

7.1 The complete list

Table 7.1 is the complete list of boundary conditions either solver ever imposes.

Table 7.1. Every boundary condition imposed by the solvers.

Boundary	Condition	Meaning
drain wall, $r = r_w$	$u = 0$ (Dirichlet)	the drain removes water as fast as the soil delivers it
drain wall, with vacuum	$u = -p_{vac}(t) \cdot f(z)$	suction applied at the drain, attenuating with depth
drain wall, drain-off row	$\partial u/\partial r = 0$	no drain present at this depth — symmetry, no flow
outer wall, $r = r_e$	$\partial u/\partial r = 0$	unit-cell symmetry: neighbouring drains are identical
top, $z = 0$	$u = 0$ (default) or $\partial u/\partial z = 0$	drainage blanket, or a sealed surface
base, $z = H$	$\partial u/\partial z = 0$ (default) or $u = 0$	impermeable base, or an underlying drainage layer
base, partial penetration	$\partial u/\partial z = 0$ (forced)	if the deepest layer is undrained but shallower ones are drained, the base is forced no-flow

7.2 The drain wall is a face, not a cell

A common way to impose $u = 0$ is to reserve a cell and hold its value fixed. This solver does not do that, and the difference is worth understanding because it affects accuracy.

The drain wall *is* the first face of the first cell. The flux through it is built from that cell's own permeability and a centre-to-face distance of **half a cell width**, with the wall value entering the stencil directly. The Dirichlet condition is therefore imposed exactly at $r = r_w$, not half a cell away from it, and no unknown is wasted on a cell whose value is already known. A pinned-cell formulation places the boundary at a cell *centre*, which shifts the effective drain radius by half a cell and biases the result — a bias that shrinks with refinement but never quite vanishes. Figure 7.1 draws the difference.

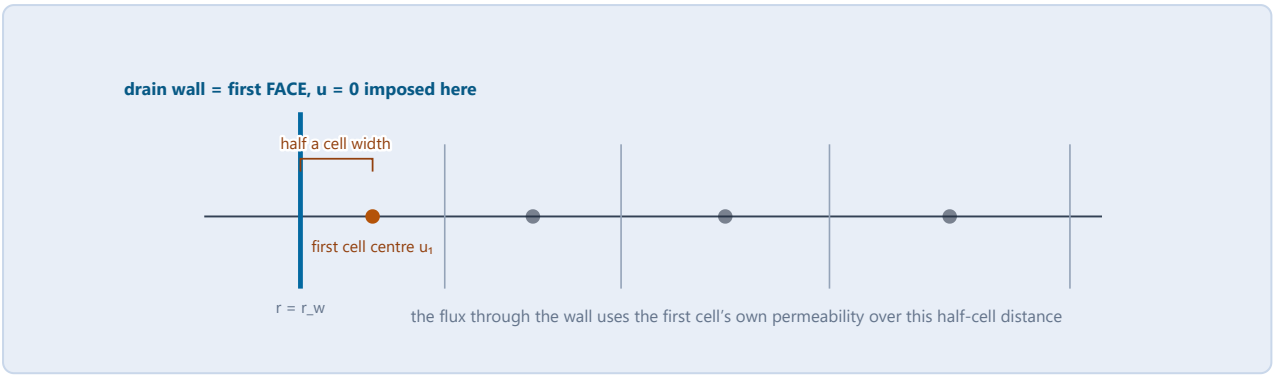


Figure 7.1. Schematic. The near-drain discretisation. The drain wall coincides with the first control-volume *face*; the first unknown sits half a cell inside it. The Dirichlet condition u = 0 is imposed exactly at r = r_w, and no unknown is spent on a cell whose value is already known.

7.3 Vacuum enters as a wall value

Vacuum preloading is not a separate mechanism in the solver; it is the same drain-wall Dirichlet condition given a negative value. The suction follows its own schedule, and Crank–Nicolson uses the wall pressure at *both* ends of the step: the old value on the explicit half and the new value on the implicit half. A vacuum that is ramping is therefore integrated to the same second-order accuracy as everything else.

Depth attenuation follows Indraratna's linear law: the suction reaching depth z is scaled by $1 - (1 - k_2)\cdot z/H$, where H is the full profile depth, so k_2 is the fraction still acting at the base of the profile. Because the vacuum enters only through a boundary value, and the equation is linear in u for fixed coefficients, surcharge and vacuum superpose exactly — which is precisely what the verification case in §11.1 confirms to within 0.03 %. The attenuation reference length, the tip interaction under partial penetration and the k_2 selection guidance are specified normatively in Appendix A of the combined book.

7.4 The outer boundary and the unit-cell idealisation

Zero flux at r_e is imposed three times over: the face conductivity is set to zero, the outward coefficient is set to zero, and the ghost value is mirrored. That is redundant but harmless, and it means no single edit can silently open the boundary.

> **WATCH OUT**
>
> The zero-flux wall is an **idealisation, not a numerical choice**. It assumes every drain has an identical neighbourhood. At the edge of a treated block, or where spacing changes, or next to an untreated zone, water genuinely does cross that surface and the unit cell over-predicts the rate of consolidation. This is a limitation of the unit-cell model itself, shared with every design chart in the literature; refining the mesh will not touch it.

7.5 Top and base

The tool uses a **draining top** and an **impermeable base**, matching the usual situation of a working platform or sand blanket over a deposit resting on rock or stiff clay. **Neither boundary is a user control in the app**: Design and Simulate fix both, and Back-calculation forces the base to no-flow on every load. The solver itself accepts either condition on each boundary; the verification cases of Chapter 11 exercise the other combinations. The application, however, does not expose the switch. An impermeable boundary is imposed by zeroing the face conductivity, which is exact — the coefficient vanishes and no flux term survives. A draining boundary places $u = 0$ on the boundary *face*, half a cell from the adjacent cell centre, mirroring the drain-wall treatment of §7.2.

CHAPTER 8

Interfaces

8.1 Radial interfaces — smear and transition

Everything needed is now in place, and it is worth seeing the three mechanisms act together, because no one of them is sufficient alone. Table 8.1 lays the three out:

Table 8.1. How a permeability jump at r_s is handled.

	Mechanism	What it prevents
1	the mesh puts a **face** exactly on r_s (§6.2)	a cell containing the jump, which would have to average two permeabilities and smear the interface
2	the face conductivity is the **harmonic mean** (§5.4)	over-predicting flow through the restriction, and so over-predicting the rate of consolidation
3	the two neighbours **share one flux** (§5.3)	water appearing or disappearing at the interface

Together these reproduce the correct interface physics: *u* is continuous, the gradient $\partial u/\partial r$ is *dis*continuous in inverse proportion to the permeabilities, and the flux is continuous. That is what a series resistance does, and the discrete scheme gets it exactly rather than in the limit of fine mesh.

8.2 Vertical interfaces — layer boundaries

The same three mechanisms apply in *z*. Faces land on layer interfaces by construction (§6.4). The vertical face conductivity is harmonic-meaned. And the two rows either side of an interface use the **same face conductivity over the same centre-to-centre distance**, so they compute one shared flux across their common face.

The **storage** term is then applied per cell: the shared flux divergence is divided by that cell's own m_v, converting a net flux into a rate of pressure change. Flux belongs to the face and is shared; storage belongs to the cell and is not. That split is exactly what makes the scheme conservative across a layer boundary.

The vertical permeability itself is recovered from what each layer already carries. Since $c_v = k_v/(m_v\gamma_w)$, the product $c_v \cdot m_v$ is k_v to within the constant γ_w. That constant cancels between the face conductivity and the storage term. So a layered profile carries a genuine vertical permeability contrast: a low-k_v band restricts flow through the harmonic mean, just as the smear zone does radially, and a layer given $c_v = 0$ closes its faces from *both* sides rather than only one.

The assumption in that construction is this: the vertical permeability contrast between layers is only as good as the m_v typed for each layer. k_v is never typed directly. It is recovered as $c_v \cdot m_v$. Figure 8.1 shows the configuration in which the recovery works and the one in which it fails. It works when each layer's c_v and m_v are individually right, because their product then really is the layer's permeability. It fails when a typed m_v is wrong in a way that happens to cancel in that layer's own settlement — the layer's response can still look right while the k_v it exports to the interface is wrong, and the cross-layer flow with it. The practical consequence: if a layered profile shows an implausible rate across a strong contrast, check the m_v values before suspecting the solver, because they set the interface permeability as surely as c_v does.

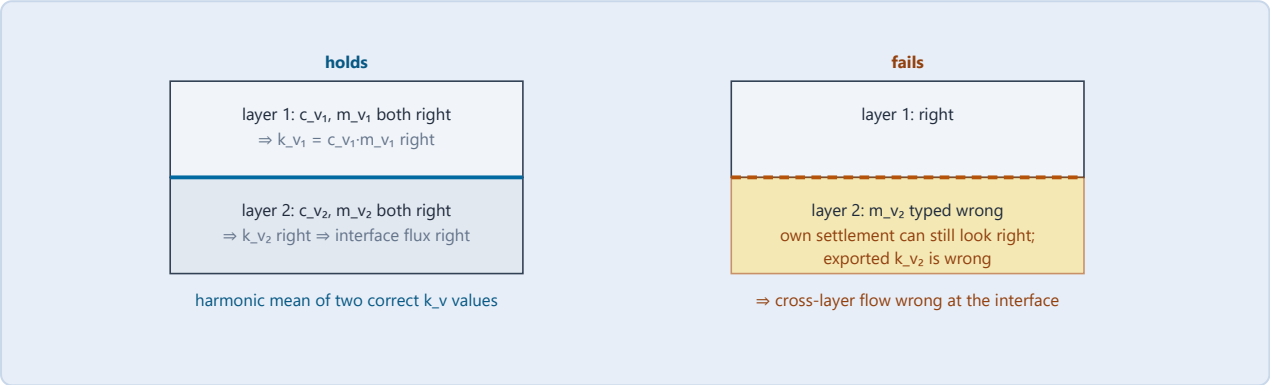


Figure 8.1. Schematic. The $k_v = c_v \cdot m_v$ assumption. Left: with each layer's c_v and m_v individually correct, the recovered permeabilities and the interface flux are correct. Right: a wrong m_v can hide inside one layer's own response while corrupting the permeability that layer presents to the interface.

8.3 This was wrong until recently, and the error was large

> **WATCH OUT**
>
> **Before 2026-08-17 the vertical face conductivity was fixed at one.** The coefficient was then $c_v(\text{cell}) \cdot \Delta u/\Delta z$. The flux across a face was therefore evaluated with the c_v of whichever cell happened to be evaluating it. The two neighbours disagreed, and the scheme was **not conservative** at a layer interface: it imposed continuity of the pressure *gradient* where the physics requires continuity of *flux*. In effect it solved $c_v\, \partial^2 u/\partial z^2$, which is valid only for uniform properties. Table 8.2 measures what that did to a layered profile.

Table 8.2. Time to 50 % consolidation for two 5 m layers with a tenfold k_v contrast and equal m_v, top drained. The reference is an independent conservative solver, itself checked against the exact Terzaghi series to 0.03 pp. Harness: `validation_2026/layered_kv_probe.js`.

Profile	Reference t_{50}	Old scheme	As shipped
uniform (control)	966 d	985 d (+1.9 %)	985 d (+1.9 %)
slow base (k_v ÷ 10 below)	1145 d	5536 d (**+383 %**)	1175 d (+2.6 %)
permeable base (k_v × 10 below)	9038 d	4535 d (**−50 %**)	9120 d (+0.9 %)

KEY IDEA

The second failure is the dangerous one: the old scheme predicted consolidation **twice as fast as reality** for a profile with a permeable base under a less permeable one. Uniform profiles were never affected — which is why every closed-form benchmark in Chapter 10 passed throughout, and why this needed a layered reference to find at all.

CHAPTER 9

Marching in time

Space is now discretised; time is not. The physical task is simple to state: the excess pore pressure at every point must be advanced from one instant to the next, over and over, from the moment load is applied until the end of the analysis. Each advance is a *step*. The art is in how the step uses what it knows: the pressure field at the start of the step is known, the field at the end is the unknown, and the governing equation connects the two. The schemes below differ only in how they make that connection, and each is introduced in plain terms before its formal statement.

9.1 Crank–Nicolson in the 1-D kernel

The radial kernel advances with **Crank–Nicolson**. The idea in plain terms: the rate of change over the step is estimated twice, once from the field at the *start* of the step and once from the field at the *end*, and the two estimates are averaged. Using only the start value (the explicit choice) drifts one way. Using only the end value (the fully implicit choice) drifts the other. The average cancels the leading error of each, and that is what buys the scheme its accuracy. Formally, the spatial operator is evaluated half at the old time level and half at the new one:

$$\left(\mathrm{I} - \frac{1}{2}\Delta t\, L_r\right) u^{n+1} = \left(\mathrm{I} + \frac{1}{2}\Delta t\, L_r\right) u^{n} + \Delta t\, S \tag{9.1}$$

where u^n is the pressure field at the start of the step and u^{n+1} at its end; I is the identity ("leave the field as it is"); L_r is the discrete radial operator — the net-flux calculation of Chapter 5, applied to a whole field at once; Δt is the step length; and S collects the sources (loading rate and creep). Every displayed equation in this book defines its terms this way at first use.

Averaging the two ends of the step makes the truncation error second order in Δt rather than first, and makes the scheme **unconditionally stable** — no time step, however large, can cause the solution to grow without bound. The price is that a very large step produces a damped, slightly lagged answer rather than a divergent one: the failure mode is graceful and visible, not catastrophic.

9.2 Peaceman–Rachford ADI in the coupled solver

In two dimensions the same averaging idea meets a practical obstacle: treating both directions implicitly at once couples every unknown to every other and forces a large sparse solve at every step. **Alternating Direction Implicit** (ADI) splitting, in the Peaceman–Rachford form, avoids that by cutting one two-dimensional step into *two one-dimensional half-steps*. In the first half-step the radial direction is treated implicitly and the vertical direction from known values; in the second the roles swap. Each half-step is cheap, and the swap restores the symmetry the accuracy relies on:

$$\begin{aligned} \text{half-step 1 (implicit in r)} \quad & \left(\mathrm{I} - \frac{1}{2}\Delta t\, L_r\right) u^{n+\frac{1}{2}} = \left(\mathrm{I} + \frac{1}{2}\Delta t\, L_z\right) u^{n} + \frac{1}{2}\Delta t\, S \\ \text{half-step 2 (implicit in z)} \quad & \left(\mathrm{I} - \frac{1}{2}\Delta t\, L_z\right) u^{n+1} = \left(\mathrm{I} + \frac{1}{2}\Delta t\, L_r\right) u^{n+\frac{1}{2}} + \frac{1}{2}\Delta t\, S \end{aligned} \tag{9.2}$$

where u^n, $u^{n+½}$ and u^{n+1} are the fields at the start, middle and end of the step; L_r and L_z are the discrete radial and vertical operators; I is the identity; Δt is the full step; and S is the source term, added at half strength in each half-step so exactly one Δt of loading enters per full step.

Each half-step is a set of independent **tridiagonal** systems — one per row, then one per column. The cost is linear in the number of unknowns instead of cubic, while the scheme keeps second-order accuracy and unconditional stability. Figure 9.1 draws the sequence.

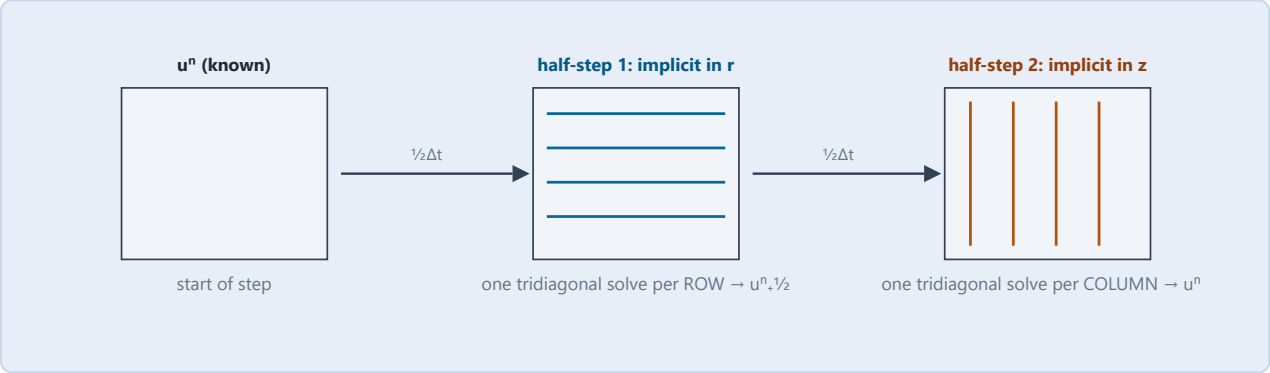


Figure 9.1. Schematic. One Peaceman–Rachford step. The two-dimensional advance is split into two one-dimensional half-steps: rows first (implicit in r, blue), then columns (implicit in z, amber). Each line is an independent tridiagonal system solved by the Thomas algorithm in one sweep.

9.3 The tridiagonal solve

Every system is solved by the **Thomas algorithm** — Gaussian elimination specialised to a tridiagonal matrix, one forward sweep and one back-substitution, exact in a fixed number of operations with no iteration and no convergence tolerance. It is used without pivoting. That is safe here because the matrices are diagonally dominant by construction: the diagonal is $1 + \theta\Delta t(a_L + a_R)$ while the off-diagonals are $-\theta\Delta t\, a_L$ and $-\theta\Delta t\, a_R$, so the diagonal always exceeds the sum of the off-diagonal magnitudes by exactly one.

9.4 Stability, and why the time step is still capped

Both schemes are unconditionally stable, so neither imposes a stability limit internally. The Back-calculation caller nonetheless caps the step (Design and Simulate use a fixed fine-step rule instead), using a diffusion-number estimate

$$\Delta t_{\text{cap}} = \frac{\mathrm{SF}\ \eta\ \Delta r_{\text{wall}}{}^2}{c_{\text{h,max}}} \quad \text{with SF} = \frac{1}{2} \text{ or } \frac{1}{4} \tag{9.3}$$

> **KEY IDEA**
>
> This cap is about **accuracy, not stability** — a distinction worth being explicit about, because the two are often conflated. An explicit scheme needs such a limit or it explodes. Crank–Nicolson does not explode; it simply resolves a rapid transient poorly if the step is too coarse for the finest cell. The cap keeps the step commensurate with the smallest cell at the drain wall, where the gradients are steepest.

9.5 Frozen coefficients — a deliberate choice

Because c_h depends on effective stress, and effective stress depends on u, the problem is non-linear. The tool handles this by **lagging the coefficients**: at the start of each step every cell's c_h is recomputed from the current pressure field, and then held fixed while that step is solved. Each step is therefore a linear solve, and the non-linearity is followed by taking small steps rather than by iterating to convergence within a step.

> **WATCH OUT**
>
> **In the ADI solver the coefficients must not be refreshed between the two half-steps.** Peaceman–Rachford splitting is derived on the assumption that the operator is the *same* in both halves of a step; updating c_h in between violates that assumption and the scheme loses its unconditional stability. When this was tried during development the result was not a small loss of accuracy but divergence — pressures reaching 10^{200} kPa within six months of simulated time. The mid-step refresh is disabled in the code with that finding recorded beside it.

The practical consequence is that **step size controls accuracy in two ways at once** in a stress-dependent run: it sets the time-integration error, and it sets how far behind the coefficients lag. Both shrink together as the step shrinks, which is why the convergence study in Chapter 10 refines Δt and the mesh together rather than separately.

CHAPTER 10

Convergence and accuracy (solution verification)

10.1 How this was measured

A numerical scheme earns trust by **converging**: as the mesh and time step are refined, the answer must approach the exact solution, and at the rate the scheme was designed for. A solver that is merely "close" on one mesh has proved nothing — it could be close by luck, or by two errors cancelling.

The **order of accuracy** is that rate, and it is worth defining before any table uses it. For a first-order scheme, halving the mesh or the time step cuts the error by a factor of two. For a second-order scheme it cuts the error by a factor of four. The practical consequence is that a second-order scheme reaches a given accuracy on a much coarser grid. The diagnostic consequence matters even more: the order can be *measured*, and comparing the measured order with the designed order is a test of whether the code is doing what its scheme claims. Exactly that test exposed a real defect in this solver (§10.4).

Two probes are used. For the **radial** direction there is no simple exact solution for the stress-dependent case, so the reference is a very fine mesh (640 cells) and what is measured is self-convergence. For the **vertical** direction Terzaghi's series solution is exact, so the measured error is genuine discretisation error against truth.

> **WATCH OUT**
>
> The combined Carrillo case deliberately is *not* used for this purpose. Carrillo's product rule is itself an approximation, at roughly the same size (~0.3 pp) as the discretisation errors being measured, so it cannot resolve them — a solver could look better or worse against it purely through error cancellation. Measuring a numerical method requires a reference that is exact.

10.2 Radial refinement

Table 10.1 reports the radial sweep.

Table 10.1. Radial refinement. Degree of consolidation at t = 365 d, band drain with smear (η = 3), against the 640-cell solution. Measured by `validation_2026/num_guide_data.js`.

N_r	U at 365 d (%)	error vs finest (pp)
20	73.882	-1.235
40	75.089	-0.028
80	75.096	-0.021
160	75.099	-0.018
320	75.113	-0.004
640	75.117	—

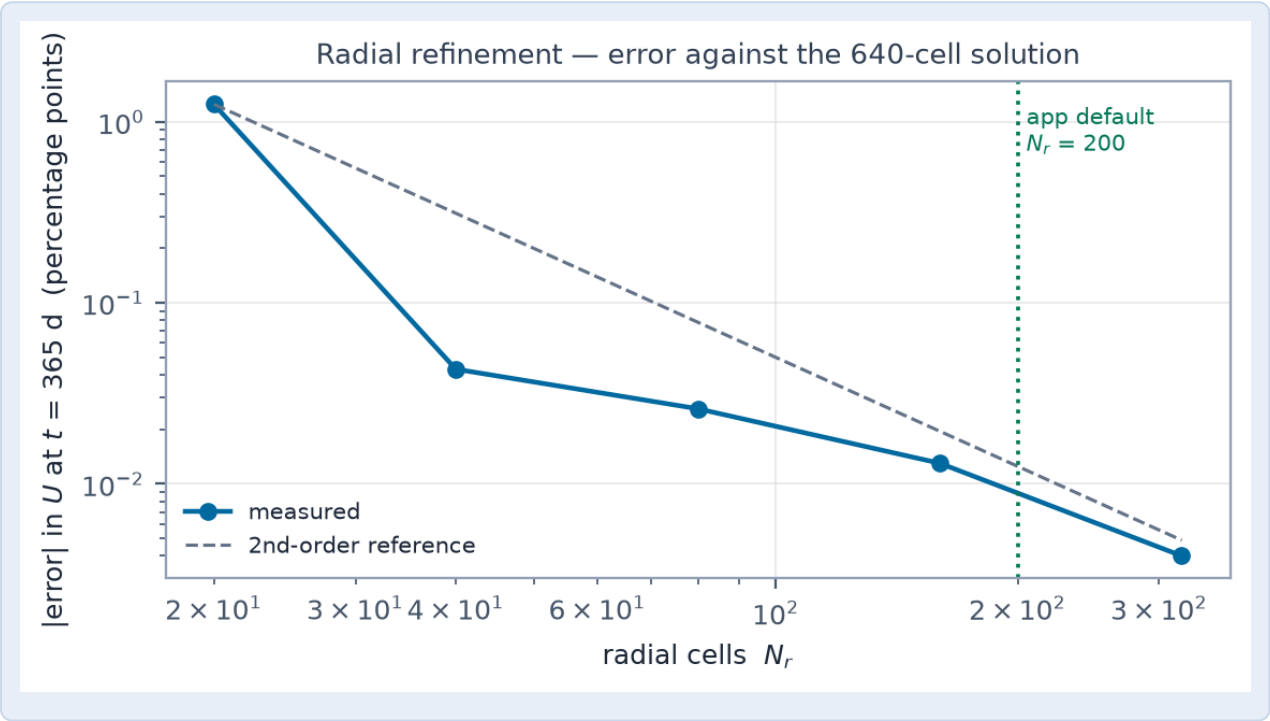


Figure 10.1. Radial refinement, log–log. The measured error falls faster than first order and broadly tracks the second-order reference.

The practical reading: at **N_r = 80**, the Design and Simulate default, the answer is already within **0.021 percentage points** of an eight-times-finer mesh. This is the same conclusion the tool's own study reached by a different route (Table 6.3): radial flow is smooth and converges early, so radial resolution is not a limiting source of error in normal use (Figure 10.1 shows the trend on log–log axes). Only the very coarse N_r = 20 case, barely more than the eight-cell-per-zone floor, is meaningfully off, at 1.23 pp.

10.3 Vertical refinement

Here the reference is exact, so both the error and the **order of convergence** can be measured. The order is the slope on a log–log plot; Figure 10.2 draws exactly that plot for this refinement. Halving the cell size should cut the error fourfold for a second-order scheme, and only twofold for a first-order one.

Table 10.2. Vertical refinement against the exact Terzaghi series, mesh and time step refined together; one row per refinement level. H = 5 m, c_v = 0.0015 m²/day, single drainage. Each error is the RMS difference between the solver's U(t) and the exact series; each "observed order" is measured between that row and the row above it. The pair of error columns compares the solver *before* and *after* the boundary fix of §10.4. What to conclude: the as-shipped column halves its cell size and cuts its error close to fourfold each time (order ≈ 2, as designed); the pre-fix column only halves it (order ≈ 1). Figure 10.2 shows the same data on log–log axes with reference slopes of 1 and 2, where the order can be read as a slope. Measured by `validation_2026/num_guide_vert.js`.

N_z (cells)	time steps	RMS error vs exact, pre-fix (pp)	observed order (vs row above)	RMS error vs exact, as shipped (pp)	observed order (vs row above)
5	1000	1.5713	—	0.5060	—
10	2000	0.8755	0.84	0.1204	2.07
20	4000	0.4603	0.93	0.0292	2.05
40	8000	0.2358	0.97	0.0072	2.02
80	16000	0.1193	0.98	0.0018	2.00

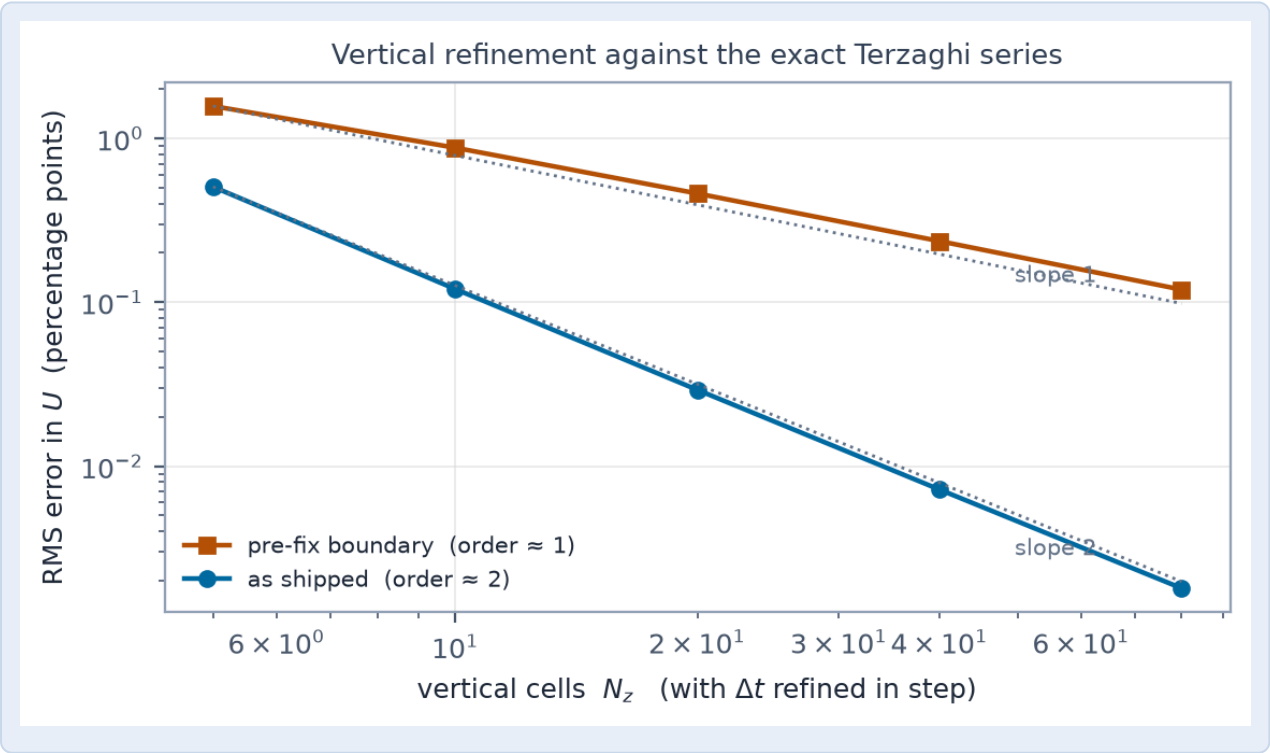


Figure 10.2. Vertical refinement against the exact solution. The shipping solver converges at second order (slope 2); the pre-fix boundary converged at only first order.

10.4 A defect this study found, and fixed

The two right-hand columns of Table 10.2 exist because writing this part uncovered a real defect. The plain-language version first. A scheme built from Crank–Nicolson and Peaceman–Rachford is second order: every halving of the grid should cut the error **fourfold**. When the convergence study was first run, every halving cut the error only **twofold**. The answers were still close; close enough, in fact, that every benchmark passed. But the *rate* was that of a first-order scheme. A second-order scheme that converges at first order has a bug somewhere on a boundary, because a single mis-weighted row is exactly the kind of error that drags the whole order down without making any one answer obviously wrong.

What the error was: in the ADI second half-step, at drained top and base rows, a ghost-cell doubling was applied on top of a coefficient that was already correct. What it did: the implicit vertical operator was twice as stiff as the explicit one for the same face, which broke the symmetry the splitting relies on and dropped the vertical scheme to first order. What fixed it: removing the duplicated factor of two. The detail follows.

The cause was in the ADI second half-step, at drained top and base rows. The distance from the boundary cell centre to the boundary face is half a cell, and the coefficient was already built with that half-cell distance — making it the correct Dirichlet-at-the-face value, and matching what the first half-step used on its explicit side. A ghost-cell doubling was then applied *on top* of it. The implicit vertical operator was therefore twice as stiff as the explicit one for the very same face, breaking the symmetry the splitting relies on. One row out of N_z being wrong by a factor of two is enough to drag the whole scheme down to first order.

> **KEY IDEA**
>
> **The fix was to remove the duplicated factor.** The measured order returned to **2.00**, and the error on the default mesh fell from **0.875** to **0.120** percentage points — better by a factor of **7.3**. The in-app Terzaghi benchmark improved in step, from 1.063 % mean absolute error to 0.116 %.

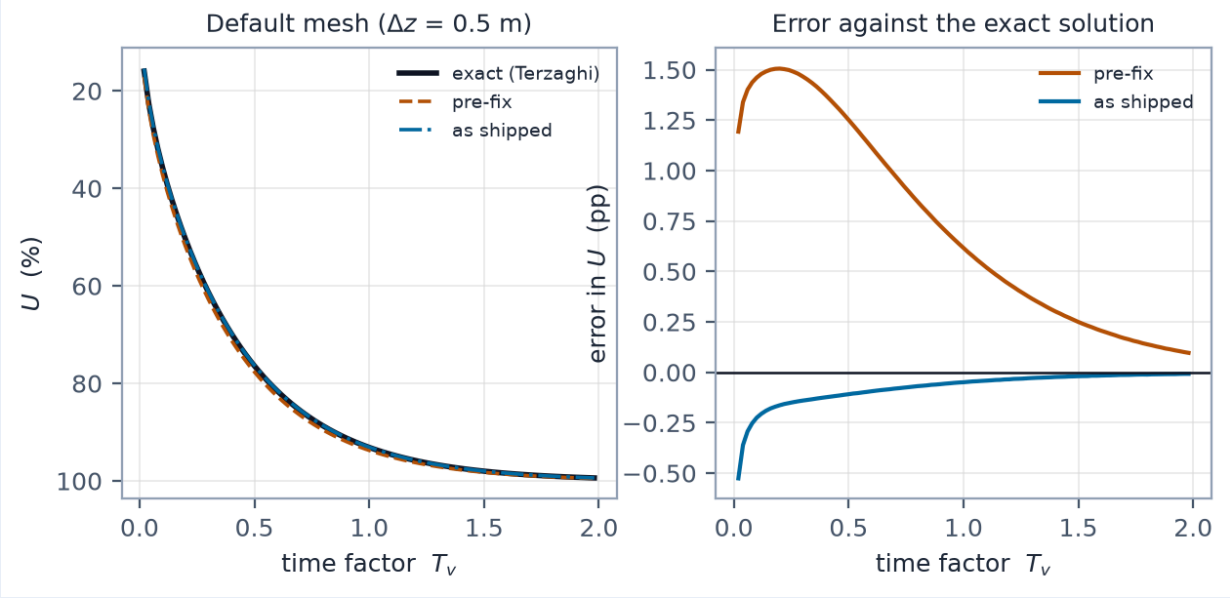


Figure 10.3. The consolidation curve on the default mesh, against the exact solution, before and after the fix. Left: the curves themselves. Right: the error, which the fix reduces several-fold across the whole history.

Two things are worth drawing out. First, the defect was invisible to every existing test: the solver still *converged*, just slowly, and every benchmark still passed its tolerance (Figure 10.3 shows the before-and-after curves). Only measuring the *rate* of convergence exposed it. Second, it affected only runs with $c_v > 0$ *and* a drained boundary — the Design and Simulate examples, which run the $c_v = 0$ radial path, returned identical answers before and after (r_e = 791 mm, unchanged).

10.5 Time-step refinement

Refining the time step alone, on a fixed mesh, shows the other half of the picture (Table 10.3 and Figure 10.4).

Table 10.3. Time-step refinement at a fixed mesh (Δz = 0.125 m).

time steps	Δt (days)	RMS error (pp)
250	133.3	0.3825
500	66.7	0.1585
1000	33.3	0.0669
2000	16.7	0.0231
4000	8.3	0.0072
8000	4.2	0.0072

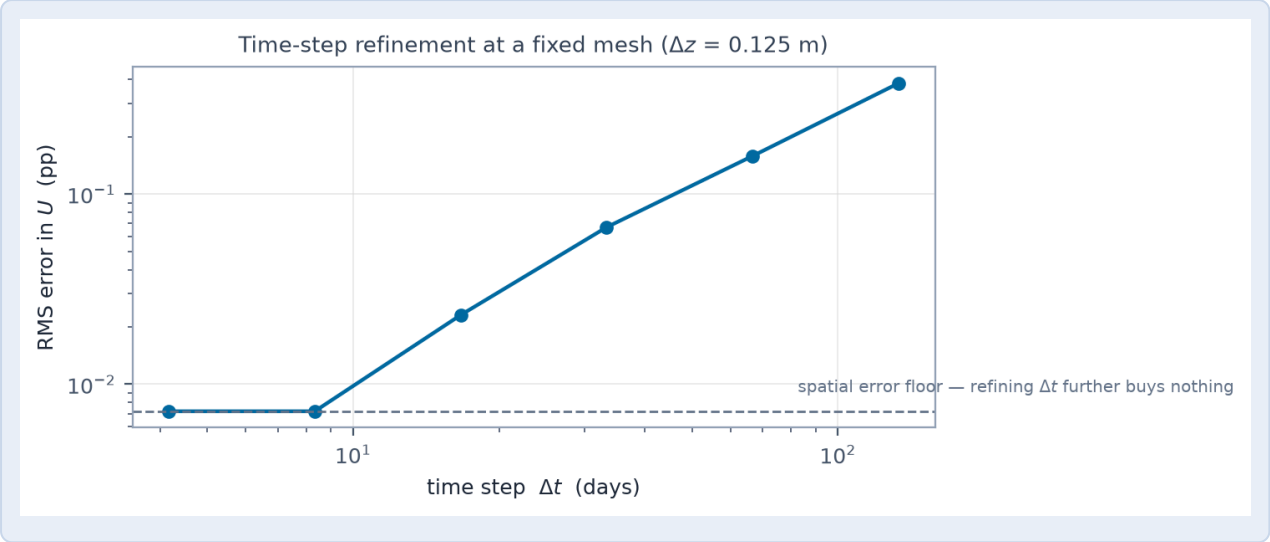


Figure 10.4. Time-step refinement at fixed mesh. The error falls until it reaches the floor set by the spatial discretisation, after which a smaller step buys nothing.

KEY IDEA

The error falls with Δt and then **stops**, at the floor set by the mesh. This is the expected and healthy behaviour, and it carries a practical lesson: refining one and not the other wastes effort. Beyond the floor, a smaller time step costs runtime and returns nothing.

10.6 What the default mesh delivers

Table 10.4 collects what the shipping defaults deliver, and against which reference each was measured.

Table 10.4. Accuracy of the shipping defaults.

Direction	Default	Measured error	Reference
radial	N_r = 80 (Design & Simulate)	0.021 pp	640-cell solution
vertical	Δz = 0.5 m (Design & Simulate)	0.120 pp RMS	exact Terzaghi series
both, inversion	40 / 1 d / 0.25 m (Grid-converged preset)	~0.5 % settlement bias	N_r 360 / Δz 0.10 / Δt 0.25 reference

Both are far below the uncertainty in the soil parameters themselves, which is where the real error in any consolidation prediction lives. A tenfold uncertainty in k_0 is ordinary; a tenth of a percentage point of discretisation error is not the thing limiting your design.

CHAPTER 11

Verification against closed-form solutions

11.1 The eight benchmarks

Convergence (the previous Part) shows the scheme solves its equations consistently; the closed-form benchmarks of this Part show it solves them *correctly*. Both are verification in the sense of the introduction's taxonomy — validation, the test against real ground, is the field comparison of Part Three. The tool ships eight benchmarks against published closed-form solutions, runnable from inside the app; Table 11.1 lists all eight with their measured errors, and the full report with curves is `validation_2026/solver_verification_report.pdf`.

Table 11.1. The eight analytical benchmarks, as measured by the shipping build. Errors are in percentage points of U.

Benchmark	What it tests	MAE	Max |Δ|
Olson–Barron	ramp loading, ideal drain	0.398 %	0.742 %
Hansbo	radial consolidation with smear	0.299 %	1.030 %
Hansbo · Axisym2D	the layered stacker, ideal drain	0.530 %	2.179 %
Hansbo Smear · Axisym2D	the layered stacker, with smear	0.254 %	0.963 %
Terzaghi 1-D	pure vertical drainage	0.116 %	1.119 %
Carrillo combined	radial and vertical together	0.000 %	0.001 %
Vacuum-assisted	suction boundary condition	0.530 %	2.178 %
Partial penetration	degenerate limits: 100 % ≡ Hansbo, 0 % ≡ Terzaghi	0.016 %	0.290 %

Every benchmark sits at or below about one percentage point of mean absolute error, and six of the eight are below 0.55. The larger maximum errors on the two radial-stacker cases are a single point on the steep initial rise, where the output sampling is coarsest; they decay immediately afterwards. For the vacuum case the residual is *identical* with and without vacuum applied, which is the cleanest possible demonstration that the suction boundary condition itself is exact and the residual belongs to the radial discretisation.

11.2 What a benchmark can and cannot prove

Each closed-form solution holds only under its own assumptions — constant coefficients, a particular loading, a particular geometry. To compare against Hansbo the solver must be run with a constant c_h, no stress-dependence, and instantaneous loading. So the benchmarks verify the **machinery**: the discretisation, the mesh, the boundary conditions, the time integration. They cannot verify the **physics extensions** (stress-dependent c_h, creep, hysteresis), because no closed-form solution exists for those.

Table 11.2 states, for each physical mechanism, which benchmark or study verifies it.

Table 11.2. What is verified how.

Feature	How it is checked
Radial discretisation, smear, mesh	closed-form (Hansbo, Olson–Barron)
Vertical discretisation, drainage BCs	closed-form (Terzaghi) — exact, hence Chapter 10
Vacuum boundary condition	closed-form, plus superposition invariance
Partial penetration	degenerate limits collapse onto Hansbo and Terzaghi
Stress-dependent c_h, creep, hysteresis	**no closed form exists** — checked by internal consistency, degenerate limits (switch the feature off and recover the classical answer), and the internal test suite

> **WATCH OUT**
>
> This is the honest boundary of what this verification establishes. The classical machinery is verified against published solutions. The advanced physics is verified to be *self-consistent and to reduce correctly to the classical case* — which is necessary, but is not the same as having been proved right against an independent exact solution, because none exists. Field back-analysis is what tests the physics.

11.3 Verifying the parallel workers

Probabilistic runs and sweep searches distribute their independent solves across a pool of web workers. This raises an obvious question of trust: is a parallel answer the same as a serial one?

It is, by construction rather than by tolerance. The workers do not carry a copy of the solver — they extract the engine source *verbatim at runtime* from between marker comments in the shipped file, so the worker numerics are byte-identical to the main thread. The workers only *solve*; the main thread does all the *judging*, with the same code the serial path uses, so the outcome cannot depend on which worker finished first. An end-to-end parity check runs both an uncoupled and a coupled case through worker and main thread and compares: the current build reports a maximum difference of **exactly zero** on settlements, pressures and plotted curves.

CHAPTER 12

Limitations, and checking your own run

12.1 Assumptions built into the numerics

Collected in one place, in Table 12.1, so a design report can state them.

Table 12.1. Numerical assumptions and their consequences.

Assumption	Consequence	When it matters
Unit cell with a no-flow outer wall	over-predicts rate where drains are not identical	block edges, spacing changes, adjacent untreated ground
Vertical permeability is derived, not typed (§8.2)	k_v is recovered per layer as $c_v \cdot m_v$, so a layered profile does carry a real contrast — but it is only as good as the m_v you supply.	profiles where m_v is left at a nominal value while c_v is varied per layer
Coefficients frozen within a step	coefficients lag the solution slightly	very rapid loading with strongly stress-dependent c_h; shrinks with the step
Pore pressure clamped non-negative	strict conservation is broken while the clamp is active	only when it would otherwise go negative; disabled automatically for unloading and vacuum runs
Small strain; the mesh does not deform	geometry is fixed as settlement proceeds	very large strains, where drain spacing and layer thickness change materially
Fully saturated, Darcy flow	no partial saturation, no threshold gradient	desiccated crusts, very low gradients in some clays

12.2 A five-minute numerical check

For a design you are going to sign, the numerical quality of the run can be established quickly.

REPRODUCE IT

1. **Read the mesh back.** Open the *Numerical mesh* results card and check $N_r \times N_z$. Confirm the layer count and thicknesses are what you intended.
2. **Halve the vertical cell size** and re-run. If the answer moves by an amount you would care about, the mesh was too coarse for your profile — most likely because a layer is thin relative to the 0.5 m default.
3. **Halve the time step** and re-run. Expect a smaller change than step 2; if it is larger, your loading is rapid relative to the step.
4. **Check the degenerate limit.** Set $\eta = 1$ (no smear) and compare against the Hansbo ideal curve — the built-in benchmark cases do exactly this, and you can reproduce it with your own geometry.
5. **Sanity-check U.** Degree of consolidation is a degree of *primary* consolidation and must never exceed 100 %. Creep and immediate settlement belong to the settlement, not to U.

RECAP

The short version. The tool discretises the conservative form of the consolidation equation with cell-centred finite volumes on annular control volumes. The mesh is built around the physical boundaries, so every permeability jump falls on a cell face. Face conductivities are harmonic means, so each face carries one shared flux and the scheme conserves water exactly. Time integration is Crank–Nicolson, or Peaceman–Rachford ADI when the vertical direction is coupled; both are second order and unconditionally stable. Convergence has been measured, not assumed: second order in both directions, with the default mesh delivering around a tenth of a percentage point of error against exact solutions. Eight closed-form benchmarks verify the machinery. The remaining uncertainty in any real prediction lies in the soil parameters, not in the numerics — which is exactly where it should lie.

PART TWO

Design and Simulation

The two everyday forward tasks: sizing drain spacing to hit a target, and forecasting what a chosen spacing will actually do — deterministically, and with the uncertainty carried through.

What you will be able to do after working through this part

1. Say precisely what Design and Simulate each take in and give back, and why Design is just Simulate run in a loop.
2. Explain why the tool derives c_h from k_0 and the compression indices instead of asking you for a single c_h — and read the resulting non-monotonic rate curve.
3. Keep U_s and settlement straight: which one creep and immediate settlement enter, and which they must never enter.
4. Build design charts for your own soil — spacing against target, and spacing against programme.
5. Set up a probabilistic run, and read the P10/P50/P90 spacing band the right way round.
6. Rank your inputs by how much they actually move the answer, and spend the investigation budget accordingly.
7. State the assumptions your design report is standing on.

CHAPTER 13

The two questions

13.1 One mode, two questions

Design and **Simulate** are one mode of the tool with two sub-modes. They share a soil table, a drain geometry, a loading schedule and a solver; they differ only in which quantity is given and which is solved for — Table 13.1 sets the two side by side.

Table 13.1. The two tasks. Everything else — soil, drain, smear, loading — is identical.

	Simulate	Design
You give	the drain spacing	a target U_s and the time to reach it
It returns	$U_s(t)$, $U_p(t)$ and settlement to t_{max}	the spacing that just meets the target
Solver calls	one	up to 32 (a bisection)
Question	“what will this design do?”	“what design do I need?”
Horizon	t_{max} = max(5 × schedule, 1825 d)	max(1.2 × t_{target}, 365 d)

KEY IDEA

Design is Simulate in a loop. There is no separate design formula and no chart lookup. The tool runs the same forward solver at trial spacings and closes in on the one that meets your target, so *every* physical effect in Simulate is automatically respected by Design. Nothing is linearised for the sake of getting a closed-form answer.

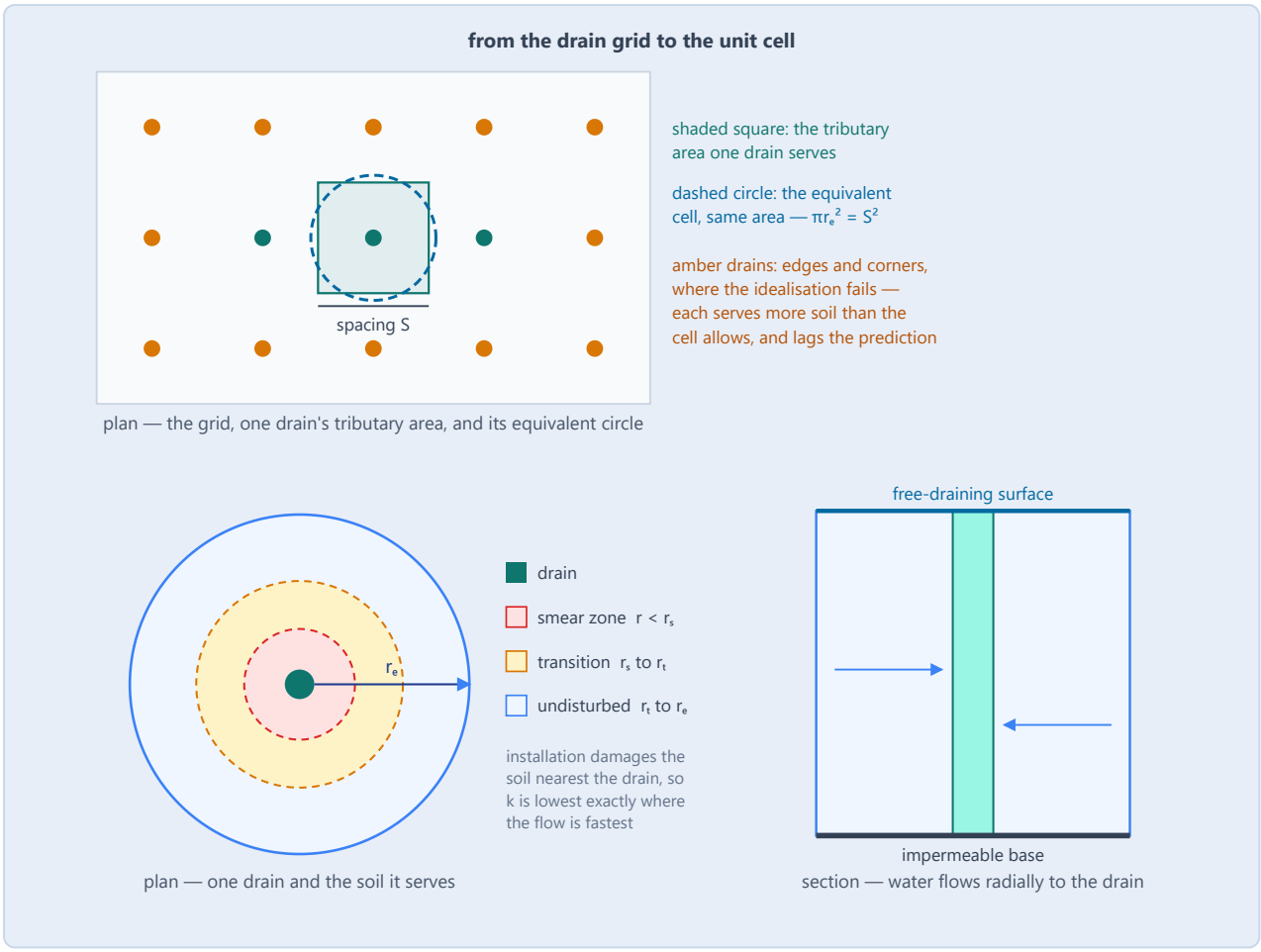


Figure 13.1. Schematic. From the drain grid to the unit cell. Top: each drain serves a tributary area, replaced by the equivalent circle of radius r_e; the idealisation fails at the amber edge and corner drains (the Watch out in §13.2). Bottom left: r_s and r_t bound the smear and transition zones, drawn exaggerated. Bottom right: the r–z section the solver works on; the fixed vertical boundary conditions act only when a layer has a non-zero c_v.

13.2 The unit cell — what is actually solved

Each drain is assumed to serve an identical cell of soil, so one cell represents the whole field — Figure 13.1 shows the construction. Within that cell the tool solves the radial consolidation equation on a cell-centred finite-volume mesh, with the drain wall held at zero excess pore pressure and the cell boundary closed. The mesh places faces exactly on r_s and r_t so no element straddles the permeability step at the edge of the smear zone.

WATCH OUT

Watch out — the unit cell assumes a large, regular field. Edge drains, corners and irregular patterns see more soil than the cell allows, and will lag the prediction. The idealisation is standard and it is what every design chart in the literature also assumes, but it is an idealisation. One further classical omission is deliberate and justified rather than assumed: the drain itself is treated as freely draining (no well resistance), which is legitimate when the discharge factor $D = q_w/(k_h \cdot l_m^2)$ exceeds about 5 (Mesri & Lo 1991) — a requirement modern band drains exceed by an order of magnitude (Hansbo 1997; Holtz et al. 1991) — and the input page can check D for your own project; the exceptions worth watching are very long drains and organic soils where capacity degrades with time (Miura & Chai 2000).

13.3 What you type, and what the tool works out

Table 13.2 lists every input, and which of them a probabilistic run can carry a range on.

Table 13.2. Inputs, and which of them the probabilistic run can carry a range on.

Input	Symbol	Role	Range?
Layer thickness	H	geometry	no
Saturated unit weight	γ_{sat}	sets $\sigma'_0(z)$	no
Initial hydraulic conductivity	k_0	**the rate**	**yes**
Permeability-change index	C_k	how k falls as the soil compresses	yes
Compression / recompression	C_c, C_r	**the magnitude**	yes
Preconsolidation pressure	σ'_p	where the soil yields	yes
Yield sharpness	β	how abruptly C_r turns into C_c	yes
Initial void ratio	e_0	scales strain	yes
Undrained modulus	E_u	immediate settlement	yes
Vertical consolidation coeff.	c_v	0 ⇒ radial only; > 0 ⇒ coupled r–z	no
Creep ratio	$C\alpha/C_c$	secondary compression	no
Drain present?	—	partial penetration	no
Smear ratio	η	near-drain conductivity loss	**yes — global**
Smear / transition radii	r_s, r_t	extent of the disturbed zone	**yes — global**

The last two rows are marked *global* because smear belongs to the installation rather than to the soil: one rig, one mandrel, one crew, so a probabilistic run draws one smear triple per sample and applies it to the whole cell. Every other ranged input is drawn per layer. §18.4 gives the reasoning and what it is worth.

CHAPTER 14

The soil model

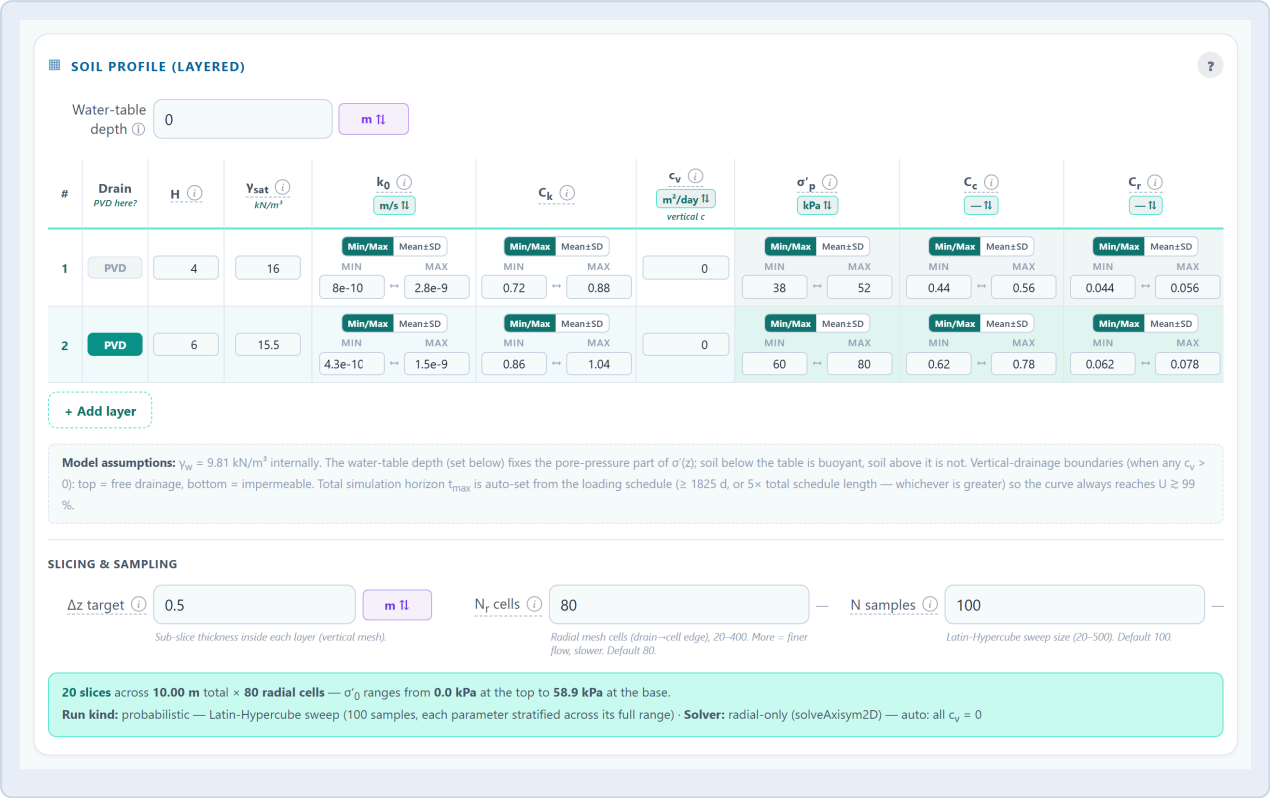


Figure 14.1. The soil table of the Design and Simulate tasks, holding the example project: one row per layer, the rate columns k_0 and C_k left of the compression magnitudes; a single value is deterministic, a Min/Max pair is ranged (what Chapter 17 samples); the smear column at the right is global (*all layers*).

Figure 14.1 shows where every number in this chapter is typed.

14.1 The rate is not a constant

Most hand methods take a single coefficient of consolidation c_h and carry it through. This tool does not. You type an **initial** hydraulic conductivity k_0 and a compression index, and the rate is *derived* at every cell from the current effective stress. The qualifier matters: k_0 is the conductivity at the initial void ratio, and the model moves k away from it through C_k as the soil compresses.

$$c_h(\sigma') = k(e) \cdot (1 + e_0) \cdot \sigma' \cdot \ln(10) \,/\, (\gamma_w \cdot C), \quad \text{with} \quad \log_{10}(k_0/k) = \Delta e \,/\, C_k \tag{14.1}$$

where C is the operative compression index — C_r below the preconsolidation pressure, C_c above it. Two things follow. The rate rises with effective stress, because c_h is proportional to σ′. And it falls as the soil compresses, because k drops with void ratio through C_k. The two effects fight each other, and which wins depends on where the soil is on its compression curve. Figure 14.2 shows the outcome on the example soil.

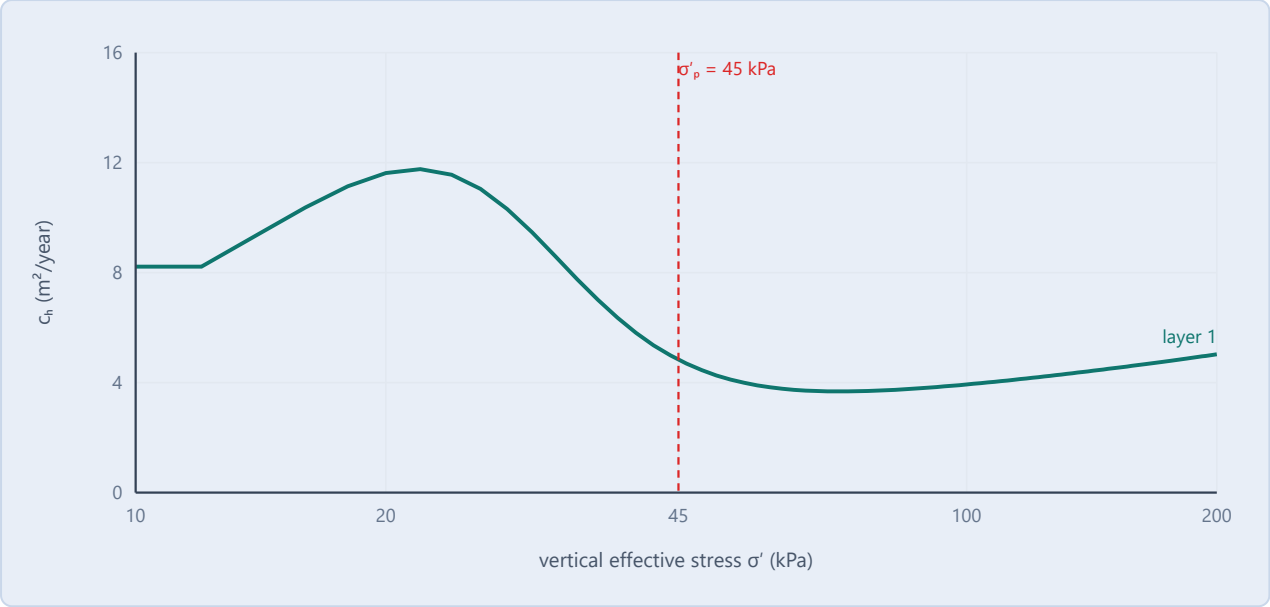


Figure 14.2. The consolidation coefficient the solver actually uses, at mid-depth of layer 1 (σ'_0 = 12 kPa): up to 11.8 m^2/yr at 22 kPa, down to 3.7 at 70 kPa as yield turns C_r into C_c, then a slow recovery (the two mechanisms are separated in §14.6).

KEY IDEA

Why this matters for design. Under an embankment the stress in the ground roughly doubles or triples. On this curve that is not a small correction — the rate varies by a factor of 3.2 across the stress range plotted, and it is not even monotonic. A design based on the initial c_h and a design based on the final c_h are different designs, and both are wrong; the tool avoids having to choose.

14.2 The compression model

Settlement comes from a bilinear e–log σ′ law: slope C_r while the soil is over-consolidated, slope C_c once it yields at σ'_p. The switch is not a corner — the parameter β blends the two slopes over a short stress range, which is both more realistic and much better behaved numerically than a hard kink. Figure 14.3 draws the law. The governing equations (the blend, the strain increment on each branch, and the permeability law) are written out term by term as Equations 1–8 in §14.5 to §14.7.

