

Solving the Model Once: Direct Semigroup Computation for Multi-Asset Exotics — Exact Risk Where Simulation Is Noisiest

Andrew Williams · Krylo

kryloquant.com · andrew@kryloquant.com

every quantitative claim in this paper regenerates from the validation repository with one command

July 2026

Abstract

Structured products with worst-of and knock-in features have produced concentrated losses in three of the last six years, and the outstanding exposure is growing. We argue the recurrence is structural: the industry risk-manages these products by re-simulating them, and simulation-based risk numbers are at their noisiest exactly at the loss events, while grid-based alternatives re-pay their cost per product and per scenario and stall beyond two underlyings. We describe a pricing engine that takes a different route: it computes the model’s transition operator — the solution operator of the pricing equation — directly and exactly to high order, once per underlier set. Every product, risk number, scenario, maturity, and counterparty-exposure profile is then a cheap read of that one object. We summarise the construction at a level intended for risk-literate readers, and present the validation program: fifteen independently gated benchmarks against closed forms, QuantLib, and ten-million-path Monte-Carlo anchors, head-to-head comparisons against Crank–Nicolson and ADI solvers, and — deliberately — the null results. The numerical stack is additionally certified end-to-end on deployment-class hardware (a 26-rung campaign on an RTX PRO 6000 workstation; zero numerical failures), reproduced in full as Appendix A; the complete test-gate inventory — every automated check in the repository, with what each certifies — appears as Appendix B. The full evidence base is reproducible with one command.

1 The problem

Worst-of autocallable notes broke their holders three times in six years: Natixis lost €260 million on Korean worst-of books in 2018; Société Générale and Natixis reported €400 million and €273 million of hedging losses on the same product class in the COVID quarter of 2020; and the 2024 knock-in wave on HSCEI-linked notes crystallised roughly \$4.6 billion of losses, with sellers reserving \$1.2 billion for compensation [1, 2]. The failure point is the same every cycle: the knock-in. Meanwhile the exposure grows — roughly \$538 billion of callable and autocallable issuance in 2025, a record [3] — and is increasingly *recycled*: dealers now transfer autocall risk to hedge funds programmatically, and autocallable ETFs have brought the payoff to fund wrappers [4]. More books, in more hands, each with less pricing infrastructure than the dealer that originated the risk.

We contend that the loss cycle is not merely a market phenomenon; it has a computational component. The industry’s working tool for these products is Monte-Carlo simulation, and its risk estimates degrade precisely where the products are dangerous. A knock-in is a discontinuity in the payoff; risk is estimated by differencing re-simulations with nudged inputs; and second differences across a discontinuity amplify simulation noise catastrophically. The number that matters most on the worst day — the note’s *correlation risk*, the exposure to both underlyings falling together — is the least reliable number on the desk. On a representative book of eight step-down notes we measured the industry-standard estimate of that quantity at -0.06 ± 0.06 : the noise equals the signal, and its *sign* is unreadable on half the book. Figure 1 shows the same phenomenon on a single real term sheet: near the knock-in, each simulation-based estimate costs about two minutes and carries error bars spanning much of the answer; the exact profile is the blue curve.

Grid-based (finite-difference) solvers remove the noise but keep the economics: each product, each scenario, and in practice each maturity requires its own solve, and the approach stalls beyond two underlyings. The