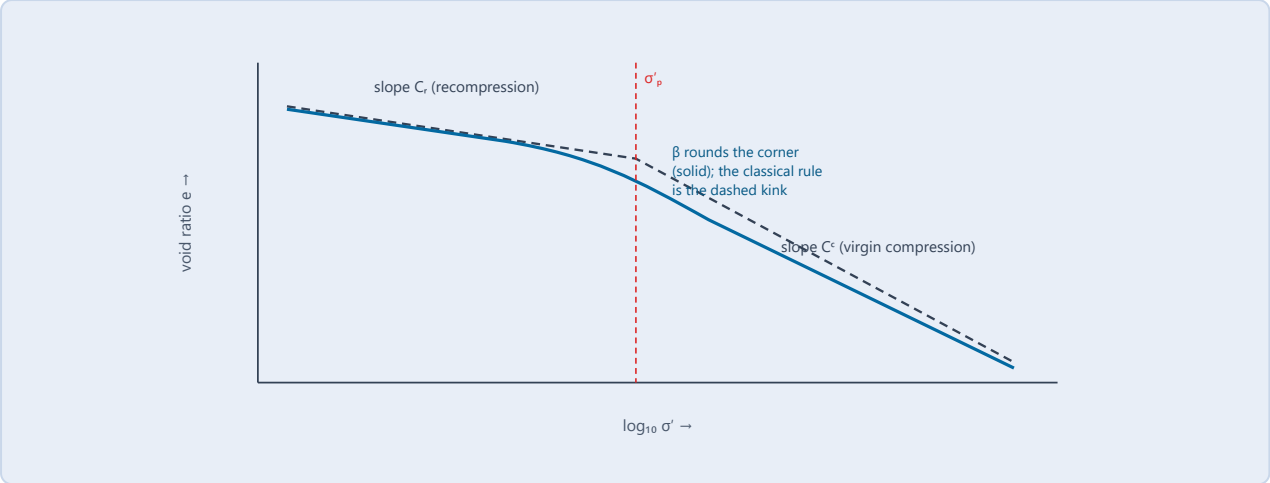


Figure 14.3. Schematic. The compression law in e–log σ′ space: the flat C_r branch, yield around σ'_p, the steep C_c branch. The classical rule (dashed) switches slope at a point; the β-blend (solid) rounds the corner, C_r and C_c still fixing the limiting slopes. Measured versions: §14.7.

14.3 What else is in the physics

Table 14.1 lists the effects the solver carries beyond primary radial consolidation.

Table 14.1. Effects the solver carries beyond primary radial consolidation. Hysteresis and creep are on by default; vacuum has its own on/off switch plus a schedule; partial penetration, smear and immediate (E_u) settlement switch on when you enter them.

Effect	Governed by	Enters
Smear and transition zone	$\eta = k_h/k_s$, r_s, r_t	reduced conductivity near the drain
Immediate (undrained) settlement	E_u	settlement only — follows the load, not time
Secondary compression (creep)	$C\alpha/C_c$	settlement only — Bjerrum equivalent time
Unload–reload hysteresis	C_r and the stress history	settlement on any unloading step
Vacuum preloading	suction schedule, k_2	drain boundary condition, with depth loss
Partial penetration	per-layer drain on/off	forces the coupled r–z solver
Vertical drainage	$c_v > 0$ on any layer	forces the coupled r–z solver

14.4 U is a degree of PRIMARY consolidation

Creep and immediate settlement are added to the reported *settlement*, and never to U_s or U_p. This is deliberate: a degree of consolidation is the fraction of the primary process that has completed, and it must not exceed 100 %. Mixing creep into the numerator while the denominator stays primary produces exactly that nonsense. Table 14.2 decomposes the worked example's settlement into the three parts. U_s and U_p are two different measures of that primary process, one from settlement and one from pore pressure. They do not coincide in this model; §15.2 explains why, which one the Design

target uses, and which one each field instrument actually measures. The exact denominator computation — span, hardening state, vacuum contribution, partial-penetration areas — is specified in Appendix A of the combined book.

Table 14.2. Settlement at t_{max} for the worked example at 1.2 m square spacing, decomposed. The three parts are what the tool reports as settlement; only the first is what U_s measures.

Component	Settlement	Share	Counted in U_s?
Primary consolidation	825 mm	60 %	**yes**
Secondary compression (creep)	304 mm	22 %	no
Immediate (undrained)	253 mm	18 %	no
Total reported settlement	**1383 mm**	100 %	—
Primary ultimate (the U_s denominator)	825 mm	—	U_s = 99.7 % of it

WATCH OUT

Watch out — do not read U_s = 100 % as "settlement has stopped". On this example U_s reaches 99.7 % while 557 mm of the 1383 mm total (40 %) is not primary consolidation at all. Creep in particular keeps running after U_s has flattened. Design to the settlement number, not to the percentage.

14.5 How every parameter follows the effective stress

§14.1 and §14.2 said the rate and the compressibility both depend on stress. This section gives the actual chain, because it is the part of the model that most distinguishes this tool from a hand calculation, and because every step of it is visible in the results.

The whole chain hangs off one dimensionless number. The **blend fraction** says how far the clay has moved from over-consolidated behaviour toward normally consolidated behaviour:

$$w(\sigma') = 1 \,/\, [\, 1 + (\sigma'_p \,/\, \sigma')^{\beta} \,] \qquad (1) \tag{14.2}$$

It is zero well below the preconsolidation pressure, one well above it, and exactly ½ *at* σ'_p — so σ'_p is the centre of the transition, not a switch. The **operative compression index** is then just the two classical indices mixed in that proportion:

$$C(\sigma') = (1 - w)\cdot C_r + w\cdot C_c \qquad (2) \tag{14.3}$$

That single quantity carries the stress dependence into everything downstream. β controls *only* the width of the transition; C_r and C_c still fix the two limiting slopes and σ'_p still fixes its centre, exactly as they always did. Figure 14.4 draws Equation 1; Figure 14.5 applies it to the indices.

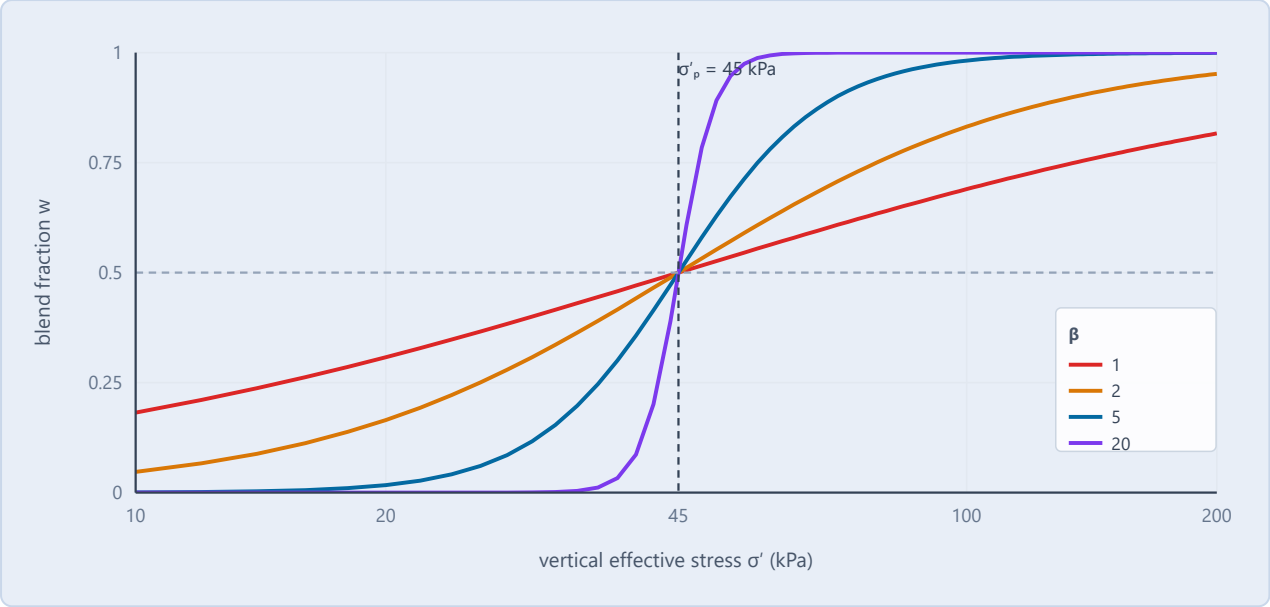


Figure 14.4. The blend fraction of Equation 1: every curve passes through w = ½ at σ'_p (the centre of yield, not a switch); β alone sets the abruptness — essentially complete within a few kPa at β = 20, spread across the whole stress range at β = 1.

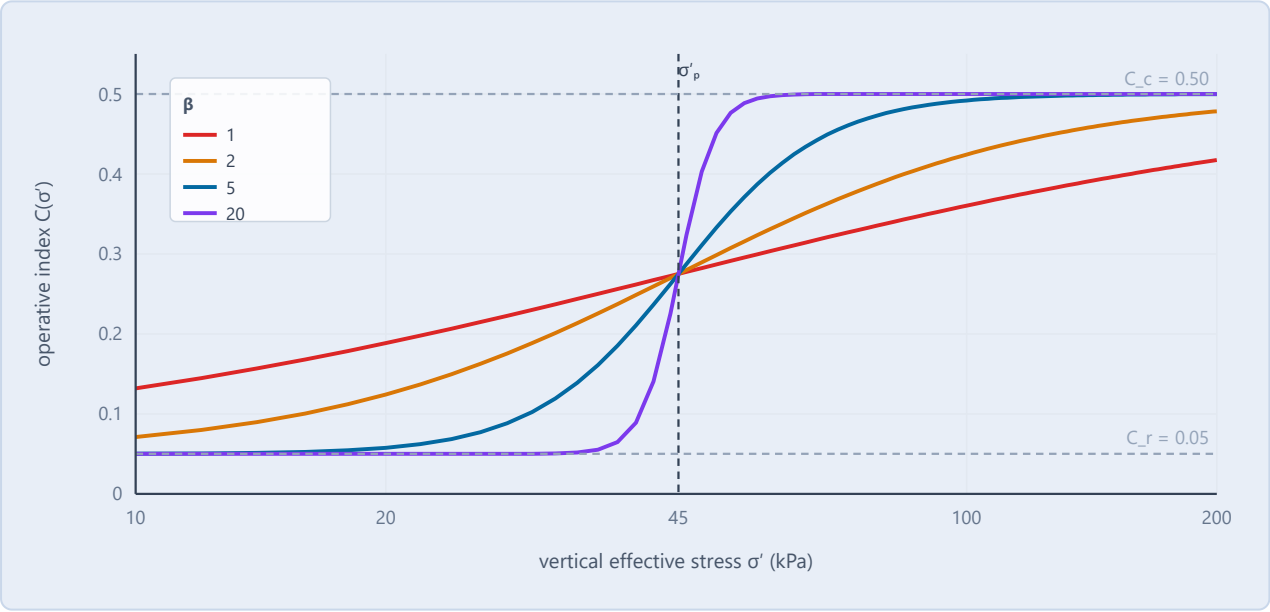


Figure 14.5. The blend applied to the compression indices (Equation 2): from C_r to C_c — a factor of 10 on this soil — over a stress span set by β; the classical bilinear rule is the β → ∞ limit.

14.6 From the operative index to the consolidation coefficient

With C(σ′) in hand the rest is bookkeeping, but it is worth following because it explains a curve that surprises people. Descending the compression curve gives the current void ratio, which drives the permeability down; the same operative index gives the compressibility; and the two combine into c_h:

$$\Delta e = C(\sigma')\cdot\log_{10}(\sigma'/\sigma'_0) \quad (3) \qquad k_h = k_0 \cdot 10^{-\Delta e/C_k} \quad (4) \tag{14.4}$$

$$m_v = 0.434 \cdot C(\sigma') / [\,(1 + e_0)\cdot\sigma'\,] \quad (5) \qquad c_h = k_h / (m_v \cdot \gamma_w) \quad (6) \qquad (14.5)$$

KEY IDEA

Two tendencies pulling opposite ways. Rising effective stress *raises* c_h through the compressibility term — m_v carries σ′ in its denominator. But the compression that same stress causes *lowers* c_h, because the void ratio falls and takes the permeability with it through Equation 4. Neither dominates everywhere, which is why Figure 14.6 below is not monotonic and why no single c_h can stand in for it.

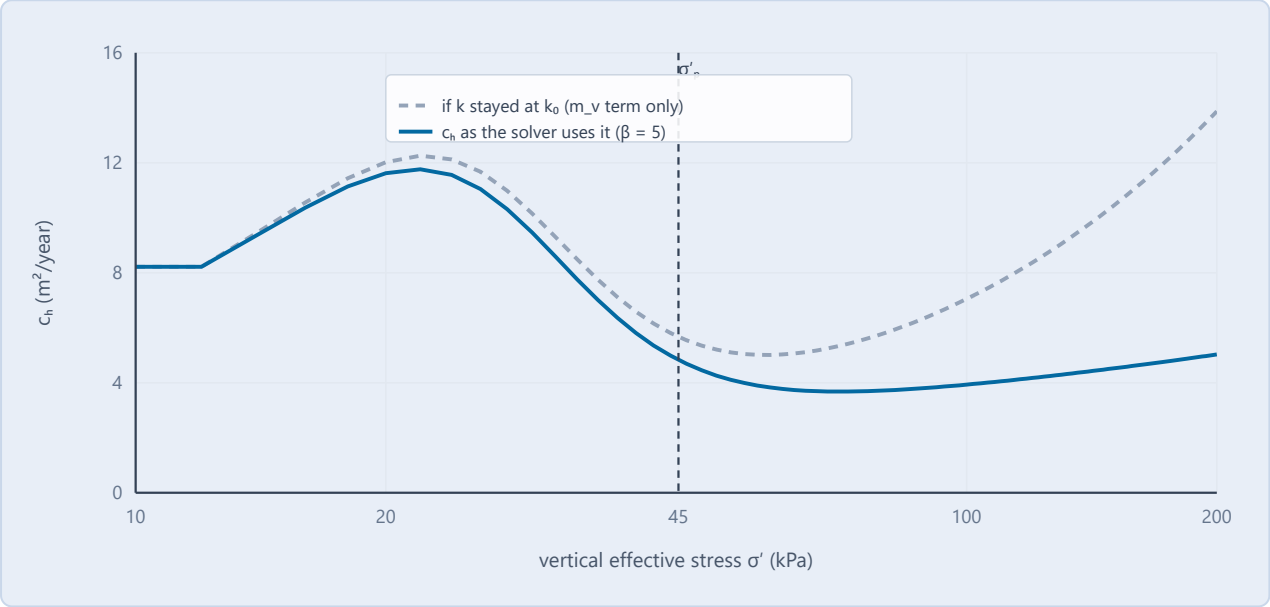


Figure 14.6. The coefficient the solver uses (solid) against what it would be if the permeability never fell (dashed): 11.8 m²/yr at 22 kPa, down to 3.7 at 70 kPa, then a slow recovery; by 200 kPa the permeability term has removed 64 % of the coefficient.

How much of that shape is β's doing? Figure 14.7 sweeps it.

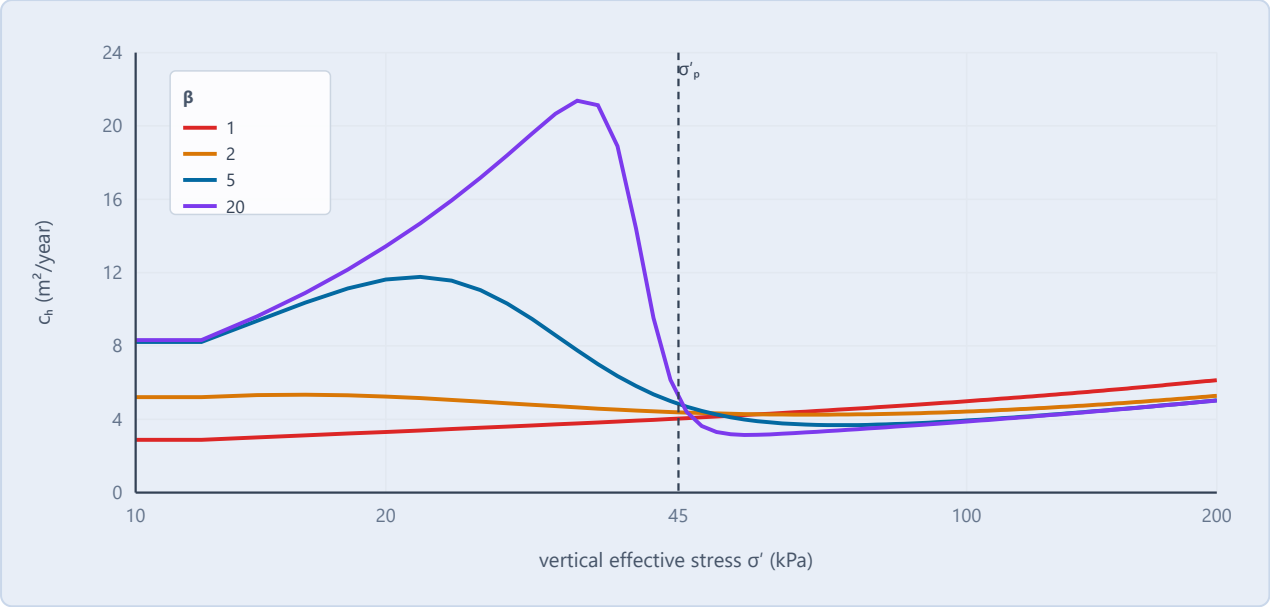


Figure 14.7. The same coefficient at four values of β. At β = 20: 21.4 m^2/yr at 34 kPa down to 3.1 at 54 kPa — a factor of 6.8 inside 20 kPa; at β = 2 only 1.3. The bilinear limit's near-discontinuous rate field is what a time-marching scheme handles worst — an artefact of the switch, not the soil.

14.7 Why the transition is smoothed, and what it costs

The classical method switches from C_r to C_c the instant σ′ crosses σ'_p. That is a modelling convenience, not an observation: real clays do not yield at a single pressure, because a natural deposit contains a spread of particle arrangements, bond strengths and stress histories, and they do not all give way at once. Smoothing is the more honest description as well as the better-behaved one.

It also matters numerically. Settlement is integrated **incrementally**, in small effective-stress steps, each evaluated at the geometric mean of its own end points:

$$\sigma'_{mid} = \sqrt{(\sigma'_i \cdot \sigma'_f)} \quad (7) \qquad \Delta\varepsilon_i = C(\sigma'_{mid})/(1 + e_0) \cdot \log_{10}(\sigma'_f/\sigma'_i) \quad (8) \qquad (14.6)$$

The geometric mean is the natural midpoint on a logarithmic compression line, and stepping this way means each increment inherits the stress the previous one left behind — so the profile stiffens as it consolidates instead of every increment acting on a fresh deposit. With a bilinear index the step that straddles σ'_p has to pick one slope or the other and is wrong either way; with the blend it simply uses the operative value at its own midpoint. Figure 14.8 shows the two rules on the example soil.

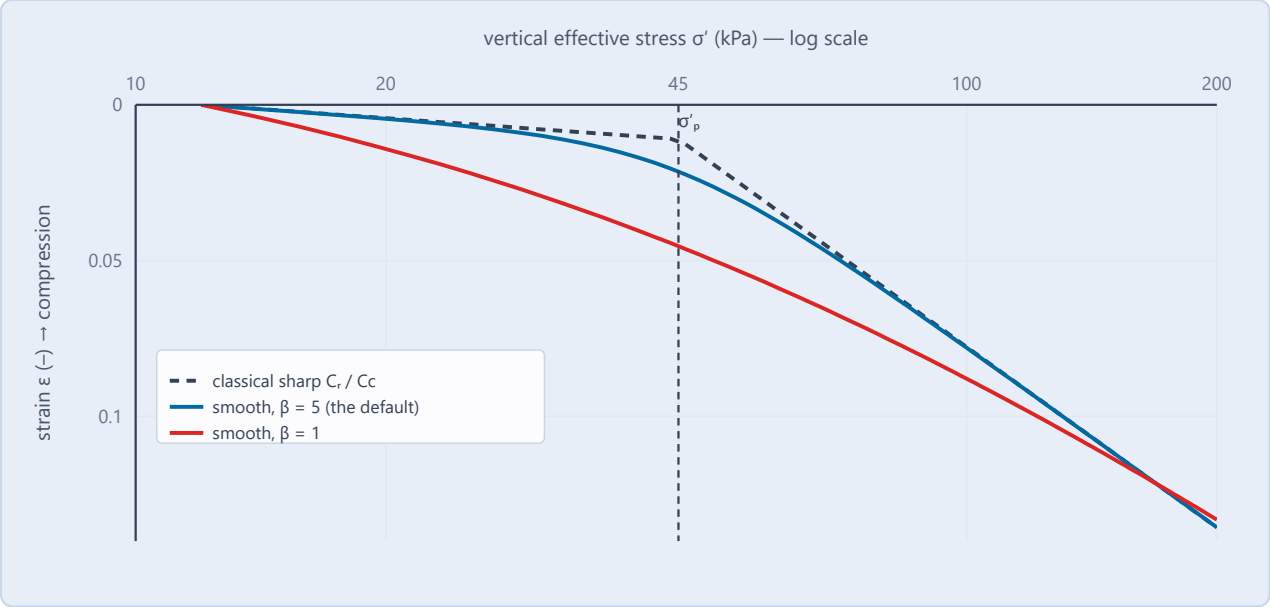


Figure 14.8. Strain against effective stress (log axis, strain downward): the sharp rule kinks at σ'_p (slope jump ×10); β = 5 rounds it. Largest gap at 44 kPa — 88 % more strain (0.020 against 0.011) — converging to 0.0 % by 200 kPa. β is a soil property, not a smoothing fudge.

14.8 How much does β actually change the answer?

A fair question about any added parameter is whether it earns its place. Sweeping β through the design search gives a direct answer: Table 14.3.

Table 14.3. Required square spacing at 90 % in one year, as β varies with everything else held at the example's mid-range values.

β	Required drain spacing	Δ from the default (β = 5)	Character of the yield
1	1.734 m	+0.294 m	very gradual — a markedly different soil
2	1.591 m	+0.151 m	very gradual — a markedly different soil
3	1.510 m	+0.070 m	gradual
5	1.440 m	—	realistic for a natural clay
10	1.412 m	-0.028 m	realistic for a natural clay
50	1.422 m	-0.018 m	effectively the classical bilinear switch

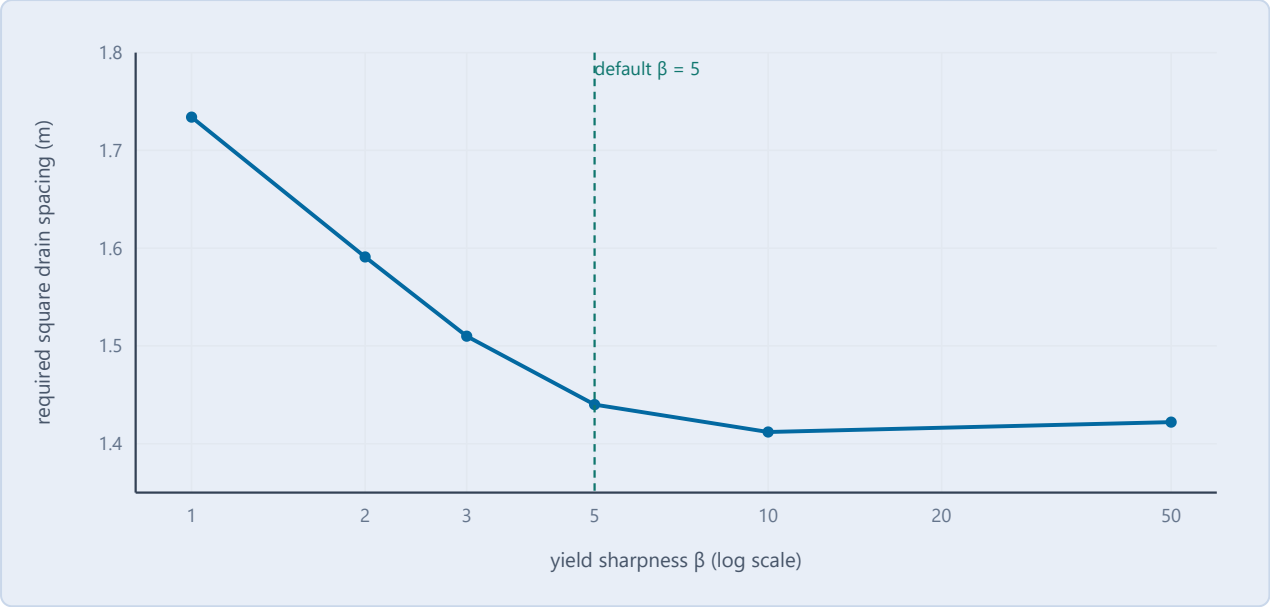


Figure 14.9. Required spacing against β, everything else at the example's mid-range values: steep below β ≈ 3, flat above — converged onto the classical bilinear result. The default β = 5 sits on the flat part.

KEY IDEA

β matters where it is small, and stops mattering once it is realistic. Figure 14.9 is the picture; the word *plateau* here means the range of β over which the computed required drain spacing is effectively insensitive to β. Across the full sweep the answer moves 0.322 m — as much as the smear parameters. But almost all of that happens below β = 3. From β = 5 upward the spacing sits within 0.028 m, i.e. the answer has *converged onto the classical bilinear result*. That is the useful property: the smooth model contains the sharp one as a limiting case, so nothing is lost, while the default β = 5 sits on the plateau where the exact value is not critical. It is why β is held constant by default rather than searched.

WATCH OUT

Watch out — do not read the plateau as "β is free". A soil that genuinely yields gradually (β ≈ 1–2) needs 0.29 m more spacing than the default assumption, which is a 31 % difference in drain count. The plateau says the answer is insensitive to β *within the normal range*, not that β can be ignored. If your oedometer curves round off noticeably through yield, say so with a lower β and let the tool respond.

CHAPTER 15

Simulate — the forward question

15.1 One spacing in, one history out

Give the tool a spacing and it returns the whole history: the settlement-based degree of consolidation $U_s(t)$, the pore-pressure-based $U_p(t)$, and the settlement in millimetres. Figure 15.1 shows what spacing does.

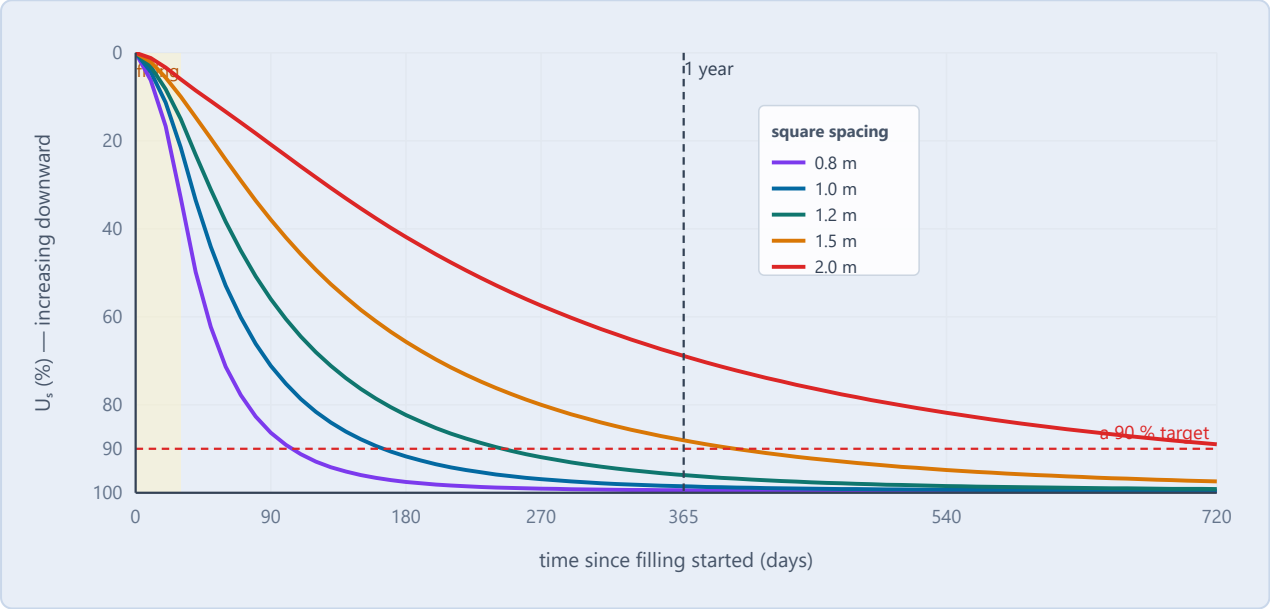


Figure 15.1. Degree of consolidation against time for five square spacings, same soil and load (U_s increasing downward). At one year the tightest spacing is long finished; 2.0 m has reached only 69 %. The drainage path length enters squared — spacing is the designer's most powerful lever.

15.2 Settlement and degree of consolidation are different questions

U_s is a *fraction*; settlement is a *quantity*. Two soils can share a U_s history and settle by very different amounts, and one soil can be at 95 % consolidation while still settling visibly, because creep continues. The tool reports both, and a design usually needs both: the percentage tells you when to stop surcharging, the millimetres tell you what the structure has to tolerate.

There are also **two different degrees of consolidation**, and they are not interchangeable. U_s is computed from settlement: the primary settlement so far over the primary ultimate. U_p is computed from excess pore-water pressure: how much of the initial excess has dissipated, averaged over the cell. In the classical constant-coefficient theory the two coincide. Here they do not, for two reasons. First, c_h varies with stress, so the soil compresses at a different pace from the pace at which pressure dissipates. Early on, the pressure near the drain drops quickly while most of the strain is still to come, so U_p typically runs ahead of U_s. Second, layering weights them differently: U_s is weighted by each

layer's share of the settlement, U_p by its share of the stored pressure. Which one matters depends on what you are doing. **The Design target applies to U_s**, because a spacing is chosen to deliver settlement performance. Field instrumentation measures the other one: piezometers read pore pressure, so a dissipation check against field data is a U_p comparison, while settlement plates track U_s. The tool reports both curves so each instrument can be compared against its own quantity (§14.4 states why neither may contain creep).

CHAPTER 16

Design — the inverse question

16.1 Bisection on the radius of influence

Design asks for exactly two numbers, shown in Figure 16.1: a date, and the degree of consolidation to have reached by then. Everything in this chapter is machinery for satisfying them.

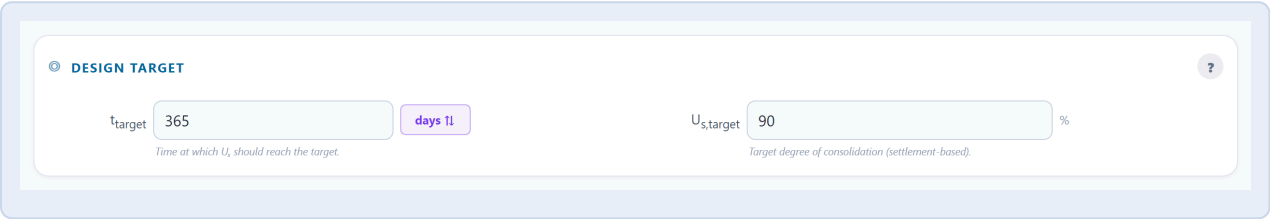


Figure 16.1. The design target: t_{target} and U_s,target (the 90 % used throughout this chapter) — the two fields that turn Simulate into Design: a spacing runs forwards; these two numbers start the search for the spacing that meets them.

Design asks the opposite question, and answers it by search. U_s at the target time falls monotonically as the drains are spread further apart, so a simple bisection converges reliably:

The principle first. U_s at the target time falls steadily as r_e grows, so if one radius meets the target and a larger one misses it, the answer lies between them. An interval known to contain the answer like that is called a **bracket**, and **bisection** means repeatedly solving at the interval's midpoint and keeping whichever half still brackets the answer — each halving doubles the precision. Here r_w is the drain's equivalent radius and r_e the radius of influence being searched for. The search brackets r_e itself, from a lower bound of 0.1 m or 1.05 × r_w, whichever is larger (a cell that tight is barely buildable), up to 3.0 m. In full:

1. **Check the tight end** (1 solver run). If even the tightest spacing cannot reach the target, the run stops and reports *too slow* with a diagnosis — no spacing will fix it.
2. **Check the wide end** (1 run). If 3.0 m still meets the target, the bracket is widened by ×1.7 until it reaches the cap at r_e = 8.0 m, which from 3.0 m takes two steps (3.0 → 5.1 → 8.0, capped); the code allows up to four — a square spacing of roughly 14 m (up to 4 runs). If even that cell still meets it, the run reports *too fast*: the design is unconstrained.
3. **Bisect** (up to 26 runs). Solve at the midpoint; keep the half of the bracket that still contains the answer; stop early once the bracket is narrower than 0.1 mm.
4. **Report**: the converged r_e, then the equivalent square and triangular set-out spacings (§16.2).

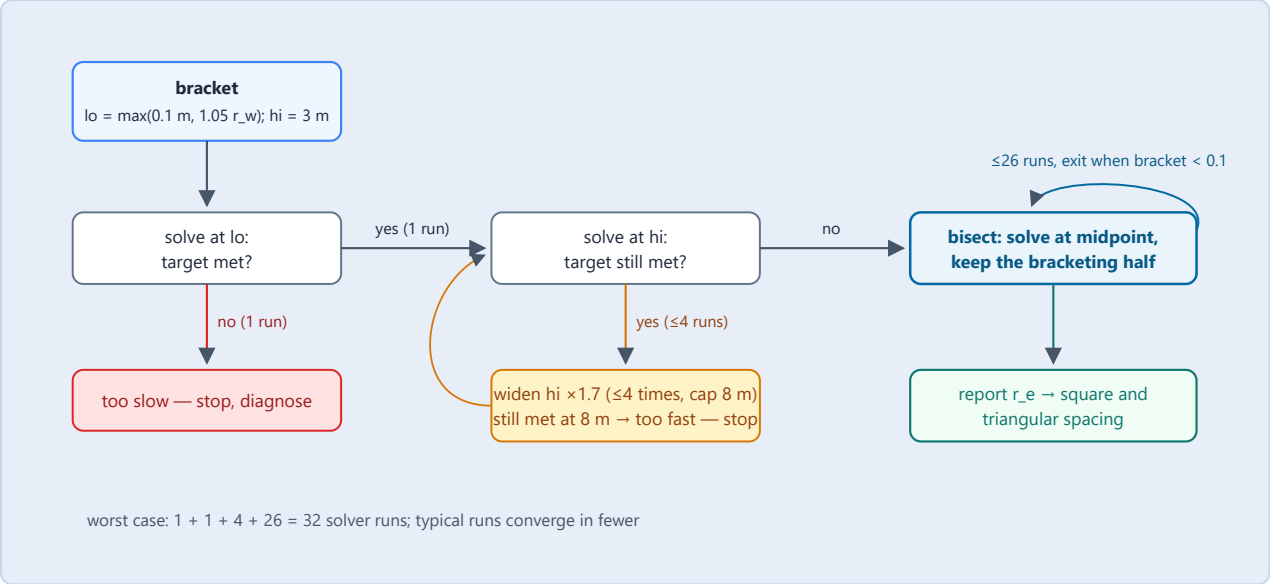


Figure 16.2. Schematic. The Design search as a flow. Each box that says "solve" is one full forward simulation; the counts on the arrows add up to the worst case of 32 runs.

KEY IDEA

Up to 32 solver runs per answer. The arithmetic is on the arrows of Figure 16.2: one check at each end of the bracket (2 runs), at most four widenings (4 runs), and at most 26 halvings (26 runs). The consequence is worth stating plainly: Design is roughly thirty times the work of Simulate, and the probabilistic Design of Chapter 17 repeats the whole search for every sample — which is what makes the parallel machinery of §17.5 necessary.

16.2 From radius of influence to a spacing you can set out

The search works in radius of influence; you set out drains on a grid. The conversion preserves the area each drain serves:

$$\text{square:}\quad S = r_e \cdot \sqrt{\pi} = r_e / 0.5642 \qquad \text{triangular:}\quad S = r_e / 0.5250 \tag{16.1}$$

A triangular pattern packs the same drain density into a spacing about 7.5 % wider. That is why it is usually the cheaper way to hit the same target.

16.3 Design charts

Because Design is cheap to repeat, the tool can be used to build the classical design charts directly for *your* soil rather than reading them off a generic figure. Figure 16.3 and Figure 16.4 are the two classical charts, built by exactly that repetition.

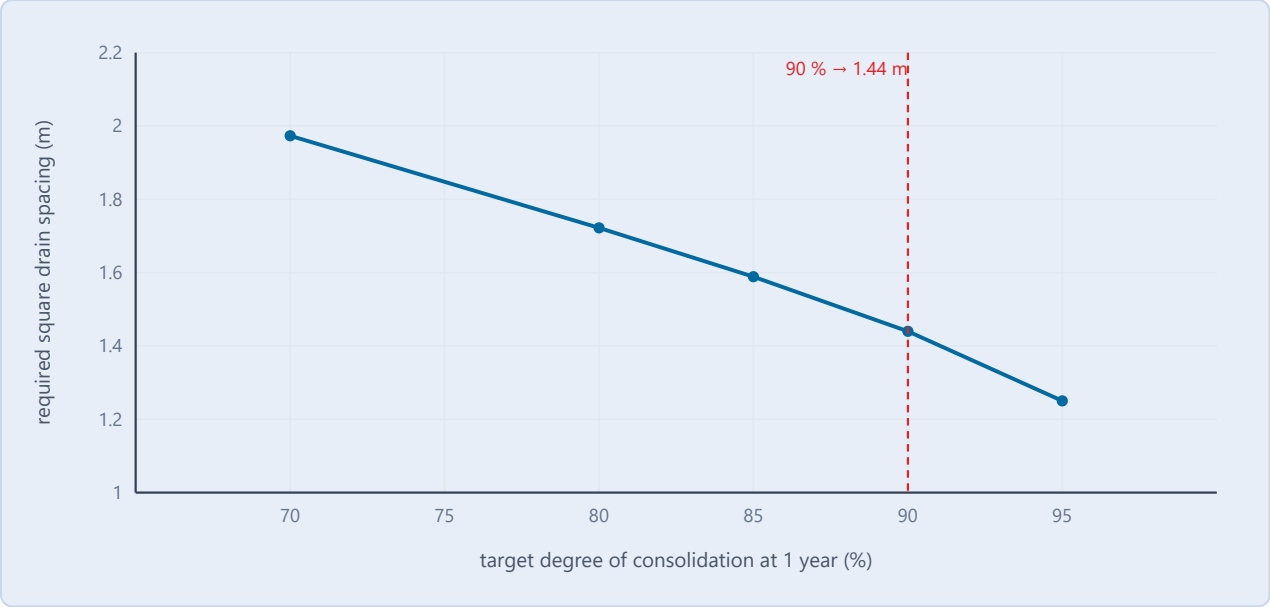


Figure 16.3. Required spacing against target at a one-year deadline: 90 → 95 % costs 0.19 m of spacing; 70 → 75 % costs far less. The last few per cent are the expensive ones.

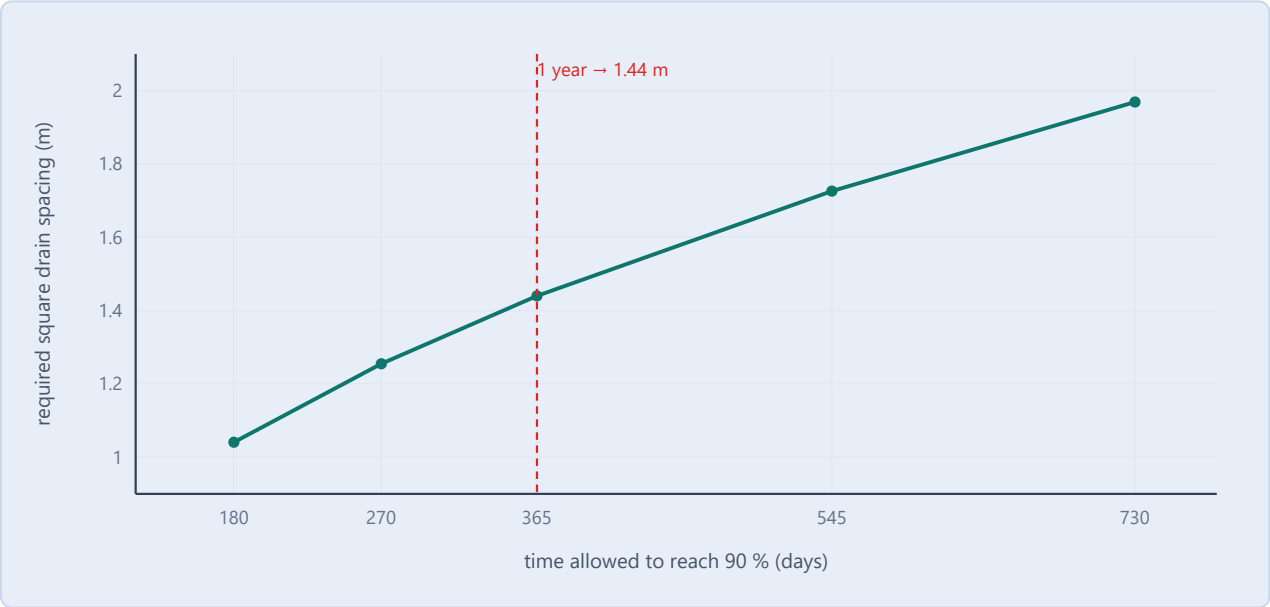


Figure 16.4. The programme trade-off: 365 → 730 days lets the spacing widen from 1.44 to 1.97 m — 47 % fewer drains (2579 against 4823 per hectare). Usually the most valuable chart in a tender.

16.4 When the target cannot be met

Two honest failures are reported rather than papered over. **Too slow** means even the tightest practical spacing misses the target — the soil is too impermeable, the deadline too short, or the target too high; the fix is a programme change, not a spacing change. **Too fast** means even a radius of influence of 8 m (a spacing of roughly 14 m) already meets it, so the design is unconstrained and PVDs may not be warranted at all.

CHAPTER 17

The probabilistic approach

17.1 Why one answer is not a design

Everything so far assumed you know the soil. You do not. A site investigation gives a handful of samples from a deposit that varies laterally and with depth. And the single most influential parameter, hydraulic conductivity, is the one measured least reliably. Laboratory oedometer permeability routinely disagrees with the field value by a factor of two or more, in either direction.

The deterministic answer of 1.44 m is the answer *if* every input happens to sit at the middle of its range. The probabilistic mode drops that assumption. You enter a plausible range for each uncertain input, and the tool returns a distribution of required spacings — which lets you choose a spacing with a stated chance of working, rather than a spacing with an unstated one.

> **KEY IDEA**
>
> **This is what makes the mode advanced.** Most tools bolt uncertainty on at the end, by taking the deterministic answer and applying a factor. This tool propagates uncertainty *through the solver*: each sample is a complete, physically consistent soil profile, and each gets its own full design search. Nothing is linearised, and no distribution is assumed for the output — the shape of the answer is whatever the physics makes it.

17.2 Latin Hypercube sampling

First, what a sample is. **One sample is one complete, internally consistent soil profile.** Each stratum receives its own draw of every ranged parameter within that sample, so a profile with 2 strata still yields 100 samples, not 2 × 100. The count is of profiles, never of individual parameter draws. (100 samples × 16 ranged dimensions — 8 rangeable inputs on each of the 2 layers, the global smear triple left blank here — is 1600 individual draws, if the distinction is ever needed.)

The sample count is a setting, not a constant: the **N samples** field accepts 20 to 500 and defaults to 100, which is what this part's figures use. The trade-off is percentile stability against runtime — each sample costs a full ≈32-run design search, so doubling N doubles the work. Resampling this part's own 100 results shows what the default buys: at n = 50 the P10 estimate wanders over a range of about 0.13 m from draw to draw, at the full n = 100 about 0.10 m. Raise N when the band edges must be quoted tightly; lower it only for exploratory sweeps.

Why Latin Hypercube rather than plain random sampling? Sampling 100 profiles at random from 16 independent ranges wastes many of them: by luck, some regions get several samples and others get none. Latin Hypercube sampling removes that luck. Each parameter's range is cut into 100 equal strata; each stratum is used **exactly once**; and the order of strata is shuffled independently for every parameter so no spurious correlation is introduced. Figure 17.1 shows the difference on two axes.

$$u = (\text{stratum index} + \text{random}) / N \quad \rightarrow \quad \text{value} = \text{Min} + u \cdot (\text{Max} - \text{Min}) \tag{17.1}$$

The jitter inside each stratum keeps the sample random; the one-per-stratum rule keeps it even. The result is a set that covers every input range completely at a sample count where plain random sampling would still be patchy.

Independence between parameters is a statement about the *prior*, not a licence for unphysical recipes: every drawn profile then passes through the same physical clamps the back-analysis enforces — C_r held inside the C_c/C_r window and below 0.95 C_c, σ'_p not below the in-situ stress at the layer's mid-depth, β and $C\alpha/C_c$ inside their physical ranges — and the results page prints how many draws were clamped, so nothing is discarded silently. Known parameter correlations (for example C_k tracking 0.5 e_0) are not yet imposed on the joint draw; a rank-correlation option is recommended future work, and until then the marginal-coverage caveat of the figure above is the honest statement of what the sample guarantees.

That mapping, the second half of the expression above, is what a **Min/Max** cell means: uniform across the range, no value inside it favoured. If instead you fill a cell in as **Mean ± SD**, the same stratified u is pushed through a normal or lognormal inverse instead, so the middle of the range is drawn far more often than the tails. Use Min/Max when you know the bounds and nothing more, which is the usual case; use Mean ± SD when you have enough measurements to justify a shape. The built-in example uses Min/Max throughout, so every figure in this Part is a uniform draw.

One dimension behaves differently by design. **η, r_s and r_t are global**: smear is a property of the installation, not of the soil, so a sample draws ONE smear triple and applies it to the whole cell instead of a fresh value per layer. It is otherwise an ordinary sampled dimension, stratified and shuffled like the rest. Leaving its Min/Max blank holds it, and the run is then exactly what it would have been before smear was rangeable.

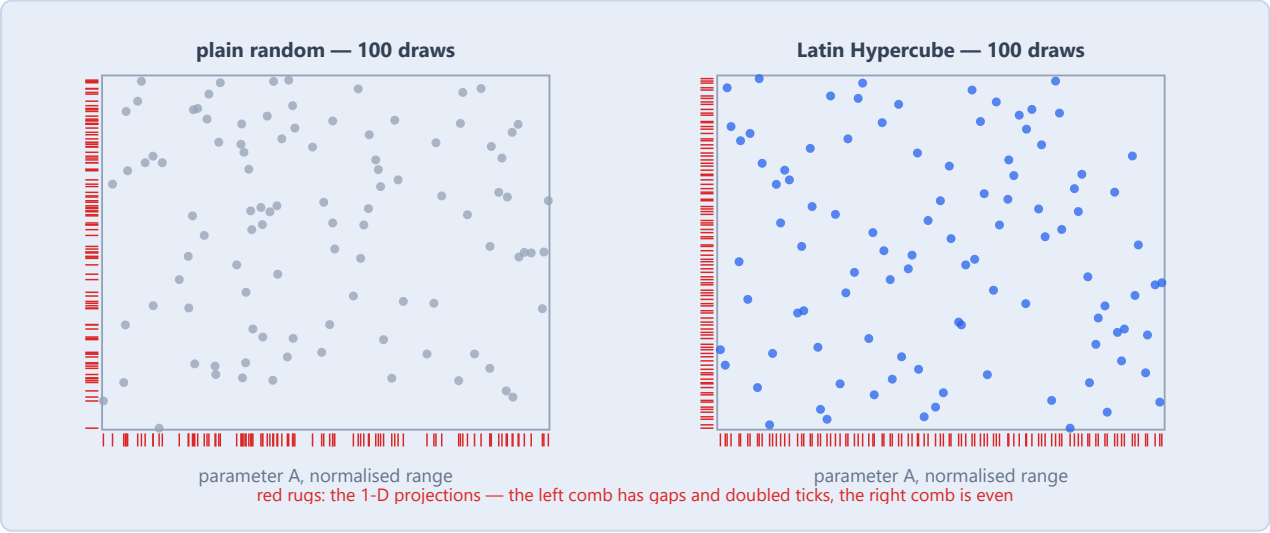


Figure 17.1. The same 100 draws by plain random sampling (left) and Latin Hypercube (right). The clouds look similar; the rugs differ: random sampling leaves per-axis gaps and clusters, while the Hypercube comb is even by construction, so percentile estimates need far fewer of the ≈32-run design searches each sample costs. The guarantee is per-axis (marginal), not joint — §17.6.

Figure 17.2 is this part's own draw, checked the same way before any solving began.

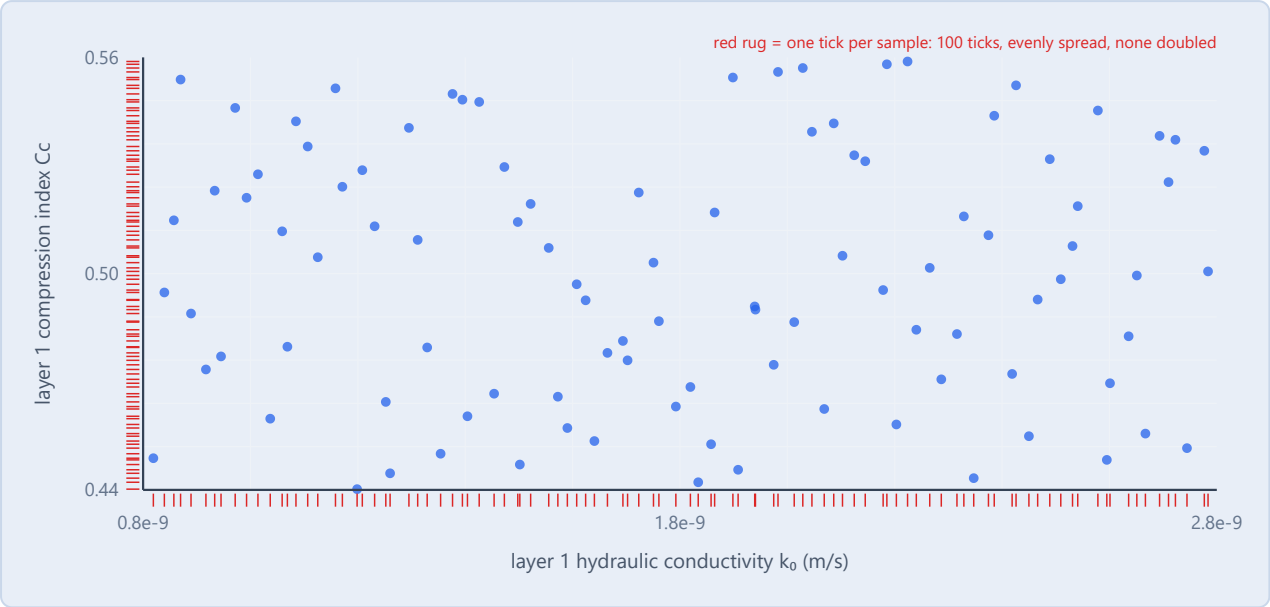


Figure 17.2. The actual 100 samples of this part, on two of their 16 dimensions. Read the red rugs: 100 evenly spread ticks per axis, every stratum used exactly once; the cloud is uncorrelated because the strata are shuffled independently. Checked before any solving: worst stratum departure 0.00499, inside the 1/(2N) bound of 0.00500.

> **WATCH OUT**
>
> **Watch out — ranges are inputs, not decoration.** The width you type is the width you get out. A range copied from a textbook because it looked reasonable produces a band that means nothing. Base each range on your own data: the scatter of your oedometer results, the spread between boreholes, the difference between lab and in-situ measurement.

17.3 The key idea — a whole search per sample

The sampling of §17.2 is one difference from a conventional Monte-Carlo pass; what happens to each sample is the other, and it is the bigger one. The mode does not sample the soil, compute 100 settlement curves and report a spread of settlements. It samples the soil and then, for each sample, **runs the entire Design bisection** — the same search described in §16.1, converged to the same tolerance. Figure 17.3 draws the pipeline.

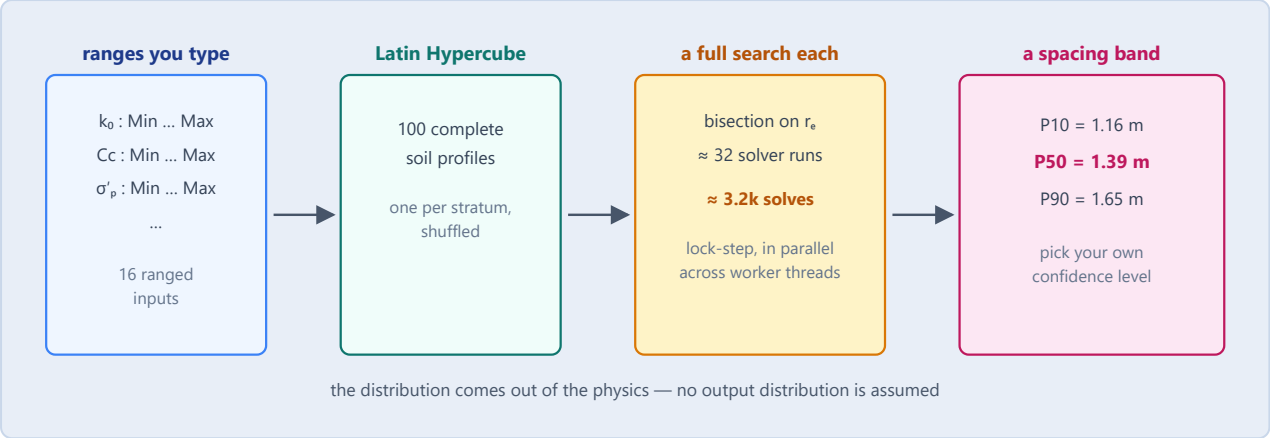


Figure 17.3. The probabilistic Design pipeline. The expensive step is the third — one converged bisection per sample — making the run roughly 100× a deterministic Design, which is why it is spread across worker threads.

Figure 17.4. The result page is a menu, not a report: each card computes its numbers when opened. Work left to right along the top row for the headline answer, then open whichever card the question of the moment needs.

The run finishes on the card menu of Figure 17.4; the spacing band is behind the first card, which also prints the spacing on its tile.

17.4 Reading the spacing band

The output is a distribution of *required* spacings. A wide spacing is a demanding requirement met easily; a tight one is a soil that needs help. So the conservative choice is the **low** percentile — design to P10 and 90 % of the sampled profiles would have been satisfied by a spacing at least that tight. Figure 17.5 is the resulting distribution; Table 17.1 tabulates the choices.

Figure 17.5. The 100 required spacings: 0.99 to 1.86 m, a factor of 1.9, from individually modest input ranges — the honest cost of not knowing the soil, invisible in a deterministic run.

Table 17.1. The spacing band, and what each choice means. The deterministic mid-range answer is 1.44 m.

Choice	Square spacing	Drains per hectare	Interpretation
P10 (conservative)	1.16 m	7406	90 % of sampled profiles reach the target
P50 (central)	1.39 m	5176	50 % of sampled profiles reach the target
P90 (optimistic)	1.65 m	3686	only 10 % of profiles reach the target
Deterministic mid-range	1.44 m	4823	**60 % of profiles would need it tighter**

> **WATCH OUT**
>
> **The single most useful line in this part.** On this example the deterministic answer of 1.44 m fails for 60 % of the sampled profiles. Mid-range inputs do not give a mid-range answer, because the search is non-linear — and they certainly do not give a safe one. Moving from 1.44 m to the P10 spacing of 1.16 m costs 2584 extra drains per hectare (+54 %). That is the price of the confidence, stated explicitly instead of hidden in a factor of safety.

17.5 Doing it in parallel, without changing the answer

Roughly 3.2 thousand solver runs would make the probabilistic Design unusably slow on one thread, so the samples are distributed across a pool of web workers: one fewer than your machine's core count, capped at 16 on a desktop and 4 on a phone, and never fewer than 2. The workers run the numerical engine *extracted verbatim from the same source file* as the main thread, so their arithmetic is identical by construction rather than by hope.

The subtlety is that Design is a search, not a single solve: a sample's next trial spacing depends on the result of its previous one. The tool handles this with a **lock-step bisection** — every round dispatches one trial for every still-active sample as a single batch, then advances each sample's search by one step. This reproduces the serial search exactly, step for step, and this has been verified: the difference

in r_e between the parallel and serial paths is zero on every branch. One caveat travels with any floating-point computation: across *different machines*, arithmetic ordering can vary the last significant figures, so pin the inputs and the solver grid when auditing a run rather than expecting bit-for-bit identity between computers.

> **WATCH OUT**
>
> **Watch out — keep the tab in front.** The lock-step search needs about 30 main-thread round trips. Browsers throttle background tabs aggressively, so a probabilistic Design left running behind another window can take many times longer than the same run in a focused tab. Simulate does not have this problem: its samples are independent and go out in one batch.

17.6 What the band does, and does not, tell you

It is worth stating plainly what kind of statement the band is, because it is easy to read more into it than it contains.

The usual way to handle an uncertain soil is to design once at your best-estimate parameters and then apply a factor. This tool does something different: it designs the works **100 times over**, each time for a different but internally consistent soil, and shows you the spread of answers. Nothing is linearised to make that affordable, no distribution is assumed for the output, and each sample gets the same converged search the deterministic answer gets. The shape of the result is whatever the physics produces.

Two things make that more than a presentational flourish. The sampling is genuinely even — every parameter's range is covered, verified to within the theoretical bound for a jittered design (§17.2). And it exposes a blind spot the deterministic run cannot: mid-range inputs gave 1.44 m, yet **60 % of the sampled profiles needed a tighter spacing than that**. Averaging your inputs does not give you an average answer, because the search is not linear.

Does the band span the true extremes? No, and it is not meant to. The Latin Hypercube stratifies each axis, so the extreme stratum of every parameter is sampled. But the *joint* worst case, with every parameter simultaneously at its unfavourable end, is vanishingly unlikely to be drawn in 100 samples, and the band does not report it. The band is a distribution over plausible profiles, not an envelope of possible ones. This was measured rather than argued. Each parameter was set to its unfavourable end, as found by the one-at-a-time sweep of Chapter 18, and one deterministic design was run at that joint corner. The required spacing there is **0.87 m**: tighter than the P10 of 1.16 m, and tighter than the tightest of all 100 sampled profiles (0.99 m). The favourable corner gives 2.06 m at the other end. If your design must survive the corner rather than the plausible range, run the corner deterministically as a separate check, exactly as was done here; the band will not do it for you.

WATCH OUT

The limit, stated honestly: the band is only as wide as your imagination. It answers one question: *if the truth lies inside the ranges I typed, how much does the answer move?* It does not cover the model being an idealisation (one drain standing for a whole field; the constitutive law itself), construction departing from the assumed programme, or any parameter you did not think to give a range. Those live in Chapter 20, and no amount of sampling will surface them.

KEY IDEA

So P10 is not a 10 % chance of failure. It means: of the soil profiles *you judged plausible*, 90 % would have been satisfied by a spacing at least that tight. That is a far more useful and far more defensible statement than an unexplained factor of safety — but it is a statement about your assumptions, not about the ground. Treat the band as a well-made map of what you told the tool, and keep engineering judgement for what you did not.

CHAPTER 18

Sensitivity — what actually moves the answer

18.1 The card, and what it sweeps

Chapter 17 gave you a band. This part answers the question that follows immediately: *which input is that band made of?* Open the **Sensitivity** card on the results page and press **Compute sensitivity**. Nothing is solved until you ask, so a run that never opens the card costs exactly what it always did. Table 18.1 says what the card sweeps and what it reports.

Table 18.1. What the card sweeps, and what it reports.

	Design	Simulate
Answer swept	required square spacing (m)	t_{90} — time to reach 90 % (days)
Cost per bar	two full bisections (≈ 64 solves)	two single solves
Soil parameters	both layers move together — each to its own Min, then its own Max	
Smear	one global triple, as in a probabilistic run — see §18.4	
Self-check	deterministic runs only: the unswept profile must reproduce the headline spacing, or the card says so (skipped when the headline is a probabilistic band)	none — the base bar is computed but there is no single headline to compare it against

KEY IDEA

Why t_{90} for Simulate. Design asks "what spacing reaches 90 % by time t"; Simulate asks "when do I reach 90 % at this spacing". They are the same question from opposite ends, so t_{90} is the directly comparable quantity. Sweeping U_s at the end of the run instead would be useless — it saturates near 100 % for almost every variant and every bar would collapse to nothing.