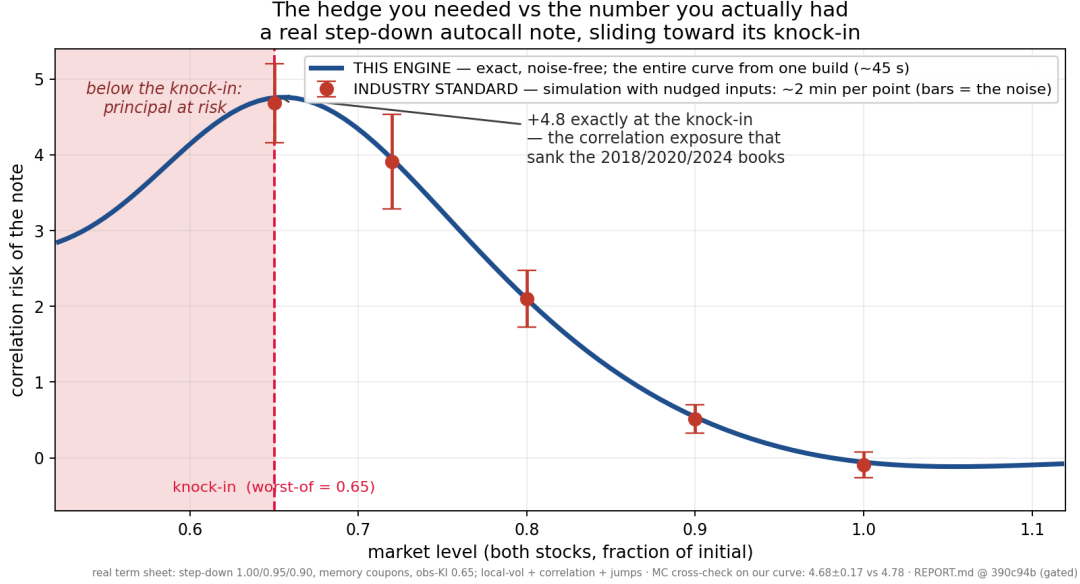


Figure 1: Correlation risk of a real step-down autocall note (memory coupons, discrete knock-in, under local volatility + correlation + jumps) as the market slides toward the knock-in. Blue: the exact profile, read from one solved model. Red: the industry-standard estimate — re-simulation with nudged inputs — with its measured noise. The independent simulation cross-check agrees with the exact curve (4.68 ± 0.17 vs 4.78 at the knock-in).

result is that the fastest-growing holders of this risk — funds and ETF managers — run it on the weakest infrastructure.

2 Every incumbent approximates around the same object

Under a risk-neutral model, the value of a European claim with payoff g satisfies a pricing equation whose solution can be written in one line:

$$V(t) = e^{(T-t)\mathcal{L}} g, \quad (1)$$

where $\mathcal{L}$ is the model's generator — the differential operator encoding volatilities, correlation, drift, and discounting. The family $e^{\tau\mathcal{L}}$ is the model's *transition semigroup*: the operator that evolves any value profile backward through time τ under the model.

Every standard method approximates *around* this object rather than computing it. Monte-Carlo samples its action one random path at a time, so every product, bump, scenario, and date pays the full sampling cost again, with statistical noise attached. Finite-difference methods (Crank–Nicolson and its ADI variants) discretise $\mathcal{L}$ on a grid and step through time, so accuracy is limited by the time-stepping scheme and the cost repeats per product and per scenario. In both cases the reusable mathematical object — the semigroup element itself — is never constructed.

3 Computing the semigroup directly

The engine's central idea is to construct $M \approx e^{\Delta t \mathcal{L}}$ — one time-step of the exact solution operator — analytically, and to make everything downstream a read of it (Figure 2). Three components matter.

The analytic kernel. Over a short step Δt , the transition density of a variable-coefficient diffusion is locally Gaussian with computable corrections. The engine evaluates this kernel in closed form cell by cell, corrects its discrete moments to sixth order, and treats the correlation cross-term *exactly* — the term that forces operator-splitting compromises in ADI schemes is just an off-diagonal entry of the local covariance here. The result is a sparse matrix M with $O(\Delta t^2)$ global accuracy in time — carried by a *regular* error expansion, so

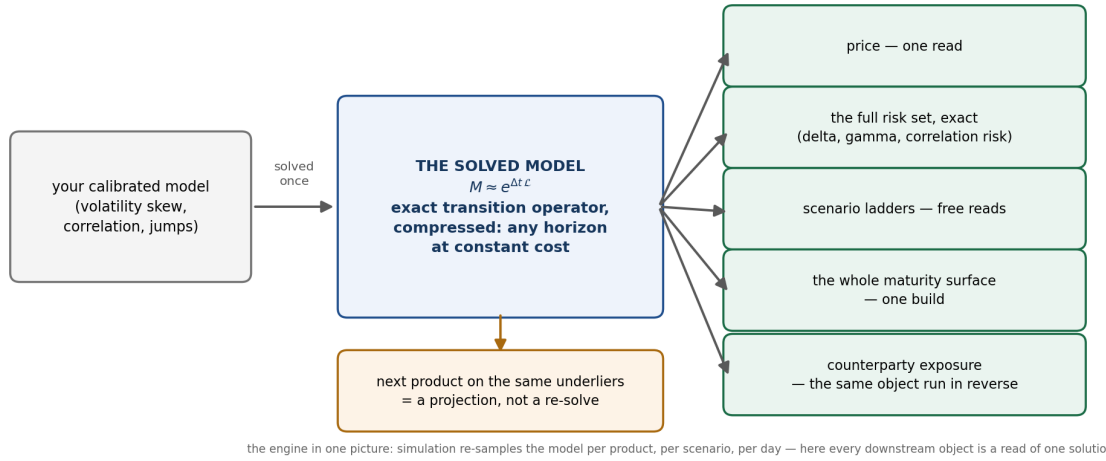


Figure 2: The engine in one picture. The model is solved once per underlier set; products, risk, scenarios, maturities and exposure are reads of the solved object, and the next product on the same underliers is a projection, not a re-solve.

a two-build Richardson extrapolation delivers $O(\Delta t^4)$ in production (certified, Appendix A) — and sixth-order accuracy in space: a one-year horizon is covered in a handful of large steps.

Compression. Repeated application of M is projected onto a small Krylov subspace, within which $e^{T\mathcal{L}}$ for *any* horizon T is a small-matrix computation. Practically: applying thirty years costs the same as applying one day (measured: $306\times$ faster than path simulation at $T = 30$, without bias), and a whole maturity surface comes from one build. Compression amortises the operator it is given but cannot repair a step’s accuracy — and we verified this by ablation: granting a competing solver the same amortisation does not close the gap (Table 2). The accuracy lives in the kernel.