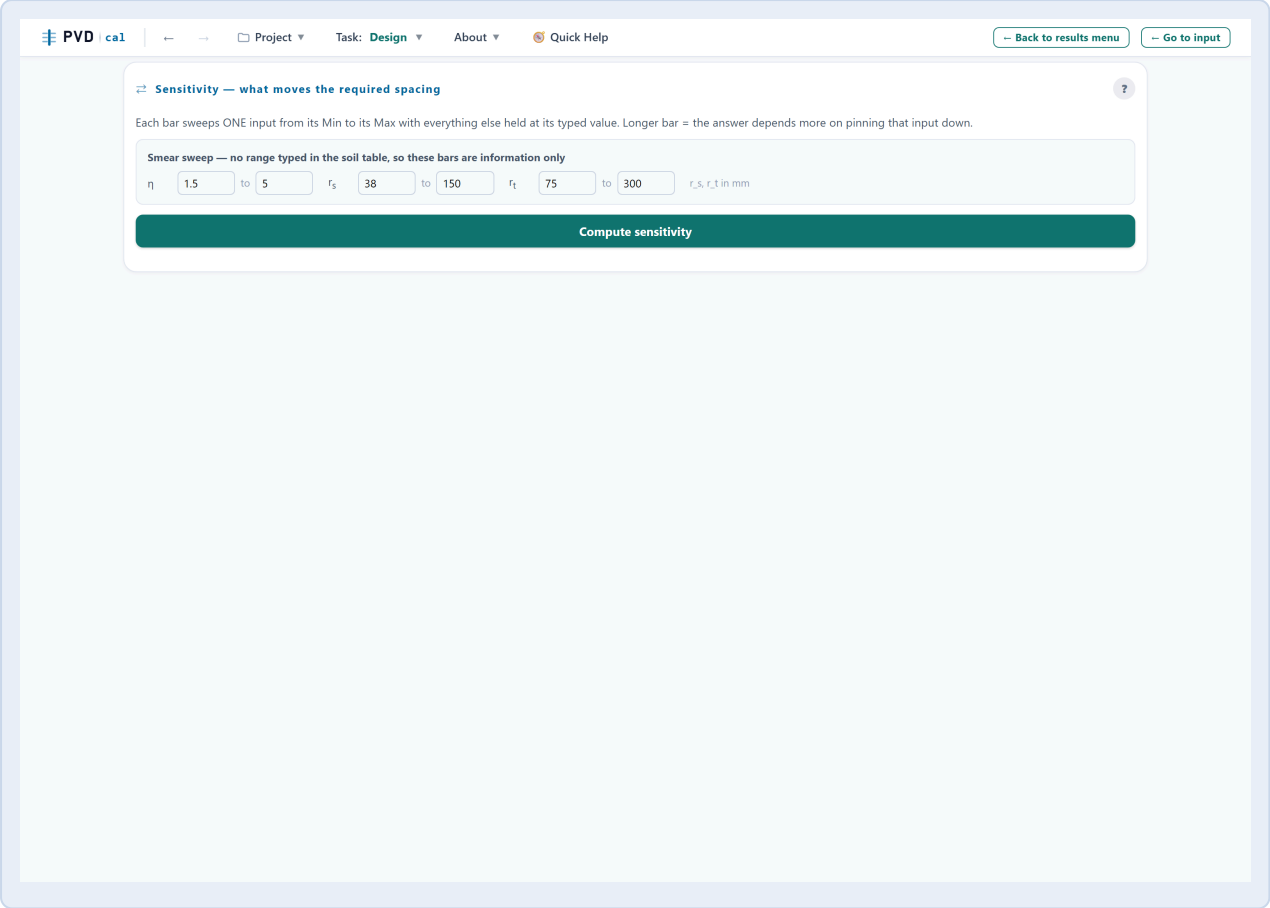


Figure 18.1. The sensitivity card on the example project: each bar sweeps one input Min → Max with the rest at base; bar length = movement of the required spacing; dashed line = the unswept answer; sorted so the input that matters most — the next laboratory test worth paying for — is on top.

Figure 18.1 shows the finished card on the example project.

18.2 Reading the ranking

Figure 18.2 is the ranking on the worked example; Table 18.2 gives the same ranking as numbers.

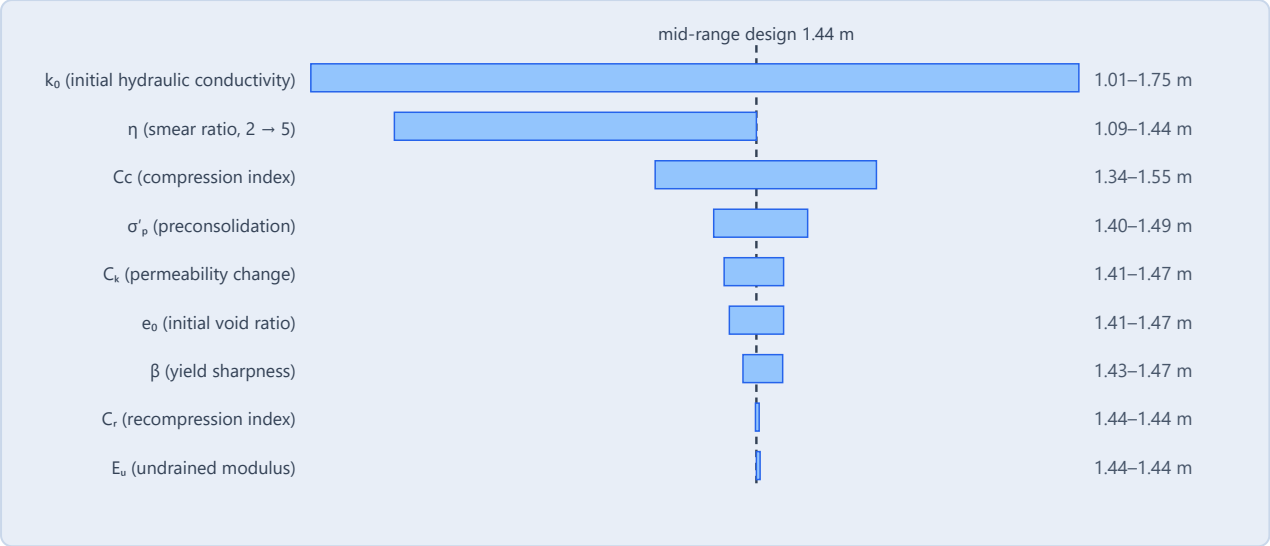


Figure 18.2. The Design ranking on the worked example: hydraulic conductivity dominates, smear second — ahead of every compression parameter — and E_u is exactly zero wide (immediate settlement sits outside U_s, §14.4). Simulate's t_{90} ranking is identical.

Table 18.2. The same ranking as numbers. Design spans are metres of required spacing; Simulate spans are days to reach 90 %. Both were produced by the card in the app.

#	Input	Design span	Simulate span
1	k_0	**0.736 m**	372 d
2	η (smear)	**0.347 m**	237 d
3	C_c	0.212 m	77 d
4	r_s, r_t (smear)	≈ 0.11 m	44 d
5	σ'_p	0.090 m	33 d
6	C_k	0.057 m	21 d
7	e_0	0.052 m	19 d
8	β	0.038 m	14 d
9	C_r	0.003 m	1 d
10	E_u	**0.000 m**	0 d

KEY IDEA

What to do with it. Spend the investigation budget where the bar is long. Another oedometer narrows C_c by a fraction of 0.21 m; an in-situ permeability test, or a well-instrumented trial embankment, narrows k_0 — worth roughly 3.5 times as much. And the parameters at the bottom can be held at a sensible value without apology: C_r moves the answer by 3 mm.

18.3 How the card computes the bars

The method is deliberately plain, because a sensitivity chart is only worth having if you can say exactly what it did.

1. **Start from the run's own profile** — the resolved soil the deterministic answer uses — and solve it untouched. That is the *base*, the dashed line on the chart.
2. **For each input in turn**, build two variants: everything at its typed value except that one input, which goes to its Min, then to its Max. A soil parameter moves in *every* layer at once, each layer to its own Min and its own Max.
3. **Solve every variant with the solver the run used** — in Design a complete converged bisection per variant, not an approximation; in Simulate one forward solve.
4. **Sort by span** — |answer at Max − answer at Min| — and draw.

KEY IDEA

It is the same machinery as the probabilistic run. A tornado is a probabilistic Design with deliberately chosen samples instead of random ones, so the card reuses the identical task builder, worker pool and lock-step bisection. There is no second solver, no linearisation and no finite-difference derivative — every bar is a real converged answer. That is also why it costs roughly two full bisections per bar, and why it computes only when you ask.

WATCH OUT

The card checks itself — on a deterministic Design run. There the untouched base variant must reproduce the headline spacing on the result page; if it does not, the card says so at the top rather than presenting authoritative-looking bars. On a probabilistic run the headline is a band, and for Simulate there is no single target, so the check is skipped. (A mismatch there would mean the sweep is measuring something other than the run.)

Two conventions are worth knowing. A **blue** bar means that input carries a Min/Max, so the range being swept is the very same range driving P10/P50/P90 — bar and band agree. An **amber** bar is always a smear parameter (η, r_s or r_t): smear is the one input swept without a typed range, using an editable default, so its influence is shown even though it is not in the band; type a range and the bar turns blue. A soil parameter with no usable range is simply skipped. And an input whose range is collapsed produces **no bar at all** rather than a zero-width one, because "you never gave me a range" and "this does not matter" are different statements.

One more bar appears on the card itself. After the ranked bars are computed, the card assembles the **joint adverse corner**: every ranged input pushed to its unfavourable end at once, solved as one more complete converged run and drawn as a single violet bar at the foot of the chart. The bars above it move one input at a time, so they cannot show how the ranges combine; the violet bar shows exactly that, and because the inputs interact it is generally not the sum of the bars above it. In Design the

unfavourable end of each range is the one that tightens the spacing; in Simulate, the one that lengthens t_{90}. The printed figure in this chapter shows the ranked bars only — the corner bar is computed live on the card.

18.4 Smear: why it is ranged, and the case it captures

Smear sits second in that ranking, above every compression parameter. Unlike the soil properties it is a property of the **installation**: one rig, one mandrel, one crew. So whatever η turns out to be, it is very nearly the same η across the whole platform. A probabilistic run therefore draws **one smear triple per sample and applies it to the entire cell**, rather than a fresh value per layer: each sample is one complete candidate site, with one installation.

The reason it must be *ranged* rather than merely typed is the combination it unlocks.

> KEY IDEA
>
> **The governing case is low k_0 together with high η.** Both act on the conductivity near the drain, so they compound — a k_0 at the bottom of its range *and* a badly smeared installation is far worse than either alone. With η pinned at one value that corner is never sampled. The low end of the band, the end you actually design to, is then capped by the k_0 range on its own.

Table 18.3 makes the corner measurable: the same hypercube, re-solved at three fixed smear levels.

Table 18.3. The same 60-sample hypercube — identical soil draws — re-solved at three fixed smear levels. Only η differs between the rows.

η	P10 spacing	P50	P90	Drains per hectare at P10
2	**1.183 m**	1.378	1.622	7145
3.5	**1.003 m**	1.173	1.388	9940
5	**0.889 m**	1.043	1.237	12653

> WATCH OUT
>
> **The size of what pinning η was hiding.** Between η = 2 and η = 5 the conservative P10 spacing falls from 1.183 m to 0.889 m, which is 25 % tighter. The drain count rises from 7145 to 12653 per hectare, an increase of 77 %. A design that held η at the optimistic end understated the drain requirement by that much.

18.5 The one caution: do not count the same doubt twice

η and k_0 are near-substitutes as far as this calculation is concerned, because both act on the same physical thing — the conductivity of the soil closest to the drain. The exchange rate is measurable: Figure 18.3 draws it, and Table 18.4 gives it in full.

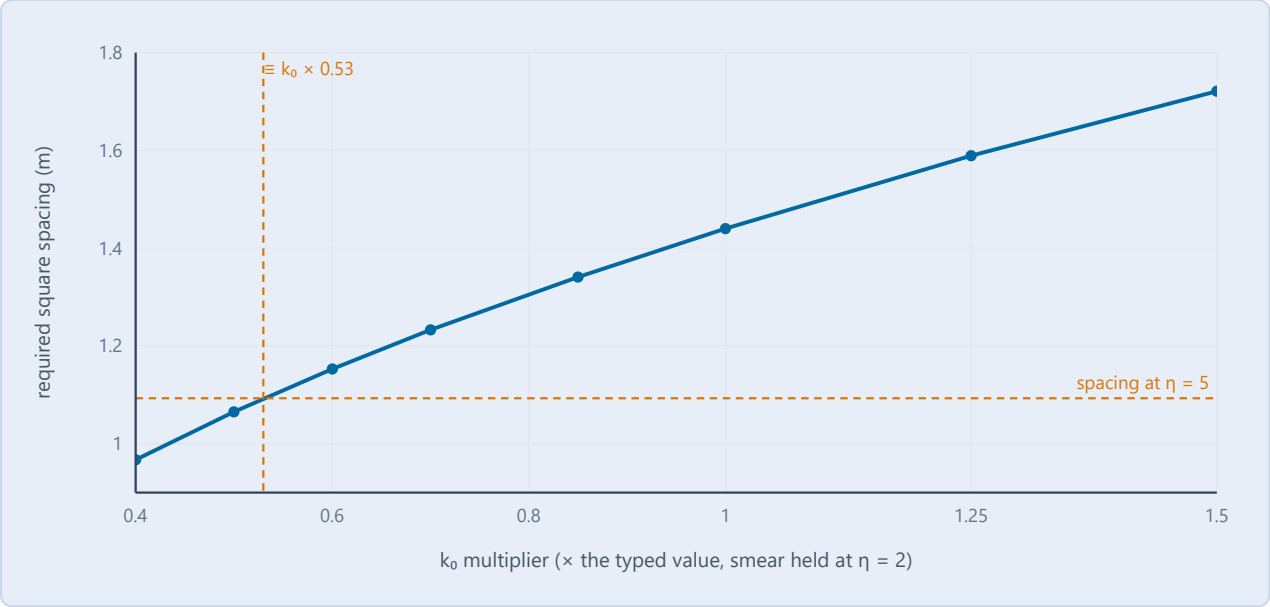


Figure 18.3. The smear–conductivity exchange rate: required spacing as k_0 alone is scaled (blue), against η raised 2 → 5 with k_0 untouched (amber). They meet at ×0.53 — η 2 → 5 moves the answer as much as halving k_0.

Table 18.4. The exchange rate in full, measured on this solver.

Change in smear	Spacing	Equivalent change in k_0
η : 2 → 3	1.287 m	**k_0 × 0.77** (−23 %)
η : 2 → 4	1.177 m	**k_0 × 0.63** (−37 %)
η : 2 → 5	1.093 m	**k_0 × 0.53** (−47 %)

WATCH OUT

Where the two ranges would overlap. Suppose your k_0 came from *back-analysis* of PVD performance, from this tool's own Back-calculation task or from an instrumented trial. Then it is an *effective* conductivity that already contains the installation damage, because that is what the field data saw. Adding an independent smear range on top of such a k_0 counts the same physics twice: the band widens, P10 tightens, and you buy drains to cover an uncertainty you had already accounted for.

KEY IDEA

At design stage that overlap usually does not exist. A k_0 from oedometer permeability, a CPTu dissipation test or a published correlation describes *intact* soil that no mandrel has been through. Smear is then a genuinely separate effect sitting on top of it, and ranging both is correct rather than duplicative. The rule of thumb: **range smear when your k_0 is a laboratory or in-situ value; fold it into k_0 instead when your k_0 was fitted to PVD field performance.**

RECAP

Why Back-calculation still holds smear. The same near-substitution that makes η and k_0 compound in a forward design makes them *inseparable* in an inversion. Field settlement cannot tell a low k_0 from a bad smear, so a search that moves both wanders along a valley of equally good fits. Degeneracy defeats identification, not propagation — which is why the forward tasks sample smear and the inverse task holds it. The two choices are consistent, not contradictory.

CHAPTER 19

Reading your results

Table 19.1 lists each reported quantity, what it means, and the misreading it invites.

Table 19.1. What each reported quantity means, and the mistake each one invites.

Reported	Means	Common misreading
$U_s(t)$	fraction of *primary* settlement completed	"settlement has stopped" — creep continues
$U_p(t)$	average dissipation of excess pore pressure	expecting it to equal U_s — it does not, because c_h varies with stress
Settlement (mm)	primary + creep + immediate	comparing it against a U_s-based hand calculation
r_e	radius of influence the search converged on	using it as a spacing
Square / triangular spacing	the set-out dimension	mixing the two patterns between design and construction
P10 / P50 / P90 spacing	percentiles of the *required* spacing	taking P90 as conservative — it is the opposite
σ′(z) and e–log σ′ plots	the stress path each layer actually followed	ignoring them — they show whether layers yielded
Mesh card	the r–z element grid used	—
A faint dashed curve	the previous option of that chart's toggle, kept so you can see what changed	reading it as a separate result — it is the same quantity, just before your last click

The σ′(z) and e–log σ′ cards deserve a specific look. They show, per layer, where the stress started, where it crossed σ'_p, and where it finished. If a layer never crosses σ'_p, its settlement is governed by C_r and the answer is relatively insensitive to C_c; if it crosses early, the opposite. This single check explains most surprises in the tornado chart.

CHAPTER 20

Caveats and good practice

Table 20.1 collects the assumptions a design report that used this tool should state.

Table 20.1. Assumptions worth stating in a design report that used this tool.

Assumption	When it bites	What to do
One drain represents the whole field	edges, corners, irregular patterns	treat the prediction as the interior of a large area
Smear η, r_s, r_t are guesses	always — they are rarely measured	state them; the effect is real but weakly identifiable
c_v = 0 unless typed	thin deposits, or drains that stop short	enter c_v; the tool then solves coupled r–z
Drain remains fully efficient	long drains, high confining stress, kinking	reduce k_0 or shorten the drain deliberately
Load is applied as scheduled	construction always slips	re-run with the real programme once known
Input ranges are yours	ranges copied from a textbook	derive them from your own scatter
Creep ratio $C\alpha/C_c$ = 0.04 by default	highly organic soils	set it from your own long-term oedometer data

KEY IDEA

A workflow that works. (1) Run **Design** deterministically to get a feel for the scale. (2) Run it **probabilistically** and pick a percentile you can defend — P10 for a critical structure, nearer P50 where you can extend the programme if needed. (3) Run **Simulate** at the chosen spacing to get the settlement history the instrumentation should follow. (4) Once field data arrives, take it to **Back-calculation & Forecast** and let the measurements narrow the ranges that Figure 18.2 says matter. Then repeat step 2 with the narrowed ranges. The uncertainty band should shrink between the tender and the first berm.

All figures were produced by the app's own solver on the soil its built-in Design & Simulate example loads. Load the example, press Calculate, and the numbers in Chapter 15 and Chapter 16 appear on screen.

PART THREE

Back-Analysis and Forecasting

Turning field monitoring into calibrated parameters, and calibrated parameters into a forecast you can defend — with an honest account of how far to trust it.

What you will be able to do after working through this part

1. Explain why back-analysing early settlement data is hard — truncation, the see-saw, look-alike answers.
2. Sort the soil parameters into the *speed* family (k_0, C_k, and the smear parameters η, r_s, r_t) and the *magnitude* family (C_c, C_r, σ'_p, β, $C\alpha/C_c$) — and say which the data can pin down, and when.
3. Describe every step of the pipeline, what it receives and hands on, and the failure it protects you from.
4. Explain the three knobs — especially why the search must move the speed together with the magnitudes.
5. Read the result page professionally: the score chart, the heat-map, the fit statistics, the speed card, the band.
6. Recognise the warning signs (pinned winners, rescue banners, dual tracks) and respond correctly.
7. Cross-check a run against the Asaoka construction and interpret disagreement.

CHAPTER 21

Understanding the problem

21.0 The problem — an embankment on soft clay

This opening section is numbered .0 deliberately: it is a preamble that sets up the problem before the numbered argument begins. Readers of Parts One and Two have met consolidation and the unit cell already and may skim to §21.1; the ground is re-laid here so that this Part also stands on its own.

This chapter assumes no prior knowledge. It builds the story in the order a newcomer needs it: first the physical problem, then why it is genuinely hard, then the simple loop and score that every method shares, then the two classical method families in the literature, and finally the method this Part uses and the rules for using it well.

The situation. An embankment is being built on soft clay. Soft clay behaves like a saturated sponge. Its pores are full of water, and when a load is placed on top the ground cannot compress until that water has been squeezed out. In clay the pores are so small that the water escapes very slowly, so the ground keeps settling for months or years after construction. This slow, delayed compression is called *consolidation*. To speed it up, engineers install prefabricated vertical drains (PVDs): plastic strips about 100 mm wide, pushed vertically into the ground on a close grid. The escaping water then travels only a short horizontal distance to the nearest drain, instead of a long vertical distance to a sand layer, and consolidation that would have taken decades happens in months.

Figure 21.1 draws the whole arrangement in one cross-section: the load, the sponge-like clay, the drains giving the water its short way out, and the two kinds of instrument that watch the process. Keep this picture in mind through the whole Part; every method discussed here is an attempt to learn the soil's parameters from what those instruments record.

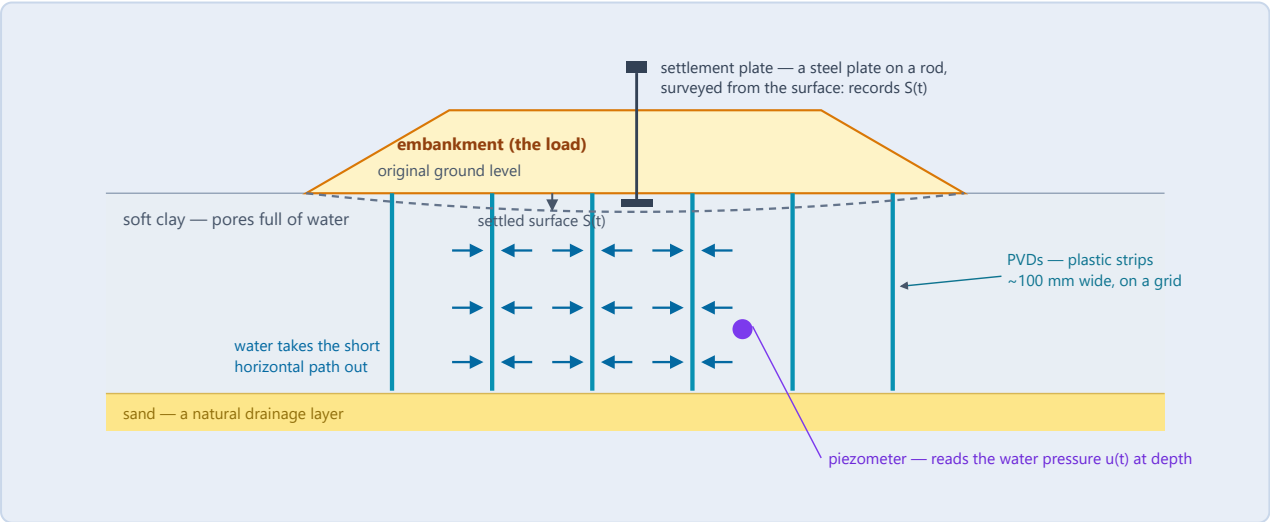


Figure 21.1. The physical situation: an embankment on soft saturated clay, PVDs (cyan) giving the pore water a short horizontal way out (arrows), settlement plates above and piezometers below watching. (Schematic, not to scale.)

The problem. Before construction, the settlement is predicted using soil parameters measured on small samples in the laboratory. Those values are uncertain, because a fist-sized sample cannot fully represent thousands of cubic metres of variable ground. So the real ground never does exactly what the prediction said. To find out what it is actually doing, the site is instrumented. Settlement plates, which are steel plates with rods reaching the surface, are surveyed regularly and record how far the ground has sunk at several depths. Piezometers record the water pressure inside the clay. The engineering question of this whole Part is: *can we use those readings to correct our soil parameters, and then trust the corrected model to forecast the rest?*

That job has a name: **back-analysis**. Think of the computer model as a machine with unknown dial settings, where the dials are the soil parameters. Back-analysis means finding the dial settings that make the machine reproduce what the instruments measured, and then leaving the dials there to predict the future. The answer matters commercially as well as technically: it decides whether the contractor can remove the temporary surcharge and hand over the site, must wait longer, or must add more load. The method this tool uses for that job is called **Sweep Search**, and this whole part is about it. The name will appear from here on, so it is worth fixing now what it means: a method that tests soil recipes in a planned sweep across the range your laboratory data allow, rather than hunting one best answer, and that reports the whole set the record cannot reject. Where it came from, and why it is built the way it is, are the subject of §21.3. The rest of this chapter builds up to it: §21.1 shows why the job is hard, §21.2 shows the loop and the score that every method shares, §21.3 surveys the two classical traditions in the literature, and §21.4 assembles Sweep Search from them.

21.1 Why early back-analysis is hard

The opening section introduced consolidation with the sponge picture (§21.0). One more level of detail is needed here, because it explains what the instruments actually see. When the embankment load is first applied, the pore water carries almost all of it, and the water pressure in the clay jumps up. The extra pressure is called *excess pore pressure*, and it is what pushes the water out towards the drains. As the water escapes, the pressure falls, the load transfers onto the soil skeleton, and the skeleton compresses. Settlement rising at the surface and pore pressure falling at depth are therefore two views of one process. That is why a well-instrumented site carries both settlement plates and piezometers, and why the method scores both.

Figure 21.2 shows the same mechanism as the classical piston-and-spring picture that most textbooks use to introduce consolidation. If the sponge image of §21.0 felt too loose, this one is exact: spring = soil skeleton, water = pore water, valve = drainage path, gauge = piezometer, and the drains of this Part are simply a wider valve.

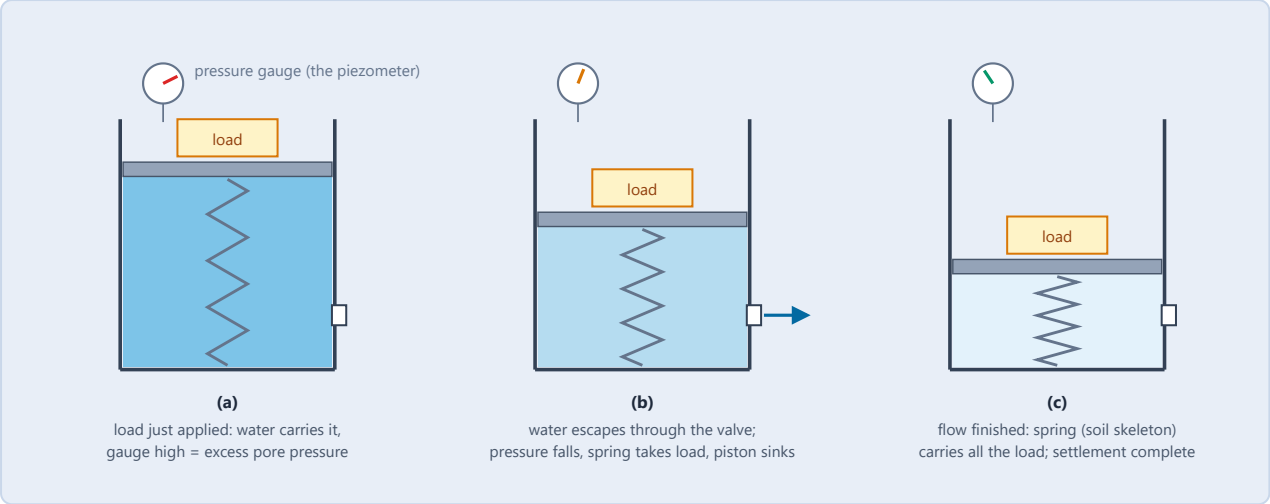


Figure 21.2. Terzaghi's piston-and-spring analogy: spring = soil skeleton, water = pore water, valve = drainage path, gauge = piezometer. (a) Load applied — the water carries it, the gauge reads high. (b) Water escapes, load transfers, the piston settles. (c) Flow stops — the spring carries all. A PVD widens the valve. (Schematic.)

The prediction is most uncertain on the rate side. The field drainage rate is routinely 2–10× different from the laboratory estimate. Part of the reason is scale: a small sample misses the thin sand seams and fabric that let a real deposit drain. Part of it is installation: pushing the drain in disturbs and remoulds an annulus of clay around it, called the *smear zone*, which slows drainage in exactly the region that matters most. So we monitor, and then we back-analyse, exactly as defined in §21.0: work backwards from the readings to the parameters that actually govern the site, so the forecast can be corrected while there is still time to act. Four things make that job genuinely hard:

- **The record is cut short (truncation).** Decisions are needed when perhaps half the settlement has happened. Early data tell you a lot about how *fast* things are going and much less about how *far* they will go. (Compare: watching the first half of a race tells you the runners' pace, not where the finish line is.)
- **The see-saw (compensation).** A softer soil draining slowly and a stiffer soil draining quickly produce almost the same early curve (Figure 21.3); they separate only later, after your record ends. Any method that fits speed and magnitude freely at once can land anywhere along this see-saw.
- **Look-alike answers (equifinality).** Because of the two points above, many different parameter sets fit the early record equally well. An honest method must say so, not silently pick one.
- **Practical limits.** The tool runs in a web browser. A run must finish in minutes, give the same answer every time (so it can be audited), and respect the ranges you typed — your lab data are knowledge, not decoration.

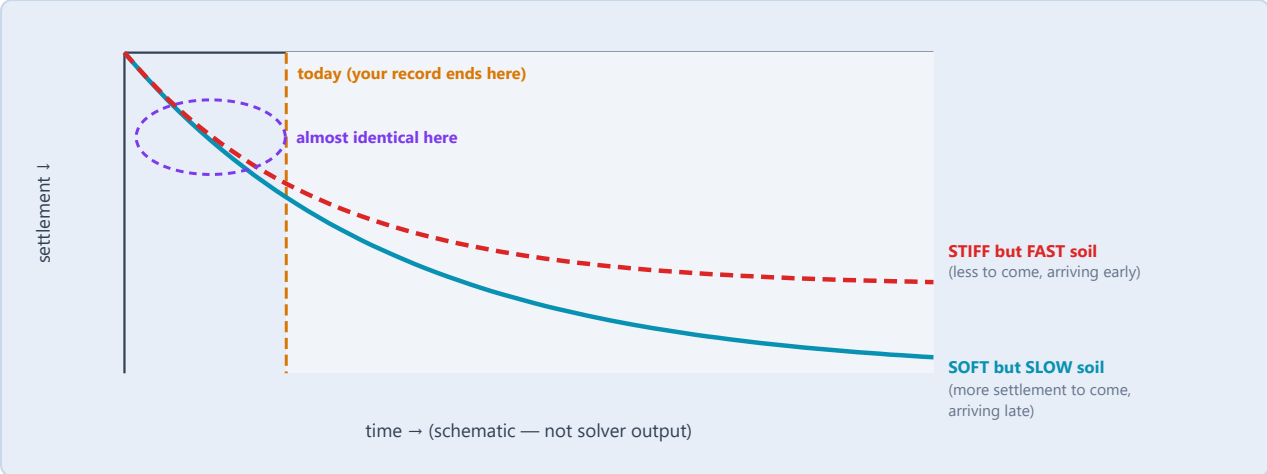


Figure 21.3. The see-saw: a soft-but-slow and a stiff-but-fast soil produce nearly the same settlement up to "today" — the record cannot tell them apart; only the future can. (Schematic.)

21.2 The back-analysis loop, and the one score

Two words carry this section, so define them first. A **forward model** runs from causes to effects. Give it a complete set of soil parameters, for example how easily water flows through each layer and how much each layer compresses under load, and it computes the settlement history those parameters would produce, day by day. It answers the question "if the soil were like this, what would we see?". An **inversion** runs the other way: it starts from the measured settlement history and works back to the parameters. Inversion is much harder than the forward run, for the same reason that it is easy to total a shopping basket but impossible to recover the individual prices from the total alone: many different combinations of inputs produce the same output. Sweep Search is an inversion wrapped around a forward model.

Every back-analysis method — those in this part, and those used in inversion generally — runs the same simple loop of four moves, whatever its sophistication. First, *propose* a candidate: one complete set of dial values, for example a drainage rate and a compressibility for every soil layer. Second, *simulate*: run the forward model with those values and let it draw the settlement history they would produce. Third, *score*: measure how far the simulated history sits from the actual readings. Fourth, use that score to *choose* the next candidate. Figure 21.4 draws the loop. Every method is a version of this one loop, and they differ only in the last move. *Random guessing* picks the next candidate at random, which needs no cleverness but wastes most of its runs. *Systematic sweeping* lays out a grid of candidates in advance and tries them all, which cannot get lost and cannot miss a second answer hiding elsewhere, but costs a run per grid point. *Gradient-following optimisation* measures which way the score improves and steps downhill, which is much faster when the score surface has a clear slope, but stalls when it is flat and can settle into the first dip it meets. Sweep Search uses the second and the third together, for reasons §21.1 has already supplied: on a short record the surface really is nearly flat.

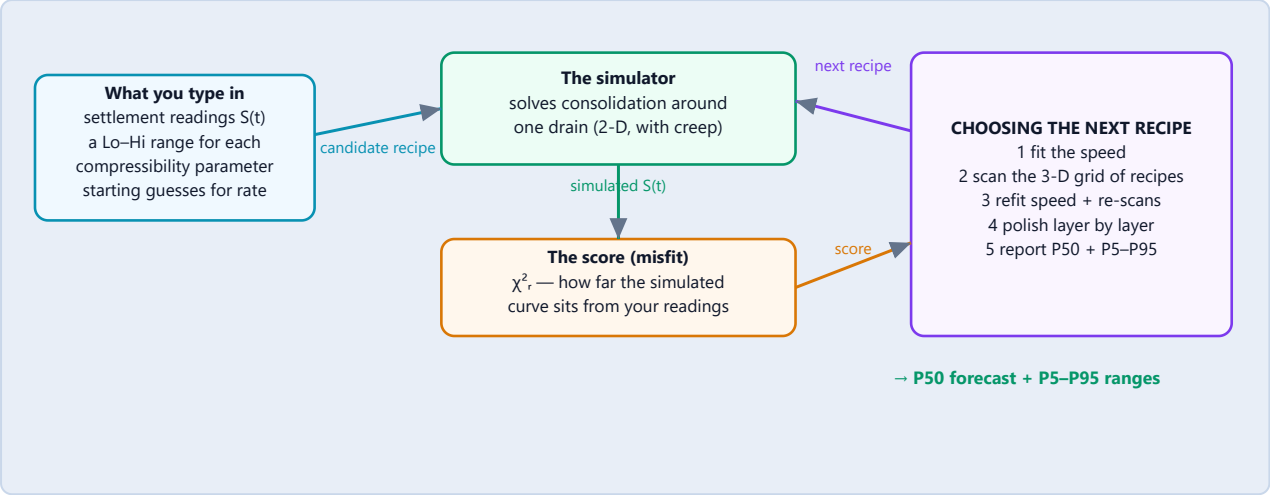


Figure 21.4. The back-analysis loop: propose a soil recipe, simulate its settlement history, score the misfit χ^2_r against the readings, choose the next recipe. (Schematic.)

KEY IDEA

the score. Everything is judged by one number: $\boldsymbol{\chi^2_r}$ ("reduced chi-squared"). In symbols, for readings S_{obs} at times t_i and a simulation S_{sim}:

$$\chi^2_r = (1/\nu)\ \Sigma_i\ [\ (\ S_{sim}(t_i) - S_{obs}(t_i)\ )\ /\ \sigma\ ]^2 \qquad (21.1)$$

It is the average squared gap between the simulated curve and your readings, measured in units of your stated measurement precision σ_s (the tool's default is 5 mm; this chapter's examples use 10 mm for easy arithmetic). $\chi^2_r \approx 1$ means the simulation misses by about one measurement error — as good as the data allow. Much larger means a poor fit; much smaller (on clean synthetic data) means the fit is tighter than the noise.

One bookkeeping definition, made once here and used everywhere a score or an interval appears in this book: the divisor is the degrees of freedom, **ν = (number of observations) − 3 × (number of layers), floored at $\nu \geq 1$**. The subtraction charges the fit for the parameters it is free to adjust — three magnitude parameters per layer — and the count is deliberately fixed rather than tracking how many coordinates a given step happens to search, so that scores from different steps stay comparable on one ruler. If a record is so short that the floor engages, the result card says so rather than dividing by a fiction.

A worked example makes the score concrete. Suppose three settlement readings, a stated measurement precision of σ = 10 mm, and a candidate whose simulation misses the three readings by 5, 10 and 20 mm. Each miss is divided by σ and squared: $(5/10)^2 + (10/10)^2 + (20/10)^2 = 0.25 + 1 + 4 = 5.25$. Divide by the number of readings to get $\chi^2_r = 5.25 / 3 = 1.75$ (this example divides by N for arithmetic clarity; the tool divides by the ν defined above, which matters when records are short). Read it as: this candidate misses by about 1.3 measurement errors on average ($\sqrt{1.75} \approx 1.3$). A rival candidate that scores 1.05 fits better; one that scores 6 is clearly worse; and none of the arithmetic needed anything more than the readings, the simulation and your honesty about σ.

Figure 21.5 draws that arithmetic. Notice two things in it. The error bars are the honesty: they say how finely the instrument can actually be read, and every miss is judged relative to them. And the squaring means one bad miss dominates: the 20 mm gap contributes 4 of the 5.25 on its own, so a candidate cannot buy forgiveness for one large error with several small ones.

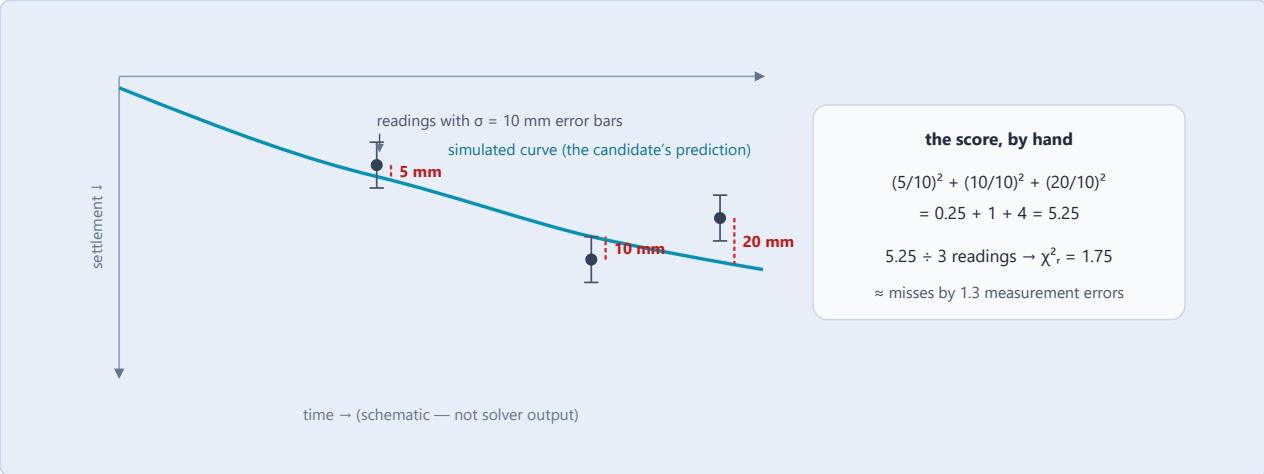


Figure 21.5. The worked score drawn: three readings, error bars at σ = 10 mm, misses of 5, 10 and 20 mm → χ^2_r = 1.75. (Schematic.)

The score does two jobs in Sweep Search. During the search it ranks candidates, so the sweep knows which recipes are worth refining. At the end it sets the weights: every admissible candidate contributes to the forecast band that §21.3 introduces, weighted by its score, so a poorly-fitting recipe fades out of the band instead of being cut by a hand-picked threshold. Both jobs depend on σ being honest. Claim your plates are more precise than they really are, and the score punishes every candidate for noise it could never fit; too few survive, and the band comes out too narrow and too confident. The caveat below deals with exactly this.

> **WATCH OUT**
>
> **Watch out — the auto-σ caveat.** The tool offers *auto-scale σ*, which rescales the assumed measurement precision until the best fit reads $\chi^2_r \approx 1$ *by construction*. That is a unit choice, not an achievement — so when auto-σ is on, judge the fit by RMSE and NRMSE (which are never rescaled). RMSE is the root-mean-square error, √(mean of the squared misses), in millimetres — the typical size of a miss, in the units you measured in. NRMSE is that same number expressed as a percentage of the largest reading on that plate, which makes plates of different sizes comparable and read the revised-σ banner. Figure 22.3 shows where σ is declared, and where the per-type balance of §21.3 can be switched off. The pipeline's internal rule is that every step optimises and reports the same metric on the same solver grid, so the score chart never advertises a "best" the delivered answer cannot reproduce.

One more honesty setting lives beside σ in the same dialogue: the **error drift** ρ, one value for settlement plates and one for probes, default 0. The score's arithmetic above treats every reading's error as independent of its neighbours'. Real records often violate that: a settling benchmark, a re-referenced survey, an instrument that drifts — each makes consecutive errors share a component, so thirty readings do not carry thirty readings' worth of information. Typing a drift ρ > 0 corrects both halves of the score at once. Each residual is whitened against its predecessor — $r' = (r - \rho \cdot r_{prev})/$

$\sqrt{(1 - \rho^2)}$, the generalised-least-squares rule for first-order autocorrelated error — and the information count feeding ν becomes $n(1 - \rho)/(1 + \rho)$ effective readings per instrument family; at $\rho = 0.7$, thirty readings count as about five. At $\rho = 0$ both reduce exactly to the classical score. You do not have to guess the value: after every run the forecast card prints the measured lag-1 drift of the winner's residuals, and a measured drift well above the ρ you used is the card telling you the intervals are optimistic — type the measured value in and re-run. The setting is not decorative — its effect is measured: re-run on the campaign's 60 correlated-noise sites with the drift typed at its true value, the record-only χ^2 region recovered from 42 % coverage to 83 %, at the honest price of widening (§32.5).

21.3 Foundations — the two traditions behind the method

The problem, its difficulty and the loop are now in hand. What remains is the biggest question: how should the loop choose what to try next, and what should be reported at the end? The literature offers two classical answers, and the method in this Part is deliberately built from both. Learn the two first; the method then makes sense as a combination.

Tradition 1: fit one best curve. This is the same idea as fitting a least-squares line through laboratory data, scaled up: write an error function that measures the gap between prediction and measurement, then adjust the parameters until that error is as small as possible. In geomechanics this is the optimisation tradition surveyed by Gioda & Sakurai (1987); in soft-ground practice its graphical cousin is the observational procedure of Asaoka (1978), which Chapter 27 uses as a cross-check. The product is a single best-fit parameter set. On this problem, that product is also the weakness. Because of the speed–magnitude trade-off (§21.1, Chapter 22), many different parameter combinations reproduce a short settlement record equally well. This is the **equifinality** that §21.1 flagged: different causes, same observed effect. To see the danger, suppose two very different soils both fit your record to within the survey error, but their forecasts of the final settlement differ by half a metre. A best-fit method hands you one of those forecasts, chosen almost by accident, and never mentions the other. The single number looks precise; the precision is an illusion.

Tradition 2: keep every candidate the data cannot reject. Work like a detective rather than a marksman. Instead of hunting one answer, line up many candidate parameter sets, test each against the record, and eliminate only the ones the evidence clearly rules out. Everything that survives is kept, and the survivors *together* carry the forecast. To make this concrete, imagine testing 1000 candidate parameter sets. Suppose 950 of them predict settlement histories clearly different from what the plates measured; they are rejected. The 50 that remain all fit the record about as well as the measurement noise allows, so the data cannot choose between them. Run each of the 50 forward in time. If their forecasts of the final settlement range from 0.9 m to 1.3 m, then that *range* is the honest answer, because nothing in the data can narrow it further. The range is reported with percentiles: sort the 50 forecasts from smallest to largest; the middle value is called P50, and the values 5 % in from each end are called P5 and P95. Saying "P5 = 0.92 m, P50 = 1.10 m, P95 = 1.28 m" means half the surviving candidates forecast more than 1.10 m, and nine out of ten forecast between 0.92 and 1.28 m.

The two traditions are easiest to grasp side by side, and Figure 21.6 draws them on the same short record: the left panel shows the single best-fit curve and the equally good alternatives it hides, and the right panel shows the survivors kept together with their spread reported as a band. Figure 21.7 then walks the 1000-candidate example above step by step, from the full candidate cloud, through the rejection test, to the sorted forecasts that the percentiles are read from.

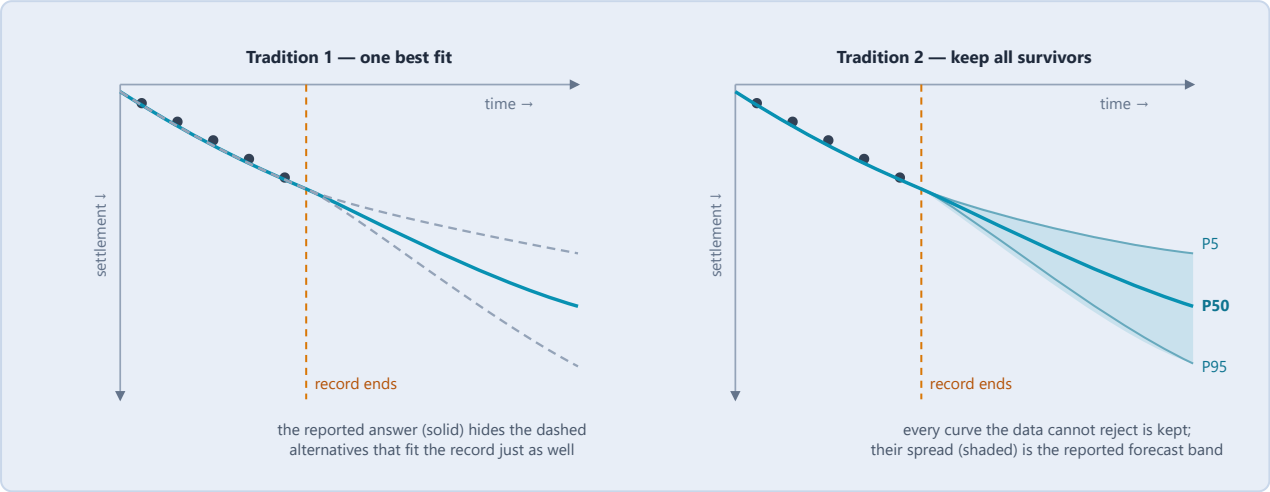


Figure 21.6. The two traditions on one short record (dots = readings; dashed orange = where the record ends; settlement grows downward). Left: one best-fit curve, its look-alikes never mentioned. Right: every unrejected curve kept, the spread reported as P5–P95. (Schematic.)

Figure 21.7. The 1000-candidate example: score all (left), reject the clear misses (middle — survivors green), sort the survivors' forecasts and read P5 / P50 / P95 off the column (right). (Schematic; dot counts reduced.)

This keep-the-survivors idea is *regionalised sensitivity analysis* (RSA; Spear & Hornberger 1980), extended into **GLUE** — Generalised Likelihood Uncertainty Estimation (Beven & Binley 1992), the most widely used member of this family in hydrology and, increasingly, in geotechnical back-analysis. GLUE calls the surviving sets *behavioural*, and reports their spread as a percentile band exactly as above. This Part's final band (§25.6) has that same shape. GLUE has two recorded weaknesses of its own, both documented in its literature (Beven & Binley 2014). First, it samples candidates blindly and uniformly, so most of its computer runs are spent on parameter combinations the physics already rules out. Second, its scoring rule is informal: the analyst chooses it, and the choice moves the answer. In the published GLUE literature that choice has been made in several incompatible ways for the same

kind of problem — the inverse of the sum of squared errors, the Nash–Sutcliffe efficiency, an exponential of the error, each raised to a power the analyst also selects, and each with a cut-off below which a candidate is discarded. Two competent analysts can therefore publish materially different bands from the same data and the same model, which is the criticism the method has never fully answered.

Sweep Search is a revision of GLUE that fixes those weaknesses using the structure of this specific problem. Four revisions, each answering one recorded criticism:

1. *Physics-bounded sampling instead of blind sampling.* The candidates come from a staged grid built on your laboratory ranges: the typed walls, the 3× expansion and the α-blend of Chapter 23. Every run is spent inside soil mechanics.
2. *A formal score instead of an informal one.* Candidates are judged by the σ-weighted χ^2 of §21.2, which is built from the instruments' own measurement precision, with an auto-σ calibration.
3. *No hand-picked acceptance cut-off at all.* GLUE keeps or rejects candidates by a threshold the analyst chooses. Sweep Search removes the binary choice: the typed walls decide what is admissible, every admissible recipe contributes to the band with a weight computed from its score, and the interval that sets the band's width uses a fixed statistical rule (the one-parameter 95 % likelihood-ratio threshold), the same for every analyst. Both mechanisms are set out step by step in Step 8 of Chapter 24: the weight each recipe carries is $\exp(-\tfrac{1}{2}\,\Delta\chi^2_r/T)$ with T a tempering factor that stops a single recipe monopolising the band, and the percentiles are weighted percentiles over all of them.
4. *Balance weights.* Each instrument family — settlement, pore pressure, local degree of consolidation — gets one vote, so forty settlement plates cannot silence six piezometers. Each family's contribution is rescaled by $(N/G)/n_g$, where N is the total number of readings, G the number of families present and n_g the readings in that family; the rescaling leaves the overall χ^2_r scale unchanged. Two consequences are worth one plain statement, because they connect across chapters: the rescaling also leaves ν unchanged by design, so the balance never alters the ruler; and the auto-σ convention drives the best fit to $\chi^2_{r,\min} = 1$, which is precisely the condition under which the multiplicative interval threshold of Step 4f coincides with the additive textbook likelihood-ratio rule. It is ON by default and can be switched to raw per-point weighting in the observation-accuracy settings, beside the σ values it belongs with.

Sweep Search also borrows Tradition 1, for a practical reason. A sweep can only report what it sampled: its answer is a grid point, and a grid fine enough to pin the speed of every layer directly would cost far more solves than a browser can afford. A line search has the opposite profile — it is precise and cheap, but it can settle into the wrong valley. So each is used where it is strong: the sweep decides *which* valley, the line search then finds the bottom of it. Between the sweeps, a deterministic line-search refines the values the sweep has located. One thing is inherited from GLUE unchanged, and it is the point of the whole approach: the product is not one calibrated model but *every parameter set the record cannot reject*, and a forecast band drawn from all of them.

Why does the ancestry matter to you as a reader? Two reasons. It tells you which parts of the method are established science and which are new. And it locates a genuine gap: the most recent systematic review of statistical back-analysis for embankments on soft clay (Jones et al. 2026) covers least squares, maximum likelihood, Bayesian updating and BUS, and the behavioural-ensemble branch — GLUE and its relatives — does not appear in it at all. That branch is standard practice next door: it is heavily used in catchment hydrology, flood and rainfall-runoff modelling, and contaminant transport, all fields that share this problem's awkward shape — an expensive model, sparse field data, and many parameter sets that fit equally well. Carrying that branch into consolidation back-analysis, revised as above, is the methodological claim of this Part. Chapter 30 then measures how far to trust the result, rather than asserting it.

The comparison can now be read side by side. Figure 21.8 draws the family tree. Table 21.1 compares the two methods dimension by dimension. Figure 21.9 shows where the solver runs are spent, and it carries the quantitative story: random draws find the see-saw ridge only by luck — 14 of the 72 drawn land on it — and on the toy problem of §21.3 that luck rate costs about 6000 blind solves to collect 30 usable ones, against roughly 170 for the planned sweep, whose regular grid crosses the ridge densely inside walls that hold the laboratory knowledge, expanded 3× so a mis-typed range cannot fence the truth out. Figure 21.10 shows who decides the acceptance cut-off — GLUE's soft spot, of which the published critique is blunt: by tuning the threshold the band "can be set to almost anything". Sweep Search removes the choice altogether: no recipe is accepted or rejected at any threshold; every recipe inside the physics walls contributes with a weight computed from its score, and the interval that closes the band uses the fixed one-parameter 95 % likelihood-ratio constant ($\Delta\chi^2 = 3.84$), the same for every analyst.

Figure 21.8. The family tree: RSA's sample–score–keep, GLUE's predictive band, the published weak points — and, at each, what Sweep Search substitutes. The optimisation branch survives inside the ensemble: Steps 3, 5 and 7 are its line searches, Steps 4, 6 and 8 the sweeps around them.

Table 21.1. Sweep Search against GLUE as published, dimension by dimension. The last row, *Product*, is what Sweep Search inherits unchanged: both methods deliver a band rather than a single answer. Every row above it is a revision, and each revision answers a criticism recorded in the family's own literature.

Dimension	GLUE (as published)	Sweep Search (this tool)	What the revision buys
Search space	wide, weakly-structured priors	typed Lo/Hi walls per layer, expanded 3×	laboratory knowledge bounds the search, while the expansion keeps the truth from being fenced out if your range was too narrow — see §23.2
Sampling	blind uniform Monte-Carlo, 10^3–10^5 runs	deterministic three-knob grid (α_A, α_B, α_S)	every run physically consistent; repeatable, auditable
Refinement	none — sample once	alternating speed refits + gated per-layer fine-tune	the see-saw is resolved, not just mapped
Likelihood	informal, analyst's choice	σ-weighted χ^2 with auto-σ calibration	the score carries the instruments' physics
Data weighting	none — long records dominate	balance weights: one vote per instrument family	piezometers heard beside plates
Cut-off	fixed, subjective	none — every admissible recipe is kept, weighted by its score; intervals close on the fixed 95 % likelihood-ratio rule	no analyst-chosen threshold anywhere in the loop
Product	likelihood-weighted band (5–95 %)	P5 / P50 / P95 band + forecast	*inherited unchanged — the family's core idea*
Guard rails	—	bound-hit flag, observability panel	says when a number is a limit, not a measurement
Cost	hours–days typical at 10^4+ runs	minutes, in a browser, on a parallel worker pool	usable while there is still time to act
Determinism	stochastic — reruns differ	same inputs → same answer, always	auditable by a checker

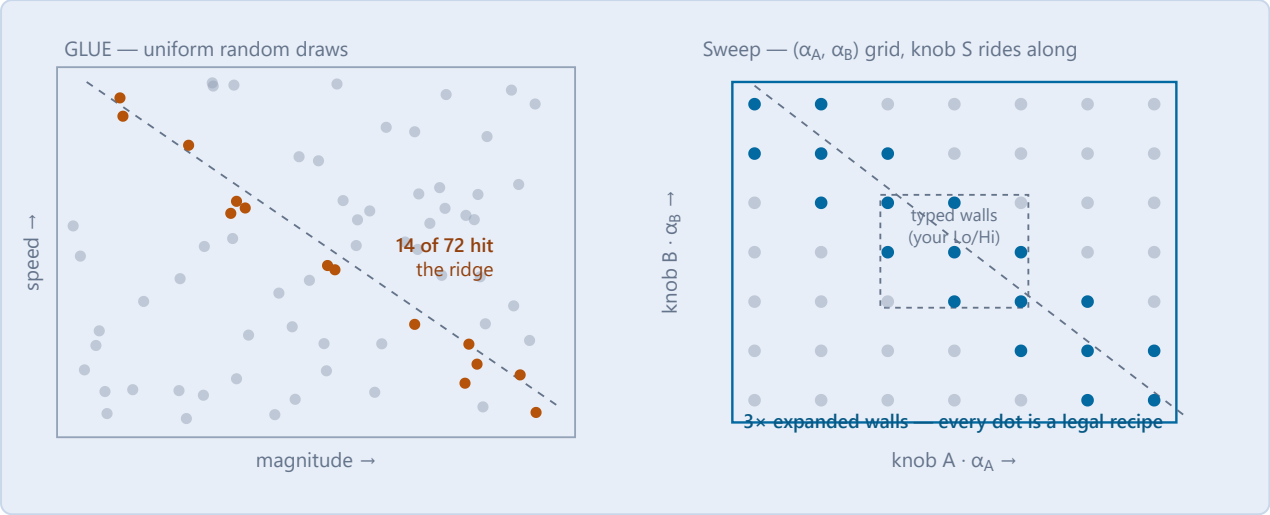


Figure 21.9. Where the runs are spent, in the magnitude–speed plane (dashed ridge = the see-saw's look-alike recipes; dashed box = the typed walls). Left: uniform random draws. Right: the walled, 3×-expanded knob grid, drawn in its (α_A, α_B) plane; the third knob α_S is an independent axis.

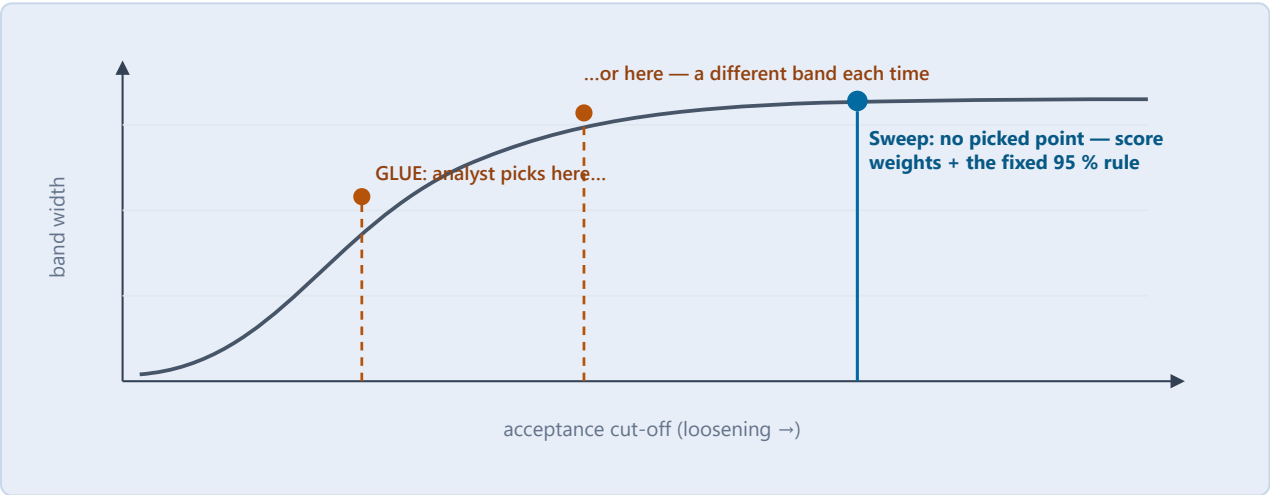


Figure 21.10. Band width against acceptance cut-off: looser thresholds admit worse fits and widen the band — steeply at first, then barely. (Schematic.)

The whole contest on a pocket calculator. The claims above are easy to state and easy to doubt, so here they are replayed as arithmetic. Take a toy soil whose settlement follows one formula: $S(t) = S\infty \times (1 - e^{-t/T})$. It has exactly two dials, the see-saw pair of §21.1: $S\infty$ is the final settlement (magnitude) and T is the time constant (speed). The true soil has $S\infty$ = 1000 mm and T = 100 days, and it is monitored at days 30, 60 and 90, giving readings of 259, 451 and 593 mm with a stated precision of σ = 10 mm. The record therefore stops at just under 60 % of the final settlement, exactly the truncation of §21.1. The toy formula stands in for the real consolidation solver; every method below treats it the same way the real methods treat the real solver, so the logic transfers even though the physics is simplified.

Now audition five candidate soils against the record, shown in Table 21.2, where each row costs three evaluations of the formula and five lines of the §21.2 arithmetic. Check any row by hand.

Table 21.2. Five candidate soils tested against the toy record (readings 259, 451 and 593 mm at days 30, 60 and 90; σ = 10 mm). Each row is one run of the model followed by the χ^2_r arithmetic of §21.2. Four candidates fit the record although their final settlements span 900 to 1200 mm; only the last, 50 % too large and 75 % too slow, is clearly rejected. The record alone cannot separate A from D; any method that returns just one of them is exercising luck, not knowledge.

Candidate	S∞ (mm)	T (days)	Misses at day 30 / 60 / 90 (mm)	χ^2_r	Verdict
A — stiffer, faster	900	85	+8.5 / +4.5 / −5.6	0.41	fits
B — the true soil	1000	100	0 / 0 / 0	0.00	fits
C — softer, slower	1100	115	−6.6 / −4.0 / +3.6	0.24	fits
D — softer and slower still	1200	130	−11.9 / −7.6 / +6.1	0.78	fits
E — far too soft, far too slow	1500	175	−22.9 / −15.8 / +9.7	2.89	**rejected**

What a best-fit method reports. On these clean readings the optimiser correctly finds candidate B, score 0.00. But a real survey carries noise. Add one realistic noise draw, +4, −7 and +3 mm on the three readings, and recompute: A scores 0.75, B scores 0.25, C scores 0.41, D scores 0.87. All four still fit, and the gaps between them are now the same size as the effect of the noise itself. Repeat the survey and a different draw reshuffles the ranking again: the located minimum wanders along the valley from draw to draw. A best-fit method therefore delivers a single number drawn essentially by lot from the range 900–1200 mm, states it with no width, and a client decision (say, when the surcharge can come off) inherits a hidden quarter-of-the-answer uncertainty.

What the sweep reports instead. Do the sweep's move: step S∞ across its range, refit T at each step, and record the best score at every stop. The result is the valley of Figure 21.11, whose floor is nearly flat: every final settlement from 860 to 1240 mm fits the record within the measurement precision. That flat floor *is* the honest answer, and the band P5–P95 is the floor reported with weights: every recipe on it contributes in proportion to how well it fits. The true soil sits comfortably inside it. Nothing about the toy favours the sweep here; the valley was always there. The difference is only that the sweep is built to find and report it, while a best-fit method is built to report one point of it.

What blind sampling costs. GLUE would find the same valley by scattering candidates at random over the prior box, say S∞ from 500 to 2000 mm and T from 30 to 300 days. Only 0.51 % of that box fits the record, so 1000 random solves yield about 5 survivors, and collecting a healthy 30-survivor ensemble costs roughly 6000 solves. The sweep's walk, one short speed refit per magnitude step, maps the entire floor in about 170 solves: some 35× cheaper for a more complete answer. That is revision 1 of §21.3 in numbers.

What the formal cut-off buys. GLUE leaves the acceptance threshold to the analyst, and on this toy the choice moves the answer materially: accept $\chi^2_r \le 0.5$ and the surviving range is 890–1160 mm; accept ≤ 1 and it is 860–1240 mm; accept ≤ 2 and it balloons to 810–1410 mm. Two honest analysts could publish bands differing by half. Sweep Search removes the choice twice over: the score is anchored to the instruments' own σ (a miss is judged against 10 mm because the survey is good to 10 mm), and no binary cut-off exists at all — every recipe on the floor contributes with a weight from

its score, and the interval that closes the band uses the fixed 95 % likelihood-ratio rule rather than a hand-picked number, so the data, not taste, fix where the floor ends. That is revisions 2 and 3 in numbers.