Products as projections. Term-sheet events are applied *between* marches as exact projections: step-down autocall calls, memory coupons, and discrete knock-ins are handled with multi-state value fields (knocked-in $\times$ coupons-in-memory), so discrete monitoring — what real term sheets specify — is priced with no continuity correction at all. Jump risk enters by operator splitting with the jump distribution’s convolution. And because M is an explicit operator, running it *in reverse* evolves a probability density forward: the same object that prices the book also produces counterparty-exposure profiles, with no simulation-inside-simulation (§5).

One further refinement is worth stating plainly, because it applies to the whole industry. Loading a kinked contract onto a grid by sampling it at grid points silently caps *any* solver — including ours, and including Crank–Nicolson — at second-order accuracy, leaving a few basis points of grid-dependent error that oscillates from grid to grid and is invisible without a convergence ladder. We measured this in our own engine (± 3 bp on a real term sheet across production grids), published it, and removed it with a sixth-order contract-loading step: a moment-preserving smoothing of the payoff and of every observation-date projection whose vanishing moments guarantee the *contract* is unchanged (the induced price perturbation is $O(\delta^6)$ against the model’s smooth density; validated against a closed form to 10^{-8}). With it, the term sheet above is converged to 0.04 bp at the *coarsest* production grid — a two-second march (Figure 3).

4 Validation: a falsification program

Claims about a novel pricing method should be cheap to check and expensive to fake. The engine’s evidence base is organised accordingly: fifteen independent benchmarks, each with a *numeric* pass gate computed inside the benchmark itself, regenerated by one command into a provenance-stamped report (git revision, pinned library versions). A regression flips the report on its own. Table 1 summarises the anchors.

Because a serious reader will ask how the method fares against the industry’s *deterministic* solvers, we ran that comparison ourselves, both ways it can be run (Table 2). On raw, node-sampled payoffs the answer is honest and unflattering: everyone is capped by the contract-loading error, and our same-grid advantage is only

claim	reference	result
real step-down term sheet (memory, discrete KI)	identical-logic Monte-Carlo	within MC noise ($\pm 0.02\%$)
worst-of autocall + jumps (desk-standard model)	converging Euler MC	-0.05% ; MC drifts toward us
correlation risk through the breach	matched-seed MC bump	4.78 vs 4.68 ± 0.17
2-asset worst-of vs closed form	Stulz (1982), exact	10^{-8} (with contract loading)
counterparty exposure profile	brute-force outer simulation	$\leq 0.12\%$ at every date
Bermudan exercise	QuantLib finite differences	$< 0.5\%$
long-maturity cost	Merton closed form + full-path MC	flat in T ; $306\times$ at $T = 30$
the whole implied-vol surface (to 30y)	Richardson-extrapolated fine CN	0.006 bp mean of implied vol; 1.6 s, flat in T
the worst-of surface (13×6)	Stulz closed form per (K, T)	0.012 bp max over 78 checks

Table 1: Selected validation anchors. Each row is an independently gated benchmark; the full report regenerates with one command.

$\sim 1.3\times$. With the sixth-order contract loading the picture inverts: $\sim 230\times$ more accurate than Craig–Sneyd ADI and unsplit sparse Crank–Nicolson at the same grid, on price *and* on correlation risk, measured against the Stulz closed form — while the same smoothing handed to the finite-difference solvers leaves them on their second-order floor. On smooth data, where no kink interferes, the gap is $37\times$ at equal step count and grows to nearly $1000\times$ across a twelve-maturity surface — including in an ablation that grants the competing solver the *same amortisation layer*, so the gap cannot be attributed to amortisation alone.

Two null results complete the program, and we report them as prominently as the wins. First, in one dimension a tridiagonal Crank–Nicolson solve is so cheap that it wins wall-clock at equal accuracy on raw payoffs; the engine’s advantages there are the amortisation and large stable steps, not per-grid accuracy. Second, a paired hedging experiment showed that *delta*-hedging off matched-seed simulation estimates costs only ~ 3 bp/yr against exact deltas: simulation deltas are fine, and we do not sell “better deltas.” The value is in the numbers simulation cannot read (correlation risk, gamma), the cost of producing its numbers (~ 28 CPU-minutes per note-year measured for the delta batch alone), and exposure.

setting	industry deterministic solver	this engine	measured gap
raw kinked payoff, same grid	CN-ADI / sparse CN at dt -floor	order-6 kernel	$\sim 1.3\times$ (a tie)
+ 6th-order contract loading	same, given the same loading	same	$\sim \mathbf{230\times}$ (price and corr. risk)
smooth data, equal step count	Crank–Nicolson	order-6 kernel	$37\times$
12-maturity surface, one build	CN + the same amortisation (ablation)	compressed kernel	$46\text{--}991\times$

Table 2: Head-to-head against industry-standard deterministic solvers, same grids, same boundary treatment, exact references (Stulz closed form; fine-solver limits). The raw-payoff tie is published deliberately: the contract-loading step is what converts the kernel’s interior accuracy into product accuracy.

5 On the desk

What the construction buys, in the units a desk plans in. All numbers are measured on the real step-down term sheet class under the desk-standard model (local-volatility skew, correlation, jumps).

The daily book. All vintages on an underlier pair share one solved model. Each note’s complete risk view — price, the full risk set, and a scenario ladder — is ~ 47 seconds of reads, exact, with zero simulation variance; a hundred-note book is about 1.3 hours on a single GPU. By simulation the same cube is 30-plus noisy re-runs per note per day, with the correlation-risk sign unreadable on half the book. (A lone price, we note, is genuinely cheap by simulation; the cube is not.)

The breach. The entire risk profile of Figure 1 — through the knock-in — comes from the same solved object, smooth and exact where hedging decisions are made on the worst day.

Counterparty exposure without the farm. Running the solved model forward as a density and pairing it with the stored backward values yields expected-exposure and potential-future-exposure profiles — including the autocall runoff staircase — in about four minutes on one GPU, within 0.12% of a brute-force nested simulation at every observation date. The standard alternatives are a simulation-inside-a-simulation farm or a regression proxy with a known bias and a reserve against it.

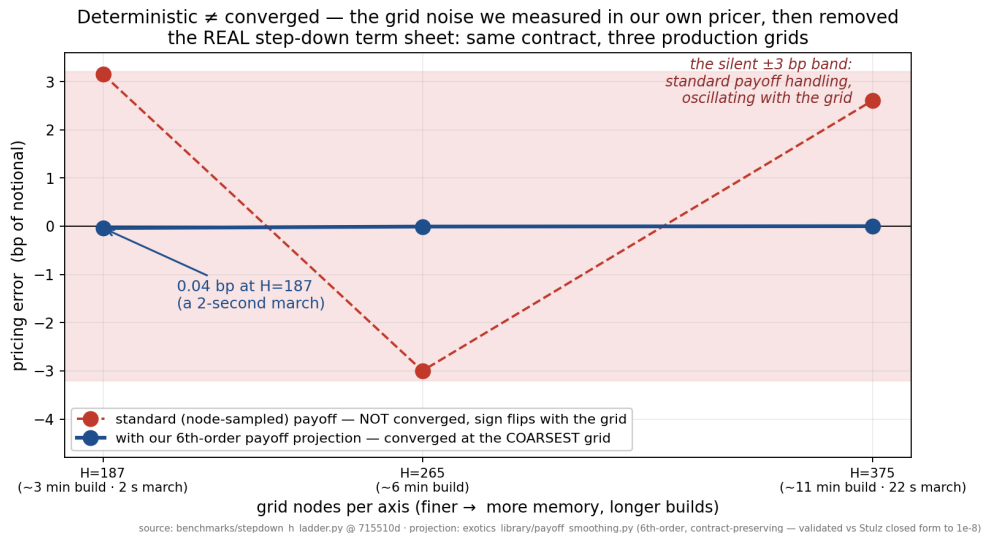


Figure 3: Deterministic $\neq$ converged. The same real term sheet priced on three production grids: standard contract loading (red) carries a ± 3 bp error that changes sign with the grid — every individual number looks plausible and noise-free. With the engine’s sixth-order contract loading (blue), the price is converged to 0.04 bp at the coarsest grid.

Long-dated and callable. Compression makes cost flat in maturity (a thirty-year note at one-year cost), and early-exercise features price by the same projection mechanism used for autocalls ($< 0.5\%$ vs QuantLib finite differences).

Operationally, the engine is a *second-opinion risk engine*: a Python library on one GPU box, beside the incumbent stack, calibrated to the desk’s own marks. Nothing is ripped out. A routing layer enforces the method’s domain of validity — handed a model outside it (for example, a non-smooth local-volatility function), the engine declines the analytic route and falls back to simulation with an explanatory note, rather than produce a confident wrong answer.

6 The volatility surface, end to end

The same architecture closes the full stochastic-local-volatility chain — vanilla surface to calibrated model to hedged exotic book — deterministically and certified at every stage. The incumbent chain (particle calibration, Monte-Carlo pricing, bump Greeks) is noisy at both ends, and we quantified each: re-running the particle calibration with a fresh seed moves the leverage function by 5–8% at the wings and the Γ of a one-year knock-in put by $\pm 7\%$ (a number the desk books as unexplained P&L); bump-Monte-Carlo Greeks near a barrier are sign-unreadable. Our chain: a deterministic forward McKean–Vlasov pass calibrates $L(S, t)$ in about a minute, self-repricing the 78-point quoted surface to ≤ 6 bp of vol against an independent Monte-Carlo referee, and holding under 10 bp across a hostile (ξ, ρ) robustness matrix and two additional surface shapes. The exotics book is then priced on the *calibrated* model by frozen-slice analytic-kernel operators spliced to a short-end scheme: a knock-in/step-down autocall book prices within Monte-Carlo noise, with Δ/Γ /vanna read from the field where bump-Monte-Carlo cannot resolve the sign.