Figure 21.11. The score valley of the toy problem: for each candidate S∞ the speed is refit and the best score plotted; the floor ($\chi^2_r \leq 1$, dashed) is nearly flat from 860 to 1240 mm. The toy uses a hard cut; the app uses score-weighting plus the fixed 95 % rule (§25.6). (Toy model; recomputable by hand.)

KEY IDEA

The toy scoreboard. Same record, three methods. Best fit: one number, drawn by luck from 900–1200 mm, no width. Blind ensemble (GLUE): the right kind of answer, but ~6000 solves to collect it and a hand-picked cut-off that can move the band by half. Sweep Search: the full 860–1240 mm floor in ~170 solves, with weights anchored to σ and a fixed statistical closing rule rather than taste. The real problem has more dials and a slower solver, which makes every one of these differences larger, not smaller.

21.4 The whole method in one breath

Everything is now on the table: the problem (§21.0), its difficulty (§21.1), the loop and its score (§21.2) and the two traditions the method is built from (§21.3). This section compresses the whole method into one summary; the Parts that follow unpack it step by step.

The forward model here is a 2-D axisymmetric (r–z) finite-volume unit-cell solver. It uses the Hansbo (1981) unit-cell geometry, with smear and transition zones and vertical drainage. It is coupled to a C_c/C_r compression law with a $C\alpha/C_c$ (Mesri) creep ratio in a Bjerrum isotache formulation. Part One of this book explains each of those terms; nothing in this Part requires them yet.

Sweep Search reads a soil's hidden parameters from early settlement readings. The difficulty is that the record is *short*: when you must decide, perhaps only half the settlement has happened. Over that short window a soft-but-slow soil and a stiff-but-fast soil trace almost the same curve, so the *speed* of consolidation and its final *magnitude* trade off against each other, the see-saw of §21.1. Fit them both freely at once and the answer can land anywhere along that trade-off.

The method's trick is to **refuse to fit speed and magnitude freely at the same time**. Hold the speed fixed. Sweep the magnitudes systematically across a grid, with only a narrow speed dial riding along. Sweeping means testing rather than guessing: write down a short list of candidate values for each magnitude parameter, spanning the range your laboratory data allow; form every combination of those values, the way a spreadsheet fills a table; solve the forward model for each combination and score it against the readings. Because the speed is held nearly fixed while this happens, the scores reflect the magnitudes alone, and the see-saw cannot act. Then swap: refit the speed at the best magnitudes, and sweep again. Repeat until the two agree, polish layer by layer, and report a *band* rather than a single number. That logic runs as a fixed sequence of steps, always in the same order, and Chapter 24 walks through every step.

Figure 21.12 sketches this alternation. One side of the see-saw is always bolted down while the other is searched, so the compensation that ruins a free simultaneous fit never gets the chance to act. The price is more forward solves, which is why the method needs a fast solver and a fixed, repeatable recipe; the reward is that each sweep answers one question cleanly.

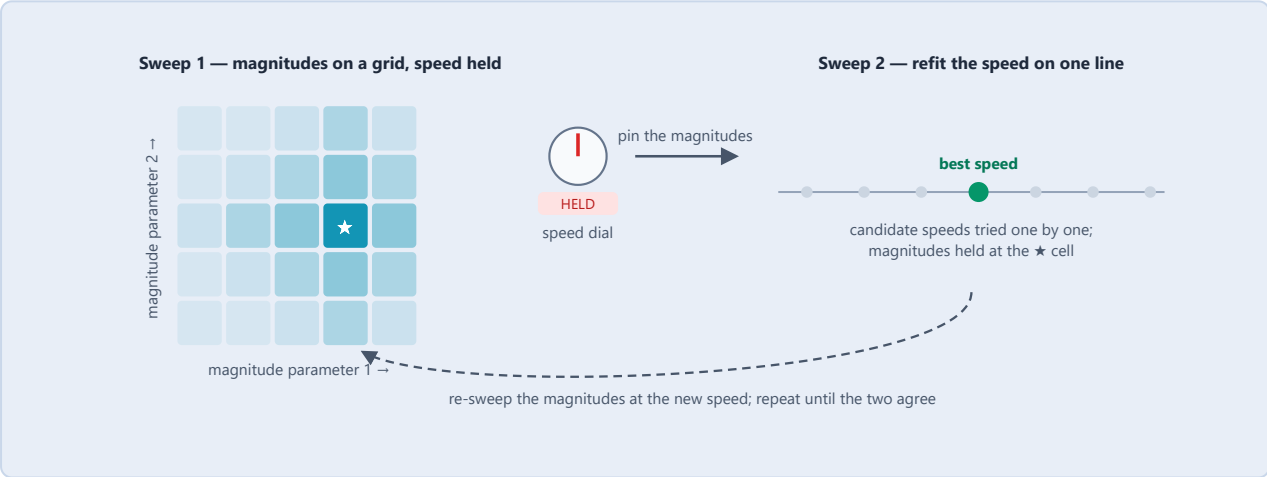


Figure 21.12. The alternation: sweep the magnitudes with the speed bolted down (left; star = best cell), then refit the speed with the magnitudes pinned (right); repeat until neither improves. (Schematic.)

21.5 Using Sweep Search well — the short version

That is the whole foundation. What remains is practice. Seven rules carry most of the value of this whole Part; each is explained in full in the Parts that follow. Treat this page as the summary sheet you would keep beside you during a real project, and come back to it once the chapters behind it make each rule obvious.

#	Rule	Why
1	**Monitor to at least half the expected settlement before trusting a forecast.**	Below that the answer is not yet located and no reported range recovers it (§30.7). Treat the threshold as necessary rather than sufficient: on the one real embankment a record covering two thirds of the settlement still forecast about 10 % low, because model error on real soil is not something this rule covers (§30.8).
2	**Type your Lo/Hi honestly.**	They are not a formality. They set the search space *and* the width of the range you will design to — a carelessly narrow entry produces a confident-looking answer you should not rely on (§23.1, §25.6).
3	**Design to the "including your input uncertainty" range.**	It contained the true settlement in 77 % of tests overall and 91–96 % with adequate monitoring. The narrower "data-resolvable" range managed 58 %, and it stops improving once the record grows — 40 % / 68 % / 66 % at a quarter, half and three quarters of the settlement (§30.3, §30.7).
4	**Read the bound-hit flag before quoting any parameter.**	A value stopped by its own search limit is set by where you put the limit, not by your data (§25.7).
5	**Check the observability panel.**	It is the block near the foot of the results page headed either *Under-observed layers held inside your typed window* or *Observability check — no layer needed holding*. It tells you which layers your record can actually see, and therefore where further site investigation pays (§25.8).
6	**Use the forecast; keep your laboratory values.**	The *forecast* is the reliable product. The individual parameters are far less trustworthy than the forecast they combine to produce. Put simply: the method is good at telling you *how much settlement is coming*, and much weaker at telling you *which soil property was wrong*. Two of them are reliable — the reported range for σ'_p and C_c contained the true value essentially as often as it should. Two are not: for k_0 and C_r the range contained the truth only about half the time, when it should have done so nine times in ten. Worse, a back-analysed C_r was actually further from the truth than the laboratory value the engineer typed in, in 12 tests out of 15, and for k_0 it was a coin toss. Quote either as diagnostic, not as a design value (§31.4).
7	Tell the tool when each instrument was installed.	A plate zeroed after filling began is missing the settlement that came before, and nothing in the numbers reveals it. Declared, the answer is recovered; undeclared, it costs about 12 % on the forecast and two thirds of k_0 (§22.4, Table 22.2).

KEY IDEA

The one-line summary. Give Sweep Search a record that reaches about half the settlement and an honest statement of your laboratory uncertainty, and it will give you a forecast good to a few per cent with a range you can design to. Give it less, and it will tell you so — if you read the panels described in Chapter 25.

CHAPTER 22

The parameters, and the three knobs

22.1 Two families of unknowns

Sweep Search's first organising move is to sort each layer's unknowns into two families by the question they answer (Figure 22.1). The *speed* family sets how fast the water escapes; the *magnitude* family sets how much the ground settles.

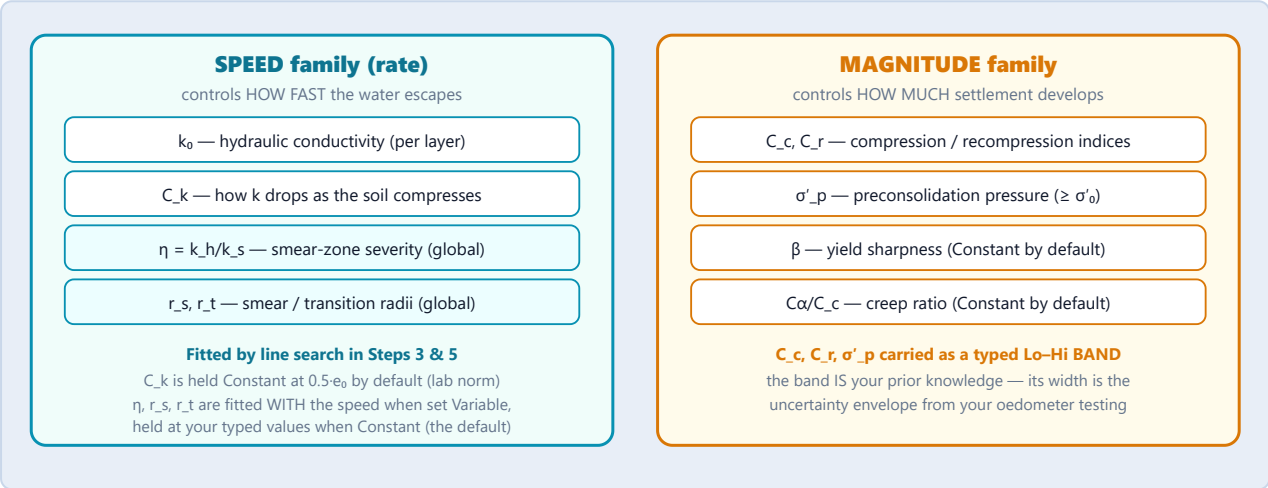


Figure 22.1. The two families. SPEED: k_0, C_k, plus the smear parameters η, r_s, r_t. MAGNITUDE: the compressibility and yield parameters. The speed family is FITTED; the magnitudes are SWEPT as a band (C_c, C_r, σ'_p — β and $C\alpha/C_c$ are Constant by default).

KEY IDEA

Why the split matters — who knows what. Your early record identifies the speed family well and the magnitude family poorly (that is what truncation means). Your laboratory data are the other way round: oedometer tests pin the magnitudes to ±10–20 %, but the field drainage rate is routinely several times the lab value. So the method *fits what the data know* (speed, by Brent line search — Step 3 of Chapter 24 explains the algorithm) and *sweeps what the lab knows* (magnitudes, as a typed band that is simultaneously the prior, the search range, and the final uncertainty statement).

WATCH OUT

Watch out — the smear parameters are speed parameters. It is tempting to file η (= k_h/k_s), r_s and r_t under "geometry" and forget them. Don't: they sit in the flow path of every drop of water leaving the clay, so they trade off directly against k_0. A higher η (worse smear) and a higher k_0 can produce the same drainage rate. That is why the method fits them *with* k_0 (same steps, same line-search machinery) whenever you set them Variable, and reports them in their own *Smear (global)* rows on the speed card, each tagged "Variable · searched" or "held".

Figure 22.2 shows where all of this is typed. Every parameter in the table below has a cell there, and the two small buttons under each cell are the whole Constant/Variable idea in the flesh. Table 22.1 lists the parameters at a glance.

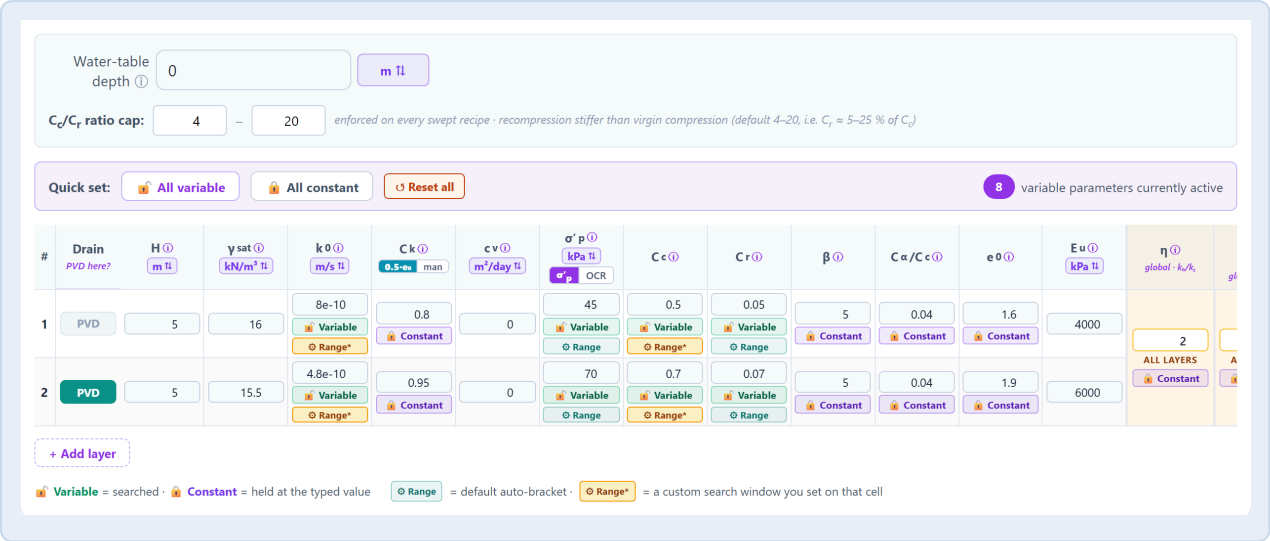


Figure 22.2. The soil table: one row per layer, one column per parameter; the Constant/Variable buttons and ⚙ Range under each cell, the variable-parameter counter top right, and the tinted global smear column at the right. (Back-calculation input page.)

Table 22.1. The parameters at a glance. "Constant" means the search must not touch it; "Variable" means it may. You set this per parameter in the input table.

Symbol	Name (unit)	Family	Default status	What it does / how the method treats it
k_0	initial hydraulic conductivity (m/s), per layer	speed	Variable	the main rate parameter — fitted by line search in Steps 3 and 5, and nudged globally by the α_s knob during the sweep
C_k	permeability-change index (–), per layer	speed	Constant at $0.5 \cdot e_0$	how fast k drops as the soil compresses. Settlement data barely constrain it, so it is held at the laboratory norm — a tested decision that costs at most ≈0.3 % of final settlement even if the norm is 40 % wrong
η	smear severity k_h/k_s (–), global	speed	Constant (typed)	how much the drain installation damaged the soil ring around it (η = 1 = no smear). When Variable it is fitted with the speed in Steps 3 and 5 — it scales the drainage rate exactly like a conductivity
r_s, r_t	smear / transition radii (mm), global	speed	Constant (typed)	how wide the damaged ring and its transition are; same treatment as η
C_c, C_r	compression / recompression indices (–)	magnitude	band (Lo–Hi)	the slopes of the compression curve — the main "how much" parameters
σ'_p	preconsolidation pressure (kPa)	magnitude	band (Lo–Hi)	the yield stress — where compression switches from stiff (C_r) to soft (C_c). Physically floored at the in-situ stress σ'_0 (OCR ≥ 1 always)
β	yield sharpness (–)	magnitude	Constant (typed)	how abruptly that switch happens (1 = very gradual, 10 = sharp)
$C\alpha/C_c$	creep ratio (–)	magnitude	Constant (typed)	secondary compression, modelled by the EVP (Bjerrum equivalent-time / isotache) law that Part One of this book sets out in full: each cell carries two hardening states, σ'_p from loading alone and from loading *and* creep. The creep clock starts at **yield** (t_e from the current OCR), not at t = 0. The previous decoupled add-on stacked creep on top of a strain that already contained the virgin compression, double-counting it; the isotache form cannot, by construction. $C\alpha/C_c = 0$ reproduces the elastoplastic answer exactly.
E_u	undrained (elastic) modulus (kPa), per layer	neither — it is *not* consolidation	Constant (typed)	sets the settlement that appears *as the fill goes on*, before any water has left. Never searched, and it does not enter the score — see §22.3

Where do the typed numbers come from?

- **C_c, C_r, σ'_p, e_0:** oedometer tests on undisturbed samples. When you have several tests in a layer, their scatter *is* your Lo–Hi range — do not average it away.

- **C_k:** from the same oedometer's permeability readings if you have them; otherwise the lab norm $C_k \approx 0.5 \cdot e_0$ (the default the tool locks it to).
- **k_0 seed:** lab permeability, CPTu dissipation, or $c_v \cdot m_v \cdot \gamma_w$ from the oedometer. Only a starting point — the field value is what the speed fit exists to find, and it is routinely several times the lab number.
- **β:** rarely measured directly; 3–7 covers most clays (default 5). Give it a generous range.
- **$C\alpha/C_c$:** Mesri's correlation — about 0.04 ± 0.01 for inorganic clays, 0.05–0.06 for organic soils and peat.
- **η, r_s, r_t:** from the drain and mandrel geometry plus published back-analyses: η typically 2–5, r_s about 2–3× the equivalent mandrel radius, r_t beyond that where the soil recovers. These are exactly the numbers site experience disagrees about most — which is why the method lets you set them Variable and fit them with the speed.

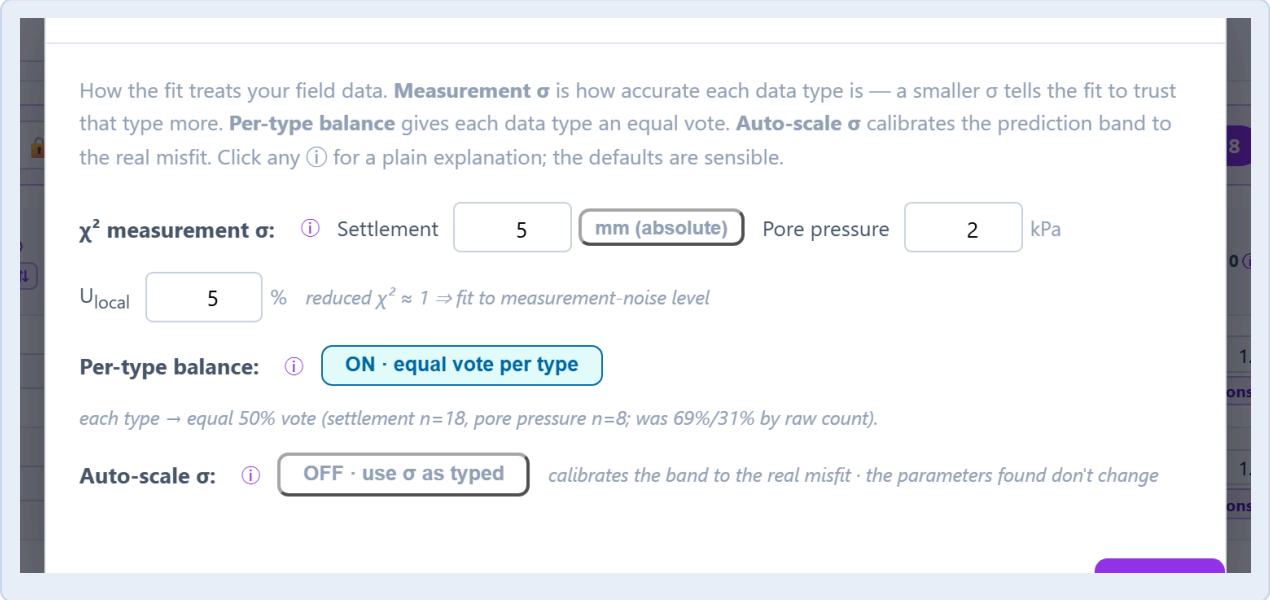


Figure 22.3. Observation accuracy: the per-instrument measurement precision σ, the auto-scale switch, and the per-type balance rule of §21.3 (one vote per family, on by default). (Back-calculation → Observation accuracy.)

22.2 The three knobs

When the sweep moves the magnitudes it does not move them one at a time. It groups them onto **three knobs**, and one turn of a knob is one complete, physically-consistent recipe:

- **Knob A** blends the settlement-*raising* group {C_c, C_r, $C\alpha/C_c$} — turn it up and the ground settles more.
- **Knob B** blends the settlement-*lowering* group {σ'_p, β} — turn it up (more preconsolidation) and the ground settles less.
- **Knob S** is one global speed factor on k_0 — the narrow dial that rides along with the magnitude sweep.

One α per knob, shared by its members. Within a knob every parameter rides a single shared α, each blending along its own wall range. So knob A's C_c, C_r and $C\alpha/C_c$ always step together, and knob B's σ'_p and β step together. But the three knobs carry three *independent* αs (α_A, α_B, α_s), swept as separate axes. That is exactly why the sweep is a 3-D grid of (α_A, α_B, α_s), not a single slider.

Ganging the members is a deliberate *search strategy*, not a limit on the answer: it keeps the global scan a cheap, robust grid, stops members from compensating one another inside the sweep, and a short record cannot separate, say, C_c from C_r anyway. Each parameter is still freed individually later, in the gated per-layer fine-tune of Step 7, once the global picture is pinned.

Knob A and knob B: why two, not one

The magnitudes are split by the *direction* they push settlement. Two knobs means the search lives on a *square*, not a line — and that matters whenever the truth needs both groups moved the same way (for example, a site that is both more compressible *and* more overconsolidated than you typed). One knob can never express that combination; it is forced to a compromise on the anti-diagonal, and no amount of later polishing recovers what the scan never offered. Figure 22.4 draws the difference.

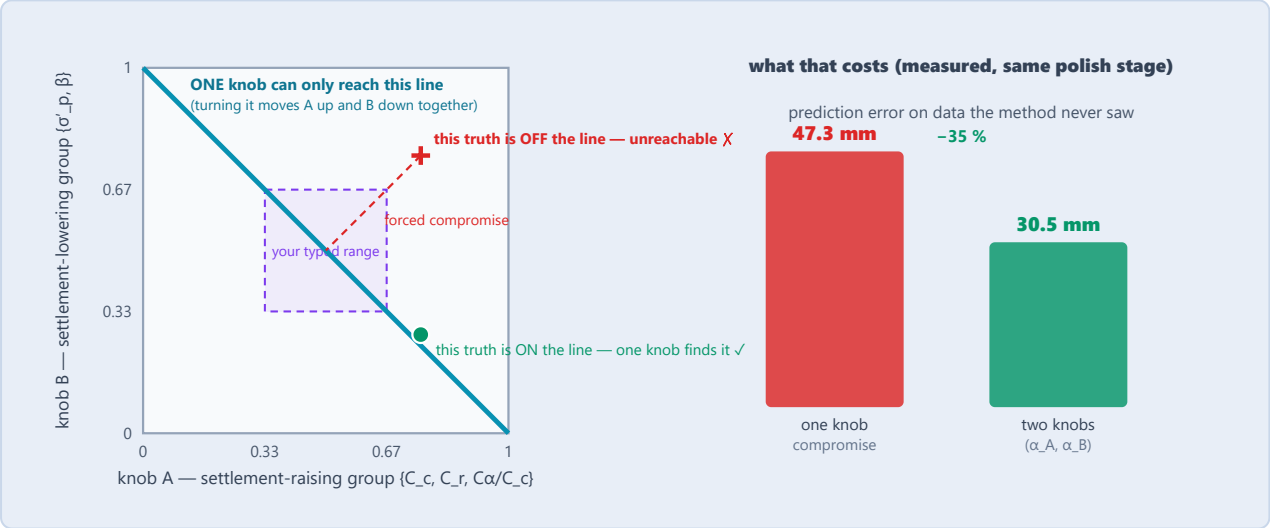


Figure 22.4. Why two knobs. Left: one knob moves groups A and B in fixed gear, so only the teal line is reachable. Right: the measured price — unseen-data error 47.3 → 30.5 mm (−35 %) when the second knob is added, all else identical.

Knob S: the joint speed factor — the most important trick in the method

Here is the subtle failure the third knob prevents. The frozen speed came from the first speed fit, which used *your typed magnitudes*. Suppose your typed values are biased in opposite directions — say you typed the compressibility too low and the preconsolidation too high. The speed fit then anchors a rate that half-compensates for a soil profile that does not exist. Every recipe the sweep tries is scored against that mis-calibrated ruler. The result is not a small shift; the whole misfit landscape tilts, and the best-looking valley can be the wrong one entirely. Figure 22.5 shows it happening on real solver output.

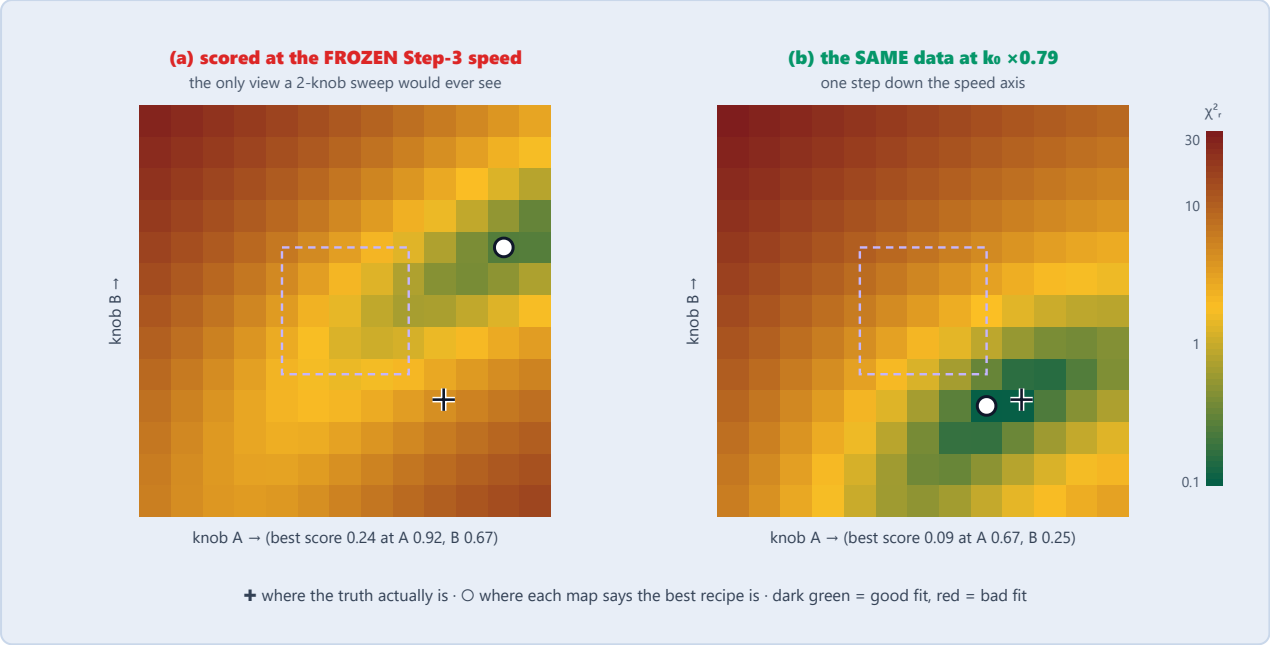


Figure 22.5. The wrong-valley failure (644 real solves): identical recipes and readings; only the assumed speed differs. (a) Frozen speed — the best recipe (○) sits far from the truth (✚). (b) One speed step down — the landscape reshapes; the best recipe moves next to the truth and fits 2.5× better.

Could we fix this afterwards, by refitting the speed at the winner and re-sweeping at fixed speed? No — and this is the crux. Refitting the speed at the *wrong* winner gives the speed that best serves the wrong valley; a frozen-speed re-scan then confirms the wrong choice. You walk in circles inside the wrong valley. **The choice of valley is made during the scan, so the speed must vary during the scan.** In a controlled experiment the frozen-speed pipeline finished with a prediction error of 89.3 mm. The three-knob pipeline, on the same problem, finished at 23.9 mm, which is 3.7× better.

The speed knob is deliberately modest, and each design choice has a reason:

- **It is narrow (k_0 ×0.625 to ×1.6).** Give the speed a decade of freedom and the see-saw comes straight back — every magnitude error could be compensated and the valley would flatten into mush. The narrow range covers a plausible anchoring error without dissolving the signal.
- **It is one global factor, not per-layer.** The dominant failure is the anchor's overall level; per-layer detail is the speed fit's job. Scanning per-layer speeds would explode the grid and re-open full compensation. Layers whose k_0 you locked Constant are never touched by it.
- **Winners are judged at their best speed ("min-over-S").** Each (A, B) recipe gets the credit of the best speed level scanned for it — so a good magnitude recipe is not punished for a poor speed pairing, and one valley cannot appear three times (once per speed) pretending to be three answers.
- **The knob's position gets its own confidence range.** The scan measures how far the speed level could move before the fit degrades beyond the 95 % rule — so speed uncertainty is a number the method reports, not something hidden inside the band.

Figure 22.6 shows that profile, measured at the winning magnitudes.

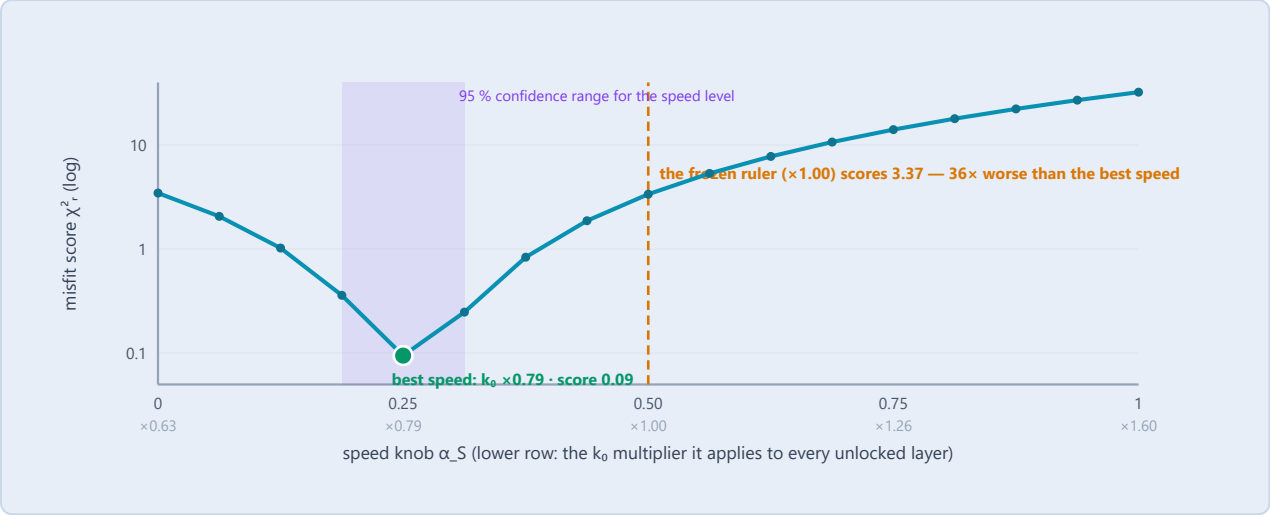


Figure 22.6. The speed knob's own profile at the winning magnitudes (real data): the frozen ruler (amber) fits 36× worse than the best level (green); the violet strip is the 95 % range for the speed level.

KEY IDEA

And when the inputs are NOT biased in opposite directions? Then the third knob finds $\alpha_s \approx 0.5$ (factor ×1.0), confirms the anchor was fine, and changes *measurably nothing*. On the coherent-bias test all architectures finish within a tenth of a millimetre of each other (29.8 mm). The knob costs a few extra solves on every run and pays for itself precisely on the runs where the inputs disagree in direction — which you cannot know in advance. The failure it prevents looks, from the inside, like a perfectly healthy run: a clean valley and a confident winner. That is why it is always on. *(Calibration note: these are the configurations that maximise the knob's benefit. Over randomly-directed bias its average benefit is close to zero — it is cheap insurance against this one failure mode, not a general accuracy gain. See Chapter 30, §30.4.)*

Figure 22.7 compresses the evidence into one picture. ("Biased in opposite directions" means the two input errors push the curve opposite ways and so hide each other — say a C_c typed 20 % high, predicting too much settlement, together with a k_0 typed three times low, predicting it arriving late: over a short record the two nearly cancel, the fitted curve still looks right, and only the forecast beyond the record is wrong, in both magnitude and timing. Same-direction errors add instead of cancelling, the misfit is obvious, and every architecture finds it — the left panel.)

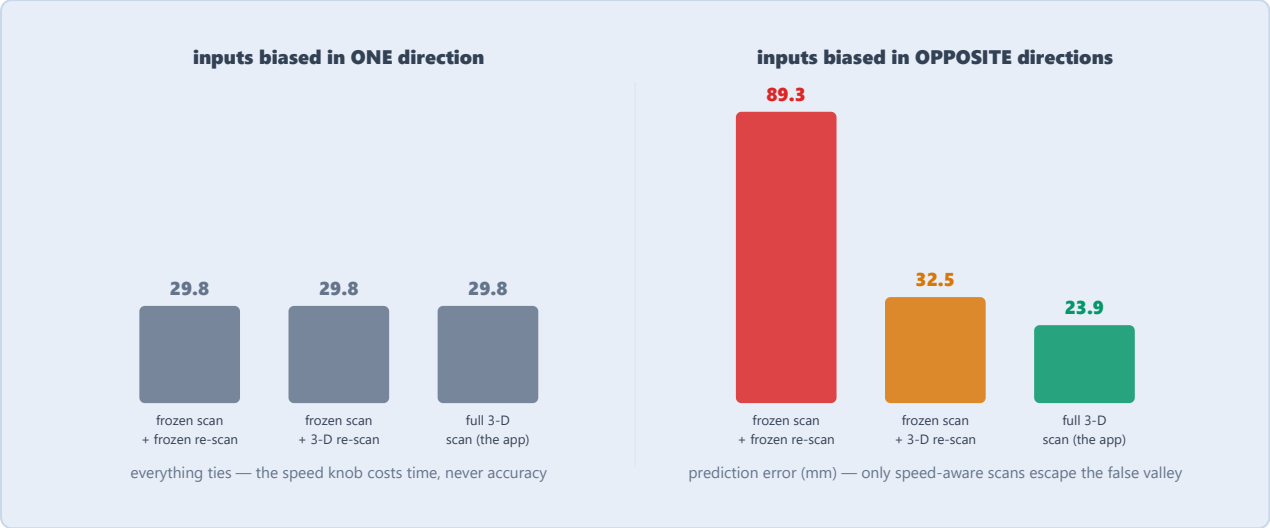


Figure 22.7. Prediction error on unseen data, across architectures (real experiments). Left: inputs biased the same way — every architecture ties. Right: opposite-direction bias — frozen-speed scan 89.3 mm; frozen re-scan stuck; 3-D re-scan 32.5; always-on 3-D scan 23.9.

22.3 Immediate settlement — modelled, but never searched

Not every millimetre a settlement plate records is consolidation. The instant load arrives, saturated clay distorts at essentially constant volume — shear, with no water leaving. That component is governed by the undrained modulus E_u, and the tool models it as $\varepsilon_u(z, t) = \Delta\sigma_z(t) / E_u$, so that $S_i(t) = \Sigma\, \Delta\sigma_z(t) \cdot H / E_u$ over the layers.

> KEY IDEA
>
> **It follows the loading schedule — it is not a step at t = 0.** Because it is driven by *total* stress, the immediate settlement appears exactly as the fill appears: it ramps up while the embankment is being placed and then holds flat. Vacuum is excluded on purpose — suction adds no total stress, so it causes no immediate distortion.

Table 22.1b. The immediate component through a 30-day construction ramp, for 100 kPa over 5 m at E_u = 4 MPa above 5 m at 6 MPa. The right-hand column is what you would see if it were (wrongly) applied all at once. Verified against the solver to the last decimal.

t (days)	Load in place	Immediate settlement	If it were all at t = 0
2	7 %	13.9 mm	208.3 mm
6	20 %	41.7 mm	208.3 mm
16	53 %	111.1 mm	208.3 mm
30 (fill complete)	100 %	208.3 mm	208.3 mm
600	100 %	208.3 mm	208.3 mm

On that example the immediate component is 208 mm against roughly 1.1 m of consolidation — about a sixth of the total, and most of it delivered before the first month is out. Leave it out of the model and the early record looks unexplainably fast; the search then has to buy that speed somewhere, and the only currency it has is the drainage parameters.

> KEY IDEA
>
> **You do not need to subtract it before searching — the arithmetic already does.** E_u is *Constant*: it is not among the search variables and cannot be varied by the sweep. So the immediate settlement is the same fixed quantity whichever way you write the residual:
>
> $$\text{residual} = \text{Obs}(t) - [\,\text{Primary}(\theta,t) + \text{Creep}(\theta,t) + S_i(t)\,] \equiv [\,\text{Obs}(t) - S_i(t)\,] - [\,\text{Primary}(\theta,t) + \text{Creep}(\theta,t)\,] \tag{22.1}$$
>
> Stripping S_i from the plate readings first and leaving it on both sides give *identical* residuals, hence an identical χ^2, the same winner and the same band. In other words the sweep is *already* fitting the primary (+ creep) response only; the elastic part cancels instead of competing for the fit.

WATCH OUT

Watch out — an error in E_u does not stay in the elastic bucket. E_u is held and never corrected by the data. Getting it wrong therefore does more than mis-state the immediate settlement: the leftover shows up in the residual, and the search absorbs it into the parameters it *can* move. The mis-fit it creates has the shape of the construction ramp: a rise, then flat. That is uncomfortably close to the shape of early primary consolidation, so the compensation usually lands on C_c and / or k_0. Pre-subtracting does *not* protect you: it is the same arithmetic. The defences are to get E_u right, and to check the datum below.

WATCH OUT

Watch out — when was the plate zeroed? Whether your record even contains the immediate settlement depends on it. A plate founded on the original ground and read before filling captures the whole of it. A plate zeroed after a working platform or drainage blanket is already in place has lost the part that occurred under that first lift. A model carrying the full $S_i(t)$ will therefore over-state the early settlement, and by the paragraph above that error will be paid for out of C_c or k_0. If the datum is uncertain, say so alongside the back-analysed magnitudes.

Choosing a value. For soft clay E_u is commonly taken as (100–500)·s_u (Duncan & Buchignani, 1976). For normally consolidated clay $s_u \approx 0.22\,\sigma'_p$ by the SHANSEP relation (Ladd & Foott, 1974). With σ'_p = 45 kPa that is $s_u \approx$ 10 kPa and $E_u \approx$ 1–5 MPa. Set E_u = 0 to switch the whole term off, which is the right choice if your plates cannot see it.

22.4 Your monitoring record — where is zero, and whose clock?

Figure 22.8 shows the card where the record is entered; the two questions of this section sit on every instrument's own row.

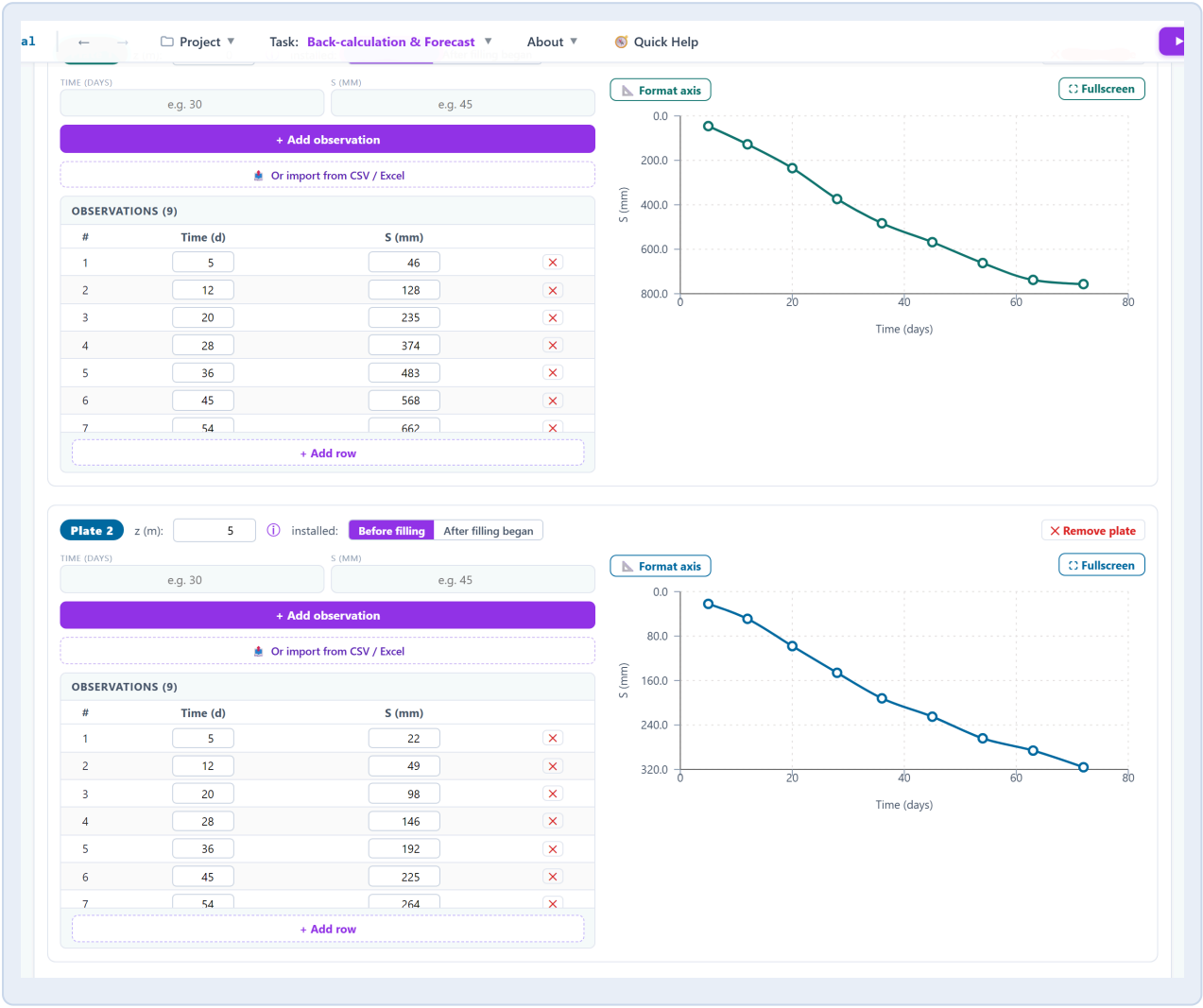


Figure 22.8. The monitoring-record entry: each plate is a block — depth, the two installation questions, a typed or imported (time, settlement) table, and a live plot where a mistyped reading shows as a kink. Piezometers follow in the same shape. (Back-calculation input page.)

Everything so far assumed the readings mean what the tool expects: cumulative settlement since filling started, timed on the same clock as the loading schedule. On a real site that is often untrue, and nothing in the numbers reveals it. The tool therefore asks two questions per instrument, and Figure 22.10 shows them on a piezometer row: *installed* — *Before filling / After filling began*, which fixes the datum, and the matching choice of clock. Both sit on the instrument's own row, beside its readings, because they are properties of that instrument and not of the project.

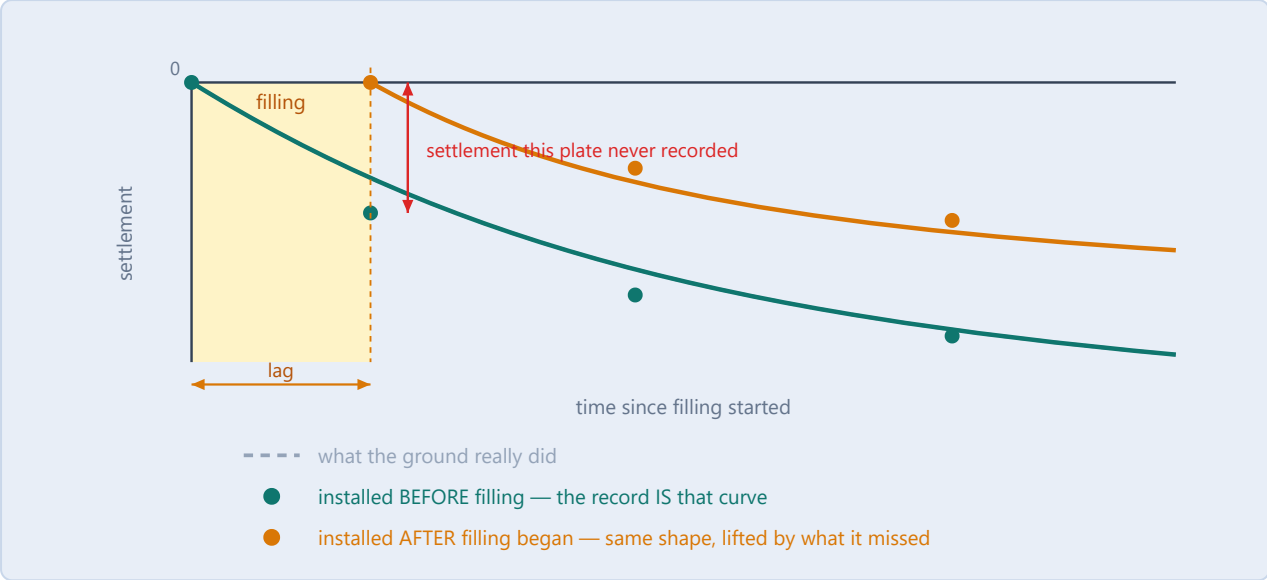


Figure 22.9. The same instrument position recorded two ways: zeroed before filling (true curve), or zeroed at first reading — identical shape, offset by the settlement already missed, which no analysis can recover but the tool can allow for once told.

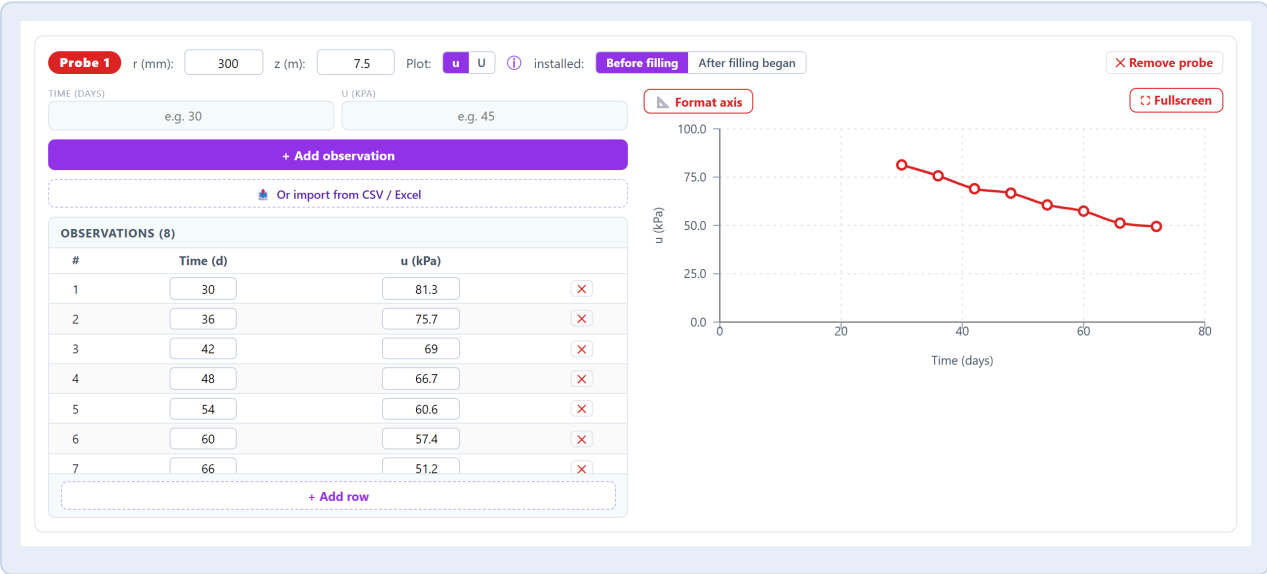


Figure 22.10. The two installation questions, on a piezometer row: *Installed* decides whether the settlement the instrument never saw is credited to the model or quietly lost. The controls sit on the instrument's own row because they describe that instrument. (Back-calculation input page.)

KEY IDEA

Installed BEFORE filling — the ideal case. The instrument was in place and zeroed before any load went on, so every reading is cumulative settlement since filling started and its times are already on the loading clock. Nothing needs correcting. *A late first reading is still this case* — what matters is where zero is, not when the readings happen to begin. A plate surveyed against a benchmark before the fill arrived belongs here.

KEY IDEA

Installed AFTER filling began. The instrument was zeroed at its first reading, typically once a working platform or drainage blanket was down. The settlement already in the ground is simply absent from the record (Figure 22.9). Told this, the tool references its own prediction to the same instant, and model and measurement are compared like with like. It needs two answers: the **lag** in days from the start of filling to that first reading, and what the **time column measures** — days since filling started (already on the loading clock) or days since the first reading (the tool adds the lag).

WATCH OUT

Watch out — the time column is the easy one to get wrong. A log written "day 30, 37, 44" is *already* on the loading clock. Declaring it as days-since-the-first-reading makes the tool add the lag again, moving those readings to day 60, 67, 74. The fit will be badly wrong and nothing on screen will say why. Check the first number in your time column against the day the fill started.

Table 22.2. What the declaration is worth. One synthetic truth, three inversions; arms B and C are given *byte-identical* observations and differ only in whether the tool is told what they mean. Errors are against the true soil; the forecast is the settlement remaining after the datum.

Case	C_c error	k_0 error	Forecast error
A — zeroed before filling (ideal)	+0.5 %	−18.0 %	**−0.5 %**
B — zeroed late, *not declared*	−13.0 %	−68.1 %	**−12.0 %**
C — zeroed late, *declared*	+0.7 %	−7.6 %	**+0.2 %**

Declaring the datum recovers the ideal answer; leaving it undeclared costs about 12 % on the forecast and two thirds of k_0. (The ≈ 18 % k_0 error in the ideal case is the ordinary equifinality of this test, which starts from a deliberately poor rate seed. It is not a consequence of the datum.)

WATCH OUT

Watch out — do not quote k_0 from a late-installed plate if E_u is uncertain. Repeating case C with the typed E_u wrong by ±30 % moves k_0 by **+67 % and −35 %**. C_c stays within 0.5 %, and the **forecast stays within 1 %**. Referencing the model to the datum removes the constant offset, so the settlement magnitude is protected and the leftover shape error is absorbed by the drainage rate instead. The number you design to is safe; the back-analysed permeability, in this situation, is not a measurement. See §25.7 on the same distinction for bound-hit parameters.

KEY IDEA

Before you start, settle three things about each instrument. When was it installed relative to the first load? What was its reading at that moment: zero, or a survey against a benchmark? And what is day 0 in the time column? Ten minutes with the installation records is worth more here than any amount of adjusting search bounds afterwards.

CHAPTER 23

The search space: building the walls

Everything the sweep does happens between two *walls* per magnitude parameter. This part builds them from scratch, with real numbers, in four moves: your typed range → automatic repairs → the 3× expansion → the α blend. Work through it once with a calculator and the sweep stops being a black box.

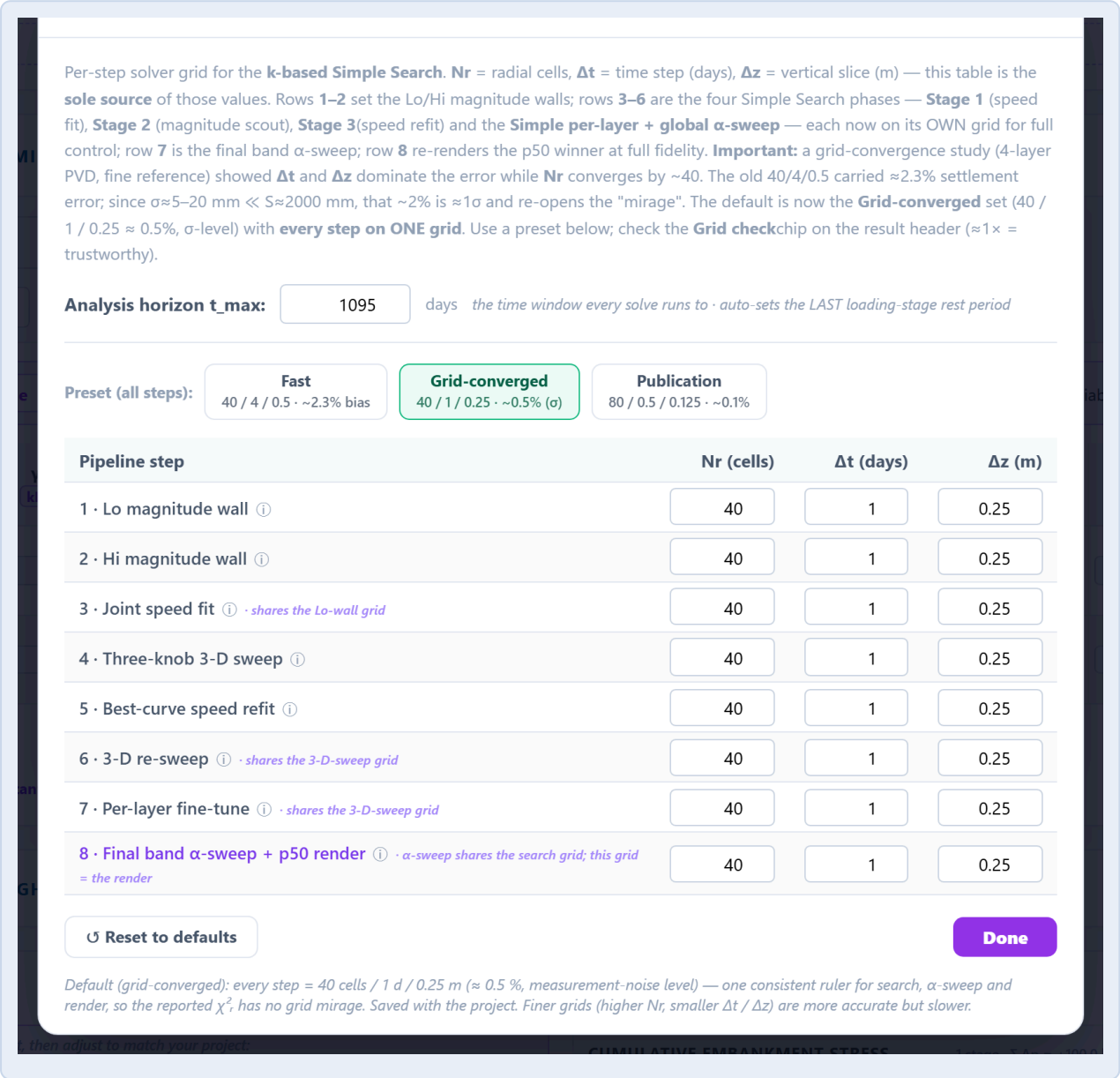


Figure 23.1. The solver grid every step shares: one setting, so no step can flatter itself against a finer ruler than the others; the defaults are the grid-converged values of Part One. (Back-calculation → Advanced solving.)

23.1 Step 1 — the typed walls (your job)

One setting sits behind every step that follows, and Figure 23.1 shows it: the solver grid, shared by the whole pipeline so that no step is scored against a finer ruler than another.

For each magnitude parameter of each layer, open the ⚙ **Range** button on that cell. The *Override bounds* popup takes either a **Uniform** Lo× / Hi× multiplier pair (it shows you the resulting minimum and maximum) or a **Normal** COV, which it converts to a ±2σ window. Set no range at all and a Variable magnitude is auto-bracketed at the recommended spread (C_c 20 %, C_r 30 %, σ'_p 20 %, β 20 %, $C\alpha/C_c$ 20 % COV). Three rules of thumb. Use the scatter of your actual tests, not a guess. Keep Lo and Hi on the same side of physics (σ'_p never below σ'_0). And resist the urge to make the range tiny to "help" the search: the band is what the final uncertainty statement inherits, and an over-confident band produces an over-confident forecast.

Two automatic repairs are then applied. Both are silent, so it is worth knowing they happen:

- **Degenerate band (Lo = Hi).** A zero-width band gives the sweep nothing to move, so any *Variable* magnitude without an explicit range is auto-bracketed at the recommended spread (about 20 % COV). This is the intended default rather than a repair, and it is not flagged on the result page. Anything you mark *Constant* is exempt — held at your typed value with a zero-width band. Note that β and $C\alpha/C_c$ are Constant by default.
- **Unphysical σ'_p.** Every candidate the solver sees is clamped to $\sigma'_p \geq \sigma'_0$ layer by layer — no recipe can be rewarded for an overconsolidation ratio below 1. If your typed Lo was below σ'_0, the clamp quietly becomes the effective wall.

23.2 Step 2 — the 3× expansion (the tool's job)

Soil parameters live on a *log* scale — the natural measure of "how different" two values are is their ratio, not their difference. So write $g = \ln m$ and your typed walls as $g(lo) = g_{mid} - w$ and $g(hi) = g_{mid} + w$, where $w = \tfrac{1}{2}\cdot\ln(hi/lo)$ is the log half-width. The extended walls simply triple that half-width:

$$g(exLo) = g_{mid} - 3w \quad \text{and} \quad g(exHi) = g_{mid} + 3w \tag{23.1}$$

Converting back out of logs gives the closed forms the tool actually computes — no logarithms at run time:

$$exLo = lo^2/hi \qquad exHi = hi^2/lo \tag{23.2}$$

(Check: $\ln(lo^2/hi) = 2\ln lo - \ln hi = 2(g_{mid} - w) - (g_{mid} + w) = g_{mid} - 3w$ ✓.) Because the tripling is in log space it is symmetric in ratio: exLo sits the same factor below your Lo as exHi sits above your Hi. **One exception:** a layer whose settlement has barely started is *not* expanded — it is searched inside your typed Lo/Hi only. Where a layer still has ≥ 1.5× as much settlement to come as the best-

observed layer in the profile, the record cannot constrain it. What would move its parameters is the fit to the *other* layers. Expanding its walls therefore discards the one reliable piece of evidence about it: your laboratory value. Its k_0 is likewise held to ×0.3–×3.0 of the typed value, because k_0 is a *speed* parameter and never passes through this expansion at all. On an evenly-observed profile nothing is held and this step behaves exactly as described above. The results page names any layer that was held, with the figures behind the decision; the benefit and its limits are quantified in §30.7. Table 23.1 walks the wall construction for two parameters of one layer.

Table 23.1. Worked wall construction for two parameters of one layer. C_c behaves normally; σ'_p shows the physical clamp taking over when the expansion dips below the in-situ stress (σ'_0 = 40 kPa here).

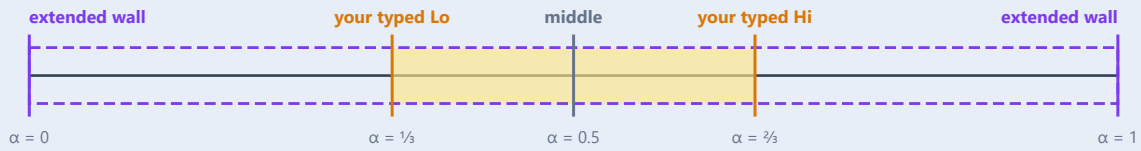
	C_c (–)	σ'_p (kPa)
typed Lo	0.360	45.0
typed Hi	0.440	55.0
log half-width w = ½·ln(hi/lo)	0.100	0.100
exLo = lo²/hi	0.36²/0.44 = 0.295	45²/55 = 36.8
exHi = hi²/lo	0.44²/0.36 = 0.538	55²/45 = 67.2
physical clamp	–	$\sigma'_p \geq \sigma'_0$ = 40.0 ⇒ effective low wall 40.0
span = exHi/exLo	1.826	1.826 (before clamp)

23.3 Step 3 — the α blend: one number, one complete recipe

Define a slider α from 0 to 1 and let every parameter travel from its extended-low to its extended-high wall along a straight line in log space:

$$m(\alpha) = exLo \cdot (exHi/exLo)^{\alpha} \tag{23.3}$$

One α value fixes every parameter of every layer at once — one complete, physically consistent recipe. Landmarks: α = 0 and 1 are the extended walls; α = ⅓ and ⅔ reproduce your typed Lo and Hi exactly; α = ½ is the geometric middle √(exLo·exHi). So the search covers your range PLUS a symmetric safety margin either side, in case your range was wrong. Figure 23.2 draws the ruler, and Table 23.2 shows the blend in action.



α is a slider from 0 to 1. α = 0 gives every parameter its extended-low value; α = 1 the extended-high value; values in between blend on a LOG scale, because soil parameters vary by factors, not by amounts. Your typed range sits at ⅓–⅔, so the search covers your range PLUS a safety margin either side, in case your range was wrong.

Figure 23.2. The α ruler: each typed Lo–Hi is stretched threefold in log-width (exLo = lo²/hi, exHi = hi²/lo), so the typed values land at α = ⅓ and ⅔ — the sweep checks your range plus a symmetric margin beyond it.

Table 23.2. The blend in action for C_c (exLo = 0.295, exHi = 0.538). Note the equal RATIO between neighbouring entries — that is what "linear in log space" means.