The differentiating primitive is the *operator library*: a built operator is an object that can be applied to a whole book, evaluated at any maturity, and — newly — interpolated across market state. Storing a lattice of compressed operator cores and interpolating them intraday reprices a held-out market state to 4×10^{-5} relative price against a direct rebuild, with a single tangent step (“yesterday’s operator plus a derivative”) good across an overnight move of several vol-points without rebuilding. The parameter-derivative is analytic (the Fréchet derivative of the operator exponential, from the generator’s closed-form dependence — never by differentiating the build), and the operator manifold is low-rank, so a multi-parameter library scales with the manifold dimension rather than the number of lattice nodes. No incumbent tool has this primitive. Full method and the gated numbers: docs/volatility_slam_dunk.md, docs/operator_library_theory.md.

The living calibration. The operator library turns calibration itself into a different kind of object: *state estimation of the pricing generator*. Because the value flow of an Itô model is a linear semigroup, the exact tangent-linear and adjoint models of variational data assimilation are the operator and its transpose — no linearisation, no window limit — so streaming quotes can be assimilated into the generator’s coefficients (the leverage surface, the variance dynamics) by windowed Gauss–Newton 4D–Var with calibrated Laplace posteriors, at minutes cadence, with the exotics book re-marked off the *same* tracked operator. This is productized and gated: the tracker reproduces the research-grade result to machine precision (1.4×10^{-15}), the assimilation window pins the surface’s *rates of change* with $8.7\times$ the information volume of any per-snapshot refit, warm operator rebuilds run at ~ 7 s on one GPU, and every innovation is decomposed — noise, drift, or an attributed shock (a weak-constraint sparse channel, validated on real exchange data: exactly silent on quiet weeks, 38 named and sized events through a real volatility episode). Against the market: every commercial surface product either fits snapshots or reprices fast; none tracks a model-consistent surface trajectory, carries calibrated uncertainty, or attributes its misfit — the gap this closes.

Three factors, one GPU. The same construction extends to the hybrid models long-dated books actually require. A stochastic-local-volatility + Hull–White generator — three state variables, a fully variable anisotropic 3×3 diffusion with spot–vol and spot–rate cross-terms, and state-coupled discounting — was built and certified at 64^3 resolution on a single consumer GPU (87-minute build, 1.25 billion nonzeros): zero-coupon bond and martingale identities to -1.9 and -1.7 bp against closed forms, discounted calls within 0.7 bp of a validated two-seed Monte-Carlo referee, and the conditional variance profile $E[v | S]$ — the object at the heart of hybrid calibration — within 0.5% everywhere. This is the regime where operator-splitting schemes surrender their cross-terms and particle methods are the only incumbent; the certification campaign, including the honest residual and its measured insensitivity to every accuracy knob, is part of the public record.

7 What it newly makes possible

Beyond doing the current job better, the solved-model architecture unlocks work that is impractical today. *Shipped:* exposure/XVA-style profiles without nested simulation; whole-surface pricing from one build — measured: the entire 30-year implied-vol surface of a local-volatility model to 0.006 bp mean accuracy, re-evaluated in 1.6 seconds flat-in-maturity (the map inside every Dupire-style calibration loop, where Monte-Carlo costs a minute per iterate at ± 6 bp of noise), and the worst-of surface as a free by-product of the operator an autocall desk already holds (0.012 bp against 78 closed-form checks); real-time interactive risk (the solution already exists — dragging spot across a screen is a sequence of free reads, something path-based methods cannot mimic); fully deterministic three-asset worst-of pricing, now certified at production resolution on deployment hardware (Appendix A); the three-factor hybrid generator of the previous section; and the living calibration — online assimilation of streaming quotes into the pricing generator with calibrated uncertainty and attributed shocks. *Validated in parts:* the conditional expectation $E[v | S, t]$ at the heart of local-stochastic-volatility calibration — the object the industry’s particle method estimates with binning that destabilises in the wings — computed exactly from one forward density (self-converged to 0.015%, and within 0.5% of a Monte-Carlo referee in the three-factor setting). Newly shipped in this revision: *semilinear and forced pricing* by exponential integrators on the compressed generator — the linear part exact via φ -functions ($k \times k$ per step), only the source staged. First vertical: American exercise as a penalty source, 0.29% against QuantLib finite differences in 1D and 0.13% cross-method (penalty vs per-step projection, two algorithms on one operator) for an American down-and-out put under 2D stochastic-local vol; coupon-stream forcings integrate by high-order Duhamel quadrature to the closed-form annuity in a single step. *Research:* the full leverage-function fixed point; simulation that samples only the jumps in the general setting.