α	0	⅙	**⅓ (Lo)**	½ (mid)	**⅔ (Hi)**	⅚	1
$C_c(\alpha)$	0.295	0.326	0.360	0.398	0.440	0.486	0.538

23.4 Step 4 — who rides which knob, and the safety clamps

Every magnitude key uses the same blend, but reads its α from its own knob: {C_c, C_r, $C\alpha/C_c$} from knob A; {σ'_p, β} from knob B. After blending, **every candidate passes through the physical clamps** before it is solved: $C_r \le 0.95 \cdot C_c$, $\beta \in [1, 10]$, $C\alpha/C_c \in [0.005, 0.2]$, and $\sigma'_p \ge \sigma'_0$. These four are **not user settings**, and deliberately so: each states a fact of soil mechanics rather than a preference. Recompression cannot be softer than virgin compression; a yield sharpness below 1 or above 10 no longer describes a yield curve; the Mesri creep ratio outside 0.005–0.2 is outside anything reported for real soil; and a preconsolidation pressure below the in-situ stress would mean the ground had never carried its own overburden. A search allowed past them would return arithmetic, not soil. One related limit *is* yours to set: the admissible C_c/C_r ratio window, default 4 to 20, which sits with the water-table depth above the soil table. It was widened from 5–10 in August 2026 because the tighter window rejected ordinary soils and silently discarded most of the input-uncertainty draws. Walls, blends, refined vertices and fine-tuned centres alike — no step can ever score an unphysical recipe.

CHAPTER 24

The pipeline, step by step

The steps run top to bottom with no gaps — the output of one is the input of the next. Each is described the same way: what it *Receives*, what it *Hands on*, *Why* it exists, its numbered substeps, its *stopping criterion*, and the exact settings. Nothing is left as "and then some magic happens". Figure 24.1 shows the whole pipeline and its rhythm.

Figure 24.1. The Sweep Search pipeline and its rhythm: fit speed (teal) → sweep magnitudes (purple) → refit and re-check (teal) → polish and diagnose (amber) → report with uncertainty (green). Steps 1–2 set up the band; Step 6 loops back on itself.

Step 1–2 — the two corner brackets

the least- and most-settlement corners of your band

IN A NUTSHELL

Before searching for anything, the tool runs two "what if" cases: the softest soil your typed numbers allow, and the stiffest. Neither is expected to be right. They exist to bracket the answer, the way you might weigh the lightest and heaviest a parcel could possibly be before guessing its actual weight. Everything the search does later is judged against these two ends.

Solve the model once at the magnitude combination that gives the least settlement your typed band allows, and once at the combination that gives the most. These two steps are purely about magnitude — the speed is not fitted, not swept, and plays no part; the solver carries your seed k_0 through so a curve can be drawn.

Receives	Your typed magnitude band (the seed speed is carried through unchanged, identical for both corners).	Hands on	The minimum- and maximum-settlement curves and their χ^2 — the two edges of the band.

Why this step exists. Before the search can start it must know how much settlement your typed numbers even allow — the least and the most. The two corner solves are opposite corners (stiffest, least-settling vs softest, most-settling), so every outcome your numbers allow lies between them. It is a pure "how much" bracket, nothing to do with rate, which is why the speed is left at its seed. The two curves become the band edges on the result page, and the gap between them is the range the later sweep searches inside.

What happens, substep by substep

1. **Least-settlement recipe:** raising magnitudes {C_c, C_r, $C\alpha/C_c$} at their *Lo* wall, lowering magnitudes {σ'_p, β} at their *Hi* wall (a high σ'_p keeps the soil on the stiff reload line and a low C_c makes the virgin line shallow — both push settlement down).
2. **Most-settlement recipe:** raising magnitudes at *Hi*, lowering magnitudes at *Lo*.
3. Pass each through the physical clamps ($\sigma'_p \geq \sigma'_0$, etc.).
4. Run one forward solve per corner at the seed speed, with nothing fitted (zero cycles).
5. Store each corner's curve and its χ^2 reference.

Stopping criterion. None — each is a single forward solve with nothing fitted, so there is no convergence loop to stop. Each step produces one curve and its χ^2, computed exactly as in §21.2: every reading's miss is divided by that instrument's σ, squared, summed, balanced across instrument families, and divided by the degrees of freedom.

Exact settings. default solver grid (40 cells / 1 day / 0.25 m); no parameter searched — speed stays at the *seed* value, meaning the starting number the step was handed (here your typed k_0), not yet adjusted by any fit.

Step 3 — Joint speed fit

fit the drainage speed against BOTH corners at once

IN A NUTSHELL

Now the tool holds the *amount* of settlement still and asks how *fast* the water escapes. It tries different drainage speeds and keeps the one that best matches both corner cases at the same time. Matching both matters: a speed tuned to the soft corner alone would be tuned to a soil we have no reason to believe in.

Find the per-layer drainage speed k_0 that matches the record as well as possible — judged against both corners of your magnitude band simultaneously, not against one guess.

`Receives`	The two corner curves; the typed walls.	Hands on	A well-anchored per-layer speed (plus any Variable smear).

Why this step exists. A short record pins the drainage rate far better than the final magnitude, so the method fixes the rate first. Why first? Because rate and magnitude can imitate each other — a soil that drains slowly but settles a lot, and one that drains quickly but settles little, trace almost the same early curve. Fixing a well-anchored speed before the magnitude sweep switches that cancelling off, so when the sweep later makes the soil softer the fit clearly gets better or worse and the data can actually choose.

What happens, substep by substep

1. Score each candidate speed by the **two-corner objective** $J = \tfrac{1}{2}\cdot[\chi^2(\text{low-settlement edge}) + \chi^2(\text{high-settlement edge})]$ — two forward solves per trial, so the winning speed must explain the record across your whole band.
2. Search by **cyclic coordinate descent**: one coordinate at a time (layer 1's k_0, layer 2's k_0, …), and if smear is Variable, its η, r_s, r_t as extra coordinates.
3. For each coordinate, open a *multiplicative* window around its current value (k_0 between ÷10 and ×10, C_k within ±20 %), so the search step is proportional rather than absolute. **Brent's method** then finds the 1-D minimum inside that window in about ten solves. Brent's method combines two ideas. *Golden-section search* keeps three points that bracket the minimum and repeatedly discards the outer third on the worse side, which is slow but can never fail. *Inverse parabolic interpolation* fits a parabola through those three points and jumps straight to its lowest point, which is very fast when the curve is smooth near the bottom — as a χ^2 valley usually is. Brent tries the parabola first and falls back to golden-section whenever the parabola misbehaves, so it is fast in the normal case and safe in the awkward one. It needs no derivatives, which matters here because each score costs a full solver run.
4. Cycle through all coordinates repeatedly until the per-cycle stop fires. Locked layers are never entered.

Stopping criterion. The **per-cycle stop**: after each cycle the fit is re-scored on χ^2_r. The search stops once the relative change $|\chi^2_r(\text{prev}) - \chi^2_r(\text{now})| \div \chi^2_r(\text{prev})$ stays below 8 % for 2 cycles in a row. Two cycles rather than one because a single flat cycle is common when one coordinate happens to be near its optimum while others are not; 8 % because tighter values bought no measurable accuracy while costing many more solves. Both numbers are yours to change — Figure 24.15 shows the panel, and §24.9 explains every field in it — and a hard cycle cap backstops them. Steps 5 and 7 reuse this exact rule and metric.

A note on names. Cycling one coordinate at a time and line-searching each is *block-coordinate descent*. It is often called "Powell" loosely, and some of the tool's own code comments do so, but Powell's method proper also builds conjugate search directions, and this pipeline does not: it searches one named parameter at a time, which keeps every step interpretable. The 1-D searches are genuine Brent.

Exact settings. Brent line-search inside a multiplicative window (k_0 ÷10…×10, C_k ±20 %); per-cycle stop = 8 % for 2 cycles (editable); smear fitted only if Variable; C_k held at $0.5 \cdot e_0$ unless unlocked (the laboratory norm of Tavenas et al., 1983). On screen: "Joint bounds speed fit" with a lock-aware parameter counter — Figure 24.2 shows the overlay during this step.

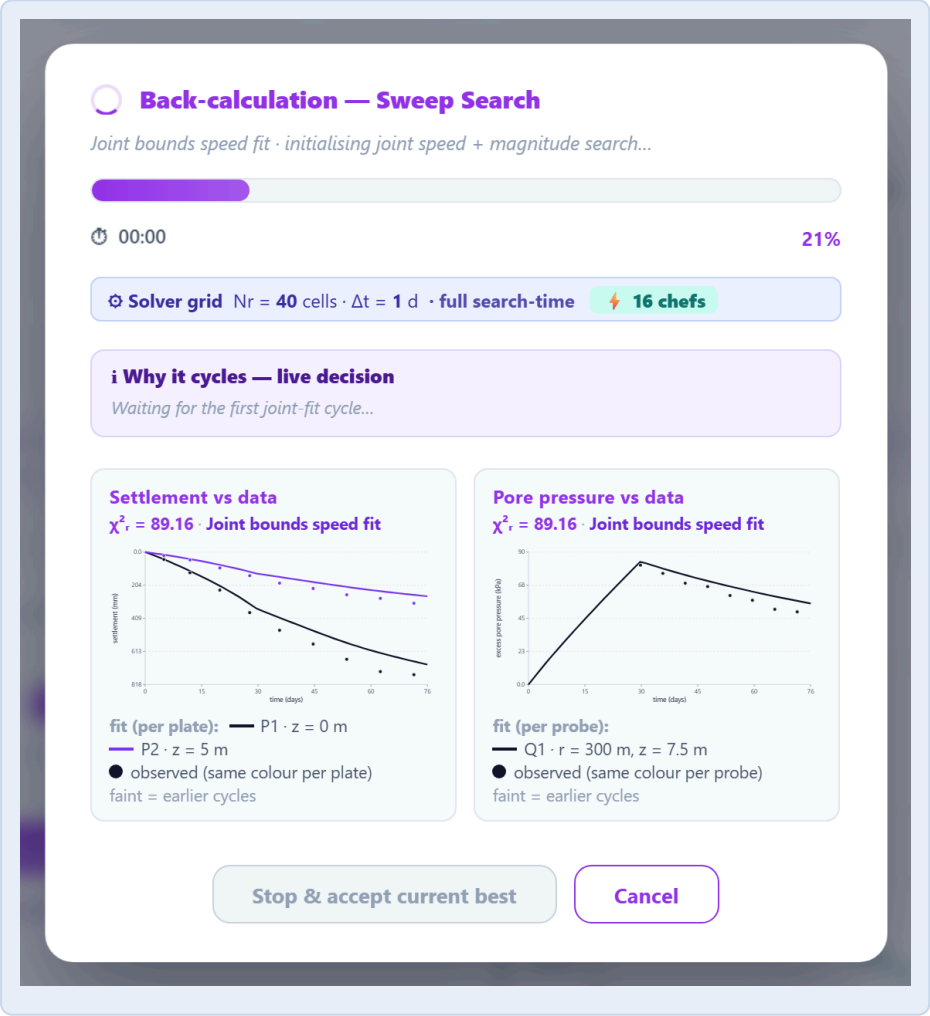


Figure 24.2. A run in progress: the stage name under the title, the shared solver grid and parallel-work chip, two live plots with earlier cycles left faint, and the two exits — *Stop & accept current best* keeps the answer so far; *Cancel* discards it. (Back-calculation, mid-run.)

Step 4 — Three-knob 3-D sweep *try every magnitude recipe on a grid; keep each one's best speed*

IN A NUTSHELL

This is the heart of the method. With the speed pinned down, the tool writes out a grid of soil "recipes" — every sensible combination of how much the soil compresses and where it starts yielding — then solves and scores every single one. It is a systematic taste test of several hundred recipes rather than a hunt for one lucky answer, which is why it cannot walk past a good recipe hiding off to one side.

Hold the per-layer speed profile from Step 3 fixed — the sweep never re-fits an individual layer's k_0 — and try every magnitude recipe on a grid. The only speed freedom is a single narrow global dial (knob S) that scales every Variable layer's k_0 together (a k_0 set Constant, or a drain-off layer, is never scaled) by ×0.79 / 1.00 / 1.26, so the sweep can also ask "was the Step-3 speed level about right?" The reason for allowing even this much freedom is the see-saw: the speed fitted in Step 3 was fitted against the corner magnitudes, so if the winning magnitudes turn out to sit elsewhere, that speed is slightly stale. A narrow three-level scan lets the sweep correct for it without re-opening the trade-off, which a wide speed range would. Then it sharpens around the best region and turns the winner into a continuous answer with a confidence range.

`Receives`	The anchored per-layer speed; the walls; the corner baselines.	`Hands on`	The winning magnitude recipe (+ its best speed level) and a 95 % interval on each knob — that is, the span of knob positions whose fit is statistically indistinguishable from the winner's, so anything inside it is a recipe the record cannot rule out.

Why this step exists. The rate is not frozen outright — the single global speed knob still moves — but no layer may change its rate *individually* any more. With that per-layer freedom removed, making the soil softer now clearly moves the score instead of being silently cancelled out — that is what makes the sweep informative. A grid is used rather than a gradient because a short record leaves the valley floor nearly flat: a gradient sees no downhill there, but a grid cannot get lost and cannot miss a second, separate answer. Figure 24.3 puts the two side by side.

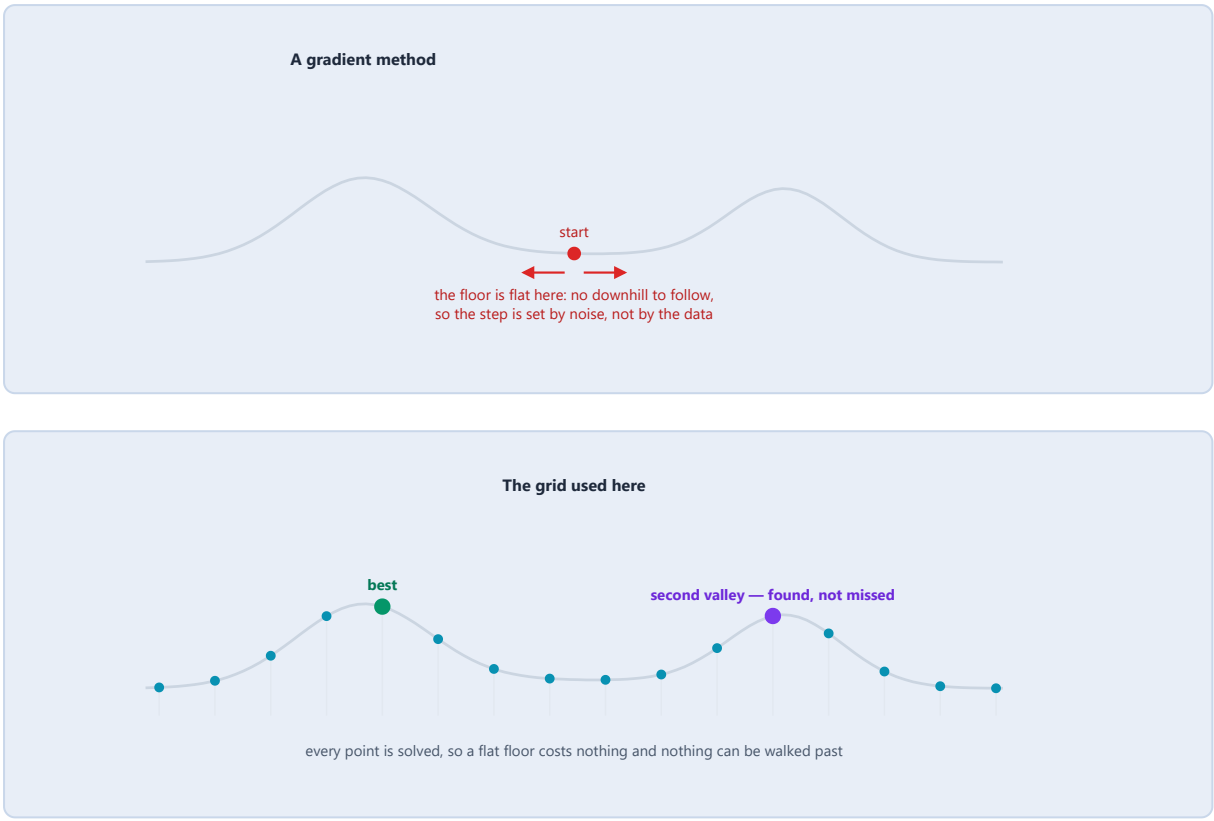


Figure 24.3. Grid versus gradient on one knob: the gradient follows slope, wanders on the flat noisy floor a truncated record leaves, and sees only its own basin; the grid solves every node, cannot get lost, and finds the second valley (violet) of Figure 24.5. (Schematic.)

What happens, substep by substep

4a · Coarse grid. Sample knobs A and B at 13 values each (α = 0, 1/12, …, 1 — so your typed walls at ⅓ = 4/12 and ⅔ = 8/12 are exact nodes) and knob S at 3 levels (α_s = 0.25/0.50/0.75 ⇒ factor = $0.625 \cdot 2.56^{\alpha_s}$ ⇒ k_0 ×0.79/1.00/1.26). That formula is just a way of writing "a multiplier that runs from ×0.625 at $\alpha_s = 0$ to ×1.6 at $\alpha_s = 1$, geometrically, passing through exactly ×1 at the mid-point" — 2.56 is 1.6^2, the total span, and 0.625 is 1/1.6, the low end. The scan therefore asks whether the fitted speed should be nudged down a fifth, left alone, or nudged up a quarter. That is 13×13×3 = 507 forward solves. 1/12 is the coarsest step that both lands exactly on your typed walls and meets the scan's resolution target. Coarser steps that also land on the walls exist — 1/6 and 1/3 — but they sample the plane too thinly to locate a valley; 1/10 is fine enough yet misses the walls altogether.

4b · Min-over-S. (Recall from §22.2 what the symbols mean: α_A is the single dial that moves all the compression magnitudes together, α_B the dial that moves the yield pair, and α_S the global speed dial. Each runs from 0 at the low wall to 1 at the high wall.) Collapse the three speed slices — that is, reduce the three stacked grids to a single one by keeping, for each cell, only its best score: each (α_A, α_B) recipe is credited with the score of its best speed level. This stops a good recipe being punished for a poor speed pairing, and stops one valley appearing three times pretending to be three answers.

4c · Valley + dual-track. Find the deepest cell on that min-over-S surface. If a second, genuinely separate valley (centres ≥ 0.15 apart in the knob plane — nearly two coarse cells) scores within 5 % of the best, the method refuses to choose: it runs the whole remaining pipeline for both and reports them side by side (Track A headline, Track B overlay), as Figure 24.5 shows.

4d · Fine window. Around the winner (and any kept second valley) refine to step 1/48 on a 7×7 window of the two magnitude knobs, plus 3 local speed levels at ±0.125. Nodes already solved are never re-solved, so the whole three-knob scan lands near 650 solves and the winner is located to about ±0.01 α.

4e · Parabolic refine. A grid returns grid values; the true valley bottom sits between nodes. Per knob, fit a parabola through the winner and its two neighbours and take the vertex α* — free, because the three scores already exist. The vertex is clamped to lie between the two neighbouring nodes, and falls back to the sampled winner if the three scores are flat or non-convex. Figure 24.6 draws both the fine window and the parabola with its guards. (A further *best-sampled guard*, which re-scores the vertex and keeps the best real node when the smooth estimate is worse, runs in the Step-6 re-sweep — that is where wall-side parabolas overshoot.)

4f · Data-resolvable range. On each knob, values of α whose score stays below threshold = $\chi^2_{r,\min}\cdot(1 + 3.84/\nu)$ are treated as indistinguishable from the winner (3.84 is the 95 % point of the χ^2 distribution with one degree of freedom — the standard statistical constant behind a 95 % confidence interval, which is why the resulting interval is a 95 % one; ν is the single book-wide definition of §21.2). Starting at the winner, walk outward along the knob until the score crosses that threshold; the two crossing points are the interval, exactly as Figure 24.4 draws it. Keep two distinctions straight when reading it. The 3.84 is a statistical constant — the 95 % point of χ^2 with one unknown — not a number tuned for this problem; and $\chi^2_r \approx 1$ is the *depth* of the valley floor (a fit as good as the noise allows), while the 95 % is about *width* measured from that floor — a flat valley yields a wide interval and a sharp one a narrow interval, from the same rule. One precision: the threshold is applied multiplicatively, $\chi^2_{r,\min}\cdot(1 + 3.84/\nu)$, the profile-likelihood form with the noise variance rescaled so the best fit reads one; it coincides with the additive textbook rule exactly when $\chi^2_{r,\min} = 1$, which the auto-σ convention arranges.

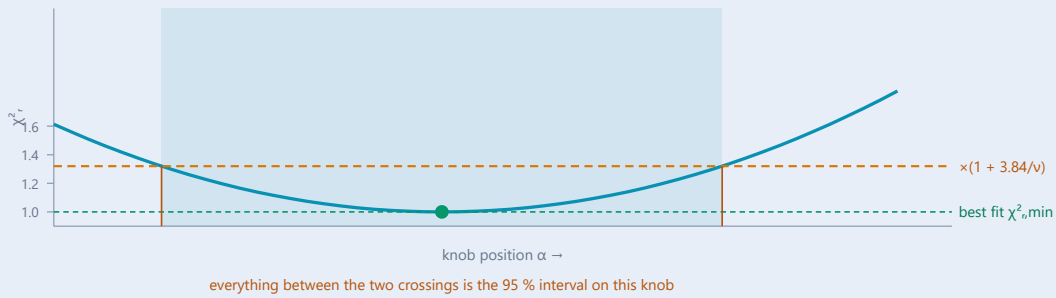


Figure 24.4. The rule drawn: the score along one knob, its minimum at the winner (green), the threshold $\chi^2_{r,\min}\cdot(1 + 3.84/\nu)$ (amber), and the 95 % interval between the crossing points. (Schematic.)

Two floors keep the answer honest. It is never reported narrower than the scan step, because nothing finer than the grid was actually tested, and never narrower than ±0.025 α (±0.0625 on the speed axis), because a interval tighter than that claims a precision this method does not have. *Call this a data-resolvable range, not a calibrated 95 % interval:* the $\Delta\chi^2$ rule assumes a parabolic likelihood and a clean parameter count, both strained here, and the sensitivity study (Chapter 30, §30.5) shows the $\Delta\chi^2$ choice sets much of the band width. Its empirical coverage is measured in §30.3.

4g · Domain extension. If the refined winner lands within the outer 5 % of any axis, the wall, not the data, may be shaping it. That side is extended by 0.5 α (magnitude knobs, capped 2/side) or 0.25 (speed knob, 1/side, with a loud diagnostic), the new strip is scanned, and the pick–refine–interval machinery re-runs.

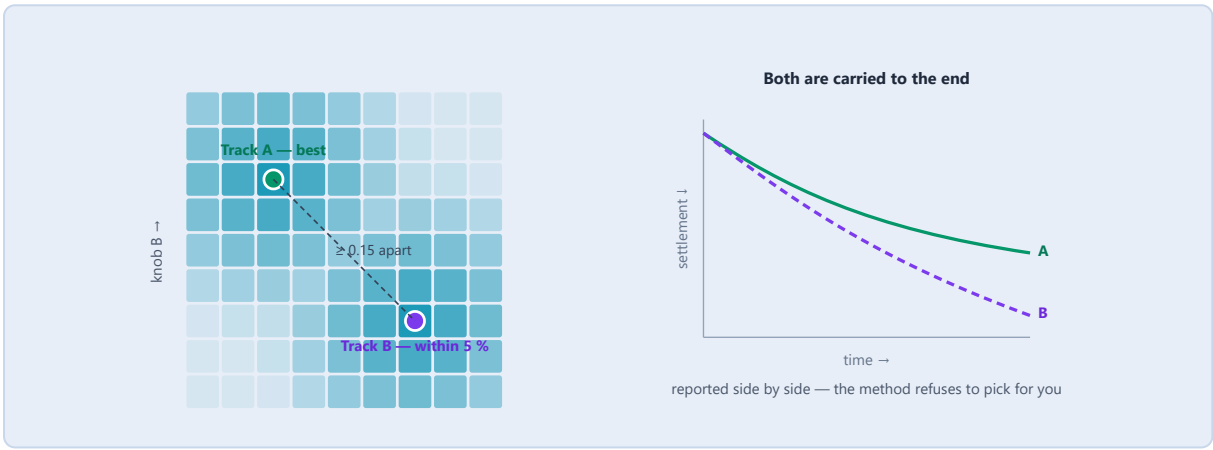


Figure 24.5. The dual-track case: two separate basins fit almost equally well (the second within 5 % of the best, centres ≥ 0.15 apart) — genuinely different soils, so the pipeline runs both and reports Track A as the headline, Track B as an overlay. (Schematic.)

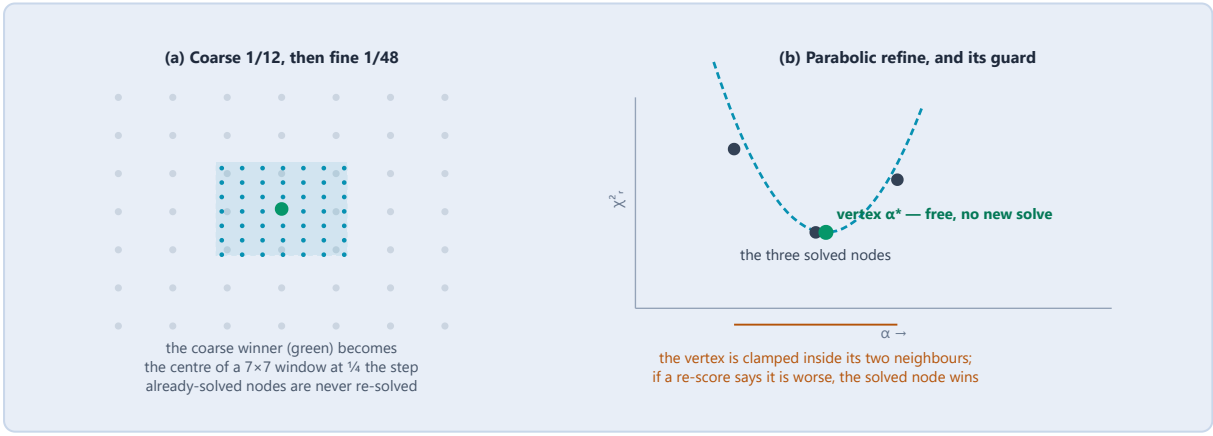


Figure 24.6. Locating the valley bottom. (a) The coarse grid finds the cell; a 7×7 quarter-step window refines it, reusing solved nodes. (b) A parabola through the winner and its neighbours gives the between-node vertex, clamped between them and re-scored where overshoot is likeliest. (Schematic.)

Stopping criterion. No early stop — it finishes the whole grid (coarse 13×13×3, then fine 7×7×3 around the winner). Two rules can add a little work: dual-track keeps a runner-up within 5 %, and domain extension pushes a crowded wall out at most twice per side.

Exact settings. coarse 1/12 (13 pts) × 1/12 (13 pts) × 0.25 (3 speed levels); fine 1/48 (7 pts) × local speed ±0.125; speed factor $0.625 \cdot 2.56^{\alpha_s}$ (×0.625 to ×1.6); dual-track within 5 %, separation 0.15; interval floor max(scan, 0.025 α); extension 0.5 α (cap 2/side) magnitude, 0.25 (cap 1/side) speed. Solver grid 40 / 1 / 0.25.

Figure 24.7 shows the coarse map at the winning speed slice.

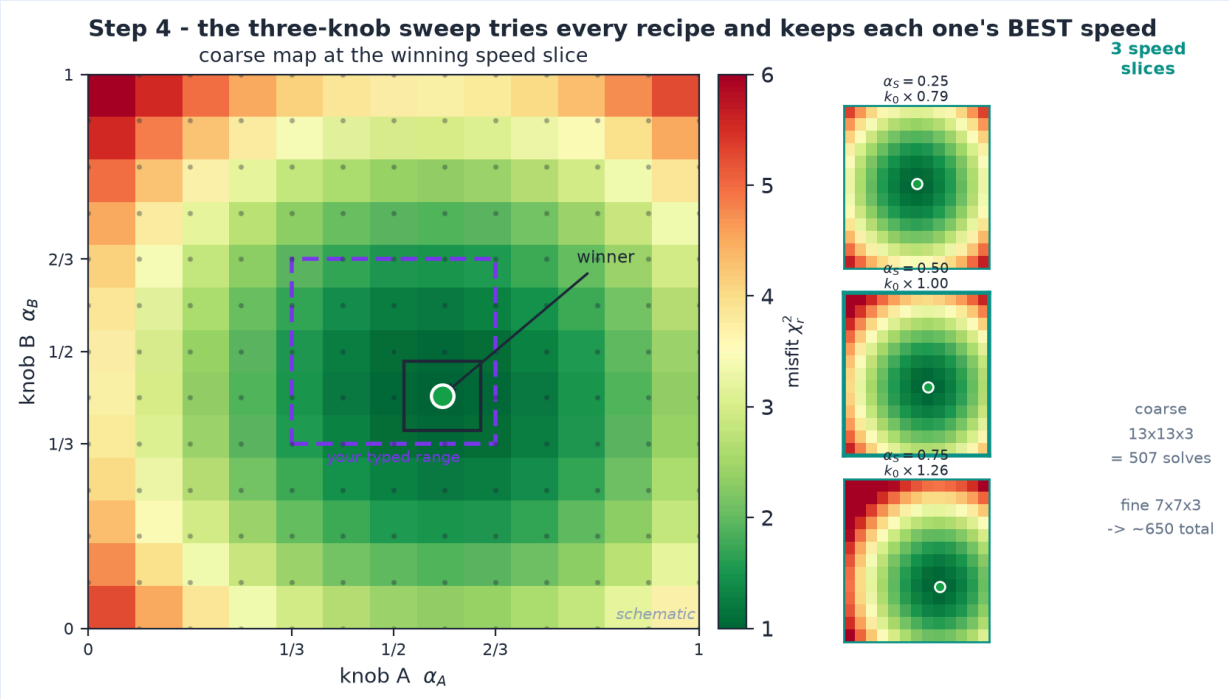


Figure 24.7. The coarse (α_A, α_B) map at the winning speed slice: the typed-range box, the winner, the fine window, and the three speed slices the sweep keeps the best of. (Schematic field.)

Figure 24.8 turns the three grid scores into a continuous minimum; Figure 24.9 reads the 95 % interval off it.

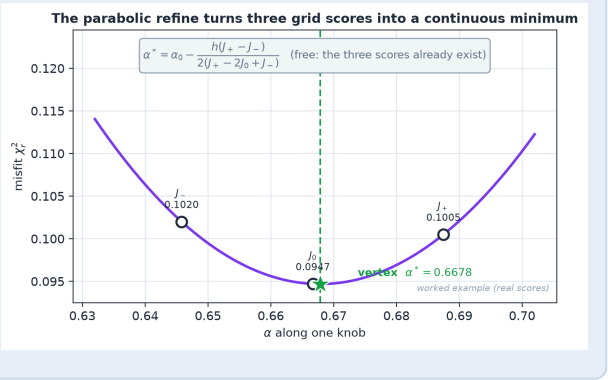


Figure 24.8. Substep 4e — three grid scores become a continuous minimum via the parabola vertex. Worked from real scores; α* = 0.6679, nudged toward the lower neighbour. (Worked example.)

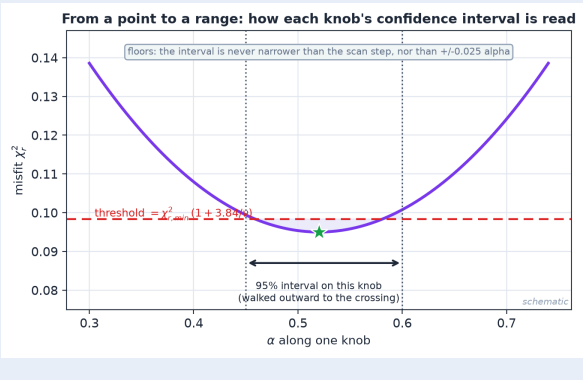


Figure 24.9. Substep 4f — each knob's 95 % interval is read by walking the score curve out to the χ^2 threshold, with honesty floors. (Schematic.)

Step 5 — Best-curve speed refit

sharpen the speed exactly where the answer turned out to be

IN A NUTSHELL

The speed used during the sweep was chosen before anyone knew which recipe would win. Now that the winner is known, the tool fits the speed again, this time at the winning recipe. It is the same idea as setting the oven properly once you have finally decided what you are baking.

Now that the winning magnitudes are known, freeze them and re-fit the speed precisely at that winner.

Receives	The winning magnitude recipe from Step 4.	Hands on	A sharpened per-layer speed (and Variable smear) at the winner.

Why this step exists. The speed from Step 3 was a compromise: it was fitted to sit between the two edges of your band, not to any single recipe. And the sweep only tried three coarse speed levels. Now that the winning magnitudes are known, the speed is re-fitted smoothly and precisely at that winner. Think of it as re-calibrating the ruler once you finally know which recipe you are measuring.

What happens, substep by substep

1. Freeze all magnitude parameters at the Step-4 winner.
2. Re-fit only the speed family (k_0 per layer, plus η, r_s, r_t if Variable), by the same windowed Brent coordinate search as Step 3.
3. Start from the winner's own coarse speed rung, so the continuous refit only travels a short way off the grid.

Stopping criterion. Stops when two more cycles barely help. The rule is the same per-cycle stop as Step 3: χ^2_r changes by less than 8 % between cycles, for 2 cycles running (editable, with a hard cycle cap as backstop).

Exact settings. speed keys only; magnitudes frozen; single-curve χ^2_r objective (the sweep's ruler).

Step 6 — 3-D re-sweep loop (alternation)

re-scan magnitudes AND speed at the fresh speed, until nothing moves

IN A NUTSHELL

If that refitted speed turned out to be noticeably different, then the whole sweep was scored with an out-of-date ruler. So the tool scans again around the winner, at the new speed. It keeps alternating — recipes, then speed, then recipes — until neither one moves any more. Taking turns like this stops the two from chasing each other in circles.

Moving the speed in Step 5 made the sweep's winner — and the band width — out of date. So re-scan a focused window of magnitudes and a local speed axis at the fresh speed, refit the speed again, and keep alternating until the two stop disagreeing.

`Receives`	The sharpened speed and the current winner (+ its intervals).	`Hands on`	A mutually-consistent (magnitude, speed) winner and updated intervals.

Why this step exists. Two things go stale the instant Step 5 moves the speed. First, the **stale ruler**. The sweep scored every recipe, and measured the band width, against the old speed. The winner is therefore judged by an out-of-date ruler. Re-scoring at the fresh speed corrects it: a single valley whose minimum simply slid. Second, and worse, the **wrong basin**: because rate and magnitude trade off, the deeper of two magnitude valleys can swap when the speed changes. A frozen-speed (2-D) re-scan re-scores recipes but stays inside the old valley — it cannot correct basin selection. Only a re-scan that carries a joint speed axis (3-D) can cross the ridge to the true basin. In the controlled test, handed the same stale winner, a 2-D re-sweep finished *unmoved* at 89 mm error while a 3-D re-sweep escaped to 33 mm. That is precisely why this re-sweep is 3-D, not 2-D.

What happens, substep by substep

1. Snapshot the speed (all k_0 and any smear) and compare before vs after Step 5.
2. **Decide whether to re-sweep:** trigger if the speed moved by ≥ 5 % — or ≥ 2.5 % for an off-centre winner ($|\alpha^* - 0.5| > 1/6$). If not, skip to Step 7. The half-size threshold for an off-centre winner is the cautious choice: a winner already leaning against one side of the scanned box has further to fall if the speed has shifted, so it is re-checked on weaker evidence. Both figures are fixed in the code, being convergence criteria rather than modelling choices. (Do not confuse this 5 % with the dual-track trigger of Step 4, which is a different test on a different quantity — that one *is* editable.)
3. If triggered, **3-D re-sweep a focused window at the fresh speed:** the two magnitude knobs over a half-width of max(3/48, 2×the knob's 95 % half-width), widened to ≥ 0.3 for a wall-hugger — in words: look twice as far as the data's own uncertainty in each direction, never less than three fine steps (so a very tight interval still gets a real window), and much wider again if the winner is pressed against a wall and may want to leave the box altogether — and a local speed axis $dS \in \{-0.125, 0, +0.125\}$ (k_0 ×0.89 / 1.0 / 1.12).

4. Adopt the re-sweep's winning local speed, then refit the speed again (a fresh Step 5).
5. Loop back to substep 1. Because each round's local speed axis is narrow, successive rounds can accumulate a larger total correction while no single round is wide enough to re-open the see-saw.
6. Stop when any fires: the winner stops moving ($\Delta\alpha < 0.01$ and $dS \approx 0$); the score plateaus (< 0.5 % gain for two rounds); or the round cap is reached, 6 rounds by default.

Stopping criterion. Three signals, not all χ^2: (1) the winner stops moving, which is a position test; (2) a χ^2 plateau, where χ^2_r improves by < 0.5 % for 2 rounds; or (3) the round cap, 6 by default and adjustable in the Stopping criteria dialogue (§24.9). It also only starts if Step 5 moved the speed by ≥ 5 % (2.5 % off-centre). When every layer's k_0 is Locked the speed axis collapses and the re-sweep reverts to 2-D.

Exact settings. trigger 5 % (2.5 % off-centre); window half-width max(3/48, 2×interval), ≥ 0.3 at a wall; local speed dS = ±0.125; stop on stalled winner / <0.5 % plateau (2 rounds) / cap 6.

Figure 24.10 gives Step 6 as a flowchart.

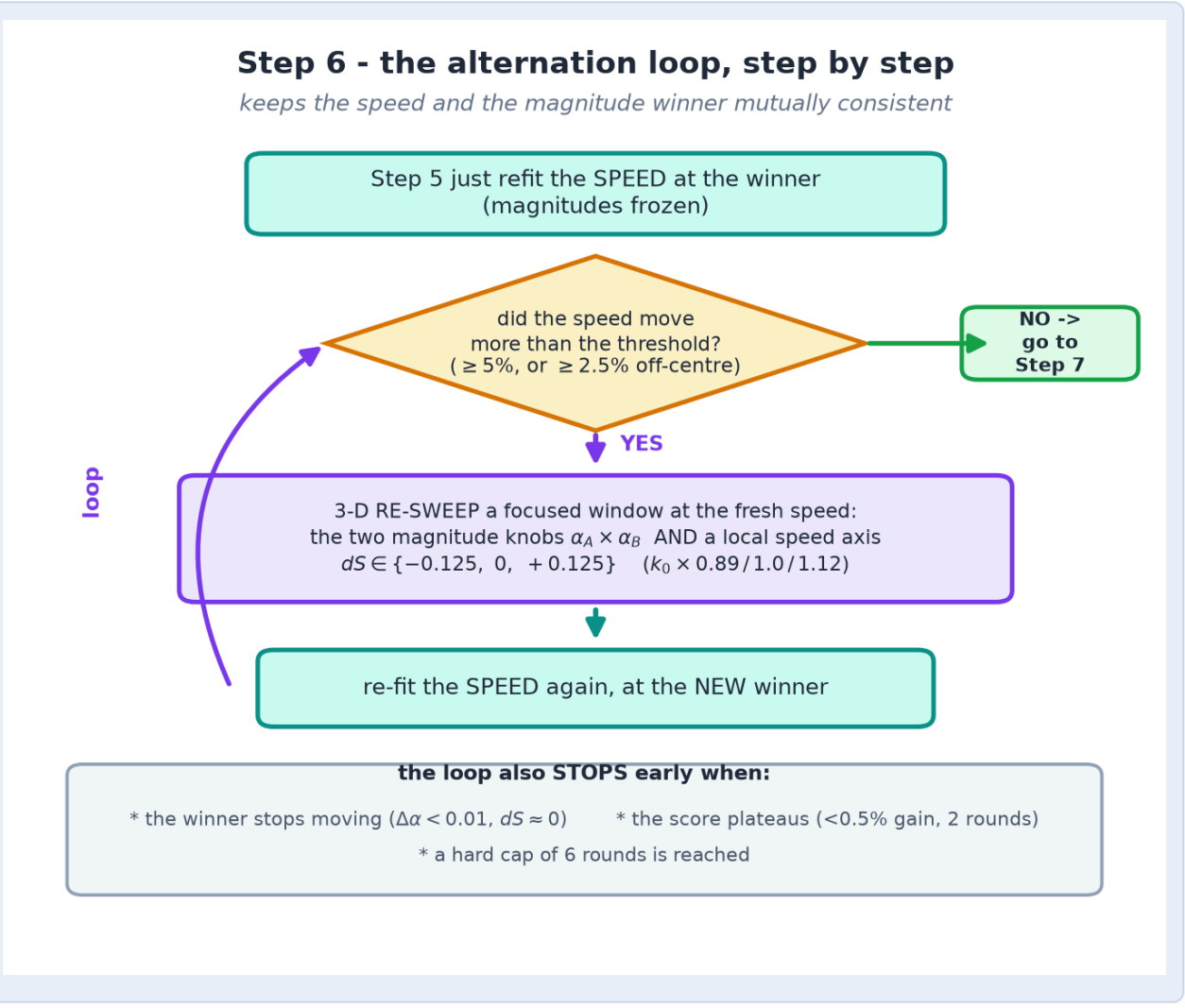


Figure 24.10. Step 6 as a flowchart. Refit speed → did it move? → if yes, 3-D re-sweep at the fresh speed → refit again → loop; stop on convergence or the 6-round cap.

How this re-sweep window differs from Step 4's

Step 4 had no prior, so its window was the whole (α_A, α_B) plane — it gridded everywhere to find where the answer even was. Step 6 is the opposite: it already knows the winner, so its window is a small box centred on that winner, with half-width set by the winner's own 95 % interval — half-width = max(3/48, 2×iv). The box reaches exactly as far as the data's uncertainty, never narrower than three fine scan steps, and is widened to ≥ 0.3 only if the winner is pinned against a wall. The speed axis narrows the same way: from Step 4's wide global scan (k_0 ×0.79–1.26) down to a local nudge (×0.89–1.12), so no single round is wide enough to re-open the trade-off. Figure 24.11 shows the two windows in the same magnitude plane.

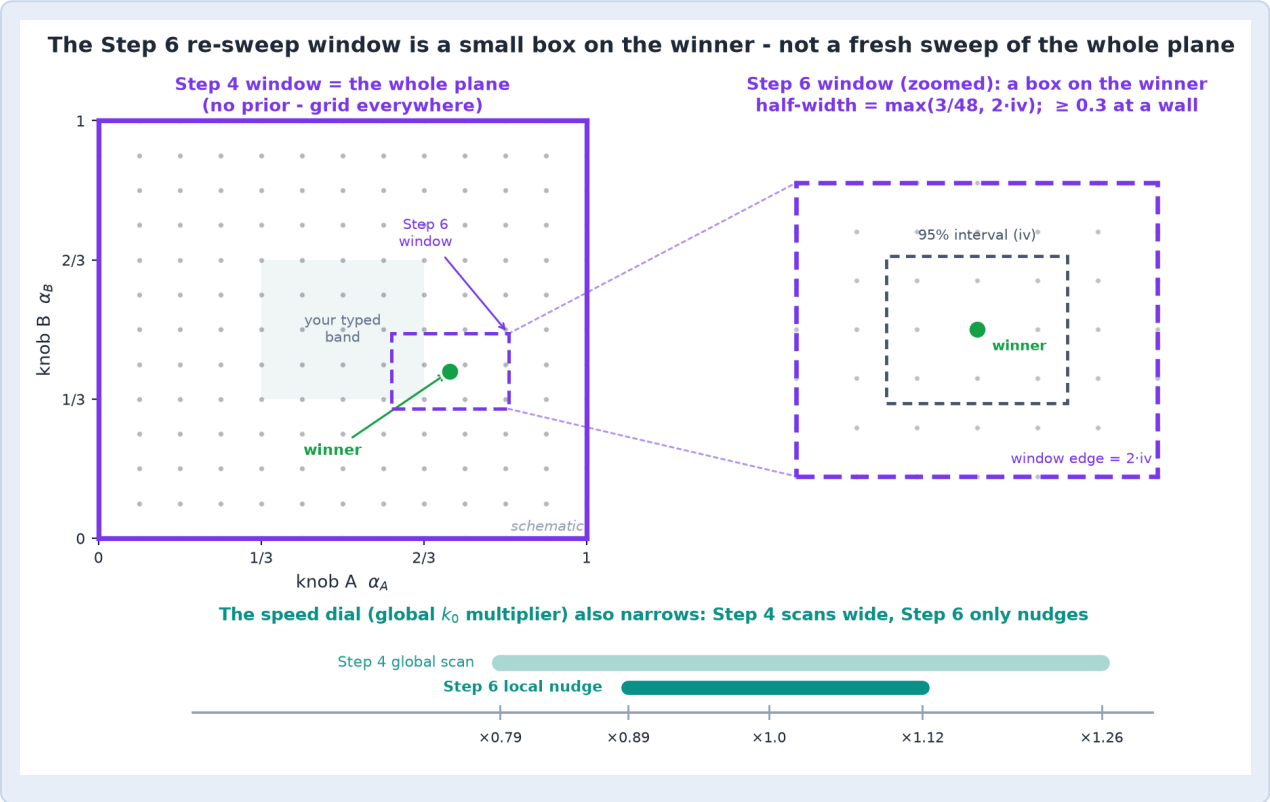


Figure 24.11. The two windows in the magnitude plane: Step 4 sweeps the whole (α_A, α_B) plane; the Step-6 re-sweep is a small box on the winner (half-width 2×iv, floored at 3/48, ≥ 0.3 at a wall); the speed dial narrows from global scan to local nudge. (Schematic.)

Special case: when the winner hugs a wall

Forcing the window wide (≥ 0.3) is only the first move of an escalating sequence the app runs whenever the winner sits within 5 % of a wall. The purpose is to let the answer move *outside* your typed band when the data genuinely demand it — while never letting it run to a physically impossible value.

1. **Wide look.** Set the window half-width to max(2×iv, 0.3) and scan coarse (~7×7×3), centred on the winner. So 0.3 is a floor, not a fixed value.

2. **Chase.** If the best recipe comes back on the window's edge, re-centre on that new best, double the span, and re-scan — follow the winner instead of re-sweeping ground already left.
3. **Move the fence.** If still on the edge, extend the ruler itself: push that wall out by 0.5 α and scan the fresh strip. Repeats up to twice per side (a hard cap of +1.0 α).
4. **Flag.** If the winner is still pinned once that cap is reached, deliver the pinned winner with an honestly wide band, and raise a diagnostic that your typed range or the Step-3 speed anchor needs attention. It deliberately does not chase the winner to a physical extreme.

> **KEY IDEA**
>
> **Does the wall reach the physical limit? No.** The search wall moves in bounded steps (0.5 α, capped at +1.0 α). If the winner is still pinned after that, the run stops and flags it rather than walking to the parameter's physical boundary. The physical boundary is a separate, always-on guard. Every candidate passes through the physical clamps ($C_r \le 0.95 \cdot C_c$, $\beta \in [1,10]$, $C\alpha/C_c \in [0.005,0.2]$, $\sigma'_p \ge \sigma'_0$), so no impossible recipe is ever scored. Stopping short of the physical wall is deliberate: a winner that wants to run to an extreme is a signal that your inputs or the model need a look, so the method reports that finding instead of silently delivering an extreme answer.

Figure 24.12 shows the escalating response.

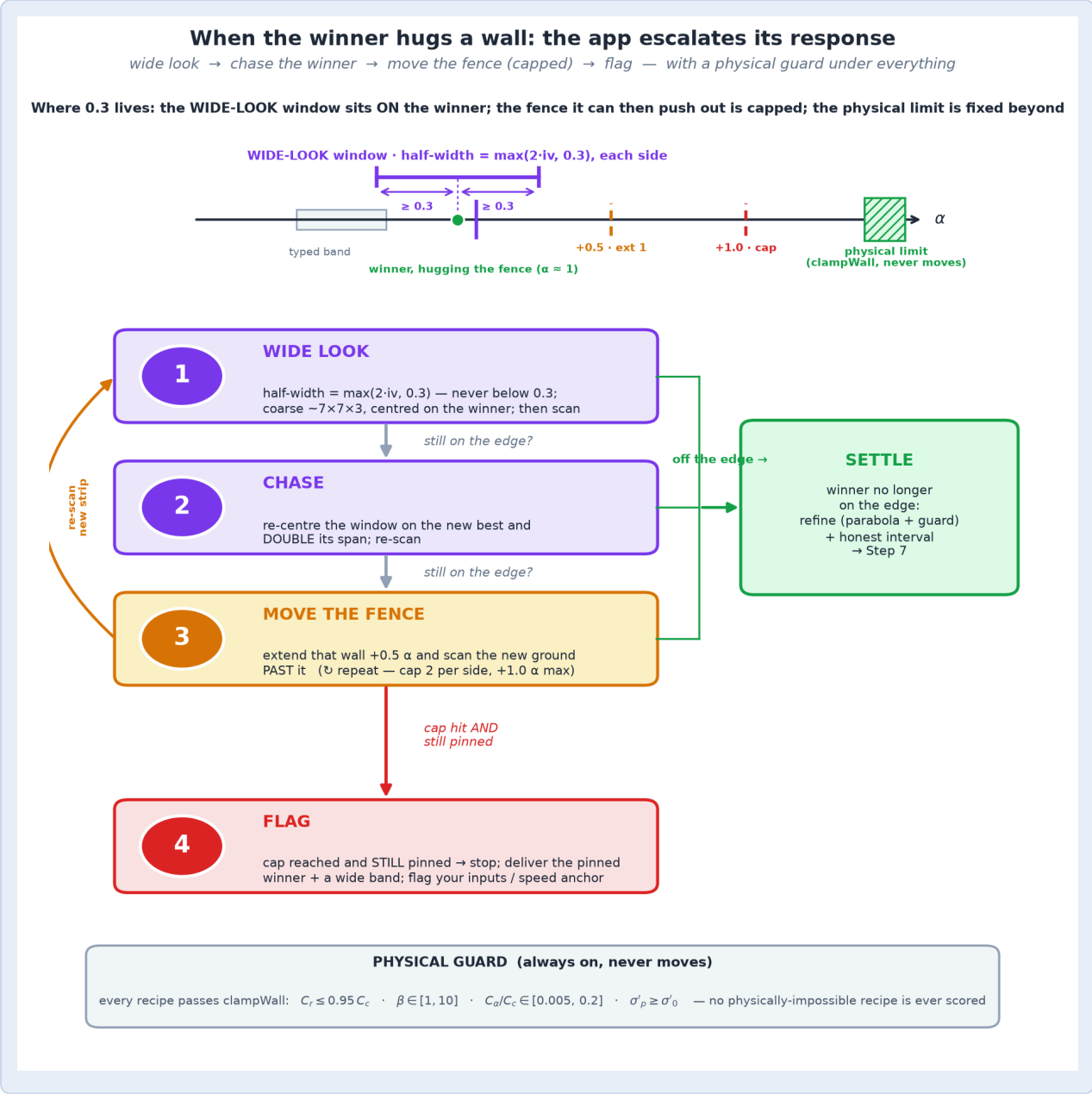


Figure 24.12. The wall-hug ladder: wide look → chase the winner → move the fence (capped at +1.0 α) → flag — all inside a separate, fixed physical guard the search never crosses. (Schematic.)

Step 7 — Per-layer fine-tune + misfit check

polish each layer alone, then locate and rescue a guilty layer

IN A NUTSHELL

Until now every soil layer has been moved together, as a group. Here the tool finally polishes each layer on its own, one parameter at a time. Then it asks where any remaining disagreement with the readings actually sits. If a single layer is responsible, that layer alone is set free and re-fitted, rather than disturbing the ones that were already matching well.

The knobs move all layers together; now free one layer at a time to polish it, then check whether any leftover misfit belongs to a single layer.

`Receives`	The global (all-layers-together) winner.	`Hands on`	A per-layer-polished answer, plus any reported Layer Rescue.

Why this step exists. The knobs move every layer together on purpose, because that is where a short record has power. But real ground varies layer by layer, and only now, with the global picture pinned, is it safe to let each layer flex on its own. It is done last, and on a tight leash, because per-layer freedom is exactly the trade-off the whole method worked to shut down; handed out any earlier, it would undo the anchoring.

What happens, substep by substep

1. Free one layer's six parameters at a time (k_0, C_k, C_c, C_r, σ'_p, β), with the centres allowed to leave the typed band — each cycle may move a *magnitude* by up to ÷1.8…×1.8, compounding over cycles, so a centre can travel well outside the typed walls (k_0 and C_k keep their Step-3 windows of ÷10…×10 and ±20 %), while all other layers stay fixed.
2. Run passes: pass 1 orders layers thickest-first; later passes order by which layer helped most last pass (greedy). A layer that stops contributing is retired.
3. **Misfit localisation.** Settlement plates nest (a plate at 4 m feels everything below it), so differencing adjacent plates isolates each interval's residual. Flag an interval that is *dominant* (> 1.5× the median), *material* (> 15 mm) and *systematic* (|ME| ≥ 0.4×RMSE — a bias, not noise).
4. **Layer Rescue.** If an interval is flagged AND the fit is still poor ($\chi^2_r > 2$) or the winner sat pinned at a wall, re-search the magnitudes of the flagged interval's layers (up to two of them) beyond the typed band. Accept the result only for a clear gain (≥ 15 % in χ^2_r). Whatever it changes is reported in an amber banner — a finding about your input, never a silent fix.

Stopping criterion. Two stops, one inside the other, both on χ^2_r. Inner (one layer): the per-cycle stop of Steps 3 and 5 — χ^2_r changes by < 8 % for 2 cycles. Outer (all layers): the Sweep runs 3 passes by default, and because that cap equals the compulsory minimum before an early stop may fire, the cap is what

normally ends the step. Raise it in the Stopping criteria dialogue and the χ^2_r target ends the step instead; §24.9 sets out both, with the other rules that govern this step. A flat layer is retired; a Layer Rescue is kept only for ≥ 15 % gain.

Exact settings. 3 passes; per-cycle open window ÷1.8…×1.8; rescue accepts only for ≥ 15 % gain; localisation flags need all three of dominant / material / systematic.

Figure 24.13 illustrates the plate-differencing trick behind those flags.

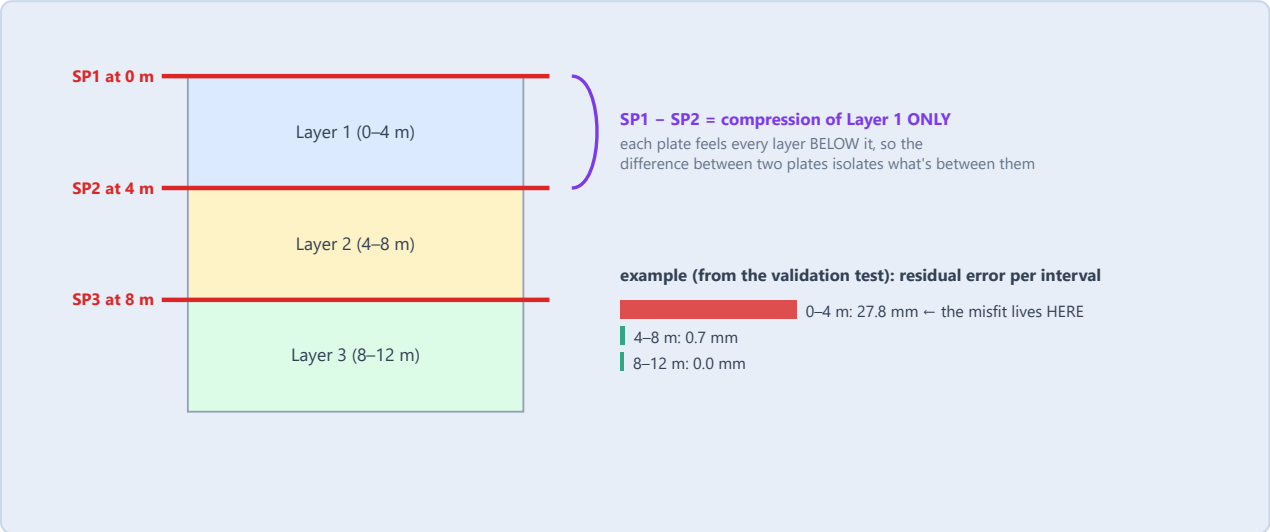


Figure 24.13. The plate-differencing trick: adjacent plates isolate the interval between them. In this real run virtually all the residual (27.8 mm) sits in 0–4 m — the defect is in Layer 1 and nowhere else.

Step 8 — Final ranges — 100-α sweep + P50

turn the winning knob positions into the two reported ranges

IN A NUTSHELL

The answer so far is one best recipe, and a single number would claim a precision the data cannot support. So the tool runs a hundred recipes near the winner, weights each by how well it fits the readings, and reports the middle forecast together with the spread around it. It then builds a second, wider range that also allows for the uncertainty in the numbers you typed — and that wider one is the range to design to.

*Sweep 100 recipes through walls whose width is set by the data, weight each by how well it fits, and read off the median forecast and the 5th-to-95th-percentile range. A second range is then formed around the same winner by drawing **200** Latin-Hypercube samples across the Lo/Hi windows you typed on the soil table, with no fit-weighting — that one is wider, better calibrated, and the one to design to (§30.3). A middle row then re-counts those same samples with each vote discounted by fit, so the lab window narrows only where the record has earned it; its measured reliability is 89–91 % wherever the record reaches half consolidation. The two counts differ because the two jobs differ: the first range only has to trace a band whose width is already fixed by the χ^2 interval, which 100 recipes do smoothly, while the second has to fill a wider space with no such guide and needs more samples before its percentiles stop moving. Both were checked by re-running with more draws and confirming the reported numbers barely shifted.*

`Receives`	The polished winner; each knob's 95 % interval.	`Hands on`	The P50 forecast, both reported ranges, and the measured speed-factor range.

Why this step exists. A single "best" number hides how little a short record can really pin down, so the deliverable is a forecast plus a range. **Read the conditioning carefully:** both ranges are *parametric*. They span parameter uncertainty *within one fixed forward model*: a 2-D axisymmetric unit-cell solver on Hansbo geometry, the C_c/C_r β-smooth compression law, and $C\alpha/C_c$ creep in a Bjerrum isotache formulation. This is the conditioning caveat set out by Beven & Binley (2014). Neither includes constitutive or structural error: a wrong drainage idealisation, unmodelled well resistance, a mis-set layer boundary. They are the ranges you defend *given the model*; the model itself is cross-checked in Chapter 27, and the residual model risk is discussed in Chapter 30.

Two ranges are produced, and they answer different questions. The *data-resolvable* range comes from the $\Delta\chi^2$ region and reports how tightly your settlement record alone pins the answer. The *input-uncertainty* range keeps the same winner at its centre and takes each magnitude's relative width straight from the Lo/Hi you typed on the soil table, applied about the winner, with a fixed ×0.3–×3.0 window on k_0. Its draws are not re-weighted by fit — laboratory scatter does not become less likely because it fits a truncated record worse — so the percentiles are read from the equally-weighted draws. (Until September 2026 the tool instead took these widths from the data-narrowed final walls and applied the first range's fit-weights. Measured on the built-in example, that construction collapsed the range to about a sixth of its validated width, left it *narrower* than the data-resolvable range, and missed the known true settlement; the

construction now shipped is the one the 270-test coverage study below actually validated. Harness: `iub_collapse_probe.js`.) A third row sits between the two — the same draws with fit-discounted votes — described in full at the end of this step. Measured over 270 paired tests the first contained the true settlement 58 % of the time and the second 77 % — 96 % once the record reaches half the settlement (§30.3, §30.7). **Design to the input-uncertainty range.** The data-resolvable range narrows as monitoring accumulates while the forecast error does not, so it stops improving as the record grows — 68 % coverage at half consolidation, 66 % at three quarters. That narrowing is measured over 270 runs in §30.7; the single field case in §30.8 is too small to confirm or refute it.

What happens, substep by substep

1. **Build data-driven walls:** each key's wall = centre·span$^{(\alpha\text{interval} - \alpha^*)}$ (span = exHi/exLo of that key). The result is *asymmetric*, inherited directly from the asymmetric interval. That is the point: the band reports what the data could distinguish, in the direction they could.
2. Sweep 100 recipes through those walls. With a speed band available, which is the normal case, these are **20 uniform-α magnitude blends × 5 speed levels** spanning the α_S interval, so the range carries rate uncertainty as well as magnitude. Only when the speed band is empty does it reduce to 100 points along a single line.
3. Weight each surviving recipe by how well it fits: $w \propto \exp(-\tfrac{1}{2}\cdot\Delta\chi^2_r/T)$, a tempered Boltzmann weight with temperature T = the minimum χ^2_r, floored at 1.
4. Order the survivors by predicted final settlement (strictly, the thickness-weighted mean of the plate settlements at the final time). Then read the cumulative-weight crossings at 5 %, 50 % and 95 % to get P5, P50 and P95. (Two different numbers live here, so they are named apart. The band edges P5 and P95 are cumulative-weight crossings of the ordered survivors, and so is the crossing median. The figure the result page prints as the headline is not that median but the *best estimate*: the single best-fitting survivor, which is always a real admissible run. On a skewed weight vector the two differ, and Chapter X's limitation about the median collapsing onto the lower edge is about the crossing median, not the best estimate.)