8 Scope, limitations, and the pilot

Two limitations are worth stating as plainly as the results. Continuously monitored barriers are priced with ~ 1 – 3% accuracy in the barrier region — decisively better than simulation-based estimates there, and on par with finite differences, but not at the sub-basis-point level of the discretely monitored contracts that dominate real issuance (those are exact). And the engine amortises a *calibrated* model: recalibration means re-solving, which is precisely the trade that makes everything after the solve free. Finally, the comparisons in this paper

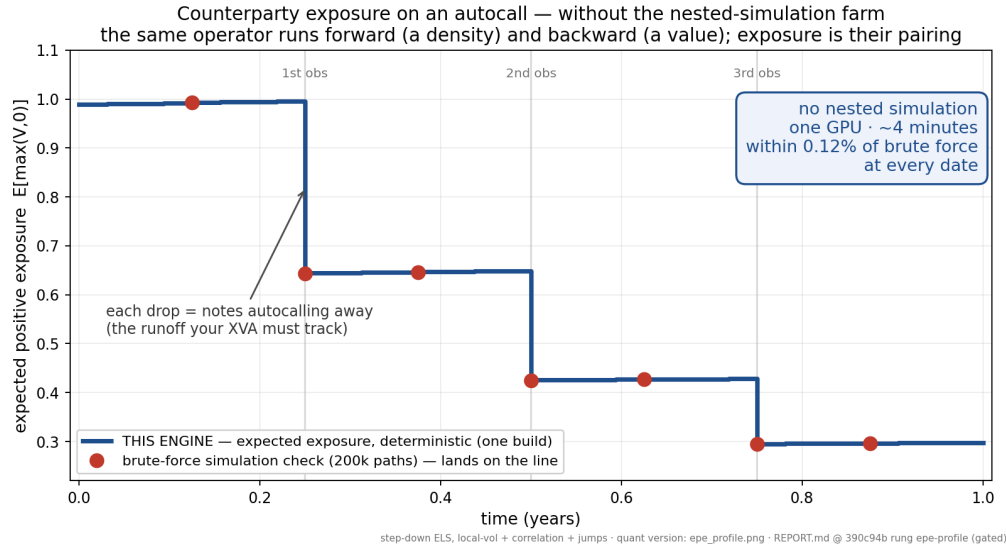


Figure 4: Counterparty exposure on an autocall, without nested simulation: the solved model run forward (a density) paired with its backward values. Brute-force simulation checks (red) land on the deterministic staircase; worst deviation 0.12%.

are against QuantLib, closed forms, and standard Monte-Carlo — not against commercial vendor stacks; the pilot below is that head-to-head, on the client's own book.

The pilot is four to six weeks: five to ten of the client's term sheets (or public/synthetic equivalents — no non-public information required), daily marks, breach risk profiles, the full per-note risk set, and optionally exposure profiles, on our hardware or a GPU box in the client's environment. Success criteria are agreed up front — prices inside the client's own simulation noise; risk numbers stable through knock-in scenarios; the full-book pack inside the daily window; and one business test: the pack is in the morning risk meeting by week three. Fixed fee; week-two review; either side stops. Contact: andrew@kryloquant.com.

A Hardware certification: the bedrock-v1 campaign

The full numerical stack — mixed precision throughout with high-precision build islands, compact banding, moment-corrected quadrature at the certified orders, and streamed (out-of-core) assembly for beyond-VRAM builds — was certified end-to-end on deployment-class hardware: an **RTX PRO 6000 Blackwell workstation** (sm_120, 96 GB), July 2026, code tagged **bedrock-v1**. Verdict: **24 of 26 rungs green, zero numerical failures across the entire campaign**; every main-pass failure was an environment-fit issue (VRAM caps and batch geometry sized for the previous-generation card, pacing ceilings), fixed and re-run green in a mop-up pass; one rung was re-classified report-tier with a documented regime finding, and one comparison was deferred to owned hardware on cost grounds. Per-rung logs are provenance-stamped in the repository.

A.1 Main pass (26 rungs, single serial GPU)

rung	verdict	time	evidence (abridged)
rails-root	PASS	1309 s	full root pytest incl. finance API + benchmark CI gate
rails-3d	PASS	704 s	3D core invariants
rails-2d-byteid	PASS	294 s	10^{-12} byte-identity vs committed references
cross-arch-d4	PASS	banked	Ampere↔Blackwell march delta 4.5×10^{-10}
gl-parity	PASS	708 s	two independent moment-quadrature rules: march delta 1.5×10^{-9}
hg-vs-resident	PASS	4786 s	host-gather streaming vs resident build: 6.9×10^{-8}
guards-fire	PASS	180 s	out-of-regime configs warn loudly and complete
order-3d-full-onestep	PASS	364 s	$O(t^3)$ one-step, variable aniso D + curl-bearing $\mu + \varphi$
order-1d-core/spatial/strang/boundary	PASS	116 s	1D closed-form exactness; $O(h^{2,4,6})$; Strang order 2
order-2d-full-onestep	PASS	176 s	full operator + Neumann walls: $\times 7.7$ – 7.8 vs $\times 8$
vs-cn (2d-worstof / smooth / krylov)	PASS	476 s	Crank–Nicolson head-to-head tables
mf-certify-all	PASS	46 s	matrix-free route: 7/7 gates
flagship-m01-mf / m01b	PASS	banked	worst-of-3 vs $4M \times 1000$ MC: -0.084% / -0.050%
big3d-h48	PASS	13344 s	$H=48$ production build repeat, self-consistency $\leq 10^{-7}$
order-2d-global / richardson, neumann-2d-rwb	mop-up	—	22 GB VRAM-cap class (Blackwell workspaces)
band-delta, finance-full, order-3d-ladder	mop-up	—	batch geometry / pacing / ladder-design classes