Stopping criterion. None to converge — a fixed 100-recipe α-sweep through the data-driven walls, run to completion, then the weighted percentiles are read off. The sample count is the only knob. In Quick mode, the mode the Sweep runs in, it is fixed at 100; it becomes editable only in Accurate mode.

Exact settings. Data-resolvable range: 100 blends; tempered Boltzmann weights; percentiles by predicted final settlement; data-driven walls per key. Input-uncertainty range: 200 Latin-Hypercube draws by default — engineer-editable (50–5000) as *Uncertainty-range recipes* on the Inversion search settings card — log-uniform per layer, spanning the typed Lo/Hi per magnitude (relative width applied about the winner), a fixed ×0.3–×3.0 k_0 window, and plain unweighted percentiles. Provisional middle row: the same 200 draws re-counted with the tempered weights $w = e^{-\Delta\chi^2_r/2T}$, T = max(1, the winner's reduced χ^2); effective sample size logged to the console; reliability unmeasured pending the next coverage round.

Figure 24.14 shows the delivered product, and Table 25.1 lists the per-plate statistics and their acceptance gates.

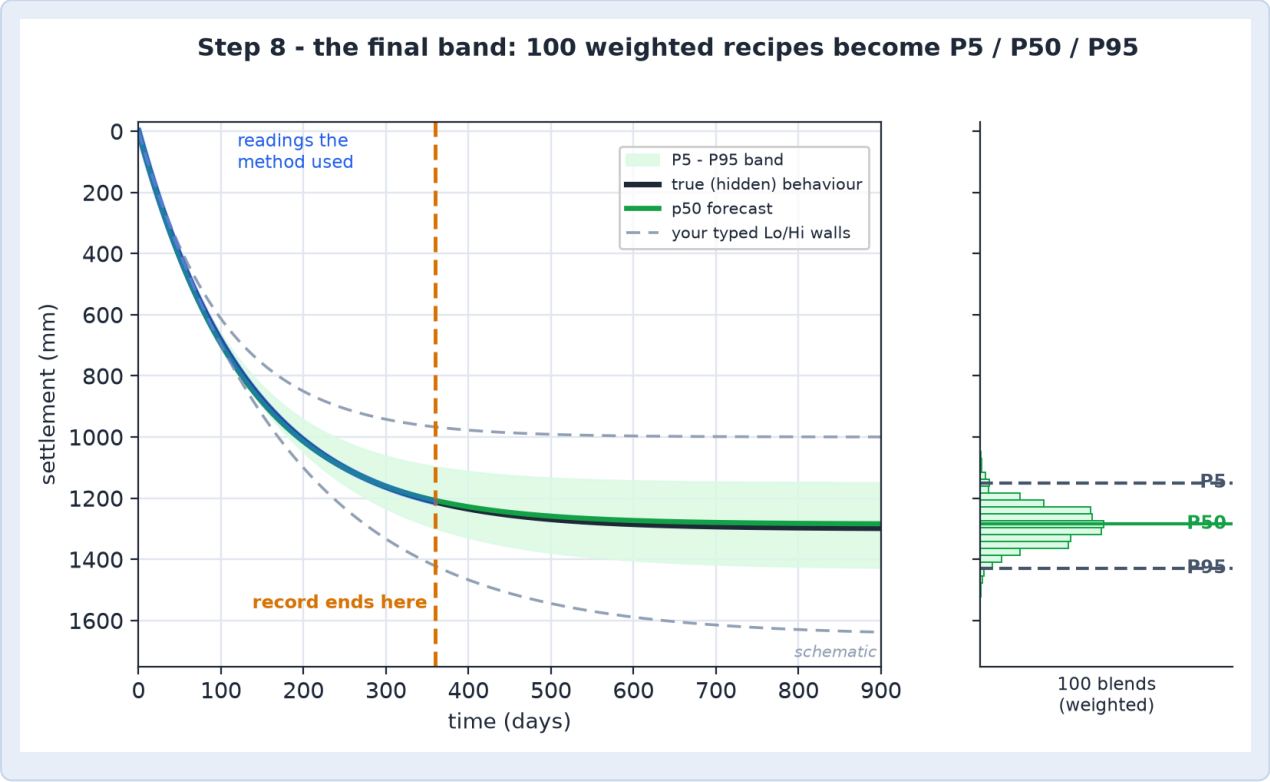


Figure 24.14. The delivered product: the P50 forecast with the data-resolvable P5–P95 range (the 100 weighted blend endpoints read as percentiles); the second range keeps this centre, takes its width from the typed Lo/Hi, and is the one to design to (§30.3). (Schematic.)

The middle row — your lab window, narrowed by the fit

The equal-vote design range above answers the engineer's uncertainty honestly, but it leaves a fair objection standing: Steps 1–7 do a great deal of fitting work, and all of it lands in the *centre* of the band — none of it in the *width*. The forecast card's middle row, *Narrowed by the fit*, added in September 2026 and since promoted from its provisional launch label, is the answer to that objection. It uses the **same 200 recipes** as the design range — no new solves, no new walls — and changes only how the votes are counted: each recipe's vote is discounted by how well that recipe fits your monitoring readings.

The misfit score and the gap. Every recipe is solved forward and compared against every reading. For each reading, take (predicted − measured) divided by the expected noise on that reading, and square it; average these over all readings, with the plates and the piezometers each balanced to a fair share. That average — over the book-wide ν of §21.2 — is the recipe's misfit score: 0 would be a perfect match, and 1 means "off by about one noise-width per reading". Every recipe's **gap** Δ is its own score minus the best score among the 200 — an arithmetic convenience only, since the vote *shares* are indifferent to where the zero of the gap sits; the number that matters, T, is anchored to the winner below. Dividing by the noise first is what lets a plate in millimetres and a piezometer in kilopascals contribute to one fair score.

$$\text{vote} = e^{-\Delta/2T} \quad \text{with} \quad T = \max(1, \text{the Steps 1–7 winner's score}) \tag{24.1}$$

Reading the formula (T = 1 for a well-fitting run): the best recipe keeps a full vote; a recipe one unit worse keeps $e^{-0.5} \approx 0.61$ of a vote; four units worse keeps $e^{-2} \approx 0.14$; nine units worse keeps $e^{-4.5} \approx 0.01$. Figure 24.14a draws the fade. Nothing is ever thrown away — a poor fit fades rather than being cut, so there is no cliff where two nearly identical recipes get opposite treatment, which is exactly the arbitrariness the $\Delta\chi^2$ wall of the data-resolvable range cannot avoid.

Recipe	Gap Δ above best	Vote kept (T = 1)
the best recipe (the yardstick)	0	1.00 — full vote
fits almost as well	1	$e^{-0.5} \approx 0.61$
clearly worse	4	$e^{-2} \approx 0.14$
badly contradicted by the readings	9	$e^{-4.5} \approx 0.01$

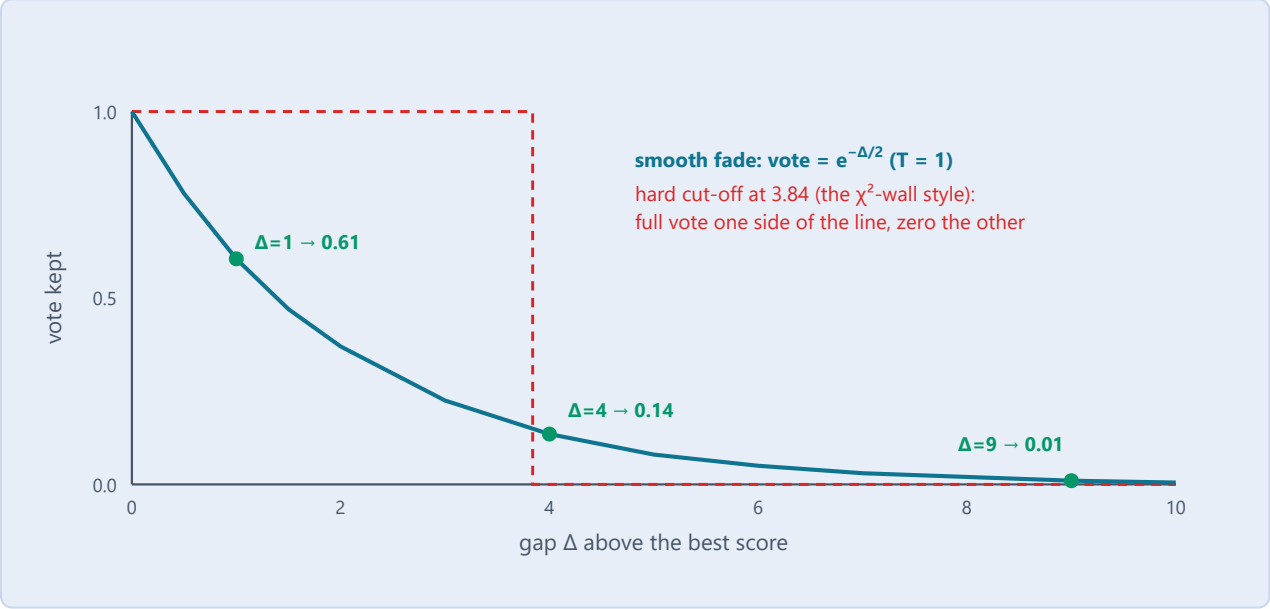


Figure 24.14a. A recipe's vote against its gap above the best score: a smooth fade with no cliff, against the hard cut-off (dashed) of the data-resolvable style, where gaps of 3.83 and 3.85 get opposite treatment.

T is the mercy factor, and it is the honest part. When even the best recipe fits poorly (score well above 1), the mismatch is bigger than reading noise alone can explain — the model or the noise estimate is off — and harsh discounting on a ruler known to be misjudged would collapse the band falsely. So T is set to the *winner's* own score, floored at 1 — an engineer's refinement of September 2026: the deep search of Steps 1–7 sets the standard, not the luck of the copies. Since the copies are perturbations of the winner, virtually all of them fit worse than it, so this anchoring lowers T and *sharpens* the discounting — measured on the built-in example it narrowed the row from 610–875 to 669–875 mm. The tempering still works as intended: the rougher the winner's own fit, the gentler the discounting. (And if a random copy ever beats the winner by a material margin, the tool prints an under-convergence warning — a 30-second nudge outperforming the deep search means the search

stopped short.) When the fit is good, T = 1 and the discounting bites at full strength. T is not a free styling knob — it is the single number the scheduled calibration study will check and, if needed, adjust by measurement.

Counting the votes. Line the 200 predictions up from smallest to largest, walk along adding up votes, and mark where the running total passes 5 % and 95 % of all votes — the same crossing rule every band in this step uses. Faded recipes barely push the markers; full-vote recipes push them hard. The result, by construction, sits between the two existing rows: it can never be wider than the equal-vote design range (discounting only removes influence), and it approaches the data-resolvable region only when the record genuinely distinguishes the recipes. Figure 24.14b shows all three rows on the built-in example, where the truth is known: best estimate 733 mm, design range 598–880, fit-narrowed row 669–875, data-resolvable 697–776 against a true 852.

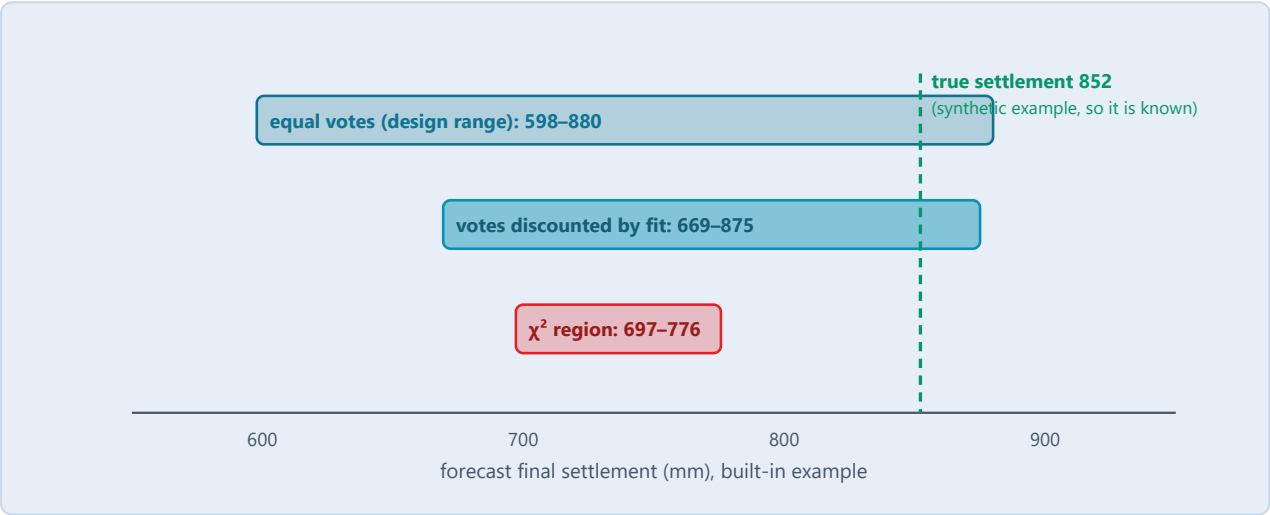


Figure 24.14b. The three rows on the built-in example (72-day record; winner-anchored T ≈ 2.2): asymmetric narrowing, 669–875 against 598–880, pruning the light-settlement edge the record testifies against. The χ^2 region excludes the known truth; both wider rows contain it. (This run's own diagnostics: the status paragraph below.)

How it narrows here — and when it will narrow more. On this example the record is only 72 days long, about half the settlement, and the winner's fit is still rough (T ≈ 2.2), so the discounting is deliberately tempered. Even so the row narrows from 598–880 to 669–875 mm, and the *shape* of that narrowing is the instructive part: the record testifies against the light-settlement recipes and prunes the low edge, while it cannot yet distinguish the heavy end, which barely moves. That selectivity is the truthful message: *the data narrow only what they have seen*. As monitoring accumulates the picture should invert, in the pattern of Figure 24.14c: the discounting sharpens, and the row closes in — but only along the directions the record can actually see, staying lab-wide where it is blind. That selective narrowing is what neither existing row can do: the χ^2 region narrows everywhere (including where it should not — its measured 58 % coverage), and the design range narrows nowhere.

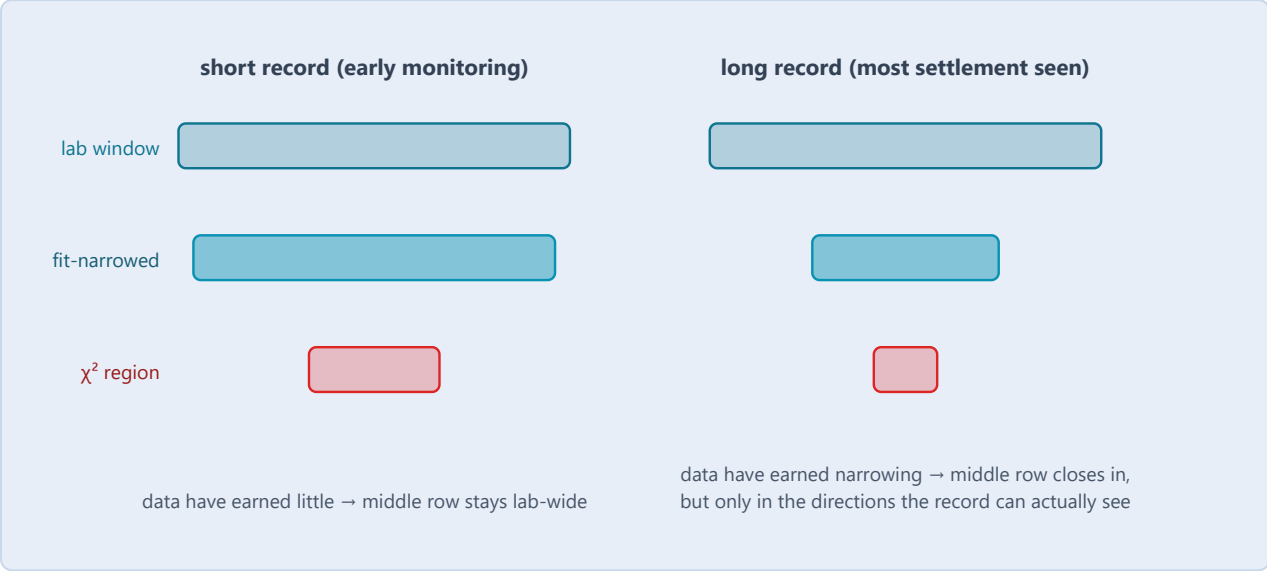


Figure 24.14c. As monitoring accumulates (schematic): the lab window never shrinks; the χ^2 region shrinks whether or not the forecast improves (its measured defect); the fit-narrowed row shrinks only where the record earns it, staying lab-wide in the blind directions.

Status — measured, and promoted. The middle row's reliability has now been measured twice, and the provisional label has been retired. The September 2026 coverage round (315 cases, the same paired star design as the original study) first measured it at 76 % pooled against a nominal 90 % — an apparent failure that a pre-registered retry then diagnosed precisely: re-running all 315 cases with 1,000 copies instead of 200, and re-counting the same draws under four discounting temperatures, showed the shortfall was *not* the discounting at all. With records reaching half consolidation or more the row covers **89–91 %** at every temperature tested — inside the design range's own confidence interval — while below half consolidation every construction alike collapses to 48 %, because there the winner itself is mislocated and no width recovers a wrong centre. Under the pre-registered promotion rule (a temperature qualifies only if it matches the design range's coverage in every truncation; the narrowest qualifying setting wins) the shipped winner-anchored setting qualified and was retained. The card now prints the measured label: 89–91 % with data to half consolidation or more, 48 % below half — like every range. Four diagnostics ship on the card itself: how far your record reaches up the predicted settlement path (the number the "below half, trust nothing" caveat turns on); T with its anchoring; the effective number of recipes after discounting, with a bold warning below about fifty; and a split-half stability stamp — each range rebuilt from either half of the copies, with the edge disagreement printed in millimetres, so "are 200 draws enough?" is answered per run rather than by convention. On the built-in example these answer differently for the two rows: the design range is stable to 22 mm (ample), while the middle row's discounting concentrates on roughly five effective recipes and its halves disagree by 65 mm — there, raise the draw count (the *Uncertainty-range recipes* box on the Inversion search settings card, default 200, up to 5000) before leaning on its edges. The retry established the division of labour these diagnostics imply: the draw count governs the *precision* of the edges (at 1,000 copies the effective count rose to several hundred and the split halves agreed within about 1 % of the settlement), while coverage was unchanged — the warning was about sharpness, never about validity.

What Steps 1–7 buy — the engineered search, against the sampling alternative

The middle row makes it fair to restate what the deep search actually contributes, because "Steps 1–7 only find the centre" understates it. Every row of the forecast card inherits something from them:

What Steps 1–7 deliver	Where it lands on the card
The **centre** — the winner recipe	every row is drawn around it
The **time-path** — the fitted speed family (k_0, C_k, and the smear parameters when set Variable)	the forecast *curve*: when the settlement arrives, not only how much — the thing staging and surcharge-removal decisions actually read
The **standard of comparison** — the winner's own fit quality anchors the mercy factor T	the middle row's discounting reflects the deep search, not the luck of the copies
The **χ^2 region** the search resolves	the bottom row — the honest diagnostic of what the record alone can pin down
The **diagnostics** — misfit localisation and layer rescue, the bound-hit flag, the observability holds, the dual-grid check	the banners and flags around the card — each exists because a stage of the search measured something
The **engineered widening** — the staged ×3.0 → ×5.1 → ×8.0 wall expansion with its amber banner	the search can leave a mistyped window, but deliberately and on the record — never by silent luck

It is also fair to set this architecture against the main alternative, sampling-based Bayesian inversion (Markov-chain Monte Carlo and its structural-reliability variants), which attacks the same problem by exploring parameter space with a guided random walk. The comparison is a genuine trade, not a rout, and both directions of it deserve stating:

	Sampling-based Bayesian inversion	This tool: engineered sweep + measured calibration
Search	guided random exploration; typically thousands of forward solves; convergence must be diagnosed, and an unconverged chain can look converged	deterministic stages along directions engineered from the physics (the speed/magnitude split, the see-saw); a fixed, inspectable solve budget; the identical answer on every run
Band width	the joint posterior — exact *if* the assumed noise model and priors hold	ranges labelled with coverage *measured* in paired synthetic studies (Chapter 30) and printed beside each row
When assumptions fail	the posterior can be confidently wrong, and nothing inside the method says so	failures surface as measured coverage misses — exactly how the construction drift documented earlier in this step was caught
Randomness	inside the search itself	confined to the propagation step, stratified and seeded — reproducible to the last digit
What it does better	the full joint posterior, with parameter correlations; principled updating of priors	solve-budget economy on an expensive forward model; an audit trail an engineer can read stage by stage

Read the last row of that table both ways. A well-tuned sampler recovers correlation structure that three engineered knobs cannot, and nothing here claims statistical superiority over a correctly specified Bayesian analysis; the method's lineage and its relation to that school are set out in §30.2. The claim is narrower and, on a construction programme, more useful: every number this pipeline reports is either deterministic or seeded; every stage has a stated purpose, budget and stopping rule; and every range carries a reliability that was *measured against known truths rather than asserted from assumptions*. A contract that can be falsified by measurement is also one that can be repaired by measurement — the September 2026 history of this very step is the proof. (§30.2 sets this argument out in full.)

24.9 When each step stops, and how to change it

Each of the eight steps has to decide when it is finished. Those decisions set how long a run takes and how hard the search works before it accepts an answer. Each step's own *stopping criterion* paragraph above states the rule that governs it; this section gathers all of them in one place and shows where each one lives in the application, because until recently most of them were constants in the code that you could neither see nor change.

They are now controls. The **Stopping criteria** button on the back-analysis input page opens the dialogue of Figure 24.15, and every value in it is the one the search already used, so opening the dialogue and closing it again changes nothing. Every criterion there has an ⓘ beside it that explains, in plain words and without any algebra, what that number does. Every worked example in this book, and all 1080 sites of the validation campaign in Chapter 32, were run at these defaults.

Two honest caveats before the table. First, **not every step ends on a convergence test**: two are single reference solves with nothing to converge, one runs a grid to completion, and one draws a fixed number of recipes. Three iterate and stop when they stop improving. Second, a stopping rule is not a quality setting. Loosening one ends a step sooner and can end it *before* the best available fit; tightening one spends solves chasing gains that may be smaller than your measurement error. Neither is free. With that said, Table 24.1 sets out all eight steps, what actually ends each one, and the control that governs it.

Table 24.1. What ends each step, and the control that governs it. Values in brackets are the shipped defaults, which are also the values every result in this book was produced with.

Step	What ends it	Control (default)
1 · Lo magnitude wall	Nothing to converge — one forward solve at the low wall.	*none*
2 · Hi magnitude wall	Nothing to converge — one forward solve at the high wall.	*none*
3 · Joint speed fit	Repeats short cycles until one more cycle barely helps: $\Delta\chi^2_r$ below the threshold for the stated number of cycles in a row, with a hard cycle cap as backstop.	**Per-cycle stop** (8 % for 2 cycles)
4 · Three-knob sweep	No early stop — the whole grid is solved, coarse then fine around the winner. A runner-up valley within the dual-track margin is carried forward as a second track.	grid size under ⚙ **Advanced solving**; **Dual-track** (5 % of the best χ^2_r)
5 · Best-curve speed refit	The same per-cycle rule as Step 3, on the winning curve.	**Per-cycle stop** (shared with Step 3)
6 · Re-sweep	Whichever comes first: the winner stops moving, the χ^2_r plateaus, or the round cap is reached. It only starts at all if Step 5 moved the speed by at least 5 %.	**Step 6 — re-sweep rounds** (6)
7 · Per-layer fine-tune	Two nested loops. *Inner* (one layer): the per-cycle rule of Steps 3 and 5. *Outer* (all layers): the pass cap — or earlier, once the compulsory passes are done, if χ^2_r reaches the good-enough target or improves by less than the per-layer threshold across the plateau window. Layers that go quiet are retired from later passes along the way.	**Step 7 — per-layer passes at most** (3) **Good-enough fit** (χ^2_r, 1.0) **Per-layer stop** (10 % after 3 compulsory passes) **Plateau window** (2 passes) **Step 7 — retire a layer below** (0.01 % for 3 passes)
8 · Final band and forecast	Nothing to converge — a fixed 100-blend α-sweep runs to completion, then the uncertainty ranges draw their recipe copies and the weighted percentiles are read off. If too few recipes carry effective weight, the draw count escalates once, five-fold; if it is still too few, the fit-narrowed row is withheld rather than printed.	**Step 8 — effective-recipe floor** (10) draw count (200), beside the method note on the input page
the whole search	All eight steps complete — or you end it yourself with *Stop & accept* on the run overlay, which keeps the best answer found so far.	**Whole-search budget** (0, meaning no limit)

COMMONLY ADJUSTED

Good-enough fit — χ^2_r target	1	χ^2_r · default 1.0
Plateau window	2	passes · default 2
Whole-search budget	0	minutes (0 = off) · default 0

PER STEP

Step 6 — re-sweep rounds	6	rounds · default 6
Step 7 — per-layer passes at most	3	passes · default 3
Step 7 — retire a layer below	0.01	% of χ^2_r · default 0.01
Step 7 — ...for this many passes	3	passes · default 3
Step 8 — effective-recipe floor	10	recipes · default 10

WHEN IS A PLATE "DONE"? (ALL FOUR MUST PASS)

R^2 at least	0.98	– · default 0.98
KGE at least	0.95	– · default 0.95
NRMSE at most	3	% · default 3
Absolute RMSE at most	20	mm · default 20

CYCLE AND PASS RULES

Per-cycle stop · $\Delta\chi^2_r$ < 8 % for 2 cyc

Per-layer stop · $\Delta\chi^2_r$ < 10 % after compulsory 3 passes

Dual-track if 2nd winner within 5 % of best χ^2_r

Every value in this dialogue defaults to what the search used before these controls existed, so a run you have not touched behaves exactly as it always did. Looser criteria finish sooner and can stop short of the best fit; tighter ones keep refining for gains that may be smaller than your measurement error.

Figure 24.15. The **Stopping criteria** dialogue, at its defaults. The four groups are the three criteria worth adjusting most often, the per-step rules, the four gates that decide when a settlement plate counts as fitted, and the cycle and pass rules shared between steps. Each criterion carries two kinds of help, answering different questions: the ⓘ explains in plain words *what the criterion is*, and hovering the label itself gives the case-based guidance on *when to change it*. The note at the foot is the trade-off in one sentence.

Why the good-enough target deserves a paragraph

Of all of these, the **good-enough fit** target carries a principle rather than a preference. A reduced χ^2 near 1 means the model misses your readings by about one measurement error, which is as well as any model can honestly do. A fit *better* than that is not more truthful, it is fitting your noise, and it forecasts worse. That is why the default is 1.0 and why the per-layer step is allowed to stop the moment it gets there.

Change it when you have reason to believe your stated precision is wrong. If the plate readings carry a settling benchmark or a re-referenced survey, their real error is larger than the σ you typed, $\chi^2_r = 1$ becomes unreachable, and the search will refine towards it for nothing: 1.5 to 2 is the honest target then. If the record is genuinely clean and you want the last available gain, 0.7 is defensible. The fit statistics of §25.3 are the evidence for which of those two you are in.

> **WATCH OUT**
>
> **At the defaults, the target never gets to fire in Step 7.** The pass cap is 3 and the compulsory minimum before any early stop may fire is also 3, so the cap ends the step and the good-enough target and plateau window are carried without being used. That is deliberate — three passes is the measured budget the campaign of Chapter 32 ran at — but it means the target only becomes the deciding rule if you raise the cap. Raise it to 5 or 6 and the step stops as soon as the fit is as good as the data allow, falling back on the cap only if it never gets there.

Choosing values for the case in front of you

The defaults suit most records, so the useful question is not “what should these be?” but “is my case one of the few that wants something else?”. Table 24.2 collects the cases that are, each with the one control to move and the reason it is the right one.

Table 24.2. When a case is a reason to change a stopping criterion. Every row below is reachable from the single dialogue shown in Figure 24.15 above.

Your situation	What to change	Why
The run is taking longer than you have	**Whole-search budget**, set to the minutes you can spare	The search stops at the budget and keeps the best answer it has, by exactly the path *Stop & accept* takes. An answer from an interrupted search is a real answer, only less refined, and the result card reports how far the search got.
You want the fit, not the pass count, to decide when Step 7 is done	**Step 7 — per-layer passes at most**, raised to 5–6	It frees the good-enough target and the plateau window to end the step, as the watch-out above explains. Expect a longer run on a record that never reaches $\chi^2_r = 1$.
A thick profile where only one or two layers are really observed	**Step 7 — retire a layer below**, raised to 0.1 %	Layers contributing almost nothing are dropped from later passes sooner, so the solves go where your record actually carries information. The observability panel of §25.8 tells you whether you are in this case.
The laboratory k_0 is badly off, and the log shows the speed still moving at the last round	**Step 6 — re-sweep rounds**, raised above 6	A strongly biased starting conductivity can take several rounds to walk in. When the speed is still moving at the cap, it is the cap ending the step, not convergence.
You intend to quote the fit-narrowed row of the forecast range in a report	**Step 8 — effective-recipe floor**, raised to 20–30	That row's edges are set by the recipes carrying real weight, so a handful of them makes the edges noisy. The floor is what turns that into an escalation and then a withheld row rather than a confident-looking number — the behaviour §25.6 describes.
A short or scattered record where the search never finishes a plate	The four plate gates, loosened *together*	A plate counts as fitted only when R^2, KGE, NRMSE and absolute RMSE all pass. On a record that cannot reach $R^2 = 0.98$, the search keeps refining a plate it can never satisfy. Loosen them as a set: one alone just moves which gate is binding.
Plates with only a few millimetres of settlement	**Absolute RMSE at most**, set near your instrument precision	A percentage is unfairly harsh when the whole settlement is a few millimetres. The absolute tolerance is the companion gate for exactly that case. The default is 20 mm; two to three times your reading error is a defensible setting, so a plate read to ±5 mm justifies 10–15 mm.
A noisy record where one flat pass could be luck	**Plateau window**, raised to 3–4	The step then needs a sustained flat stretch, not a single quiet pass, before it calls itself converged. Only useful once the pass cap has been raised.

KEY IDEA

Nothing here is hidden from the record. Every criterion is written into the project file with the rest of the inputs, so a project that produced a particular answer also records the rules the search was following when it produced it. That is the difference between a tuned run and an unrepeatable one: reopen the file and you get both the answer and its terms.

CHAPTER 25

Reading your results like a professional

25.1 The score-vs-cycle chart

The live chart plots the best χ^2_r so far against "cycles" (step stages), colour-banded by stage. Read it like a descent log: the joint speed fit does the big early drop (rate is what the data know best); the sweep and fine-tune buy the rest. Two things are normal and often mistaken for bugs. The very last point ticks up slightly, because the final band sweeps 100 *discrete* recipes through the data-driven walls, so its best survivor need not sit exactly on the continuous optimum the fine-tune reached. And a stage may hold flat for several cycles before its stop rule fires.

25.2 The sweep heat-map

The (α_A, α_B) heat-map is shown at the winning speed slice. Learn its furniture. The solid violet box is your typed range ($\alpha \in [⅓, ⅔]$ on both knobs). The amber *dashed* box marks the original 3×-extended walls whenever the domain was extended beyond them. The green dot is the winner. (Extension zones are shaded only on the 1-D $\chi^2_r(\alpha)$ fallback plot, not on this map.) A winner inside the violet box says your typed ranges were consistent with the data. A winner outside it, or in an extension zone, is the method telling you which direction your inputs were off.

25.3 The fit statistics

Figure 25.1 shows the card these statistics live on, from a real run of the built-in example.

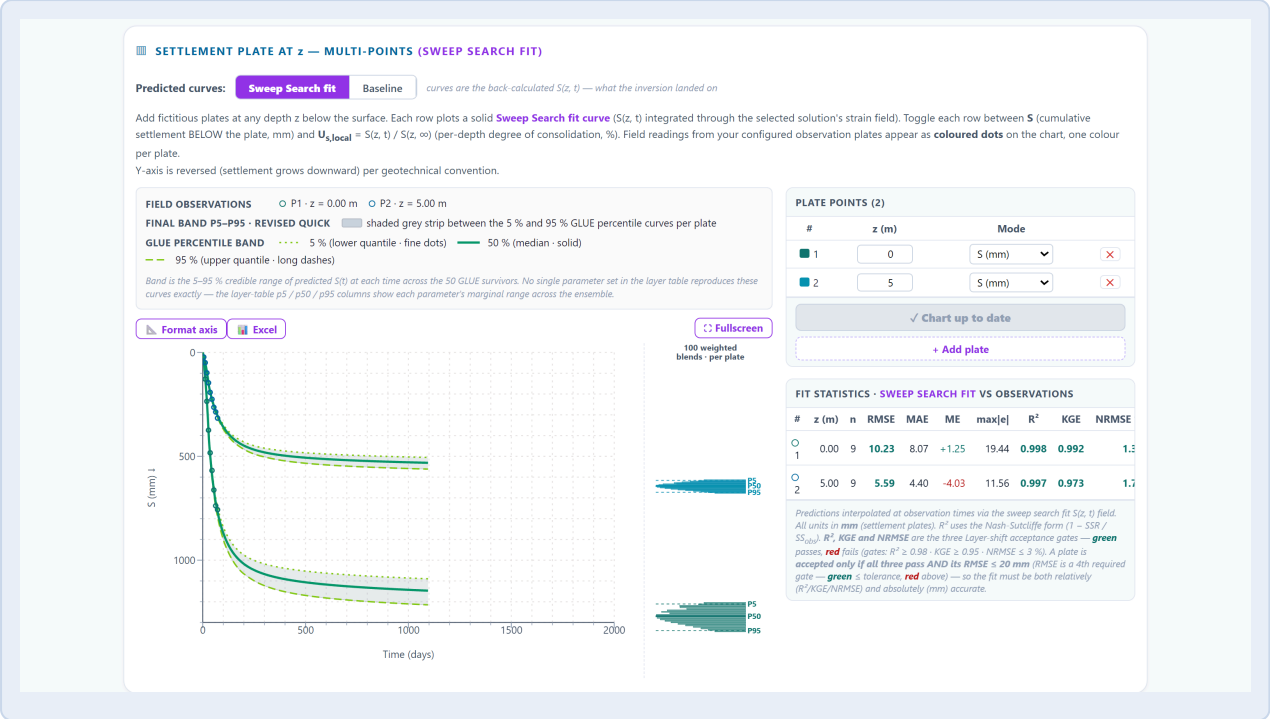


Figure 25.1. The settlement-plate card after a run: fitted S(t) per plate with the readings as dots and the 5–95 % strip (settlement downward, forecast continuing past the last reading); the per-plate statistics on the right, passing values green, with the acceptance rule stated under the table; per-plate histograms of the hundred blends beside the chart. (Back-calculation results → Settlement plate at z.)

Table 25.1. The per-plate fit statistics and their acceptance gates. ME is the signed mean error. In this table it is mean(predicted − observed), so **POSITIVE** ME means the model predicts MORE settlement than measured (it over-compresses). Every signed number the tool shows you now uses this one convention, including the residual-diagnostics card (positive bias = the model over-predicts settlement). The *misfit-localisation* banner of Chapter 26 works internally in the opposite convention but never shows a signed number — it prints the size of the miss with the direction in words ("over-compresses" / "under-compresses"). The one place you will still meet the reverse sign is the older plate-detail residual table, which labels its own formula (obs − pred) beside every value.

Statistic	Meaning	Gate	How to read it
RMSE	typical miss, in mm	≤ 20 mm	compare with your σ_s; the workhorse number
NRMSE	RMSE as % of the plate's largest reading	≤ 3 %	scale-free version of RMSE
R^2	fraction of the variation explained	≥ 0.98	harsh on plates with a small settlement range — a deep plate can "fail" R^2 while missing by 5 mm; check RMSE first
KGE	balanced skill (correlation + bias + spread)	≥ 0.95	catches compensating errors R^2 misses
ME	signed bias, mm	—	the diagnostic gold: its SIGN tells you the direction of model error, plate by plate

> **WATCH OUT**
>
> **$\chi^2_r \approx 1$ under auto-σ is not a compliment.** If auto-scale σ is on, the precision is rescaled until $\chi^2_r \approx 1$ by construction — a unit choice, not an achievement. Judge fit quality by RMSE and NRMSE (never rescaled) and read the revised-σ banner.

25.4 The speed card — including the smear zone

The speed results table, headed *Speed forward models*, shows the designer's guess per layer against the P5 / P50 / P95 values carried by the final-band models, plus the change in P50 against your guess, for k_0 and C_k. Because the band varies the magnitudes about a frozen speed profile, those three columns are equal by construction — the note above the table says so. The measured speed uncertainty is the k_0 factor and its 95 % range, quoted in the note above the table.

> **The smear values have their own rows.** A tinted *Smear (global)* block sits at the foot of this table: one row each for η, r_s and r_t, showing your typed value, the searched value and the change, tagged "Variable · searched" or "held". Set any of them Variable and it is fitted alongside k_0 in Steps 3 and 5, with the fitted value reported in its row; left Constant (the default), the row shows your typed value with a "held" tag. The rows appear whenever a smear zone exists ($\eta > 1$). Set smear Variable if the drainage rate looks wrong.

25.5 What to write down

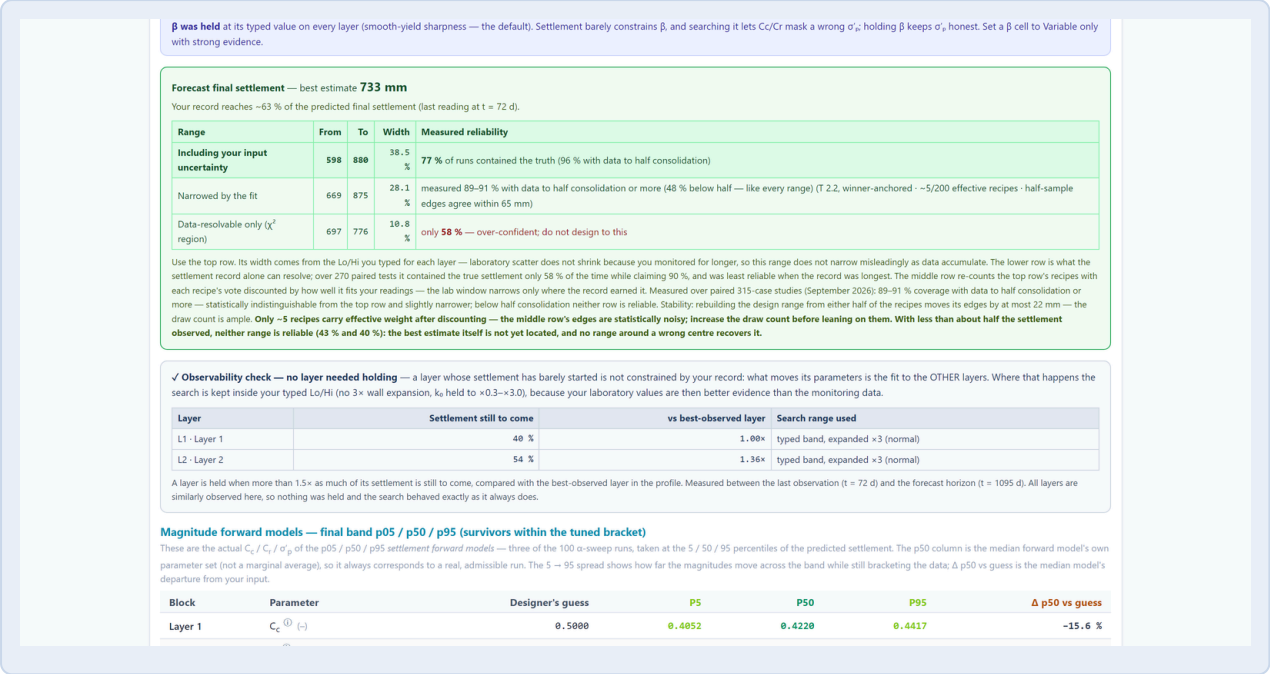
For each settlement plate, report five things:

1. the **P50 final settlement** as the forecast;
2. the **"including your input uncertainty" range** as the design range — *not* the narrower data-resolvable one (§25.6 explains why, and §30.3 gives the measured reliability of each);
3. the **record's truncation** ("based on data to U ≈ 50 %"), because below about half consolidation neither range is dependable;
4. the **speed-factor range** from the α_s interval;
5. if the run was dual-tracked, **both tracks**, and what future observation would separate them (usually: more time).

Quoting the data-resolvable P5–P95 alongside is useful and honest, because it states what your monitoring record by itself resolves. It must be labelled as that, though, and not offered as the design envelope.

25.6 The forecast range — the number you design to

Figure 25.2 shows this part of the card as the tool delivers it.



β was held at its typed value on every layer (smooth-yield sharpness — the default). Settlement barely constrains β, and searching it lets Cc/Cr mask a wrong σ'_{vm}; holding β keeps σ'_p honest. Set a β cell to Variable only with strong evidence.

Forecast final settlement — best estimate **733 mm**

Your record reaches ~63 % of the predicted final settlement (last reading at t = 72 d).

Range	From	To	Width	Measured reliability
Including your input uncertainty	**598**	**880**	38.5 %	**77 %** of runs contained the truth (96 % with data to half consolidation)
Narrowed by the fit	669	875	28.1 %	measured 89–91 % with data to half consolidation or more (48 % below half — like every range) (T 2.2, winner-anchored · ~5/200 effective recipes · half-sample edges agree within 65 mm)
Data-resolvable only (χ^2 region)	697	776	10.8 %	only **58 %** — over-confident; do not design to this

Use the top row. Its width comes from the Lo/Hi you typed for each layer — laboratory scatter does not shrink because you monitored for longer, so this range does not narrow misleadingly as data accumulate. The lower row is what the settlement record alone can resolve; over 270 paired tests it contained the true settlement only 58 % of the time while claiming 90 %, and was least reliable when the record was longest. The middle row re-counts the top row's recipes with each recipe's vote discounted by how well it fits your readings — the lab window narrows only where the record earned it. Measured over paired 315-case studies (September 2026): 89–91 % coverage with data to half consolidation or more — statistically indistinguishable from the top row and slightly narrower; below half consolidation neither row is reliable. Stability: rebuilding the design range from either half of the recipes moves its edges by at most 22 mm — the draw count is ample. **Only ~5 recipes carry effective weight after discounting — the middle row's edges are statistically noisy; increase the draw count before leaning on them. With less than about half the settlement observed, neither range is reliable (43 % and 40 %): the best estimate itself is not yet located, and no range around a wrong centre recovers it.**

✓ **Observability check — no layer needed holding** — a layer whose settlement has barely started is not constrained by your record: what moves its parameters is the fit to the OTHER layers. Where that happens the search is kept inside your typed Lo/Hi (no 3× wall expansion, k_0 held to ×0.3–×3.0), because your laboratory values are then better evidence than the monitoring data.

Layer	Settlement still to come	vs best-observed layer	Search range used
L1 · Layer 1	40 %	1.00×	typed band, expanded ×3 (normal)
L2 · Layer 2	54 %	1.36×	typed band, expanded ×3 (normal)

A layer is held when more than 1.5× as much of its settlement is still to come, compared with the best-observed layer in the profile. Measured between the last observation (t = 72 d) and the forecast horizon (t = 1095 d). All layers are similarly observed here, so nothing was held and the search behaved exactly as it always does.

Magnitude forward models — final band p05 / p50 / p95 (survivors within the tuned bracket)

These are the actual C_c / C_r / σ'_p of the p05 / p50 / p95 *settlement forward models* — three of the 100 α-sweep runs, taken at the 5 / 50 / 95 percentiles of the predicted settlement. The p50 column is the median forward model's own parameter set (not a marginal average), so it always corresponds to a real, admissible run. The 5 → 95 spread shows how far the magnitudes move across the band while still bracketing the data; Δ p50 vs guess is the median model's departure from your input.

Block	Parameter	Designer's guess	P5	P50	P95	Δ p50 vs guess
Layer 1	C_c (–)	0.5000	0.4052	0.4220	0.4417	-15.6 %

Figure 25.2. The delivered answer, from a real run of the built-in example: the β-held note; the forecast final settlement with its three rows and measured reliabilities; the observability check; and the head of the magnitude table, typed guess against P5 / P50 / P95. The card states its own limitations. (Back-calculation results → Forecast & back-calculated parameters — the first card on that menu.)

Above the parameter tables the result page reports the forecast final settlement as a best estimate plus **three** rows, each with a measured reliability printed beside it. They answer different questions and the difference matters.

Row	What it answers	Typical width	Measured reliability
Including your input uncertainty	Given the Lo/Hi you typed for each layer, how far could the forecast move?	~18 % of the settlement	**77 %** of runs contained the truth; 96 % with data to half consolidation (§30.7)
Narrowed by the fit	Same draws as the design range, each vote discounted by how well the recipe fits the readings (see the end of Step 8)	between the two	**89–91 %** with data to half consolidation or more; 48 % below half, like every range (§32.5)
Data-resolvable only (χ^2 region)	How tightly does the settlement record *alone* pin the answer?	~5–17 % (7.5 % pooled)	only **58 %** — over-confident (§30.7); under drifting error, typing the measured drift repairs it (§32.5)

KEY IDEA

Design to the top row. Its width comes from your own laboratory scatter, which does not shrink because a settlement plate was read for another six months — so it does not narrow misleadingly as the record grows. The lower row is genuinely informative about *what the data resolve*, and is worth reading for that, but it contained the true settlement only 58 % of the time while being labelled 90 %, and it stops improving as the record grows — 40 % / 68 % / 66 % at a quarter, half and three quarters of the settlement, so its lowest coverage is at the *shortest* record. On the one real embankment tested it covered two of three cut-offs (§30.8) — three cut-offs on one site cannot settle a coverage question either way, so judge this row on the 270-run evidence of §30.7, not on Ballina.

WATCH OUT

Neither range helps below about half consolidation. There the central estimate itself is not yet located (43 % and 40 % coverage), and no width around a wrong centre recovers it. If your record has not reached U ≈ 50 %, treat the whole result as provisional and keep monitoring.

25.7 The bound-hit flag — when a number is not a measurement

A red banner appears if any recovered **magnitude** (C_c, C_r, σ'_p, β or $C\alpha/C_c$) finishes *on the edge* of its search range. Read it literally: the data were pulling that parameter further and the boundary stopped them, so the number reported is set by where you placed the limit, not by your monitoring.

- **Do not carry a bound-hit value** to another site or design stage.
- If the value looks physically wrong, widen that parameter's ⚙ Range and re-run. **If it then moves a long way, your data do not constrain it** and the forecast is resting on your typed value — which is useful to know.
- A parameter you set to *Constant* is never flagged: it was held deliberately, not stopped by a wall.
- **The speed family is mostly not covered.** C_k and an unpinned k_0 are fitted by line search rather than swept between walls, so they raise no banner — check them against your ⚙ Range by eye. The one exception is a layer held by the under-observed-layer clamp (§25.8): its k_0 is hard-clamped to ×0.3–×3.0 of the typed value, and sitting on that clamp does raise the banner, labelled "k_0 (held by the under-observed-layer clamp)".

> This is why the flag exists: a validation run once returned k_0 = 15 m/s — faster than free-falling water — and reported it as a calibrated permeability without complaint. At Ballina, layers finish sitting on the C_c/C_r cap (§30.8).

25.8 The observability panel — which layers your record can see

The panel lists every layer with the share of its *own* settlement still to come after your last reading, and says whether the search was held inside your typed window for that layer.

A layer whose settlement has barely started is not constrained by your record — what moves its parameters is the fit to the *other* layers. Where one layer is at least 1.5× worse observed than the best-observed layer in the profile, the tool keeps its search inside your typed Lo/Hi: no 3× wall expansion, and k_0 held to ×0.3–×3.0. Your laboratory value is then the better evidence. On an evenly observed profile nothing is held and the panel says so.

KEY IDEA

How to use it. Read the "still to come" column as a map of where your forecast's magnitude comes from. A layer owing a large share of the remaining settlement is where extra site investigation pays for itself — and where a held (pinned) layer tells you the answer is resting on your laboratory value rather than on the monitoring record.

WATCH OUT

What the hold does and does not do. On a fast-shallow-over-slow-deep profile it roughly halves the median held-out forecast error and improves the recovered magnitudes of the held layers. It is provably inert where it does not fire. **It does not rescue the worst cases** — treat it as an accuracy gain on typical projects, not as a safety margin (§30.7).

CHAPTER 26

The safety nets

Master these and you master the method. Each is a small guard that keeps a specific failure from reaching your design note.

Domain extension — never trust a winner leaning on a fence

Trigger: the winner lands within the outer 5 % of any knob's range. *Action:* the range is extended (half a band-width per step, capped at two steps per side; one step on the speed axis, with a loud warning). *Your job:* if the final winner still sits at a capped wall, treat the value as a **bound, not an estimate** — and go back to your inputs: a range, a layer boundary or a load is pushing the answer out of reach.

The best-sampled guard — smooth curves lie near walls

The refine step fits a parabola through the winner and its two neighbouring grid points to estimate the true valley bottom. Against a wall the parabola extrapolates and can "improve" to a point that actually scores worse. The guard re-scores the refined point and keeps the best sampled point if the refinement lost ("keeping the sampled winner" in the log). Lesson: whenever you interpolate an optimum near a constraint, re-evaluate it.

The alternation principle (Step 6)

Whenever you re-measure with a changed ruler, the previous winner is suspect; and whenever you re-scan, the scan must be able to move the same quantities that made the ruler change. A re-scan frozen in the very dimension that shifted is theatre, not verification (89.3 mm unmoved vs 32.5 mm rescued, in the controlled test).

Misfit localisation — the plate-differencing trick (learn to do this by hand)

Settlement plates nest: a plate at 4 m feels the compression of everything below 4 m, so SP1 − SP2 isolates the compression of the soil between 0 and 4 m. Apply the same differencing to the residuals (observed − predicted) and ask: where does the leftover misfit live? If one interval carries nearly all of it, with a consistent sign, the model error is localised to that layer. The ME sign then tells you the direction. Negative (the model over-compresses) means too much compressibility or too little preconsolidation; positive means the opposite. The app runs this automatically after the fine-tune and may trigger a Layer Rescue (Step 7).

> **KEY IDEA**
>
> **Rescues are findings, not fixes.** A rescued value outside your typed band appears in an amber banner naming the layer, the released values and the improvement. The method is handing you evidence that an input deserves revision — ratify it by updating the input and re-running, or reject it. If the rescue fails, the banner says the defect is probably outside the searched parameters: check that layer's thickness, unit weight, e_0 or the load history. And if a flagged interval persists after everything — suspect the model, not the parameters.

The two consistency rules that keep every number honest

Two rules run through the whole pipeline, both paid for with real debugging pain. (i) **Every step optimises and reports the same metric.** Otherwise a step can look like it made things worse when it merely changed the yardstick. (ii) **Every step scores on the same solver grid as the delivered result.** Otherwise the chart advertises a "best" the final answer can never reproduce. When you build your own analysis tools, steal both rules.

CHAPTER 27

Checking against the field: the Asaoka cross-check

Everything above shares one forward model. In the field the model itself is imperfect. So it is worth running a second, nearly model-free analysis on the same readings: the classical **Asaoka construction**. Plot each settlement reading against the previous one at equal time steps and fit a straight line. Its intersection with the 1:1 line is the ultimate settlement S_{∞}^{A}, and its slope back-figures an effective c_h^{A}. Because Asaoka carries almost no compression physics, the two analyses fail in *partly* different ways, which is what gives the comparison value.

> WATCH OUT
>
> **You run this one yourself.** The tool does *not* perform the Asaoka construction. A dedicated Asaoka mode existed in earlier versions and was retired: its assumptions — constant load, time-invariant c_h, no stress dependence — conflict with the multi-stage loading and stress-dependent c_h this tool is built on, so keeping it would have meant shipping a cross-check that quietly disagreed with the model for the wrong reasons. Plot it by hand or in a spreadsheet from the same readings, then compare it with the numbers on the result page using the three gates below.

> WATCH OUT
>
> **How independent is this check, really?** Only partly — and the guide is explicit about it because over-claiming here would undo the honesty of the band. Asaoka is model-*light*, not model-*free*: it still assumes one-dimensional-style, exponential-approach consolidation and the same drainage idealisation as the primary method. A *structural* error in that idealisation biases *both* analyses in the same direction and can pass all three gates undetected. Wrong smear geometry, unmodelled well resistance, drain clogging and a mis-set drainage boundary are all errors of this kind. So agreement raises confidence but does not *prove* the model; a genuinely structure-independent check (e.g. an independent measured settlement-rate, piezometer dissipation, or a second constitutive engine) is what closes that gap. Treat G1–G3 as necessary, not sufficient.

Figure 27.1 draws the field workflow.

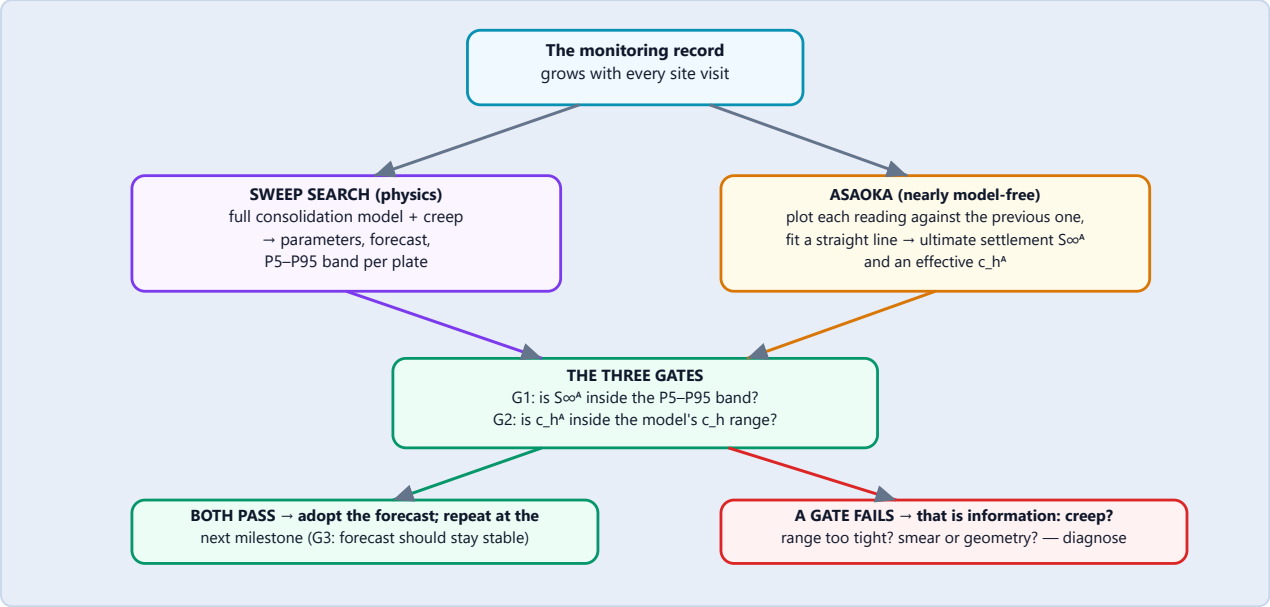


Figure 27.1. The field workflow. Run both analyses at every monitoring milestone and apply the three gates. Agreement promotes the forecast; disagreement is treated as information, not averaged away.

- **G1 (magnitude):** Asaoka's S_∞^A should land inside the Sweep P5–P95 band, plate by plate.
- **G2 (rate):** the back-figured c_h^A should land inside the range of the model's stress-dependent $c_h(\sigma')$ over the record. A G2 failure with a healthy G1 usually points at the smear inputs (η, r_s) or the as-built drain geometry — the moment to set smear Variable and re-run.
- **G3 (stability):** re-run at every milestone; the P50 should drift within its own previous band as the record grows.

KEY IDEA

The creep signature. Asaoka assumes settlement decays exponentially; creep never lets it. So under active creep the Asaoka asymptote keeps growing as the fitting window moves later, while the Sweep P50 (which models creep explicitly) stays put. A growing S_∞^A over a stable P50 is not a contradiction to be averaged — it is positive evidence of secondary compression; look at the recovered $C\alpha/C_c$ instead of panicking. Conversely, Asaoka applied before U ≈ 60 % under-estimates S_∞ — discount an early G1 failure from below.

CHAPTER 28

A worked self-consistency example you can reproduce

WATCH OUT

What this example is — and is not. This is a *solver self-consistency* demonstration: the "truth" is generated by the *same* forward model the estimator uses to score candidates. It shows that the optimiser converges and recovers a truth it is structurally capable of reproducing — it does *not* establish predictive skill on real soil, where model error is present. Read it as "the machinery works as designed," and see Chapter 30 for what field- and cross-model evidence adds (and for the measured behaviour once the truth is generated by a *different* model).

Benchmark Project A: 12 m of soft estuarine clay in three layers, staged loading, drains on a triangular grid. The "truth" was generated by the same simulator (with creep active), 60 readings per plate were taken, and everything after 50 % consolidation was hidden from the method. The typed ranges were set ±15 % around truth for the main magnitudes, and the speed seed was deliberately wrong (k_0 halved). Figure 28.1 shows what the recovery looks like.

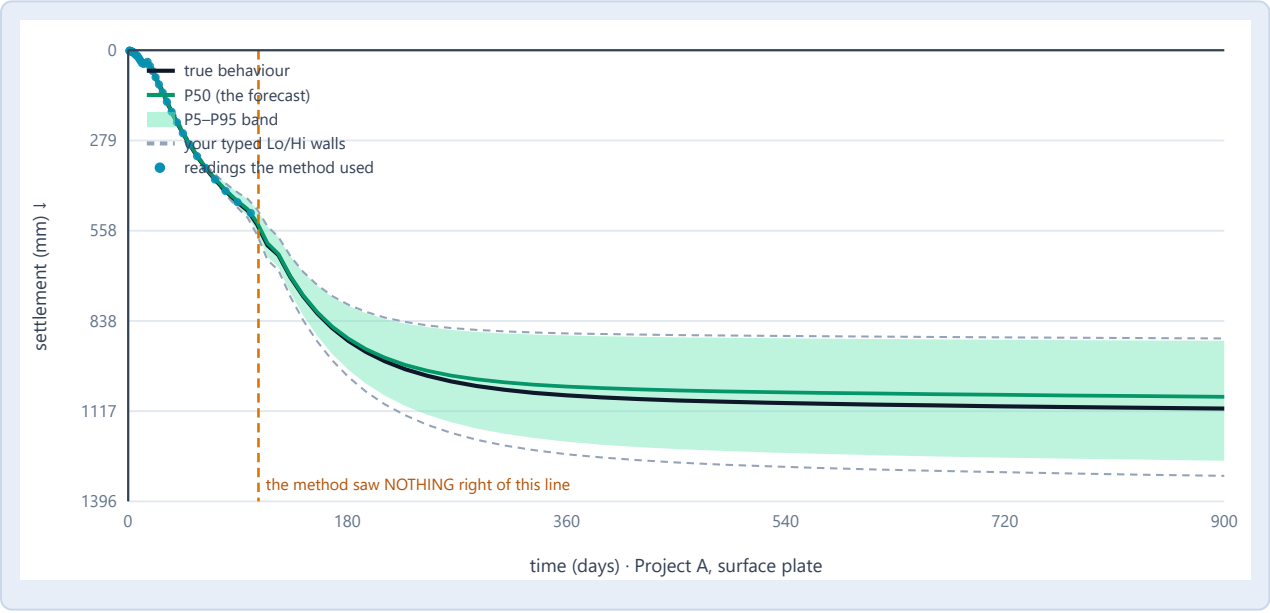


Figure 28.1. Self-consistent recovery: the method saw only the readings left of the amber line, yet P50 (green) tracks the hidden truth (black) inside the P5–P95 band for the remaining 88 % of the timeline. The truth shares the estimator's forward model, so this checks convergence and identifiability, not skill under model error.

Numbers to check when you re-run it: winner α ≈ 0.5 on both magnitude knobs, because the ranges were centred on truth, so the middle is right; fine-tune $\chi^2_r \ll 1$ on the noise-free run; and a held-out error of the P50 around **4 mm** at the surface on the noise-free run, which is a few tenths of a per cent of the final settlement. Add σ = 10 mm Gaussian noise (auto-σ OFF, so χ^2_r is interpretable) and the winner barely moves: held-out error about **19 mm** (1.7 %) with $\chi^2_r \approx 1.07$ — the noise floor.

But note what that number is. With identical forward physics and Gaussian noise, a correctly implemented estimator *must* land near $\chi^2_r \approx 1$ on held-out points. So $\chi^2_r \approx 1.07$ confirms the optimiser converged; it is not independent evidence of accuracy on real ground. (If auto-σ had been ON, $\chi^2_r \approx 1$ would be true by construction and carry no information at all. Judge by RMSE/NRMSE then.)

REPRODUCE IT

Try it. Open the tool → Back-calculation & Forecast. Sweep Search is the default (and only) method. Enter a layer profile and plate readings (or load the demo), set your Lo–Hi ranges with the ⚙ **Range** button on each cell, leave C_k and the smear trio Constant on a first pass, and press **Calculate**. Watch the step names of Chapter 24 go by; then read the result page with Chapter 25 open beside you. For a second pass, set η Variable and see whether G2 improves.

CHAPTER 29

Common mistakes, and a self-test

Table 29.1 is the troubleshooting guide: symptom, likely cause, and the fix.