A.2 Mop-up dispositions

rung	verdict	time	evidence
order-2d-global	PASS	2928 s	global $O(\Delta t^2)$, $H=100$, both BCs (80 GB cap)
order-2d-richardson	PASS	1026 s	Richardson $O(\Delta t^4)$ exposed
neumann-2d-rwb	PASS	1513 s	Neumann flux with boundary reweight
finance-full	PASS	3 passes	all 30 sub-claims green (28 in-run + 2 standalone DA)
order-3d-ladder	report tier	banked	regime finding (limitation 2 below)
band-delta $H=96$	deferred	>30 h	full-band comparator cost; claim stands on 3 independent legs

The two long DA benchmarks inside **finance-full** (reduced 4D-Var + EKF on real 2D SLV operators; the calibrated Laplace-at-the-mode filter) exceeded the per-rung pacing cap on the box’s weak single-thread cores and were completed standalone, uncapped, same scripts and environment — both **rc=0**. The deferred band-delta comparison is a diagnostic whose *comparator* (the unclipped full band) is an order of magnitude costlier than the production build it checks; the band-truncation claim meanwhile stands on the analytic $\leq 10^{-8}$ /row far-field bound, the measured guard-owned out-of-regime divergence, and march-level agreement across three independent consistency rungs.

A.3 What each rung class certifies

The engine builds a sparse one-step propagator $M(\Delta t) \approx e^{\Delta t \mathcal{L}}$ for $\mathcal{L}u = \nabla \cdot (D \nabla u) + \mu \cdot \nabla u + \varphi u$ by evaluating the analytic short-time kernel in closed form entry by entry and correcting the discrete row moments to the certified spatial order. **Rails:** the implementation is the one that was validated — byte-identity against committed references, plus operator identities (the const- D eigenfunction check $M \cos \pi x \cos \pi y \cos \pi z \approx e^{-3\pi^2 d \Delta t} u_0$: the kernel realizes the correct operator, not merely a mass-conserving one). **One-step certs:** local $O(t^3)$ against

exact Taylor references for fully variable coefficients, including a deliberately curl-bearing (non-gradient) drift so the test cannot pass by accident of potential structure, and with Neumann boundary images participating (2D). **Temporal ladders:** local $O(t^3)$ composes to global $O(\Delta t^2)$, and Richardson extrapolation exposes $O(\Delta t^4)$ — certifying the error is a *regular asymptotic expansion*, which is what makes two-build fifth-order marches usable in production. **Spatial ladders:** the order- p reweight delivers global $O(h^p)$, $p \in \{2, 4, 6\}$, interior and boundary. **1D core:** where the kernel admits a provably exact closed form, rowsums match the analytic erf mass to 10^{-12} — the one doctrinally sound rowsum test — and the marched operator with drift and potential matches an independent Crank–Nicolson solve under both boundary conditions. **Consistency legs:** quadrature-rule independence, streaming-vs-resident builds, and cross-silicon (Ampere vs Blackwell) agreement, all at march level on smooth vectors — never rowsums, which are ≈ 1 by construction and certify nothing. **Guards:** outside the validity envelope the builder warns loudly and completes rather than silently producing a plausible-looking wrong operator — the failure *mode* is certified, not just the success mode. **Matrix-free route:** the const- D path (exact Fourier symbol + semi-Lagrangian transport) has its own 7-gate cert and flagship. **Finance layer:** 30 end-to-end sub-gates against references of increasing independence (closed forms, QuantLib analytic and finite-difference engines, converged bridge-corrected Monte-Carlo, bumped-MC vega). **Scale tie:** two independent $H=48$ production builds ($110k \times 110k$) agree at march level under GPU-nondeterministic scheduling.

A.4 Coverage and documented limitations

Coverage spans dimensions (1D/2D/3D), coefficient classes (variable anisotropic D , curl-bearing μ , potential φ), boundary conditions (Dirichlet, Neumann, free/far-field, mixed barrier walls), orders ($O(t^3)$ one-step; $O(h^6)$ global; Richardson $O(t^5)$ -class), cross-method references (Crank–Nicolson, QuantLib, closed forms, Monte-Carlo), and both routes (banded builder; matrix-free const- D). Five limitations are on the public record, verbatim in the certification document: a 1D boundary order-2 plateau on a non-production path; the 3D *global* temporal ladder is measured infeasible below $H \approx 96$ (the in-regime window is empty — regime physics, with the graceful degradation itself measured and the guards firing throughout; 3D temporal evidence of record is the gated one-step cert, the 2D ladders, and flagships vs MC); 3D Neumann and mixed per-axis walls implemented and wall-physics-gated (the earlier “not implemented” note was stale), with the 3D Neumann *order* certificate the remaining owned-box item; the full-band diagnostic mode’s cost; and two flagship ladder points blocked by rented-container limits. Infrastructure lessons (a silent CPU-fallback class among them) were burned into the harness: the certification runner now refuses to start without verified CUDA devices.