Table 29.1. Troubleshooting guide. Most "method problems" are input problems wearing a disguise.

Symptom / mistake	What is really happening	What to do
Typed Lo = Hi to "fix" a magnitude	the band IS the search space; a zero band would freeze the sweep, so the tool auto-widens it and tells you	give every magnitude an honest range. Any magnitude you mark *Constant* is held at your typed value with a zero-width band — that applies to C_c, C_r, σ'_p, β and Cα/C_c alike, and β and Cα/C_c are Constant by default
σ'_p typed below the in-situ stress	physically impossible (OCR < 1); the solver clamps it to σ'_0 and your typed number silently stops mattering	type $\sigma'_p \geq \sigma'_0$ per layer; think in OCR
Winner pinned at a wall after extensions hit their cap	the data are asking for values outside anything the search may reach — usually an input inconsistency (often one layer's)	read the misfit-localisation banner; revise the named layer's inputs; re-run
Trusting $\chi^2_r \approx 1$ with auto-σ on	auto-σ makes $\chi^2_r \approx 1$ by construction	judge by RMSE / NRMSE and the revised-σ banner
Reading a deep plate's R^2 failure as a bad fit	R^2 is harsh when the settlement range is small; a 5 mm miss can "fail"	check RMSE (mm) first; R^2 second
Ignoring a dual-track result and quoting Track A only	the data genuinely cannot separate two stories	report both; say what future data will separate them; design for the worse one if you must decide now
Accepting a Layer-Rescue value without follow-up	the rescue is a finding about your INPUTS, not a silent correction	ratify it: update the input, re-run, confirm the flag clears
All layers' k_0 locked Constant, expecting a speed search	with every speed locked, the α_s axis collapses (by design — locks are contracts)	leave at least the uncertain layers' k_0 Variable
Smear left Constant while G2 keeps failing	the drainage rate is throttled by a smear severity the search was not allowed to question	set η (and r_s) Variable on the merged smear cells and give them a window with ⚙ Range; re-run; read the fitted values off the *Smear (global)* rows of the results table (§25.4)
Comparing runs made on different solver grids	different rulers — small systematic offsets look like parameter changes	keep the Advanced Solving grid identical across runs you compare

Test yourself

1. Why is the per-layer speed frozen during the magnitude sweep — and what would happen if it were free?

2. Your site is more compressible AND more overconsolidated than your typed mid-values. Why can a single-knob sweep never represent it, and which knob pair fixes that?
3. The speed anchor is fitted at your typed magnitudes. Explain in one sentence each: (a) the mild consequence of a biased anchor; (b) the severe one; (c) which the alternation loop can fix.
4. Why is the α_s axis narrow (×0.625–×1.6) instead of a full decade?
5. What does "min-over-S" winner selection prevent?
6. Name the two speed-family parameter groups fitted besides k_0, and state when they are actually searched.
7. ME at the surface plate is −30 mm; the deeper plates fit well. Which depth interval is guilty, and in which direction should its σ'_p (or C_c) move?
8. χ^2_r reads 1.02 with auto-scale σ enabled. What may you conclude about fit quality?
9. The Asaoka asymptote grows every month while the Sweep P50 stays put. Failure of which method?
10. A layer's σ'_p is typed Lo = 60, Hi = 90 kPa, in-situ σ'_0 = 45 kPa. Compute exLo and exHi and state the effective low wall the solver uses.

Answers. 1. Freezing severs the speed–magnitude see-saw so changing the magnitudes visibly changes the score; with speed free, every magnitude candidate could compensate and the sweep would return an arbitrary point on the valley floor. **2.** One knob raises group A and lowers group B in fixed gear (the anti-diagonal); a same-direction truth lies off that line. The (α_A, α_B) pair spans the whole square. **3.** (a) the valley minimum shifts (stale ruler); (b) the wrong valley can look deepest (wrong basin); (c) alternation fixes only (a) — basin choice happens at scan time, so only a scanned speed axis fixes (b). **4.** A wide speed range re-opens the see-saw and flattens the valley the sweep depends on; the narrow range covers plausible anchoring error without dissolving the signal. **5.** Two artefacts: a good recipe punished for a poor speed pairing, and one valley appearing at two speed levels as a fake dual-track. **6.** C_k (held at $0.5 \cdot e_0$ unless unlocked) and the smear trio η, r_s, r_t (searched only when Variable). **7.** The interval between the surface plate and the next plate down; negative ME = model over-compresses → raise σ'_p/OCR (or lower C_c) for that layer. **8.** Almost nothing — auto-σ rescales precision until $\chi^2_r \approx 1$ by construction; judge by RMSE/NRMSE. **9.** Neither — it is the creep signature (Asaoka's exponential-decay assumption is violated); check the recovered $C\alpha/C_c$. **10.** exLo = $60^2/90$ = 40.0 kPa; exHi = $90^2/60$ = 135.0 kPa; but the clamp $\sigma'_p \geq \sigma'_0$ = 45 makes the effective low wall 45 kPa.

CHAPTER 30

How far to trust the answer

This part states plainly what the guide's numbers do and do not establish. It positions the method against the prior literature, and it reports the empirical studies run to answer the questions two rounds of independent review raised. That includes the **band-coverage calibration** (§30.3), which a good first revision surfaced as the load-bearing issue. All drive the *real* forward solver; the re-runnable scripts live in `validation_2026/` (`audit_coverage.js`, `audit_bias.js`, `audit_sensitivity.js`, `audit_crossmodel.js`, plus the Round-2 `audit_calibrate.js`, `audit_knobs_regime.js`, `audit_crossmodel2.js`).

A note on resolution. For replicate throughput these studies run on a deliberately coarse solver grid, and it matters which result that helps or hurts. Band *coverage* (§30.3) is a calibration property a too-tight interval exhibits at any resolution, so the N-1 finding is robust and if anything conservative. The speed-axis benefit (§30.4) and the structural-error signal (§30.6) instead depend on a *sharp* misfit landscape a coarse grid smooths — so null results there are reported as *unconfirmed at production resolution*, not refutations, with the production-grid re-run named as the next step.

30.1 What the evidence covers — and what it does not

> **WATCH OUT**
>
> Every synthetic result up to this point is a **solver self-consistency** check: the "truth" is generated by the same forward model the estimator scores against. Benchmark Project A, the three-knob evidence figures and the "$\chi^2_r \approx 1$ noise floor" are all of this kind. That configuration (an *inverse crime*) can show the optimiser converges, that parameters are identifiable from a truncated record, and that the machinery is internally consistent — but it cannot, even in principle, expose *model inadequacy*, the dominant error in real back-analysis. That is why the one test against an instrumented case history, the Ballina embankment of §30.8, carries more weight than everything before it. It is also why a second such site remains the single most valuable addition to this tool's evidence base.

30.2 Where this sits in the literature (what is, and isn't, new)

The speed–magnitude trade-off the method "discovers" is the well-known rate–compressibility coupling of consolidation inversion, not a new phenomenon; freezing one family while sweeping the other is coordinate descent with a nuisance-parameter profile. The standard alternatives are Asaoka's observational method (1978), direct Hansbo unit-cell inversion, and Bayesian or Kalman-filter settlement updating. Against them, the contribution claimed here is narrow and specific: *robustness of the magnitude scan to a mis-anchored drainage rate under coupled, oppositely-directed input bias*. It is obtained by carrying a bounded global speed axis through the scan and alternating (Chapter 22, knob S). That is a measurable design choice (§30.4), not a general claim of superior accuracy; establishing the latter needs head-to-head comparison against those baselines on common data.

Chapter 31 is one such comparison, against a published Bayesian back-analysis of the same Ballina embankment. But it is a single site against a single alternative, which is a long way short of a general claim.

The method's ancestry, set out in Foundations at the opening of Chapter 21, bears on the evidence too. The coverage shortfall measured in §30.7 is not a private defect of this implementation. It is the documented property of behavioural bands, debated in print for twenty years: such bands are conditioned on the model structure and on the threshold, and they are not formal credible intervals (Beven & Binley 2014). The measurements in §30.7 calibrate the band empirically. That calibration is this part's answer to a limitation the family has always carried.

Naming the method precisely. With the September 2026 middle row in place, Step 8 contains a genuine — if deliberately small — Bayesian computation, and it is worth stating exactly how far that reading goes. The typed Lo/Hi windows act as the prior; the tempered fit-discounting acts as the likelihood; the vote-weighted percentiles approximate the posterior forecast by importance sampling from the prior. A fair one-line description of the whole pipeline is therefore *a solve-budget-efficient engineered search, followed by a quasi-Bayesian propagation step with measured calibration*: the engineering lives in Steps 1–7, the Bayes lives in Step 8, and the measured-calibration claim is the part neither a plain sweep nor a textbook Bayesian analysis carries on its label. The word "search" is deliberately kept out of the Bayesian half — Steps 1–7 are deterministic optimisation and carry no probability.

Three caveats on the Bayesian reading, and what answers each. First, the discounting is *tempered*: $T > 1$ softens the likelihood below the exact Bayes rule. But tempering by T is algebraically identical to exact Bayes with every reading's noise standard deviation inflated by $\sqrt{T}$, and since T is the winner's own reduced χ^2, that inflation is precisely "re-estimate the noise floor from the best fit achieved" — a standard, defensible move when the nominal noise understates model error; the card prints T so the reader sees the inflation applied. Second, the prior samples each parameter *independently*. In truth it is independent-then-truncated: the C_c/C_r ratio cap and the admissibility rules impose the dominant known physical dependences on every joint draw, and the *posterior* cloud carries whatever correlations the data induce regardless, because the weights act on complete recipes rather than on parameters one at a time. Third, 200 draws is a deliberately small importance sample, and this caveat is answered on the card itself: the effective number of recipes after discounting and the split-half stability stamp are printed beside the row. That diagnostic bit on its first outing — on the built-in example the winner-anchored discounting concentrates the weight on roughly five effective recipes, the split halves disagree by 65 mm, and the card says so in bold rather than presenting fragile edges as solid. A method that reports its own weak runs is the spirit of everything in this Part.

Calibration as a falsifiable contract. The deepest difference between this pipeline and a conventional Bayesian analysis is not algorithmic but epistemic: it lies in what kind of statement each makes about its own reliability, and in what it would take for that statement to fail. A posterior credible interval is exact *conditional on its premises* — the assumed error model, the prior, and the structural adequacy of the forward model. That conditional exactness is a genuine strength, and nothing here disputes it; but the analysis itself contains no trial of those premises, so when one fails the interval remains fluent and confident while its guarantee has quietly lapsed. This is not a defect of

Bayesian reasoning — it is the price of deductive guarantees, and a rigorous Bayesian practitioner pays it down with posterior-predictive checks and sensitivity studies external to the inference proper. The reliability statements in this part are constructed the other way about. Every printed coverage figure is the outcome of a designed trial against known truths that was free to return a refuting answer, under acceptance criteria registered *before* any run stating what result would count as failure. The claim "this range contained the truth in 77 % of trials" is therefore falsifiable in the strict sense: a definite procedure existed that could have contradicted it, was executed in full, and did not.

The record shows this contract being exercised in both directions, which is what distinguishes it from ceremony. The $\Delta\chi^2$ region's nominal 90 % was measured at 58 % — and the finding was printed on the result card as a warning rather than absorbed into a footnote. The September 2026 construction drift was exposed precisely because a measured truth fell outside a range that claimed to contain it, and the repair was itself re-measured before shipping. The fit-narrowed row's own on-card diagnostics flagged, on their first outing, that its edges rested on too few effective recipes to be quoted as solid — and the card said so in bold. Three reliability claims exposed to refutation; three refutations found; three repairs, each re-verified by the same instrument that found the fault. A calibration that can fail out loud — and demonstrably has — is the only kind whose successes carry evidential weight. That is this part's considered answer to the fair observation that behavioural methods lack the deductive guarantees of the formal Bayesian family: what this method forgoes in guarantee it repays in tested, repairable, and continuously re-measured performance, with the test protocol published alongside the number it produced.

30.3 How reliable is the reported range?

The result page reports two ranges around the forecast. They answer different questions, they have very different reliabilities, and the difference decides which one you design to.

Range	The question it answers	Typical width	Measured coverage
Including your input uncertainty	Given the Lo/Hi you typed for each layer, how far could the forecast move?	~18 % of the settlement	**77 %** [71–81]
Data-resolvable only (χ^2 region)	How tightly does the settlement record *alone* pin the answer?	~5–17 % (7.5 % pooled)	**58 %** [52–64]

Both are labelled P5–P95, so both nominally promise 90 %. Measured over 270 paired tests spanning five soil profiles, three record lengths, three noise levels and three grades of laboratory input, the data-resolvable range delivered 58 % and the input-uncertainty range 77 %. A second coverage round (September 2026; 315 cases, same star design, same seeds) re-measured every construction through the *shipped* code path rather than a stand-alone replica implementation, after an implementation drift had shown the difference matters. It replicated this study almost exactly — design range 79 % pooled against 77 % here, χ^2 region 60 % against 58 %, paired agreement on 90 % of matched cases — and added four findings. The pre-fix band construction measured 41 %, with 102 paired cases where the repaired construction covered and the old one did not, against one reversal.

Omitting Steps 1–7 entirely (the same propagation centred on the typed values) measured 37 % against 79 % — the deep search's centring is worth roughly forty points of coverage, and its centre error falls from 10 % of the settlement at quarter-records through 6 % at a third to below 3 % at half, while the typed guess stays near 14 % throughout. Under AR(1)-correlated noise (ρ = 0.7 — the realistic case of drifting instrument error) the χ^2 region's coverage fell further to 42 % while the design range held 73 %: the assumption-dependent band degrades when its noise assumption breaks; the typed-width band barely notices. The failure is repairable, and the repair is now shipped: type the drift into Observation accuracy and the score whitens the residuals and counts the effective readings honestly, recovering the χ^2 region to 83 % on the same 60 sites at more than double its width (§32.5). And a dedicated one-third-record slice, run at the same breadth as each main truncation (105 cases: all profiles, all noise and laboratory-quality excursions), graded the truncation advice: the design range's coverage runs 48 % at a quarter, 63 % at a third, 93 % at half — so "below about half, treat every range as provisional" stands, with one-third records usable only with caution. The fit-narrowed middle row measured 89–91 % wherever records reach half consolidation (its promotion is described under Step 8). Broken down by how much settlement had been observed when monitoring stopped:

Record reaches	Data-resolvable range	Including your input uncertainty
a quarter of the settlement	40 %	43 %
half	68 %	**96 %** [89–98]
three quarters	66 %	**91 %** [83–95]

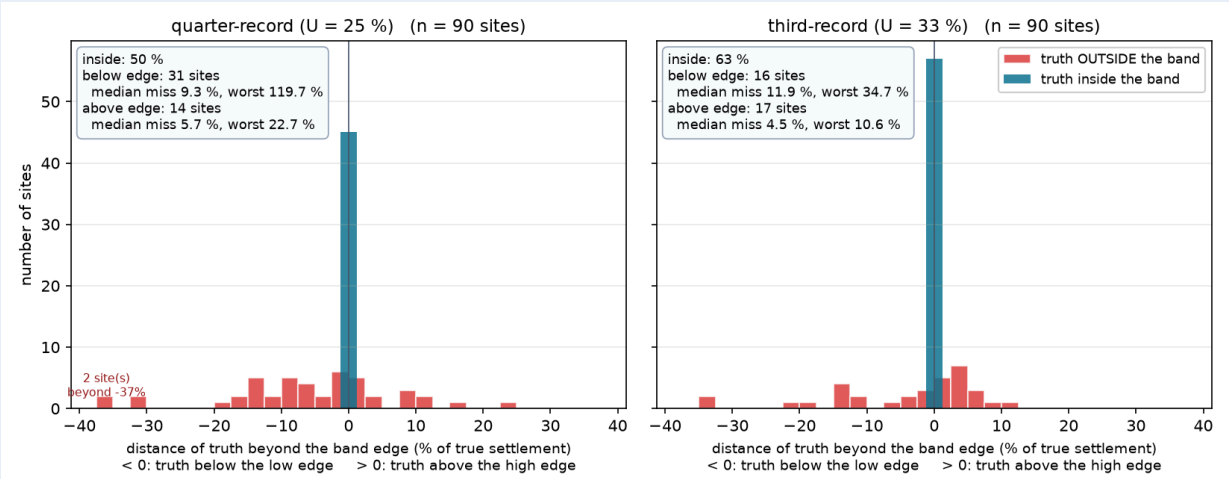


Figure 30.1. Where the truth sits relative to the design range at quarter-records (left) and third-records (right), 90 independent-noise cases each: teal = inside; each red bar a miss, placed by its distance beyond the nearer edge (negative = the range over-predicted). Correlated-noise cases excluded to match the 48 % and 63 % headline coverages.

The anatomy of a miss. Figure 30.1 answers the question the coverage percentages hide: when the range misses, by how much — and why. Four regularities stand out. *First, misses are mislocation events, not width events.* Among the cases the range contained, the winner's centre error is small — median 5.1 % of the settlement at quarter-records, 3.4 % at a third — while among the misses it is

19 % and 14 %: roughly twice the band's own half-width of 9–10 %. The band almost never misses because it is too narrow; it misses because the search, starved of record, parked it in the wrong place, and most misses are accordingly grazes of 5–12 % just beyond an edge with a thinner tail of larger errors. *Second, laboratory quality is the dominant controllable factor at short records.* At quarter-records the range contained the truth in 60 % of good-laboratory cases, 53 % of standard ones and only 27 % of poor-laboratory ones; by a third the same split has recovered to 80 % / 62 % / 53 %. With little record the search remains anchored near the typed values, so a biased laboratory drags the centre with it — and the study's single pathological case (a quarter-record, poor-laboratory profile whose range sat at more than double the truth, centre error +154 %) is exactly that mechanism at full strength. *Third, instrument precision matters far less than record length.* The low- and high-noise excursions differ from one another by no more than their small-sample uncertainty at either truncation, while a quarter-to-third increase in record length moves containment by thirteen points and a quarter-to-half increase by forty-five: a longer record with noisy plates beats a shorter record with precise ones, which is worth remembering when a monitoring budget is being argued over. *Fourth, the miss directions are asymmetric in the direction that favours safety.* At quarter-records two thirds of the misses are over-predictions — the range sitting above the truth, the conservative error for settlement design — while the dangerous side, real settlement above the range's top, has both fewer cases and shorter reach: its worst case is 23 % beyond the edge at quarter-records and 11 % at a third. Lengthening the record cuts these tails before it repairs the averages — the worst over-prediction falls from 120 % to 35 % between a quarter and a third — and that tail-cutting, as much as the coverage staircase itself, is what the "wait until about half" advice buys.

> **KEY IDEA**
>
> **Design to the input-uncertainty range.** With a record reaching at least half the settlement it contains the truth about nine times in ten — close to the 90 % it claims. The data-resolvable range is genuinely informative about *what your measurements resolve*, and worth reading for that, but it is not a design envelope.

Why the narrow range fails, and why more data stops helping it

The data-resolvable range is a profile-likelihood region: the set of parameter combinations that fit the record about as well as the best one. It therefore measures how tightly *the measurements* constrain the answer — not how much you do not know. Those are different quantities, and the gap between them is where the error lives.

As monitoring accumulates the record pins the parameters ever more tightly, so this range narrows — from about 17 % of the settlement early on to about 5 % once three quarters has occurred. But the forecast error does *not* shrink as fast, because part of it comes from soil variability and model limitations that more settlement readings cannot resolve. The result is the pattern in the table above: **coverage stops improving as the record lengthens**. It is 68 % at half consolidation and 66 % at three quarters, while the interval keeps tightening. (Its lowest coverage, 40 %, is at the *shortest* record, where the centre itself is not yet located.) At three quarters consolidation it reports roughly ±2.5 % and is right about two thirds of the time.

The input-uncertainty range does not have that failure mode. Its width comes from the laboratory scatter you typed, which does not shrink because a settlement plate was read for another six months. It stays near 18 % and its coverage rises with data rather than falling.

The failure is in the range, not in the search

One innocent explanation would be that the hidden truth simply lies outside the region the sweep can reach. It does not: across the replicate suite the true C_c and σ'_p both lay inside the expanded search walls in all 12 replicates of the calibration study (100 % [76–100]). **The truth is almost always inside the searched domain but outside the reported interval** — so the shortfall is a calibration fault of the interval, not a failure of the search to look in the right place. That is why the remedy was a different width, not a wider search.

> **WATCH OUT**
>
> **Two limits that remain.** Below about half consolidation neither range is reliable (43 % and 40 %): the central estimate itself is not yet located, and no width around a wrong centre recovers it — keep monitoring. And the *per-parameter* ranges (C_c, σ'_p and the rest) are narrower than their own errors justify; read them as diagnostic, never as a design envelope.

> **How this was arrived at.** Earlier releases multiplied the *reported interval* by a fixed empirical factor to compensate. (This is a different mechanism from the ×3 expansion of your typed walls in §23.2, which is unchanged and still in use — that one widens where the search *looks*; this one widened what the tool *reported*.) The reporting factor was retired once its cause was identified: the interval swept magnitude at a fixed rate, so it carried no rate uncertainty at all, and mid-consolidation — where a design forecast lives — rate is the dominant unknown. Sampling the joint magnitude × speed region supplied the missing dimension directly and made the multiplier unnecessary. The remaining shortfall, and the input-uncertainty range that addresses it, are measured in §30.7; §30.8 repeats the test on a real embankment.

30.4 Does searching the speed axis actually help?

The headline "89.3 → 23.9 mm, 3.7× better" figure compares the method only against a crippled version of itself at two hand-picked bias corners. We re-ran the contrast over randomly-directed input bias and added a genuine non-ablation baseline ("use your lab values, just fit the rate"). Over **18** randomised bias directions, median held-out surface RMSE was **58 mm** (naive: typed centre + fitted rate), **33 mm** (frozen-speed scan) and **28 mm** (full three-knob); the full method beat its frozen-speed ablation in **7 / 18** runs, median improvement **-0.2 mm**.

Two distinct conclusions, and it matters which is which. First, the magnitude *sweep* is clearly worth doing. Both sweep variants roughly halve the median held-out error of the naive baseline. More importantly, they cut off its long tail: the naive forecast can be very wrong when the typed centre is biased. That is a real benefit of the core method against a non-ablation alternative. Second, and contrary to the headline, the *speed-axis / 3-D* refinement gives essentially **no average benefit** over the frozen-speed ablation across randomised bias — the median improvement is within noise of zero and the full method wins only about half the time. The large gains quoted in Chapter 22 are real but

confined to the specific opposite-direction-bias configuration that maximises them. The honest reading: knob S is best understood as insurance against one particular failure mode (a mis-anchored rate under oppositely-biased inputs), not a general accuracy improvement.

Does the insurance actually fire when the covered event happens? The average benefit being ~zero is only defensible if the benefit concentrates in the regime knob S is for. So a second suite was run with the bias direction controlled — 8 runs in each regime, separate from the 18-run study above. In the **opposite-direction-bias** regime, where the settlement-raising group is typed low while the lowering group is typed high (the exact mode knob S targets), the full method beat its frozen ablation in **2/8** runs with a median gain of **-0.0 mm**. In the **coherent-bias** regime the win rate was **2/8** and the median gain **-0.1 mm**. So the safeguard is **not demonstrably firing** here, which on its face argues for making knob S opt-in rather than always-on.

> **WATCH OUT**
>
> **One honest caveat before acting on that.** The wrong-basin failure knob S exists to fix is a *sharp-landscape* phenomenon — two distinct misfit valleys that swap depth when the speed changes. The deliberately coarse study grid used here (chosen for replicate throughput) smooths the misfit landscape and may not reproduce those distinct basins. This null result is therefore *consistent with* knob S having nothing to fix at this resolution. It does not show the knob is useless at the production grid, where the original 89 → 24 mm gap was measured. The correct next step is to repeat this conditional test at the app's production solver grid before deciding opt-in vs always-on; until then knob S's value is best called *unconfirmed*, not refuted.

30.5 Does the answer depend on the tool's internal settings?

Four load-bearing constants were previously asserted with a plausibility story but no sensitivity evidence. Varying each one-at-a-time on a representative noised case (others held at their shipped values) gives these swings in the delivered forecast and band width — the larger the swing, the more the constant is a genuine user-facing knob rather than a settled default:

- **dChi2**: forecast-band-width swing **53 mm** · held-out-RMSE swing 6 mm (over the tested range).
- **speedR**: forecast-band-width swing **40 mm** · held-out-RMSE swing 9 mm (over the tested range).
- **expandK**: forecast-band-width swing **2 mm** · held-out-RMSE swing 17 mm (over the tested range).
- **boltzT**: forecast-band-width swing **0 mm** · held-out-RMSE swing 0 mm (over the tested range).

Constants with a small swing are safe as fixed defaults. The $\Delta\chi^2$ threshold is the one that matters: it sets band *width* directly, with much the largest swing of the four. The expansion factor matters most for accuracy (a 17 mm RMSE swing against the speed ratio's 9 mm, even though it barely moves band width). None of the three is a user setting — each is fixed in the code, which is exactly why this test matters: it shows the fixed choices are benign. The final-band temperature likewise moved nothing measurable and is also fixed.

30.6 Testing against a different model — still an open question

The cleanest fix for the inverse crime is a field case history. Ballina (§30.8) is one, but with a year or more of record its residual model error is absorbed. At the shortest cut, where it is not absorbed, a single site cannot isolate the mechanism. Absent a field case that clearly exercises a non-absorbable error, the second-best is a truth generated so that structural error is present by construction, then inverted with the default model. We tried two structural errors. Importantly, we **report this section as inconclusive**, because both proved largely *absorbable* by the free parameters for this radially-dominated PVD problem:

- **Unmodelled smear (η=3) + a 30 % void-ratio error.** With the truth matched to the model (inverse crime), the median over **12** replicates was $\chi^2_r \approx$ **1.20**, held-out RMSE **32 mm**, band coverage **58 %**. Regenerating the truth WITH a smear zone (η=3) and a 30 % void-ratio error, neither directly representable by the default inversion, gave $\chi^2_r \approx$ **1.19**, held-out RMSE **24 mm**, band coverage **83 %**. Did not separate: a smear zone is mostly a drainage-*rate* effect the speed family soaks up, and a void-ratio error is partly masked by the $C_c/(1+e_0)$ degeneracy.
- **A wrong drainage boundary (truth double-drained, model single-drained).** Over **12** replicates, the matched control fit to $\chi^2_r \approx$ **1.20** with held-out RMSE **34 mm** and band coverage **83 %**; generating the truth with DOUBLE drainage moved these to $\chi^2_r \approx$ **1.19**, held-out RMSE **31 mm**, band coverage **75 %**. Also did not separate — for a radially-dominated unit cell the vertical drainage boundary moves the settlement curve only slightly, so the radial speed knob absorbs the rest.

> **WATCH OUT**
>
> **What this does and does not show.** An unfalsified hypothesis from tests that could not falsify it is not evidence of robustness. Neither attempt exercised the failure mode it was built to probe, so **the cross-model question remains genuinely open**. It is quietly informative that several *plausible* structural errors turned out absorbable for a radial PVD unit cell. The test that would settle it is a decisively non-absorbable error: finite drain *well resistance*, which the unit-cell model has no term for at all; an unmodelled thin soft lens; or a second real instrumented field record. Ballina (§30.8) supplies the field data but does not exercise a non-absorbable error, so the decisive case remains outstanding. This section should be read as "not yet demonstrated either way," not as reassurance.

> **KEY IDEA**
>
> One signal here *is* firm, and it is independent of the structural-error question. The reported band covered the truth in *well under* 90 % of runs **even in the matched inverse-crime case**. That independently corroborates §30.3: the band, at its shipped width, overstates its own coverage.

30.7 The reported range, measured properly

The band was measured on **270 paired runs** spanning five soil profiles, three truncations (U = 25 / 50 / 75 %), three measurement-noise levels (5 / 10 / 20 mm) and three laboratory-quality levels (×0.5 / ×1 / ×2 disturbance of the typed values). Every run starts from *disturbed* inputs, never

from the answer.

KEY IDEA

The band is over-confident, and more data does not fix it.

Condition	P5–P95 coverage of the true final settlement
U = 25 %	40 %
U = 50 %	68 %
U = 75 %	**66 %**
noise 5 / 10 / 20 mm	51 / 57 / 67 %
laboratory quality ×0.5 / ×1 / ×2	71 / 56 / 51 %
pooled	**58 %** [52–64] against a nominal 90 %

Against a nominal 90 %. Note the third row: with the *most* monitoring the interval is narrowest (median width ~5 % of the settlement) yet covers the truth no better than at half consolidation (66 % against 68 %). The lowest coverage of all is at the *shortest* record (40 %), where the centre itself is not yet located. The point of this row is different: as data accumulate, the band narrows faster than the error shrinks, so it stops improving precisely where it looks most credible. **Report the P5–P95 range as indicative; do not use it as a design envelope.**

The range you are shown is the survivor of a systematic search for a better one. Eleven candidate constructions were built and measured against this suite; the two that held up across every noise level and every laboratory-quality level are what ship, and they are described below. The others improved matters in the conditions they were designed for and failed when those conditions changed.

That is worth one sentence of guidance: treat any uncertainty estimate, in this tool or any other, as unproven until it has been measured across different data quantities, noise levels and input quality. An interval that has only been checked under one set of conditions is a claim, not a measurement.

The band that replaced it — width from YOUR uncertainty, not from the data

The $\Delta\chi^2$ band answers “how tightly do my *data* pin this?” After three quarters of the settlement has occurred, the data pin the parameters very tightly, but to slightly the wrong place. The band therefore reports ±2.5 % around an answer that is 4 % out. A second band was therefore built (engineer’s proposal, 2026-08-06): **same winner at the centre, width taken from the Lo/Hi the engineer typed**. Laboratory scatter does not shrink because a settlement plate was read for another six months, so this band cannot collapse as data accumulate. 200 independent Latin-Hypercube draws per layer, log-uniform over the typed range.

	Δχ² band (data-resolvable)	**Input-uncertainty band**
U = 25 %	40 %	43 %
U = 50 %	68 %	**96 %** [89–98]
U = 75 %	66 %	**91 %** [83–95]
pooled	**58 %** [52–64]	**77 %** [71–81]
median width	7.5 %	18.5 % (×2.5)

KEY IDEA

The defect is removed. The $\Delta\chi^2$ band narrows 16.6 % → 5.1 % as data accumulate while its coverage *falls*; the input-uncertainty band holds ~18–20 % and its coverage *rises* to 91 % against a nominal 90 % — essentially calibrated. Paired McNemar over 270 runs: 61 runs covered where the $\Delta\chi^2$ band missed, 10 the other way, **p < 0.0001**. Better at every noise level and every laboratory-quality level tested, including deliberately poor laboratory data (67 % vs 51 %). It is now the headline range on the result page; the $\Delta\chi^2$ band is retained beneath it, labelled as what the settlement record alone can resolve.

WATCH OUT

It does not help below about half consolidation (43 % vs 40 %). There the winner itself is mislocated, and no width around a wrong centre recovers it. This strengthens rather than softens the rule that a usable back-analysis needs the record to reach U ≈ 50 %.

A third defect, found only by running the real app

WATCH OUT

The C_c/C_r admissibility cap was set tighter than real soils go. Every recipe the search considers must satisfy $C_c/C_r \in [c_{min}, c_{max}]$, defaulted to **[5, 10]** — i.e. C_r between 10 % and 20 % of C_c. Published ratios run roughly C_r/C_c = 0.05–0.2 (Mesri & Godlewski and successors), i.e. C_c/C_r ≈ 5–20, so the upper limit of 10 *rejected ordinary soft clays*. Two consequences, both invisible until the inversion was driven end-to-end in the shipped build: **(i)** ~74 % of the input-uncertainty draws were discarded, collapsing that band from ~18 % to ~4 % width and making it look far more precise than the data support; **(ii)** the TRUE parameters of four of the five internal validation profiles (C_c/C_r up to 11.25) were themselves inadmissible — *the search could not have accepted the correct answer.* The default is now **[4, 20]** (C_r ≈ 5–25 % of C_c), still editable in the caps modal. The validation programme now applies exactly the same rule as the tool, so the measured figures below describe the estimator you actually run.

Two defects fixed

- **k_0 search bounds.** A generic "never let a positive parameter reach zero" floor of 1×10^{-9} was written when rates were entered as c_h (m^2/day, typical 0.1–10). After the tool moved to permeability, k_0 in m/s runs 10^{-11}–10^{-8} — so that floor sat *inside* the physical range. For any layer with $k_0 < 10^{-9}$ m/s the search range excluded the engineer's own typed value and could only increase it; below ~3×10^{-10} k_0 was dropped from the search entirely, silently. On the tool's

own worked example the true k_0 of the second layer was unreachable. k_0 now carries its own bracket, 10^{-13}–10^{-4} m/s.

- **Inert parameters no longer searched.** A layer with the drain switched off has a no-flow inner boundary, so its *horizontal* permeability cannot influence a single observation. k_0 and C_k were nevertheless being fitted there, wasting a search dimension and reporting an artefact as a calibrated value — one validation run returned k_0 = 15 m/s, faster than free-falling water. Pinning that value to a sane range changed the forecast by *exactly zero*, confirming the parameter was inert. Such layers now hold k_0 and C_k constant.

Two transparency additions

- **Bound-hit flag.** Any recovered value sitting on the edge of its search range is now flagged on the results page. A parameter stopped by its own limit is not a measurement. The data were pulling further and the boundary stopped them, so the number reported is set by where the limit was placed. This is how the 15 m/s value above was reported without complaint.
- **Under-observed layers held inside the typed window.** A layer whose settlement has barely started is not constrained by the record: what moves its parameters is the fit to the *other* layers. Where a layer has ≥ 1.5× as much settlement still to come as the best-observed layer in the profile, its search is now kept inside the typed Lo/Hi (no 3× expansion, k_0 held to ×0.3–×3.0). The results page lists every layer, its remaining fraction and whether it was held.

> **WATCH OUT**
>
> **What the layer-hold does and does not do.** On profiles with a strong observability imbalance — a fast shallow layer over a slow deep one — it roughly halves the median held-out forecast RMSE — **161 → 74 mm** over six paired runs (167→83, 155→27, 94→64, 58→34) — and improves the recovered magnitudes of the held layers (C_c error 32 %→14 % and 27 %→13 %; the deepest layer 18 %→7 % and 28 %→8 %). Where the profile is evenly observed it does not fire at all — verified bit-identical in 9 of 9 paired runs. **It did not rescue the worst cases**: the worst run stayed at ~18–20 % forecast error at both truncations (439→395 and 419→434 mm). Treat it as an accuracy gain on typical projects, *not* as protection against the outlier. An earlier *absolute* version of the same rule, which fired on every layer when the whole profile was early, measurably worsened the tail (worst error 23.5 %→28.1 % over 15 paired runs) and was replaced by the relative form for that reason.

30.8 Ballina — the test against a real embankment

Everything in §30.3–§30.7 is an *inverse crime*: the same forward model both generated and inverted the data, so model error is zero by construction. The Ballina NFTF trial embankment (Pineda et al. 2016; Kelly et al. 2018; Lim et al. 2018) is the one test with real soil, real construction and real measurement error. It is a 3 m embankment on PVD-improved marine clay, on a 1.2 m square grid, monitored for about three years. The tool was given the measured record up to a cut day and asked to forecast to day 1080.

Which measured record, and why it matters. Two published records exist for this embankment and they do not agree (§30.8.1). The results below are scored against **M0**, the top extensometer magnet, whose final reading is **1427 mm** at day 988 (the digitised tail wobbles by ±16 mm — day 974 reads 1444 mm — vector-extraction noise, not real heave). M0 is used because its coordinates are extracted exactly from the vector artwork of Huang, Kelly et al. (2025) rather than traced by eye from a printed figure; because it is a single defined instrument rather than a mean over four plates at different positions; and because it is the record the comparator of Chapter 31 targets, which makes that comparison like-for-like.

Record given	Readings used	Forecast at day 1080	Error vs the final reading	$\Delta\chi^2$ band	Input-uncertainty band
to day 200	11	1293 mm	**−9.4 %**	1.7 % wide — misses	13.3 % wide — misses
to day 400	25	1406 mm	**−1.5 %**	7.4 % wide — covers	16.0 % wide — covers
to day 700	34	1416 mm	**−0.8 %**	4.3 % wide — covers	17.5 % wide — covers

How these errors are scored. The M0 record ends at day 988; the forecast runs to day 1080 and is compared against that last reading carried forward. Scored strictly at day 988, where a real measurement exists, the three errors are −10.3 %, −2.4 % and −1.9 % — about a percentage point worse than the table, and the honest figures to quote if you need a like-for-like epoch.

KEY IDEA

What the field test does show. With a year or more of monitoring the forecast lands within about 2 % of the measured settlement two years ahead (held-out RMSE 26 mm at the 400-day cut, 24 mm at 700 days; every held-out point inside ±20 %). With six months it is about 10 % low — broadly the record-length dependence the synthetic work predicts, though on real soil, where model error is *not* zero, the two cannot be separated from a single site (see the caveat below).

WATCH OUT

What it does *not* show — and this cuts against an earlier claim.

- **Ballina does not corroborate the over-confidence finding.** An earlier version of this section reported that the $\Delta\chi^2$ band missed at all three cut-offs and cited that as the strongest evidence in the guide. Scored against M0 it *covers* two of the three, and its width does not narrow monotonically (1.7 → 7.4 → 4.3 %). With three cut-offs on one site this test was never able to resolve a coverage question either way. **The over-confidence result now rests on the 270-run study of §30.7 alone**, and that is where it should be judged.
- **Both ranges still fail at six months.** At the 200-day cut the central estimate is about 10 % low and neither range contains the truth — and note that this cut already covers **two thirds** of the settlement, so it is *not* the below-half regime the synthetic work identifies. A band cannot rescue a mislocated centre. What put the centre in the wrong place here cannot be settled from one cut-off: real model error is a candidate the synthetic suite cannot produce, but §30.7's under-observed-layer study shows a short record can do it too on a fast-shallow-over-slow-deep profile — which is what this site is. Read it as a warning that the half-settlement rule is necessary rather than sufficient, not as a diagnosis.
- **The un-inverted laboratory parameters over-predict by 8.4 %** (+121 mm at day 974). Measured at day 974 the inversion beats them at the 400- and 700-day cuts and is beaten by them at 200 days. So back-analysis earns its place here only once the record is long enough. (Over the four held-out checkpoints of the 200-day cut, days 398, 691, 900 and 1080, with the last against the final reading carried forward, the inversion is still ahead: 115 mm against the typed baseline's 129 mm, and still ahead on the three measured days alone.)
- **Recovered parameters are not physically clean.** k_0 spans several orders of magnitude across a fairly uniform marine clay, and layers sit on the C_c/C_r cap. Use the forecast; do not carry these values to another site. The bound-hit flag exists for exactly this.

30.8.1 Two records for one embankment

The settlement of this embankment has been published twice, and the two records disagree by far more than either paper suggests:

Day	SP1–SP4 plate mean	M0 extensometer	Difference
100	633 mm	719 mm	+86 mm (+13.6 %)
200	837 mm	942 mm	+106 mm (+12.6 %)
292	942 mm	1118 mm	**+176 mm (+18.6 %)**
500	1119 mm	1270 mm	+151 mm (+13.5 %)
700	1264 mm	1317 mm	+53 mm (+4.2 %)
974	1431 mm	1443 mm	**+12 mm (+0.9 %)**

The pattern is specific and worth reading carefully. The two records disagree about *how fast* the ground settles, by as much as 18.6 % in mid-record. They agree to within 1 % about *how much* it finally settles. They are not offset by a constant; the gap opens and then closes.

That is what lateral variability across a single footprint looks like. M0 sits at one point; SP1–SP4 are plates at different positions averaged together. Where the soil profile, the drain performance or the fill thickness varies across the site, different locations reach the same final settlement along different time paths. A single instrument is therefore a measurement of *that location*, and a plate average is a measurement of nothing in particular — which is the practical argument for scoring against a defined instrument.

> **WATCH OUT**
>
> **Stated plainly, because it affects every number above:** reading error in the hand-digitised plate record cannot be excluded as a contributor, and with two records we cannot separate real spatial variability from digitisation drift. What we can say is that *a back-analysis is only ever as good as the record it is fitted to*. Fitting the plate mean instead of M0 moves this tool's day-1080 forecast by 10 mm at the 200-day cut, 157 mm at 400 days and 12 mm at 700 days — and the 400-day figure alone is larger than any parameter effect measured anywhere else in this part. Before trusting a back-analysed forecast, be sure you know which instrument produced your settlement record and what it represents.

30.9 What this tool does not model

- **Well resistance** is neglected, and for most modern PVD installations that is justified rather than assumed. It is negligible when the discharge factor $D = q_w/(k_h \cdot l_m^2)$ exceeds about 5 (Mesri & Lo 1991), which for typical clays requires only 2–80 m^3/year of discharge capacity; band drains marketed today deliver of the order of 100–500 m^3/year (Hansbo 1997; Holtz et al. 1991; Rujikiatkamjorn & Indraratna 2007; Abuel-Naga et al. 2015). The tool therefore imposes $u = 0$ at the drain wall. Check D for your own project — the input page computes it from your typed k_h, the drain length and the specified q_w — and treat the answer with caution when $D < 5$: very long drains (the length enters squared), organic soils where discharge capacity degrades with time (Miura & Chai 2000 measured up to fivefold reduction over 100–300 days), and kinked drains are the cases where the neglect can bite, and a G2 (rate) mismatch that smear cannot explain may then be well resistance. The exact loading-schedule, U_s-denominator and vacuum-attenuation semantics are specified normatively in Appendix A.
- **Loading history.** Multi-stage loading is supported as an input schedule, but how the solver treats stages falling *inside* the truncated window deserves explicit documentation before relying on staged-load back-analysis.
- **Minimum usable record.** $U \approx 50$ % is quoted as typical, not a floor: below $U \approx 40$–50 % the magnitude family is essentially unconstrained and the band collapses to the prior; the Asaoka cross-check is itself unreliable below $U \approx 60$ %.
- **Determinism.** Deterministic *up to solver tolerance*: same inputs + same solver grid reproduce the same answer, but parabolic refinement, best-sampled guards and floating-point line searches make the last significant figures order- and platform-sensitive. Audit by pinning the grid and inputs; do not expect bit-for-bit identity across machines.

CHAPTER 31

Sweep Search compared with Bayesian back-analysis

The natural comparator is Huang, Kelly et al. (2025), *Efficient Class C prediction of PVD-improved soft clay settlements* (*Engineering Geology* 346, 107903). It back-analyses the same Ballina embankment using BUS (Bayesian Updating with Structural reliability methods), wrapped around a 1-D consolidation model with the general simplified Hypothesis B creep formulation. It is a strong, carefully executed piece of work, and the comparison below is not a defence of one tool against another — it is an attempt to say precisely *where each method places its trust*, because that is what determines which one suits your site.

31.1 What actually makes the paper's 76-day result work

The paper's headline is that 76 days of monitoring suffice. At day 76 the embankment is roughly 46 % consolidated, barely past end-of-fill, so this deserves scrutiny rather than acceptance. The mechanism is visible in its priors. **λ (compression index) is lognormal with COV 0.3, derived from CRS tests; k_v is lognormal with COV 3.5**, the latter so wide as to be nearly non-informative. The division of labour is therefore:

- **Settlement magnitude** is supplied largely by the *laboratory prior*.
- **Rate** is what the 76 days of field data actually determine — and early curve shape does constrain rate well.

The paper is consistent with this reading: posterior λ comes out “close to the prior values… as expected, as the prior mean of λ at each layer is derived from the CRS test results”. The paper also concedes the laboratory→field gap directly, introducing a fitted correction factor $\alpha \in [0.6, 1.0]$ because laboratory strain rates “lead to the overestimation of OCRs”. And no sensitivity study on the prior COVs appears anywhere in the paper, so the dependency is untested by its authors.

NOTE

How firmly is this established? Partially. The attempt to settle it is worth reporting honestly. A Bayesian inversion was built using the paper's prior *structure* (C_c COV 0.30, k COV 3.50), with the compression-index prior deliberately biased by ±30 % to see whether 76 days of settlement would correct it. Within that setup the day-1080 forecast moved by **41 % of baseline** — the data pulled every arm in the same direction but did *not* bring them back together, which supports the prior-driven reading.

But that exercise did not reproduce the paper's result, and the claim should be read accordingly. Its baseline arm returned +63 % error at day 1080 where the paper reports ≈ +1.6 %. Four things differed from the original. Only **four** digitised observations were available in the day-20–76 window, against the paper's full monitoring record. The forward model was this tool's `solveRC2D`+EVP rather than 1DSimp_B with the generalised simplified Hypothesis B. The parameterisation was two global scale factors rather than per-layer λ, κ, φ, k_v, OCR with correction factors. And the sampler was a short Metropolis chain rather than converged subset simulation. Changing the data, the physics, the parameterisation and the sampler and then obtaining a different number is not evidence against the original. **The mechanism is plausible and directionally supported; it is not established.**

One byproduct of that attempt does stand on its own, and it favours the paper. A first pass using prior-sampling importance weights collapsed to an effective sample size of **1–2 out of 400** — the "posterior" was one lucky draw. A likelihood that concentrated against a COV 3.5 prior is precisely the regime BUS and subset simulation were designed for. **Their choice of sampler is well-justified**, and a naive Bayesian implementation would fail where theirs works.

NOTE

Reproducibility caveat — applying to this comparison and to both methods. The paper publishes its predictions only as figures; neither the monitoring record nor the code is provided. Every number attributed to it in Chapter 31 was therefore obtained by extracting vector path coordinates from its Fig. 25 — exact coordinates rather than traced pixels, and cross-checked, but still a reconstruction. That limits how far any independent audit of its inversion can go, and it is why the reproduction above could not be made faithful.

This is a criticism of *publication practice*, not of the work, and it is near-universal in this literature — **it applies to this tool too** until its data and code are released. It should not be read as doubt about the paper's result, which is independently corroborated. The same figure carries two further analyses of the same embankment by another team, both tracking the measured record alongside it: Tian et al. 2022 (RSM-based, 26.8 mm RMSE) and Tian et al. 2023 (FEM-based, 34.7 mm). Read that corroboration carefully, though. All three share the same inference engine (BUS) and differ only in their forward model. The two Tian papers share four authors. And both used data to day 682, so over days 200–600 their curves are largely in-sample and are weak corroboration of a 76-day forecast.

31.2 The Sweep Search's argument, stated properly

The three knobs are not an arbitrary simplification. They are a **mechanism-aligned decomposition of the subspace the data can actually resolve** (`sweep_inverter_k3.js`):

- **α_A** — parameters that *increase* settlement (C_c, C_r, C_α/C_c)
- **α_B** — parameters that *decrease* it (σ'_p, β)
- **α_S** — rate (k_0 only; C_k is held or line-searched, never ridden by this knob)

A settlement record resolves magnitude-up, magnitude-down and speed — and very little else. Rather than sampling a high-dimensional posterior that is mostly null-space, the Sweep Search **parameterises the identifiable manifold directly and declines to search the directions the data cannot see**. The evidence for that stance is in this part's own validation work: a 150 % error in k_0 (the $\eta \leftrightarrow k_0$ see-saw) produces a **0.1 % settlement error at day 900**. Parameters are non-identifiable; predictions are identifiable. The paper meets the same physics from the other side and reports **multimodal posteriors**, which it acknowledges but does not resolve.

Two further choices are statistically principled rather than expedient:

- **Min-over-S is a profile likelihood.** For each magnitude recipe the score is taken as the best over speed (`project()`, `sweep_inverter_k3.js`) — profiling out the rate as a nuisance direction, which for *prediction* is arguably better suited than marginalising over it.
- **The two-corner objective optimises the band, not a point.** $J = \tfrac{1}{2}[\chi^2(\text{low-settlement corner}) + \chi^2(\text{high-settlement corner})]$ means the search optimises *the interval it intends to report*. Most inversion schemes optimise a point estimate and attach uncertainty afterwards.

31.3 The typed band is a hypothesis, not a weight — and it is falsifiable

This is the difference that matters most in practice, and it is easy to miss. The typed Lo–Hi range is **not a hard constraint**. Per-edge domain extension (`sweep_inverter_k3.js`) widens any wall the winner crowds within 5 %, up to `EXT_CAP = 2` steps of `EXT_STEP = 0.5` on the magnitude knobs. Two things are worth separating here. The **3× expansion is always on**: a typed ±15 % band is searched over roughly **×0.63…×1.56** on every run, whether or not the data ask for it. The **domain extension** is the part that fires only under pressure, and with two steps per side it carries the search out to about **×0.25…×3.85**. Comparing either to the paper's lognormal λ prior is not like-for-like, because a hard search interval and an unbounded distribution are different objects. But the searched domain is certainly not narrow.

The consequences are worth spelling out:

- **Good laboratory data are not wasted.** The winner sits comfortably inside the typed window, no extension fires, and the search converges quickly in a small trusted region. The method performs like the paper's because the problem genuinely *is* easier.
- **Poor laboratory data are survivable.** The extension fires and the search relocates to where the data point.
- **A wrong typed range announces itself.** The extension is logged — "`[SWEEP] winner in the 5 % zone of the αA-LOW wall — EXTENDING knob-A domain to [-0.5, 1.0]`", or for the speed axis "`[SWEEP] winner's SPEED in the 5 % zone of the αS-LOW wall`". That is an explicit, visible signal that your typed knowledge disagreed with the measurements.

Contrast a Bayesian prior. A prior that is wrong does *not* announce itself: it biases the posterior quietly and in proportion to its stated confidence, and the credible interval remains narrow and authoritative. Detecting it requires a prior-sensitivity study, which the paper does not perform and which no practitioner will run routinely. **The Sweep Search's prior is a falsifiable starting hypothesis; the Bayesian prior is a fixed weight.** On sites with thin or disturbed sampling, which is most sites, that is a material advantage. It is the strongest practical argument for this method.

31.4 Limitations — the real ones, and the ones that dissolve on inspection

Earlier drafts of this section listed five weaknesses. Three did not survive examination, and two of those were *strengths* misread as faults. What follows separates them, because a list of limitations that quietly applies a different standard to each method is worse than no list at all.

Real limitations.

- **The band can degenerate on a sharply-peaked misfit surface.** When one survivor dominates the weight vector, the cumulative weight crosses the 5 % and 50 % thresholds at the same sample and P50 collapses onto P05 — observed on real data. Percentiles taken over a near-degenerate weight vector are not percentiles. This is a genuine defect, not a caveat.
- **One real site for the inversion itself.** Shared with the paper, but real. The forward model is checked against closed forms and degenerate limits, and the inversion against a synthetic recovery suite — but synthetic cases generated by the same solver that inverts them validate the machinery, not the physics.
- **No published cross-site study of fitted-versus-laboratory parameters.** The mechanism runs on every inversion (below), but its results have not been collected across many sites and written up. That is a gap in the *evidence*, not in the tool.

Not limitations.

- **"No sequential updating."** Sequential updating is an *approximation* adopted when re-processing the full record is too expensive; carrying a sample-based posterior forward as a prior degrades it at every step. Re-inverting on the complete dataset is the exact answer and costs minutes here. Where batch re-inversion is cheap, starting from scratch is **better**, not worse.
- **"It returns no posterior."** A posterior over parameters the data cannot identify is dominated by the prior exactly where evidence is weakest — more authoritative-looking, not more informative. The deliverable is a settlement forecast, and on that object this tool reports a **measured** coverage: 77 % over 270 tests, and 96 % once monitoring reaches half the settlement (§30.3). The paper reports mean ± 1 SD (≈ 68 % nominal) that is never coverage-checked. A distribution is genuinely required only for formal reliability-based design.
- **"Held parameters carry no uncertainty."** C_k and β are held here *by default*; the paper's parameter set (Tables 6–7) contains **no analogue of either** — k_v is one constant per layer and yield is not smoothed. Only one of the two methods gives the engineer a choice: every parameter here carries a per-layer *Constant / Variable* control (`CkLocked`, `betaLocked`,

`CaCcLocked`, `CcLocked`, `CrLocked`, `sigmaPLocked`, `koLocked`). Set it Variable and the search fits it. (E_u deliberately renders with no Constant/Variable toggle at all: it is a typed stiffness, never a search variable.) In the paper's formulation no such switch exists, because the physics is absent from the model form. **Modelling a parameter and letting the user hold or release it is a strictly larger capability than not modelling it at all.**

- **"Per-parameter bands under-cover."** They do — measured on the A–E suite against a nominal 90 %: C_c 95 %, σ'_p 100 %, k_0 57 %, C_r 48 %. But this is a property of *the inverse problem*, not of this tool: settlement records resolve magnitude-up, magnitude-down and rate, and little else. The paper meets the identical physics and reports **multimodal posteriors**; its per-parameter coverage is *never measured*, so nothing suggests it fares better. The asymmetry is only in the reporting. And the deliverable is a settlement forecast, on which measured coverage is 77 % (96 % with a record reaching half the settlement). **Measuring your own weakness and publishing the number is a scientific virtue, not a defect** — and it is why these figures appear in this part rather than being left for a reviewer to find.
- **"Recovered parameters are never checked against the laboratory."** They are checked on *every run*. The typed Lo–Hi range *is* the laboratory data (§22.1: oedometer scatter within a layer is your band), so the fitted values are reported directly against it, per layer, every time you invert. When the fit cannot be reconciled with the lab range the domain extension fires and **says so in the log**. The paper performed the equivalent check once, as a published figure, for one site. A standing, automatic, per-layer lab-versus-field comparison that flags its own disagreements is *more* transparent than a one-off validation exercise, not less.

Evidence base — a fair accounting. On the *inversion* side this tool carries a five-case synthetic parameter-recovery suite (with noise and mis-seeded-smear variants), measured forecast and per-parameter coverage, and the identifiability and equifinality studies of Chapter 30. On the *forward model* side it is checked against the Hansbo closed form (< 1 % U RMSE), the Carrillo combination, the Terzaghi limit, cross-kernel agreement to 0.2 %, and 81 unit tests. The paper validates its forward program against a second real embankment (HKIA Chek Lap Kok, Yin et al. 2022), which is a genuine strength this tool does not match. It then demonstrates its inversion at Ballina. **For the inversion itself, both rest on one real site.**

31.5 What the Ballina comparison does and does not show

All methods are scored against the identical measured record (M0, the top extensometer magnet, extracted from the paper's Fig. 25 by vector path extraction) at the same four checkpoints — days 200, 400, 600 and 800. Every method clears the ±20 % criterion at every checkpoint. The complete results:

Method	Record used	RMSE over the four checkpoints	Checkpoints genuinely held out	RMSE over held-out points only
Paper (BUS + Hypothesis B)	to day 76	**11.1 mm**	**4 of 4**	**11.1 mm**
Tian (2022), RSM-based	to day 682	26.8 mm	1 of 4	32.7 mm
Tian (2023), FEM-based	to day 682	34.7 mm	1 of 4	**7.3 mm**
Sweep Search	to day 292	26.9 mm	3 of 4	**30.4 mm**
Sweep Search	to day 400	18.4 mm	2 of 4	8.3 mm
Sweep Search	to day 700	4.9 mm	1 of 4	6.4 mm

> WATCH OUT
>
> **Read the last two columns before the third.** The four checkpoints are fixed, but the cut-off is not, so for the longer Sweep Search runs most of those points lie *inside* the calibration window. The day-700 row's 4.9 mm is three-quarters an in-sample fit statistic, and its "held-out RMSE" is a single residual — not a quantity that can be set against the paper's four-point value. Quoting it as a win would be the same error as quoting only the cuts that flatter the tool, pointing the other way.

Read plainly. On the only near-comparable row (day-292, three of four points held out) this tool gives **30.4 mm against the paper's 11.1 mm**. And it needed *nearly four times* the monitoring record to get there. The paper's result is the better one on this site by a clear margin, and the margin is *wider* than a bare four-checkpoint comparison suggests. Against the Tian baselines the reading depends on which column you use, and the honest one is the harder one. On the four-checkpoint statistic Sweep Search overtakes both once its record reaches 400 days (18.4 mm against 26.8 and 34.7). But those two rows are 682-day analyses, so three of their four checkpoints are in-window too — and on the *held-out* day-800 point Tian 2023 scores **7.3 mm against this tool's 8.3 mm**. It does not overtake there. Against the paper it does not overtake on any column: 76 days against 292.

What it does not show is a data-efficiency gap in this tool's favour. An earlier draft of this comparison reported one; that claim is withdrawn. Sweep Search could not be tested at day 76 on the M0 record at all, because Fig. 25 does not plot the data behind the paper's own day-76 curve — its markers begin at t = 63 d, leaving a single usable observation. Comparing the paper at 76 days with this tool at 400 is a statement about what could be tested, not about what either method requires.

> **Provenance, because this table has been wrong once.** The figures come from `h2h_numbers.json`, regenerated on 2026-08-08 by `regen_h2h.py` from `ballina_cmp_m0_{292,400,700}.json`. An earlier edition of this section quoted 39.9 / 25.3 / 22.6 mm from a derived file that had not been regenerated after its inputs were re-run; those numbers described a superseded configuration of the tool. If the Ballina runs are repeated, regenerate the derived file before quoting from it.

31.6 Would better inputs close the gap? A direct test

The obvious reading of §31.5 is that the Bayesian route wins on this site because it *started* from better soil parameters — and that this tool, given the same numbers, would land in the same place. That is a testable claim, so it was tested rather than argued.

The paper's prior means (its Tables 3, 6 and 7) were converted into this tool's parameter set and used as the typed baseline. The conversion: $C_c = \lambda \cdot \ln 10$, $k_0 = k_v / 86400$, $C_r = C_c/\alpha_1$, $C\alpha/C_c = \alpha_2$, and $\sigma'_p = \alpha \cdot OCR \cdot \sigma'_0$, each at the midpoint of the paper's uniform prior ($\alpha_1 = 7.5$, $\alpha_2 = 0.04$, $\alpha = 0.8$). Geometry, drain, loading schedule and the measured record were left untouched, so the soil baseline was the only thing that changed.

Record given	This tool's own laboratory baseline		The paper's prior means	
	day-1080 error	held-out RMSE	day-1080 error	held-out RMSE
to day 292	**−3.4 %**	**46 mm**	+8.9 %	91 mm
to day 400	**−1.5 %**	**26 mm**	+6.6 %	79 mm
to day 700	**−0.8 %**	**24 mm**	+5.6 %	61 mm

> **KEY IDEA**
>
> **The explanation fails.** Transplanting the paper's parameters into this tool multiplies its held-out RMSE by two to three times at every cut-off (×2.0, ×3.0, ×2.5). Whatever produces the paper's day-76 result, it is not a set of numbers that can be lifted out and dropped into a different forward model.

Why, in one table. Running both baselines forward with no fitting at all, against the same measured record:

Typed forward, no fitting	day 200	400	600	800	974	RMSE
This tool's laboratory table	−75	+58	+112	+153	+121	109 mm
Paper's prior means	**−329**	−181	−81	**+8**	**+19**	172 mm

The paper's prior is *excellent* on final magnitude, coming within 20 mm at days 800 and 974. It is *poor* on rate, running 329 mm slow at day 200 through this solver. That is exactly what its own prior specification says it should be: λ is given a COV of **0.30**, k_v a COV of **3.5**. The magnitude is pinned a priori; the rate is left almost free for the data to place. A prior mean that loose was never intended as a point estimate, least of all in someone else's model.

> **An aside that corroborates §31.4.** The paper constrains $\alpha_1 = \lambda/\kappa$ to [5, 10], which forces C_c/C_r into that range. Ballina's measured C_c/C_r runs 15.4–17.4 through the soft clay below about 3 m (7.8–17.4 across the full profile, from Lim et al. 2018 Table 2), so for most of the profile the paper's prior *excludes* the measured ratio — and it still forecasts well. That corroborates this part's own finding that the forecast is *insensitive* to C_c/C_r. Two cautions, though. It is not independent evidence: it rests on this part's reconstruction of the paper's figures and on its own λ/κ mapping. And insensitivity of the forecast is not the same as identifiability of the parameter. Note also that the same [5, 10] window, used as an *admissibility cap* in this tool, was measured in §30.7 to reject four of five validation profiles and 74 % of band draws — harmless as a prior on a forecast the ratio barely moves, harmful as a hard filter on sampled recipes.

What survives. The gap in §31.5 is real and is not explained away by input quality: on the nearest like-for-like comparison the paper achieves 11.1 mm from a 76-day record against this tool's 30.4 mm from 292 days, and this test removes the most comfortable excuse for it. The mechanism worth taking seriously is not better numbers but *fewer free parameters in the directions that do not matter*. The paper updates two variables and hard-bounds three ratios. The per-layer sweep, in these runs, moved three magnitudes (C_c, C_r, σ'_p) and one rate parameter (k_0); β, $C\alpha/C_c$ and C_k are all Constant by default, and all were held in these runs.

31.7 Choosing between them

The two methods are not really competitors; they suit different *states of knowledge*.

- **Use the Bayesian route** when you have high-quality CRS data from good samples, need an early call from limited monitoring, want parameters you can defend and reuse, or need a posterior to feed reliability-based design.
- **Use the Sweep Search** when site characterisation is thin or sampling is suspect, when the deliverable is a settlement forecast rather than a parameter set, when you need geometry the 1-D route must proxy (smear, partially-penetrating drains, vacuum), or when you are practising the observational method and need to re-invert as data arrive — minutes per run supports that workflow; subset simulation over hours does not.

The honest synthesis is that the best tool is neither: BUS-style inference wrapped around a 2-D resolved forward model, with the identifiability-driven knob decomposition used to define the sampling subspace. Both halves exist; nobody has joined them.

CHAPTER 32

The validation campaign: 1080 sites against known truths

The reliability numbers quoted throughout this part — the design range's 77–79 %, the χ^2 region's 58–60 %, the middle row's 89–91 %, the 48 / 63 / 93 truncation staircase — all come from one September 2026 measurement campaign. This Part assembles the whole campaign in one place: how a synthetic site is built, what ground the test bed spans, what was measured, how the counting works, the complete results, and — stated as plainly as the successes — what the campaign does and does not establish. Acceptance criteria were registered in the campaign scripts before any production run, and every result is reported, including the one construction that failed its first exam.

IN A NUTSHELL

You cannot check a weather forecaster with one storm — you need many storms whose outcomes you know. The same holds for a settlement range: its claimed reliability can only be tested against sites where the true final settlement is known. Real sites never reveal their truth on schedule, so the campaign builds 420 synthetic ones from known soils, hands each one's noisy, cut-short monitoring record to the app exactly as an engineer would, waits for the app to commit to its ranges, then opens the envelope and counts. Ninety-plus percent of everything this part claims about reliability is those counts.

32.1 How one site is built — and one worked all the way through

Figure 32.1 shows the anatomy of a single campaign site. A *true soil* is drawn and kept hidden. Its forward solution — the settlement history that ground would really exhibit — is sampled by three settlement plates on a logarithmic schedule, corrupted with reading noise, and cut off where the record reaches a quarter, a third, a half or three quarters of the final settlement. Separately, the true soil is passed through a simulated laboratory: each parameter the campaign perturbs is multiplied by a lognormal error, producing the *typed values* — the plausible-but-wrong numbers an engineer would enter — and their Lo/Hi windows. The pipeline receives exactly what a real user has (the typed soil, the windows, the truncated noisy record), runs Steps 1–7 and Step 8 unmodified, and commits to its best estimate and ranges. Only then is the truth revealed and the hit or miss recorded.

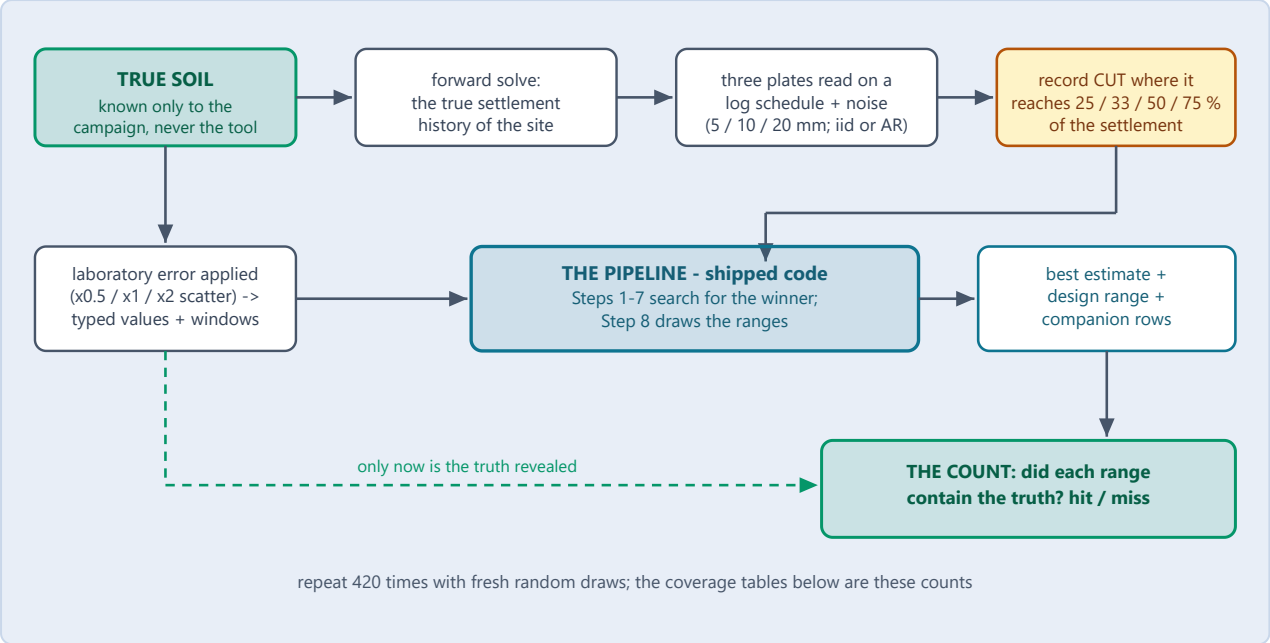


Figure 32.1. How one campaign site works: a hidden true soil generates the noisy, truncated record; its laboratory-disturbed twin supplies the typed values; the shipped pipeline sees only those two, commits to its ranges — and only then is the truth revealed and the hit or miss counted.

Figure 32.2 shows one such site worked end to end — profile A at a half-record, standard noise and laboratory. The grey dashed curve is what the typed values alone predict: visibly wrong from the start, drifting to a final settlement 13 % high. The dark readings are all the pipeline is given. From them, Steps 1–7 recover the teal winner curve, which hugs the hidden green truth so closely beyond the cut that the two are hard to separate — and Step 8's design range (991–1168 mm) comfortably brackets the true 1113 mm. This one picture is the entire campaign in miniature; the tables that follow are simply this exercise repeated 420 times and tallied.

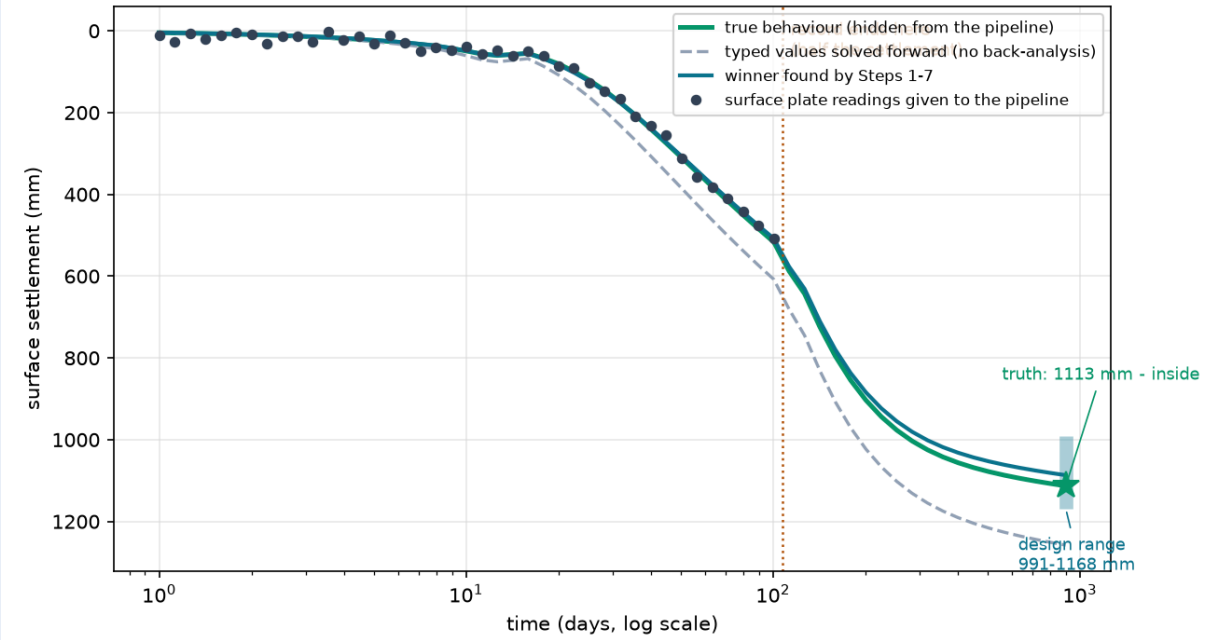


Figure 32.2. One campaign site end to end (profile A, half-record): from the dark readings and the grey-dashed typed start, the pipeline recovers the teal winner, and its design range brackets the hidden truth (star). Dotted line = where the record was cut.

32.2 The test bed: nine soils, and what varies site to site

Five base profiles (Figure 32.3) span the ground conditions the tool is built for, deliberately including its harder features — multi-stage loading everywhere, vacuum assistance in one profile, a partially penetrating drain in another, and a deep peat–clay stack whose parameters range over an order of magnitude:

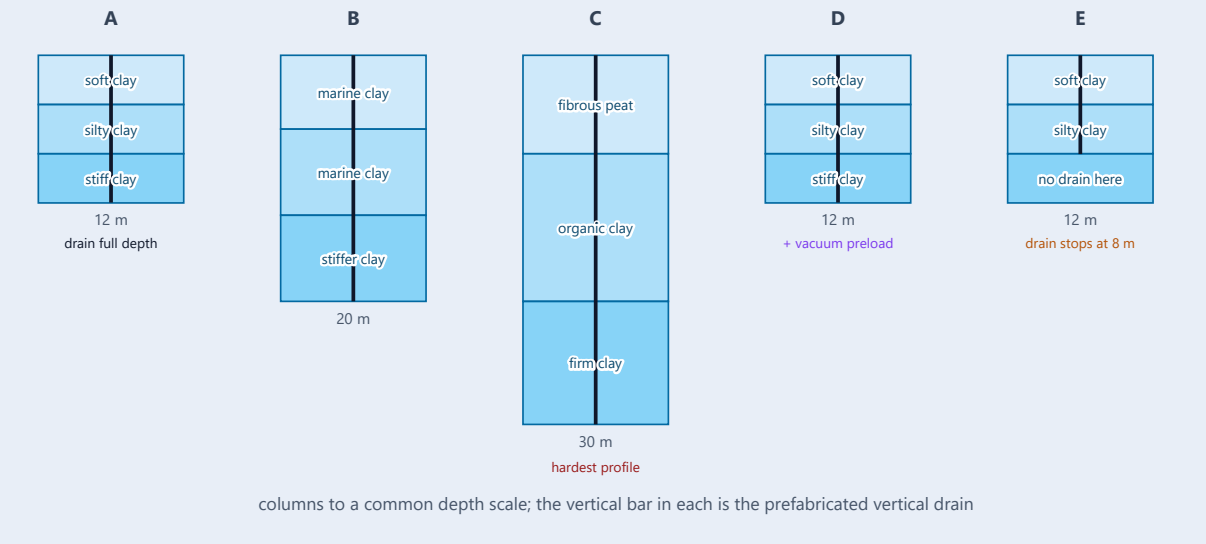


Figure 32.3. The five base profiles to a common depth scale: A, shallow estuarine clay; B, deeper marine deposit; C, the hostile 30 m peat–clay stack; D, profile A under vacuum preloading; E, profile A with the drain stopping at 8 m.

Profile	Character	Depth (layers)	C_c	C_r	σ'_p (kPa)	e_0	k_0 (m/s)	η	Loading
A	shallow soft estuarine clay	12 m (3)	0.20–0.45	0.020–0.040	11–93	1.0–1.8	7×10^{-10}–2×10^{-9}	3	2 stages
B	medium-depth marine clay	20 m (3)	0.28–0.55	0.030–0.060	14–132	1.3–2.2	4×10^{-10}–1×10^{-9}	4	3 stages
C	deep stratified peat–clay	30 m (3)	0.30–1.20	0.040–0.150	4–129	1.4–5.0	3×10^{-10}–3×10^{-8}	5	3 stages
D	vacuum-assisted preload (Profile-A soil)	12 m (3)	0.20–0.45	0.020–0.040	11–93	1.0–1.8	7×10^{-10}–2×10^{-9}	3	2 stages + vacuum
E	partially penetrating drain (8 m into 12 m)	12 m (3)	0.30–0.45	0.030–0.040	11–42	1.4–1.8	1×10^{-9}–2×10^{-9}	3	2 stages
F	sand-interbedded soft clay (wide spacing)	18 m (4)	0.12–0.50	0.020–0.050	13–139	0.9–2.0	2×10^{-10}–2×10^{-9}	3	2 stages
G	overconsolidated crust over soft clay	16.5 m (3)	0.25–0.50	0.030–0.050	41–103	1.1–2.0	3×10^{-10}–2×10^{-9}	3.5	2 stages
H	deep drain, weakly observed at depth	28 m (3)	0.28–0.60	0.030–0.060	23–177	1.4–2.3	2×10^{-10}–5×10^{-10}	4	3 stages
I	interrupted vacuum (Profile-A soil)	12 m (3)	0.20–0.45	0.020–0.040	11–93	1.0–1.8	7×10^{-10}–2×10^{-9}	3	2 stages + vacuum lost d253–283

A test bed is only as good as its coverage of the awkward cases, so each profile is in it for a stated reason and puts something under stress that no other profile does. The field references below name *deposit classes and regions established in the literature*, not specific projects: the only claim a synthetic profile can support is that ground of this character is commonplace there. None of these profiles reproduces a particular site.

Profile	Why it is in the bed — and what it stresses	Where ground of this character is found
A	The baseline, and deliberately easy: every layer settles enough to be seen and drains fast enough to be resolved. It isolates the statistical machinery from any identifiability difficulty, so if A ever falls the cause is in the method, not the ground.	Soft Holocene estuarine and deltaic clays under road and rail embankments: the Bangkok plain, the estuaries of coastal New South Wales and Queensland, the Thames, Severn and Tees, and the Holocene coastal clays of the Netherlands.
B	Depth and patience. **Three** loading stages with rests over 1200 days — the pattern that makes records hard to read, because each stage restarts the dissipation and a mis-inferred rate shows up as a mis-timed shoulder rather than a wrong final value.	Marine clay under reclamation and airport platforms: the Kallang Formation beneath Singapore, Hong Kong marine deposits, the Korean west coast around Incheon, and the Bothkennar research site in Scotland.
C	The hard case, and the bed's honest failure. The bottom 10 m contributes **13.2 %** of the settlement and carries the lowest permeability, so it is barely visible to any surface instrument. C's low coverage is the correct answer, not a defect.	Holocene peat-over-clay: the Dutch polders and the Randstad, the Irish midland raised bogs crossed by motorways, organic deposits in northern Germany and Denmark, and organic estuarine sequences in eastern Australia.
D	Vacuum is a distinct loading, not a variation of surcharge: it imposes a **suction boundary condition at the drain wall** rather than a stress at the surface. D asks whether parameters are still recovered with that boundary active. The vacuum is a known loading, never a fitted parameter.	Membrane vacuum consolidation on large soft-ground reclamations: Bohai Bay and Tianjin in China, airport and highway platforms in Thailand, trial embankments in coastal New South Wales, and highway works in the Mekong delta.
E	Drains frequently stop short, and the un-penetrated zone then drains *vertically only*. E demonstrates the split-layer inheritance tie working: without it the un-penetrated zone's horizontal permeability would be wholly unconstrained.	Wherever treatment depth is limited by plant or economics, and wherever a dense layer stops the mandrel — common in layered alluvial sequences.

Profile	Why it is in the bed — and what it stresses	Where ground of this character is found
F	Tests the layered vertical-permeability fix **through the estimator**: that defect put t_{50} out by +383 % and −50 % on a 10× contrast and every benchmark passed through it, because every benchmark was single-layer. It had been validated forward only. F also produced the bed's sharpest identifiability failure: the sand band carries the **highest permeability and 7.3 % of the settlement**, so it governs the drainage rate and is invisible to a settlement plate.	Interbedded alluvial and deltaic sequences with sand lenses and partings — arguably the commonest soft ground of all: the Chao Phraya and Mekong deltas, the Yangtze delta beneath Shanghai, the Mississippi and Gulf Coast deltaic sequences, and the Po plain.
G	In every other profile the ground is at OCR 1.0–1.6 and reaches the virgin line almost at once, so the C_r branch and the smooth β yield transition never decide anything. G is the case where they govern: a 2.5 m crust at OCR 5 needs about 42 % of the load's effect to yield, so it settles on C_r through the first half of the record and on C_c through the second.	Desiccated and weathered crusts are close to universal on soft clay: Norwegian and Swedish marine clays with 1–3 m crusts, the weathered crust of the Bangkok plain, the Champlain Sea clays of eastern Canada, and UK and Irish estuarine deposits. Crust OCR is commonly 3–10.
H	Depth beyond B without C's extreme compressibility: the same weak-observation condition reached by a different route. Its result is the useful one — depth alone is survivable; it is depth *plus* extreme compressibility contrast, as in C, that is not.	Deep soft deposits routinely treated with 25–35 m drains: Incheon and Busan in Korea, the Yangtze delta beneath Shanghai, Bohai Bay reclamations, and the thicker parts of Singapore's marine clay.
I	D's vacuum ramps once and holds, so the inverter integrates a smooth curve. Real systems fail. I imposes a 65 kPa suction **lost at day 253 and restored 30 days later**, leaving a shelf in the record the inverter must track rather than smooth over — and, unlike D, a suction that falls with depth.	Wherever vacuum consolidation is used at scale. Interruption — punctured membranes, pump or power failure, perimeter seal loss — is an operational reality of the method, and is why vacuum projects monitor suction continuously.

Two things about this bed are stated rather than implied. Profile F's 2.0 m drain spacing is at the outer edge of practice (0.8–1.5 m is typical) and was chosen to stop radial flow swamping the vertical path; its *stratigraphy* is ordinary, only its geometry is a stress choice, and §32.5 reports a control profile at a normal 1.20 m. And profile H does **not** test well resistance: the solver screens for it with the Mesri–Lo discharge factor against $D \geq 5$, which is published literature and not a claim of this campaign. H tests the geometry, and nothing here should be read as validating that screening rule.

Around each profile, every site draws independently the things an engineer cannot control. The perturbation design in full:

What varies	How	Levels
typed soil values	true values × lognormal laboratory error — standard deviations 45 % on k_0, 30 % on C_r, 25 % on C_c, 15 % on σ'_p (largest where laboratories really are least certain)	×0.5 good lab · ×1 standard · ×2 poor
plate noise	Gaussian reading error on every observation	5 / 10 / 20 mm; independent, or AR(1)-correlated ($\rho = 0.7$) imitating drifting instruments
record length	observations kept only until the record reaches the stated fraction of predicted final settlement	25 / 33 / 50 / 75 % — 105 sites each
held true	layer thicknesses, unit weights, e_0, C_k, β, $C\alpha/C_c$, loading history	typed exactly; see the limits in §32.7
windows given to the search	the guide's standard recommendation	magnitudes ×0.85–1.15; k_0 ×0.3–3.0
recipes per range	the app default	200 (1,000 in the temperature re-run of the 315-case main grid)

The campaign at a glance: 5 profiles × 4 record lengths × (6 standard-cell seeds + 4 excursion cells of 3 seeds + 3 correlated-noise seeds) = **420 distinct sites**; five constructions measured on every one; two independent studies plus a 315-case temperature re-run; over half a million forward consolidation solves on the drawn recipes alone.

32.3 The five constructions measured

Construction	Built how	The question it answers
Design range	200 recipes across the typed windows about the winner; equal votes (Step 8's top row)	does the shipped headline range keep its printed promise?
Fit-narrowed row	the same recipes, votes discounted by fit with winner-anchored T (Step 8's middle row)	can Steps 1–7 legitimately narrow the band, and at what reliability?
χ^2 region	the data-resolvable band (Step 8's bottom row)	how far over-confident is the record-only band, quantified?
Pre-fix construction	the August–September 2026 defect resurrected: data-narrowed widths, harsh weights, out-of-cap rejection	how much did the repair matter, case by case?
Prior-centred propagation	Step 8 alone, centred on the typed values, no search at all	what do Steps 1–7 actually buy?

All five are assembled by the shipped band-construction functions, extracted verbatim from the application source at run time and pinned bit-for-bit by a dedicated test category — the lesson of the September drift, in which a validated *replica* and the shipped *product* quietly parted ways, applied structurally.

32.4 How the counting works

Three statistical choices carry the campaign, and each deserves a plain statement. *Coverage.* Every range claims, implicitly, "the truth is inside me 90 % of the time" (P5 to P95). The campaign simply counts: on what fraction of sites did the range contain the true final settlement? *Wilson intervals.* A count from 90 sites is itself uncertain — 84 hits out of 90 could easily have been 80 or 87 on a different draw of noise — so every percentage is printed with the interval a statistician would attach to it: 93 % [86–97] means the data are consistent with any true rate in that bracket. Where an interval straddles the nominal 90, the range is consistent with its promise; where the whole interval sits below, the shortfall is real. *Paired comparison.* When two constructions are compared, they are compared on the *same* sites — same soils, same noise, same records — so site-to-site luck cancels, and only the cases where the two disagree carry information. The full count for the repaired design range against the pre-fix construction, on their 270 shared main-truncation sites:

	pre-fix hit	**pre-fix missed**
repaired hit	110 (both cover — uninformative)	**102 (repair rescued the site)**
repaired missed	**1 (repair lost the site)**	57 (both miss — centre wrong, see §32.6)

The 110 and the 57 say nothing about which construction is better; the verdict lives entirely in the 102-to-1 split of the disagreements, whose probability of arising by chance is about 10^{-29} (exact McNemar test). The same discipline — count everything, judge on the disagreements — is applied to every comparison in this Part. And every acceptance rule (what would count as a pass, what would trigger a re-test, what would forbid a promotion) was written into the campaign scripts before their production runs: a result cannot be reinterpreted after the fact by the person reporting it, because the interpretation was committed first.

32.5 Results

The campaign was rebuilt and re-run in September 2026. It now stands at 1080 sites over nine profiles, and it carries two studies the original did not: a model-discrepancy arm, and a measurement of what piezometers add. The original five-profile results are printed first and kept in full — a campaign whose headline only improves when it grows is a campaign that selected its ground, so the pre-registration required the old numbers to stay on the page whatever the new ones did.

The original five-profile campaign, 420 sites. The master table — coverage of the true final settlement (nominal 90 %), with Wilson-95 intervals and the median range width as a fraction of the settlement. Ninety independent-noise sites per truncation; the correlated-noise excursion is pooled across its four truncations at the foot, and Figure 32.4 draws the same numbers as a staircase.

Record reaches	Design range	Fit-narrowed	χ^2 region	Pre-fix	No Steps 1–7
a quarter	50 % [40–60] · 20 %	50 % [40–60] · 19 %	42 % [33–53] · 17 %	28 % [20–38] · 11 %	39 % [29–49] · 20 %
a third	63 % [53–73] · 18 %	59 % [49–68] · 19 %	53 % [43–63] · 13 %	36 % [26–46] · 8 %	39 % [29–49] · 19 %
half	**93 % [86–97]** · 18 %	88 % [79–93] · 17 %	73 % [63–81] · 9 %	49 % [39–59] · 5 %	37 % [27–47] · 20 %
three quarters	**92 % [85–96]** · 19 %	89 % [81–94] · 16 %	66 % [55–75] · 5 %	47 % [37–57] · 3 %	36 % [26–46] · 20 %
pooled, independent noise	75 % [70–79]	71 % [67–76]	59 % [53–64]	40 % [35–45]	38 % [33–43]
correlated noise (all truncations)	73 % [61–83]	72 % [59–81]	42 % [30–54]	37 % [26–49]	30 % [20–43]

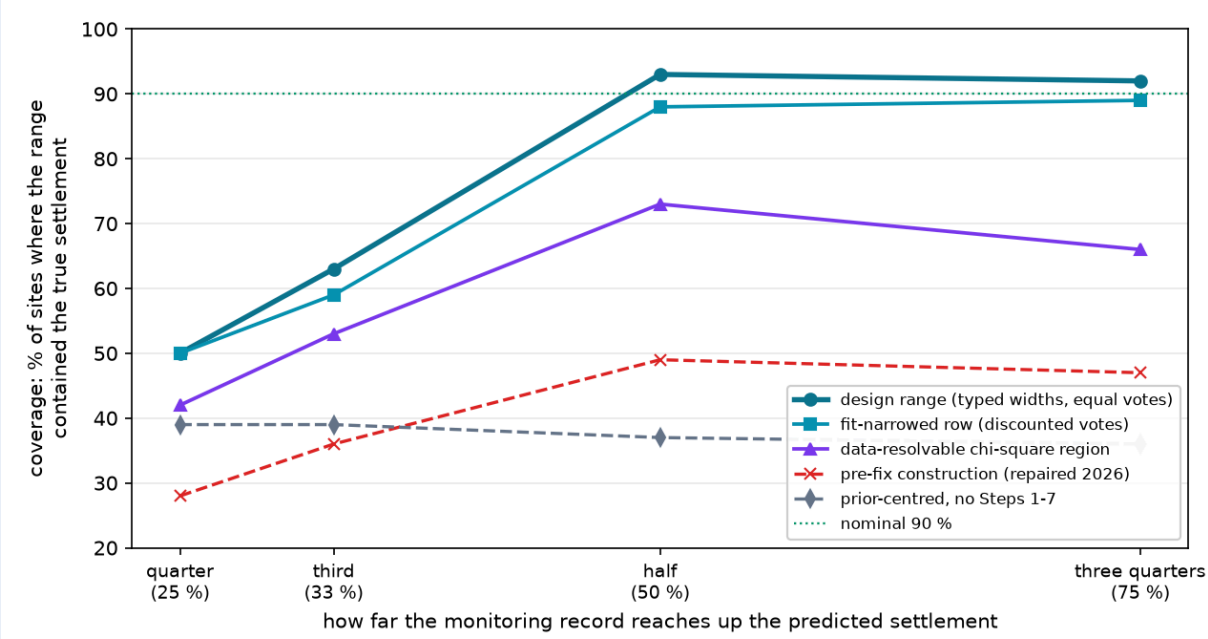


Figure 32.4. Coverage against record length for the five constructions (90 independent-noise sites per truncation; nominal 90 % dotted): the shipped rows climb onto nominal at half-records; the χ^2 region peaks then falls; the pre-fix construction never recovers; the no-search band is flat.

The correlated-noise foot row was re-measured after the error-drift repair of §21.2 was built. The same 60 sites — identical profiles, seeds and noise streams — were re-run with the score whitened at the generator's $\rho = 0.7$ and the degrees of freedom counting $n(1 - \rho)/(1 + \rho)$ effective readings. The χ^2 region recovered from 42 % [30–54] to **83 % [72–91]** — by truncation, 73 % at a quarter, 93 % at a third, 87 % at half, 80 % at three quarters — with its honest price printed beside it: the median width grew from 10 % to 28 % of the settlement, wider than the design range, as a record-only band under drifting error should be. The recovery is not a draw of luck: on the 60 paired sites the whitened region covered on 25 sites where the unwhitened one missed, with zero reversals (exact McNemar

p ≈ 6×10^{-8}). Whitening also sharpened the search itself — the winner's median centre error fell from 5.1 % to 3.3 % — while the typed-width design range, already robust to drift, moved only from 73 % to 78 %, and the no-search construction did not move at all. One honesty note bounds the claim: these re-runs whitened at the *known* generator drift; in the field ρ must come from the record itself, which is why the result card measures and prints the winner's residual drift for the engineer to type back in.

Robustness, cell by cell (design range, pooled over all four truncations): plate noise 5 / 10 / 20 mm gives 78 / 77 / 73 % — flat within sampling error; laboratory quality ×0.5 / ×1 / ×2 gives **85 / 77 / 58 %** — the strongest lever in the campaign; correlated noise gives 73 % against the χ^2 region's 42 %. Figure 32.5 splits the design range by profile: with adequate records, four of the five profiles sit at 97–100 % — including the vacuum and partial-penetration cases, whose extra physics evidently cost nothing — while the deep peat–clay profile C manages only 69 %, for a reason worth understanding (§32.6).

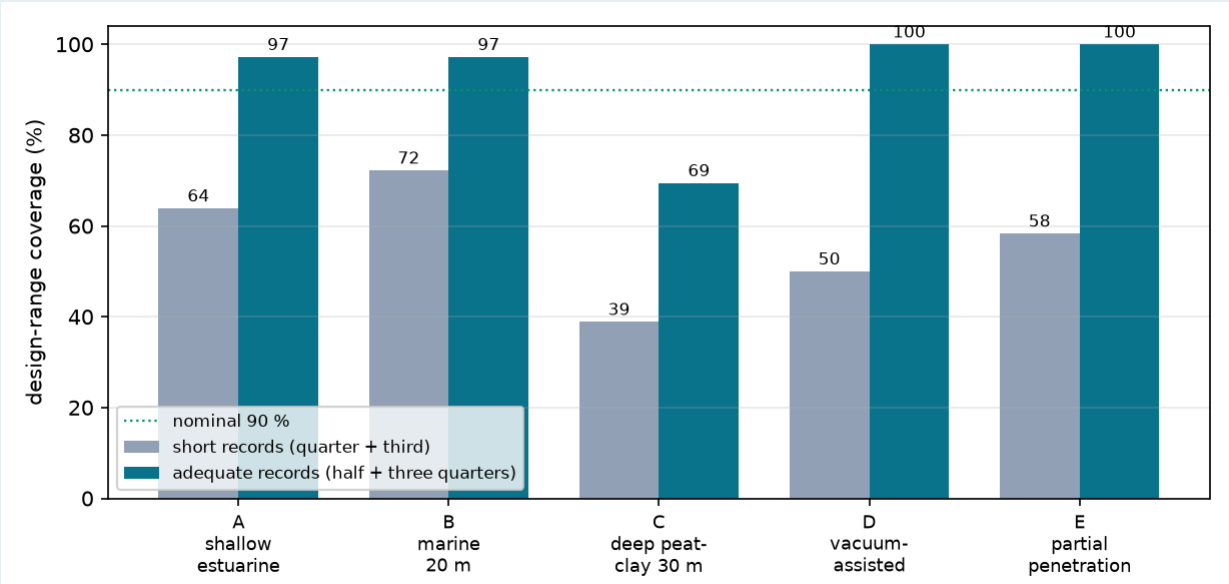


Figure 32.5. Design-range coverage by profile — short (quarter + third) against adequate (half + three-quarter) records, 36 sites per bar: vacuum (D) and partial penetration (E) reach 100 % with adequate records; the deep peat–clay C lags at every length (the observability lesson of §32.6).

Three paired results complete the picture. Against the pre-fix construction, the repaired design range won 102 paired cases and lost 1 (exact McNemar p ≈ 10^{-29}; the full table in §32.4). Against the record-only χ^2 region it won 54 and lost 5 (p ≈ 10^{-11}). And on the 270 sites shared with the original study it reproduced that study's coverage — 79 % against 77 % — agreeing case-by-case on 90 % of them: the direct evidence that the printed numbers describe the shipped product rather than a research prototype. Finally, the Step 8 temperature study (the 315-case main grid re-run at 1,000 recipes, four discounting temperatures on identical draws) found every temperature's coverage inside

the design range's own confidence interval at every truncation, promoted the fit-narrowed row from provisional under its pre-registered rule, and established a clean division of labour: the draw count governs only the *precision* of the band edges; the coverage it leaves untouched.

The rebuilt campaign: 1080 sites over nine profiles. Four profiles were added (F–I above) and the original 420 were *re-run rather than pooled*. The reason is worth stating, because it is the kind of thing that usually goes unnoticed: the harness solver had been edited two days after the original campaign finished, and a 20-case reproduction check found that **not one stored case reproduced byte-identically**. The forward change was tiny — about 1.6×10^{-4} of settlement, exactly the size the face-weighting change predicts — but the inverted winner moved by up to 5.9 %, some 250× the forward perturbation, because the sweep is a *discrete* search whose winner can flip on a change far below any tolerance one would think to set. The pre-registered rule was byte-identical or re-run everything, and it earned its keep: a numerical tolerance would have passed a change that moved the bands by percent. The campaign now sits on one solver version rather than two.

Record reaches	Design range, nine profiles	χ^2 region	No Steps 1–7
a quarter	51 % [44–59]	—	—
a third	64 % [57–71]	—	—
half	86 % [80–90]	—	—
three quarters	93 % [88–96]	—	—
adequate records (U ≥ 50 %)	**90 % [86–92]**	61 % [56–67]	54 % [48–59]
pooled, all four truncations	74 % [70–77]	52 % [48–56]	54 % [51–58]
AR(1) correlated noise	67 % [57–75]	43 % [34–52]	55 % [45–64]

Adding four harder profiles did not move the headline. Compared like for like on the three truncations the original campaign pooled, nine profiles give **77 % [73–80]** against the published **79 % [73–83]** on five — statistically indistinguishable. That is a stronger result than an improvement would have been: the reliability claim survived a near-doubling of the ground it is tested on. The comparison has to be like for like, because the one-third column was added after the original pooling and is the second-hardest of the four.

Where the loss sits. Two profiles carry almost all of it, and both are identifiability problems rather than physics failures. At adequate records A, B, D, E, G and I return 97–100 %; H returns 89 %; **C returns 67 % and F 58 %**. For F the winner is displaced a median 8.1 % against an 11.1 % half-width, and the displacement is *scatter, not bias* — the centre lands far from the truth in random directions, which is what equifinality looks like. The fit is untroubled throughout (χ^2_r 1.19): many parameter sets explain the record equally well.

Is F the ground or the geometry? Profile F's 2.0 m spacing was chosen to make the vertical path compete, so a control profile — the same stratigraphy at an entirely ordinary 1.20 m — was run to separate the two. It recovers to **75 % [59–86]** from 58 % [42–73]. The wide spacing does make it worse; but 75 % is still short of the 90 % nominal and far short of the 97–100 % the well-behaved

profiles return, so **interbedded ground is a genuine weak case, not an artefact of the geometry**. The two intervals overlap, so at 36 sites each the difference between them is suggestive rather than established; what is established is that neither is in the same class as the rest of the bed.

Model discrepancy: what happens when the model itself is wrong. The campaign's standing limitation was that its truths came from the same physics the estimator uses. Three mechanisms were therefore added that the estimator *cannot express at any parameter value*, at three severities each, 324 sites in all: a length of drain that does not work (crimped or kinked); installed spacing that scatters, so the site is a mixture of unit cells rather than the one the estimator solves; and held parameters — C_k and C_α/C_c, which the pipeline does not search — set wrong. Each mechanism's mildest rung acts as a null control against matched no-discrepancy cases, and all three pass.

The lie told to the estimator	Coverage as it grows	χ^2_r as it grows
Held parameters wrong *(not structural)*	91 → 82 → 47 %	1.33 → 1.90 → **5.45**
Drain impairment *(structural)*	92 → 56 → 61 %	1.17 → 1.19 → **1.28**
Spacing scatter *(structural)*	94 → 83 → **31 %**	1.19 → 1.18 → **1.25**

> **The diagnostics detect wrong parameters. They do not detect a wrong model.** When the parameters are wrong, χ^2_r reaches 5.45 — a fourfold rise nobody could miss, and the engineer knows to distrust the answer. When the *model* is wrong, coverage falls to 31 % while χ^2_r reads 1.25, which any engineer would call a good fit. The reported range does not widen in any of the three cases. So a fit of $\chi^2_r \approx 1.2$ beside a normal-looking range cannot distinguish "all is well" from "a length of drain is crimped and this range is 31 % reliable". The defence is not a better statistic: it is checking installation records and drain continuity, and treating a design range as conditional on the drains being what the drawings say.

Two qualifications keep that finding the right size. The severe spacing rung is a 35 % coefficient of variation — installed r_e spanning 0.36 to 2.49 times nominal, which is not a real site; at a realistic 10 % scatter coverage holds at 94 % and at 20 % it holds 83 %. And the drain-impairment ladder reads as a drop then a plateau rather than a reversal, its last two rungs' intervals overlapping almost entirely. Each rung carries about ±14 percentage points, so the arm is powered to establish a *trend*, not to certify any single figure.

Piezometers: what the campaign had been leaving out. Every coverage number published before September 2026 — including every figure in the first table of this section — was measured with **settlement plates alone**. The application scores plates *and* piezometers, balancing the two data types so that a dense plate record cannot drown a sparse piezometer; the validation harness simply had no piezometer support. The published figures therefore describe a weaker instrument set than an engineer installs. A paired study — 126 cases, identical seeds, noise and laboratory error, differing only in whether piezometers were scored — measures the difference.

Profile	Plates only	Plates + piezometers	Winner mislocation	Range width
A *(control — nothing to gain)*	98 % [88–100]	100 % [92–100]	2.2 → 2.1 %	18.2 → 18.1 %
C *(deep, weakly observed)*	67 % [52–79]	**95 % [84–99]**	5.7 → 2.1 %	17.6 → 17.8 %
F *(sand band governs the rate)*	60 % [44–73]	**93 % [81–98]**	7.9 → 3.1 %	22.1 → 21.3 %
pooled	**75 % [66–81]**	**96 % [91–98]**	4.5 → 2.6 %	18.6 → 18.5 %

Piezometers rescued 28 cases and cost 1 (exact McNemar, $p < 0.0001$), and the *way* they do it matters as much as the size. The range width does not change — 18.6 % against 18.5 %. The gain is not a wider hedge; it is a better-located centre, with the winner's displacement roughly halving. On profile F the layer that governs the drainage rate contributes 7.3 % of the settlement: it is invisible to a plate and loud in pore pressure, and measuring it turns the campaign's worst profile into one of its better ones.

The practical reading is a site recommendation rather than a statistic: **on interbedded ground, or where a layer drains far faster than it settles, instrument the pore pressure** — the settlement record alone cannot see the layer that sets your programme. The qualification is that these are idealised piezometers: 2 kPa reading error, read on the plates' own schedule, at the depths the profile specifies. Real instruments drift, fail, and are often installed at fewer depths. What is measured is that piezometers *of this quality at these depths* recover the weak profiles, not that piezometers always will.

32.6 Discussion: what the campaign establishes

The headline claim survives its second, harder exam. A reliability label is only as good as the trial behind it, and the design range's label has now been earned twice — once by the original 270-run study, once through the shipped code path on the same sites plus 150 new ones — with the two measurements statistically indistinguishable. The label is honest in both directions: the same campaign that certifies 92–93 % at adequate records prints 50 % at quarter-records on the card rather than hiding it.

The engineered search is the load-bearing component. Strip Steps 1–7 away and re-centre the identical propagation on the typed values, and coverage falls from 75 % to 38 % pooled — and, more telling, the no-search band is *flat across record length*: at three-quarter records it still manages only 36 %, because nothing in it can learn from data. The search's centring error, by contrast, falls from 10 % of the settlement at quarter-records through 6 % at a third to under 2 % at three quarters. Combined with the miss anatomy of §30.3 — misses are mislocation events, with the centre errors of misses roughly twice the band's half-width — the mechanism is unambiguous: the band's width philosophy is sound; everything turns on where the search parks the centre, and the search earns its keep by parking it well once the record can steer it.

Robustness lands where theory predicts it. The typed-width construction is insensitive to instrument noise level, degrades only mildly under correlated noise (75 → 73 %), and depends most strongly on laboratory quality at short records — exactly the profile expected of a range whose width carries laboratory information and whose centre carries record information. The χ^2 region shows the mirror image: it collapses under correlated noise (59 → 42 %) because its width is derived from a noise model the correlation violates. The repair is as instructive as the failure: whiten the score for the

typed drift and count the effective readings honestly (§21.2), and the same region recovers to 83 % on the same sites — the collapse was never a defect of the χ^2 machinery, only of feeding it an error model the record did not obey (§32.5).

The hard profile is a lesson, not an embarrassment. Profile C's 69 % at adequate records (Figure 32.5) traces to depth, not to peat: when a 30 m column's surface record reaches half the total settlement, its deep firm-clay layer — farthest from the drains and slowest — has barely begun its own consolidation, so the record carries almost no information about the layer that controls a large share of what remains. This is the observability problem of §25.8 appearing at campaign scale, and it is why the result card's observability check reports settlement-still-to-come *per layer* rather than for the profile as a whole: on deep stratified sites, the surface record's maturity flatters the true state of knowledge.

The protocol polices its own components. Within this one campaign the pre-fix construction was resurrected and shown to fail 102-to-1 against its repair; the fit-discounted row failed its first pooled exam, was diagnosed by truncation, re-tested at five times the draws under a rule frozen in advance, and only then promoted; and the one-third slice was first sketched on 15 sites, challenged, and re-measured at full breadth — whereupon its coverage moved from the sketched 67 % to the confirmed 63 %. None of these corrections was prompted from outside the protocol. A validation exercise that only ever confirms is ceremony; this one bit three times, and each bite improved the product or the numbers printed on it.

IN A NUTSHELL

For a practising engineer the campaign compresses to four rules. Design to the top row of the forecast card — it has kept its printed promise twice. Do not trust any range, from any method, before the record reaches about half the predicted settlement; at a third, use with caution; at a quarter, treat every number as provisional. Spend a monitoring budget on duration, not on precision — and spend a laboratory budget generously, because at short records laboratory quality is worth more than anything the monitoring can yet tell you. And on deep stratified profiles, read the observability panel before believing a mature-looking surface record.

32.7 What the campaign does not establish

Honesty about scope is part of the credibility being claimed, so the limits are stated as plainly as the results. *First, this is largely self-consistency rather than field validation:* the same forward physics that generates most sites' records also inverts them. *That statement needed narrowing in September 2026*: the discrepancy arm of §32.5 now puts three mechanisms into the truth that the estimator cannot express at any parameter value, and it found that the diagnostics do not detect a wrong model. What remains outside the campaign's reach is everything those three mechanisms do not cover — unmodelled well resistance above all, which the solver screens for with the Mesri–Lo discharge factor against $D \geq 5$ rather than solving, on published literature and not on any claim made here. The cross-model checks of Chapter 27 and the single field comparison of Chapter 31 are the complementary evidence on that axis, and one field case remains far short of a field-scale coverage claim. *Second, the campaign perturbs what engineers most often get wrong* (k_0, C_c, C_r, σ'_p) *but types the rest true:* layer

thicknesses, unit weights, e_0 and the loading history are exact on every site, so the measured coverages are conditional on those being right — a mis-surveyed fill height or a mis-picked layer boundary is a separate hazard the campaign does not price. *Third, the band constructions ran through the shipped code, but the forward solver behind the campaign is the tool's verified numerical twin;* a subsampled end-to-end run through the full application interface is the campaign's one remaining scheduled step. *Fourth, the instrument set matters, and is now measured rather than assumed:* the tables in the first half of §32.5 are **plates-only** inversions, and the paired study at the end of that section shows piezometers lifting the weak profiles from 75 % to 96 %. A reader comparing a project against these figures should ask first which instruments that project has. *Fifth, coverage is a population frequency, not a per-site guarantee:* 93 % at half-records means roughly nine sites in ten drawn from ground like this test bed's — it does not promise any single embankment, and the per-site diagnostics on the card (record reach, observability, stability) exist precisely because averages do not transfer to individuals.

32.8 Reproducing, extending — or attacking — the campaign

Every number in this Part regenerates from scripts and result files that ship in the tool's validation directory: the campaign runner and its temperature-study companion carry the pre-registered criteria in their headers, the per-cell result files carry every site's truth, ranges and diagnostics, and the analysis scripts print the tables above verbatim. The full 1080-site campaign is roughly twelve hours of computing on a 32-thread desktop (about 9 minutes per site at 26-way concurrency, and 4–8 minutes run alone); a single cell reruns in under two hours. The invitation is deliberate: a reader who distrusts any number here can rerun it, vary the test bed against it, or add a competing construction as a new arm and let the same counting judge it. A calibration that ships with its own test rig is the strongest form of the falsifiable contract of §30.2 — and the easiest kind of claim to take at face value, because taking it apart requires no one's permission.

CHAPTER 33

Appendix A: computational specification

This appendix is normative where the rest of the book is tutorial: it states exactly what the shipped solvers compute for the three questions that most often need a precise answer — how the loading schedule is applied, how the degree-of-consolidation denominator is formed, and how vacuum attenuates with depth. Every statement here was read from the solver source, and the worked example regenerates from `appendix_staged_example.js` in the tool's validation directory. "Reproduce it" boxes elsewhere in the book may cite this appendix as their stable reference for these semantics.

A.1 Loading-schedule semantics

A schedule is an ordered list of stages, each *(Δp, ramp, rest)*: the total applied stress rises linearly by Δp over the ramp and is then held for the rest, with the next stage starting where the previous ended. The applied-stress history p(t) is therefore piecewise linear and cumulative. Negative Δp (unloading) is legitimate and is carried by the hysteresis law (C_r on the unloading path, with the running-maximum σ′ as the anchor).

The rule that matters for back-analysis: the solver always applies the complete typed schedule, on every solve of every pipeline step. Truncation of the monitoring record affects only which predicted instants are scored against readings — corner solves, scans, line searches, re-sweeps and every Step-8 draw all simulate the full schedule to the project horizon. Stages that lie wholly beyond the last reading are therefore fully present in every prediction and every forecast; the record's end has no influence whatever on the loading the model feels. An engineer who wants a what-if without a future stage must edit the schedule, not rely on the record ending before it.

Worked staged example (profile A of the validation campaign; two stages, 60 kPa ramped over 15 days then +40 kPa from day 90 over 12 days; record cut at day 60, *between* the stages): the solved surface settlement is 364 mm at the cut, 485 mm when stage 2 begins, 841 mm at day 180 and 1064 mm at the 900-day horizon. A back-analysis of this site would score only readings up to day 60, yet every candidate's curve — and the entire forecast — carries stage 2 exactly as typed.

A.2 The U_s denominator, exactly

The settlement-based degree of consolidation is $U_s(t) = S_{prim}(t) / S_\infty$, and S_∞ is a closed-form ultimate, not a solve at large time. Per element:

$\sigma'_\infty = \sigma'_0$ + span, where the *span* is the maximum cumulative applied load the schedule ever reaches (for a monotonic schedule, simply the final net load; for schedules with unloading, the maximum-historical value, so U_s remains well defined) *plus* the ultimate vacuum contribution at that element's depth, p_{vac}·df(z) of §A.3. The ultimate strain is $\varepsilon_\infty = \varepsilon_\beta(\sigma'_\infty)$ evaluated with the *typed* (geologic) σ'_p: neither loading-induced nor creep-induced hardening of σ'_p enters the denominator, which keeps U_s

a property of the stated problem rather than of the solution path. S_∞ composites these strains over depth ($\Sigma\, \varepsilon_\infty \cdot dz$), and under partial penetration each row uses its true plan area — the annulus outside the drain on drained rows, the full disk including the core on undrained rows.

Companion definitions, for completeness: U_p and U_r normalise pore pressure by the driving pressure $p_{net} + p_{vac}$ (for U_p, the thickness-weighted mean of the attenuated vacuum), so with combined vacuum and surcharge both tend to exactly 100 %. Creep and immediate (E_u) settlement are added to reported *settlement* only and never to any U — the invariant stated in Part Two.

A.3 Vacuum attenuation with depth

The applied suction is attenuated linearly with depth: $df(z) = 1 - (1 - k_2) \cdot z / H$, floored at zero, where z is the element's mid-depth and **H is the full compressible-profile depth — not the drain length**. The suction enters as the drain-wall pressure $u = -p_{vac}(t) \cdot df(z)$ on drained rows only; under a partially penetrating drain, rows below the tip receive no suction at all — the vacuum, like the drain, stops at the tip, while the attenuation ramp continues to be reckoned against the full profile depth for the rows above it. k_2 is the fraction of the surface suction still acting at the base of the profile: $k_2 = 1$ means no loss, and field observations of suction loss along drains (Indraratna & Redana 2000, the source already cited for the smear model) motivate values below 1 for long drains. Where paired piezometers exist near the drain top and tip, k_2 is best taken from their suction ratio; absent measurements, values near 1 for short drains and progressively lower for long drains reflect the published observations, and the sensitivity card will show how much the choice matters for your profile.

Sections that rely on these semantics point here; any future change to them must update this appendix and its regenerating script in the same edit.

CHAPTER 34

Quick reference

Table 34.1 collects every axis and every setting in one place.

Table 34.1. Every axis, every setting. "scan"/"res." = the scan resolution at that step; iv = the knob's 95 % interval half-width. All solves use the same solver grid (40 cells / 1 day / 0.25 m by default) so no step advertises a score the delivered answer cannot reproduce.

Axis / step	Coarse step	Fine window	Continuous refine	95 % interval	Extension
Knob A — α_A	1/12 (13 pts; walls at 4/12, 8/12)	1/48 (7 pts, winner ±1/16)	parabola vertex (clamped between neighbours; sampled winner if flat)	max(scan, 0.025)	0.5 α/step, cap 2/side (max [−1, 2])
Knob B — α_B	1/12 (13 pts)	1/48 (7 pts)	same	max(scan, 0.025)	0.5 α/step, cap 2/side
Knob S — α_s	0.25 (3 levels: ×0.79/1.00/1.26)	±0.125 (3 levels)	parabola on the S-slice	max(0.0625, 0.025)	0.25, cap 1/side + diagnostic
Step 6 window	—	max(1/48, span/6) — ~7 pts/axis; hw = max(3/48, 2×iv)	same + best-sampled guard	fresh each round	≥0.3 at a wall → extend, cap +1.0 α, else flag
Final band	100 solves = 20 uniform-α magnitude blends × 5 speed levels, through data-driven walls	—	—	carried per key	—

RECAP

The whole pipeline in eight short sentences. Bracket your band with two corner solves (Steps 1–2). Anchor the speed across both corners (Step 3). Sweep every magnitude recipe with a speed dial riding along, and turn the winner into a confidence interval (Step 4). Sharpen the speed at that winner (Step 5). Re-sweep and refit in a loop until speed and magnitude agree (Step 6). Polish each layer and diagnose any guilty one (Step 7). Report a data-driven P5–P50–P95 band (Step 8).

Glossary

α (alpha)	a 0-to-1 slider blending one group of parameters from its extended-low to its extended-high wall; one α = one complete recipe
band / walls	the Lo–Hi range you type for each magnitude parameter; simultaneously prior knowledge, search space and the seed of the final uncertainty statement
basin / valley	a region of recipe-space where the misfit is low; “wrong basin” = a low-misfit region that does not contain the truth
χ^2_r	the misfit score: average squared miss in units of measurement precision; ≈ 1 means fitted to the noise level
code verification	evidence that the programme solves its equations as written (convergence order, exact-solution benchmarks) — the introduction’s taxonomy, and most of Part One
inverse crime	testing an estimator on synthetic data generated by its own forward model — checks convergence and identifiability, never model error; every synthetic result here is labelled accordingly (§30.1)
data-driven walls	the final band’s walls, set by each knob’s measured 95 % interval rather than your typed width
dual-track	running and reporting two answers when two valleys score within 5 % — the data cannot separate them
equifinality	many parameter sets fitting the same record equally well — the disease; bands and dual-tracks are the treatment
held-out	data hidden from the method and used afterwards to measure true predictive skill
knob A / B / S	the three sweep axes: settlement-raising magnitudes, settlement-lowering magnitudes, and the global speed factor
min-over-S	judging each (A, B) recipe by the best score across the scanned speed levels — treating speed as a nuisance parameter
pinned	a winner sitting in the outer 5 % of an axis — the wall, not the data, may be shaping the answer
ruler	the speed level a sweep is scored at; a “stale ruler” is a speed fitted at magnitudes that later changed
smear zone	the ring of soil disturbed by drain installation — reduced permeability (severity $\eta = k_h/k_s$) over radius r_s, transitioning back to intact soil at r_t; a speed-family feature
solution verification	evidence that a particular run is numerically converged on its mesh and time step (the mesh card, the grid presets)
truncation	the record ending early (typically U ≈ 50 %) — the root cause of most difficulties in this subject
validation	the test against real ground — field records the model never saw; here the single field case of §30.8, plus the recommended future cases of the introduction (§30.1)

REFERENCES

Works cited in the text. Entries are listed alphabetically by first author. Where a reference underpins a specific piece of the tool, the role is noted.

Abuel-Naga, H.M., Bergado, D.T. and Gniel, J. (2015) Design chart for prefabricated vertical drains improved ground. *Geotextiles and Geomembranes*, 43(6), 537–546. https://doi.org/10.1016/j.geotexmem.2015.04.021

A chart-based design procedure; cited in §1.1 as representative of the class.

Asaoka, A. (1978) Observational procedure of settlement prediction. *Soils and Foundations*, 18(4), 87–101. https://doi.org/10.3208/sandf1972.18.4_87

The graphical settlement extrapolation used in Part Three as an independent check.

Barron, R.A. (1948) Consolidation of fine-grained soils by drain wells. *Transactions of the ASCE*, 113, 718–742.

The original radial consolidation solution; the equal-strain case is the tool's Olson–Barron benchmark.

Beven, K. and Binley, A. (1992) The future of distributed models: model calibration and uncertainty prediction. *Hydrological Processes*, 6(3), 279–298. https://doi.org/10.1002/hyp.3360060305

Generalised likelihood uncertainty estimation, the ancestor of the final band.

Beven, K. and Binley, A. (2014) GLUE: 20 years on. *Hydrological Processes*, 28(24), 5897–5918. https://doi.org/10.1002/hyp.10082

A retrospective on the method's known weaknesses, several of which Part Three measures.

Bjerrum, L. (1967) Engineering geology of Norwegian normally-consolidated marine clays as related to settlements of buildings. *Géotechnique*, 17(2), 83–118. https://doi.org/10.1680/geot.1967.17.2.83

The delayed-compression picture underlying the creep model.

Carrillo, N. (1942) Simple two- and three-dimensional cases in the theory of consolidation of soils. *Journal of Mathematics and Physics*, 21, 1–5.

The product rule combining radial and vertical consolidation.

Crank, J. and Nicolson, P. (1947) A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43(1), 50–67.

The time-integration scheme of the radial kernel.

Duncan, J.M. and Buchignani, A.L. (1976) *An Engineering Manual for Settlement Studies*. Department of Civil Engineering, University of California, Berkeley.

Source of the undrained modulus range for soft clay.

Garlanger, J.E. (1972) The consolidation of soils exhibiting creep under constant effective stress. *Géotechnique*, 22(1), 71–78.

Bjerrum's time lines cast as the equivalent-time law the solver uses (§3.4).

Gioda, G. and Sakurai, S. (1987) Back analysis procedures for the interpretation of field measurements in geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 11(6), 555–583. https://doi.org/10.1002/nag.1610110604

The optimisation tradition in geomechanical back-analysis (§20.3).

Hansbo, S. (1981) Consolidation of fine-grained soils by prefabricated drains. *Proceedings of the 10th International Conference on Soil Mechanics and Foundation Engineering*, Stockholm, 3, 677–682.

The smear-zone solution against which the radial solver is validated.

Hansbo, S. (1997) Aspects of vertical drain design: Darcian or non-Darcian flow. *Géotechnique*, 47(5), 983–992.

Modern band drains generally deliver q_w > 150 m^3/yr, high enough for well resistance to be neglected in design (§29.9).

Holtz, R.D., Jamiolkowski, M., Lancellotta, R. and Pedroni, R. (1991) *Prefabricated Vertical Drains: Design and Performance*. CIRIA / Butterworth-Heinemann, London.

Typical required discharge capacities of 25–100 m^3/yr — comfortably exceeded by modern products (§29.9).

Huang, S., Huang, J., Kelly, R., Jones, M., Wong, P., Yuen, S., Kamruzzaman, A.H.M. and Jiang, S.-H. (2025) Efficient Class C prediction of PVD-improved soft clay settlements. *Engineering Geology*, 346, 107903.

The Bayesian (BUS + general simplified Hypothesis B) comparator examined in Part Three.

Indraratna, B. and Redana, I.W. (2000) Numerical modelling of vertical drains with smear and well resistance installed in soft clay. *Canadian Geotechnical Journal*, 37(1), 132–145.

Smear and well-resistance modelling; the vacuum depth-attenuation law.

Jones, M., Huang, S., Kelly, R. and Huang, J. (2026) Review of statistical back analysis methods for road embankments built on soft clay soils: progression to Bayesian. *Intelligent Transportation Infrastructure*, 5, liaf033. https://doi.org/10.1093/iti/liaf033

The recent systematic review against which this book positions its method (§20.3).

Kelly, R.B., Sloan, S.W., Pineda, J.A., Kouretzis, G. and Huang, J. (2018) Outcomes of the Newcastle symposium for the prediction of embankment behaviour on soft soil. *Computers and Geotechnics*, 93, 9–41.

The Ballina prediction symposium; a second published settlement record used in Part Three.

Ladd, C.C. and Foott, R. (1974) New design procedure for stability of soft clays. *Journal of the Geotechnical Engineering Division, ASCE*, 100(GT7), 763–786.

SHANSEP; source of the normalised strength ratio used to size E_u.

Lambe, T.W. (1973) Predictions in soil engineering. *Géotechnique*, 23(2), 151–202.

The prediction-class taxonomy used throughout this book (§1.7).

Leroueil, S., Kabbaj, M., Tavenas, F. and Bouchard, R. (1985) Stress–strain–strain rate relation for the compressibility of sensitive natural clays. *Géotechnique*, 35(2), 159–180.

The isotache framework the equivalent-time form belongs to (§3.4).

Lim, G.T., Pineda, J.A., Boukpeti, N. and Carraro, J.A.H. (2018) Predicted and measured behaviour of an embankment on PVD-improved Ballina clay. *Computers and Geotechnics*, 93, 204–221.

Source of the measured Ballina soil profile used in Part Three.

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), 239–245.

Latin hypercube sampling, used for the probabilistic runs.

Mesri, G. and Godlewski, P.M. (1977) Time- and stress-compressibility interrelationship. *Journal of the Geotechnical Engineering Division, ASCE*, 103(5), 417–430. https://doi.org/10.1061/AJGEB6.0000421

The C_α/C_c ratio used by the creep model.

Mesri, G. and Lo, D.O.K. (1991) Field performance of prefabricated vertical drains. *Proceedings, International Conference on Geotechnical Engineering for Coastal Development (GEO-COAST '91)*, Yokohama, 1, 231–236.

Origin of the discharge factor $D = q_w/(k_h \cdot l_m^2)$ and the $D \geq 5$ criterion for negligible well resistance (§1.9, §29.9).

Miura, N. and Chai, J.-C. (2000) Discharge capacity of prefabricated vertical drains confined in clay. *Geosynthetics International*, 7(2), 119–135.

Measured up to fivefold reduction of q_w over 100–300 days — the basis of the organic-soil / long-drain caveat (§29.9).

Olson, R.E. (1977) Consolidation under time-dependent loading. *Journal of the Geotechnical Engineering Division, ASCE*, 103(1), 55–60.

Ramp loading; the Olson–Barron validation case.

Peaceman, D.W. and Rachford, H.H. (1955) The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1), 28–41.

The alternating-direction implicit scheme used by the coupled solver.

Pineda, J.A., Suwal, L.P., Kelly, R.B., Bates, L. and Sloan, S.W. (2016) Characterisation of Ballina clay. *Géotechnique*, 66(7), 556–577.

The site characterisation on which the Ballina stratigraphy is built.

Rujikiatkamjorn, C. and Indraratna, B. (2007) Analytical solutions and design curves for vacuum-assisted consolidation with both vertical and horizontal drainage. *Canadian Geotechnical Journal*, 44(2), 188–200.

With Abuel-Naga et al. (2015): the contemporary statement that well resistance is generally negligible for modern high-capacity PVDs (§29.9).

Rujikiatkamjorn, C. and Indraratna, B. (2009) Design procedure for vertical drains considering a linear variation of lateral permeability within the smear zone. *Canadian Geotechnical Journal*, 46(3), 270–280. https://doi.org/10.1139/T08-124

A chart-based design procedure; cited in §1.1 as representative of the class.

Spear, R.C. and Hornberger, G.M. (1980) Eutrophication in Peel Inlet — II. Identification of critical uncertainties via generalized sensitivity analysis. *Water Research*, 14(1), 43–49. https://doi.org/10.1016/0043-1354(80)90040-8

Regionalised sensitivity analysis — the origin of the keep-the-survivors idea (§20.3).

Tavenas, F., Jean, P., Leblond, P. and Leroueil, S. (1983) The permeability of natural soft clays. Part II: permeability characteristics. *Canadian Geotechnical Journal*, 20(4), 645–660. https://doi.org/10.1139/t83-073

The source of the $C_k \approx 0.5 \cdot e_0$ default the solver locks to.

Terzaghi, K. (1925) *Erdbaumechanik auf bodenphysikalischer Grundlage*. Deuticke, Vienna.

One-dimensional consolidation theory; the exact series used to measure vertical convergence.

Thomas, L.H. (1949) *Elliptic problems in linear difference equations over a network*. Watson Scientific Computing Laboratory report, Columbia University.

The tridiagonal algorithm both solvers use.

Tian, H., Li, D., Cao, Z. and Du, W. (2023) Auxiliary Bayesian updating of embankment settlement based on finite element model and response surface method. *Engineering Geology*, 323, 107244.

The FEM-based baseline in the Part Three comparison.

Tian, Y., Wu, W., Wang, L. and Jiang, G. (2022) Response surface-based back analysis of soft clay parameters. *Acta Geotechnica*, 17, 1273–1294.

A response-surface baseline in the comparison of Chapter 30.

Yin, J., Chen, Z. and Feng, W. (2022) A general simple method for calculating consolidation settlements of layered clayey soils with vertical drains under staged loadings. *Acta Geotechnica*, 17, 3647–3674.

The case against which the comparator's forward program is validated.

Authors of the works listed in the References, and of the methods named in the text. Page numbers point to where the work or method is used, not only to where it is cited.

Y

Concepts, parameters and features. Where a term is treated at length, the page on which that treatment begins is listed.

U

V

W