B The complete gate inventory

Every automated check in the repository, current as of this draft. Three layers: the fast CPU benchmark report (regenerated by one command, provenance-stamped), the pytest suite (default gate + opt-in GPU/MC tier), and the campaign runs banked with their artifacts. Names are the repository’s own.

B.1 The reproducible benchmark report (15/15 PASS)

rung	benchmark	reference
ladder-1	BS vanilla price + Greeks	QuantLib analytic (exact)
ladder-2	1D barrier price + Greeks near the barrier	QuantLib analytic + FD + MC-bump
ladder-3	worst-of put, const-vol, 2 & 3 asset	Stulz closed form + 10M-path MC anchor
ladder-berm	Bermudan put via projection	QuantLib FD (800 × 800)
greeks-vega	worst-of parallel vega	bumped-MC vega (8M paths)
ladder-4	Merton jump vanilla (shock-only)	Merton closed form + full-path MC
H2-unlock	local-vol jump-diffusion (no closed form)	Merton CF + converged full-path MC
vol-surface	the whole implied-vol surface to 30y	Richardson-extrapolated fine CN + MC
slv-calibration	LSV leverage fixed point (2D)	input SSVI + independent MC refereee
ssvi-roundtrip	SSVI → Dupire → march → implied vol	the input surface (identity)
operator-library-1d	operator interpolation + analytic Fréchet tangent	direct builds at held-out θ
da-twin-1d / da-ekf-1d	reduced 4D-Var / streaming EKF+RTS	synthetic twins, known truth
da-4dvar-1d	TRUE windowed non-autonomous 4D-Var (exact adjoint)	drifting- θ twin
da-eta-1d	weak-constraint sparse- η shock detector	3-way attribution battery

B.2 The pytest suite (42 files, 158 test functions)

The default gate runs the CPU tier on every invocation; GPU/Monte-Carlo tests are the opt-in `slow` tier. One line per file; each file's own docstring states its claim in full.

file	#	tier	certifies
test_api_a567	6	CPU	API completeness: surfaces on every route, generic scenario cube, market overrides, one-build maturity ladders
test_api_endpoints	7	CPU	public API safety net: symbols, surface amortization, library online-vs-rebuild speed
test_asian_vertical	1	CPU	Asian (split-route) vertical end-to-end
test_assimilation_api	4	CPU	living-calibration finance surface: vega-weighted R , quote windows, validated construction
test_basket3_vertical	4	CPU	3-asset worst-of vertical (matrix-free FFT route)
test_bcaxes_port	1	GPU	per-axis wall-BC interface regression (2D)
test_bc_3d_mixed	2	GPU	3D mixed-axis walls (Dirichlet×Neumann×free) wall physics + refusal
test_benchmark_report	2	GPU	the one-command report regenerates and gates
test_block_seed_compression	4	CPU	block-seed compression of a stationary-barrier propagator
test_build_propagator (+3d, +mf)	7	CPU	the construction facade: 1D/2D wiring, 3D wiring, matrix-free path
test_build_report	3	CPU	per-build manifest / provenance
test_compression_fold	3	CPU	compressed-semigroup backend fold
test_dd_rectangular	1	CPU	high-precision kernel-build rails (rectangular grid)
test_engine	10	CPU	the Engine conductor, evolve semigroup, operator algebra
test_etd_integrators	4	CPU	exponential integrators on the finance operator: k-space $\equiv$ oracle, orders 2/4, Duhamel $\equiv$ annuity
test_american_semilinear	3	CPU	American put via penalty ETD vs QuantLib FD (0.29%); structure + stability guard
test_semilinear_gpu	1	GPU	American-DO put under 2D SLV: penalty vs projection cross-method (0.13%)
test_event_projection	3	CPU	event-projection hook (autocall/coupon mechanics)
test_facade_finance_parity	1	GPU	finance 2D builders $\equiv$ numerical facade
test_finance_verticals	6	CPU	priced verticals through <code>api.price/greeks</code>
test_gpu_verticals	4	GPU	the GPU finance verticals in the gate
test_helpers_geometry / readout	8	CPU	coefficient-geometry helpers; the shared sub-pixel readout
test_jumps_mc	2	GPU	MC jumps vs the Merton closed form (A2)
test_lookback_vertical	1	CPU	lookback vertical end-to-end
test_matrixfree_walls	4	CPU	Neumann/mixed conormal-image walls + corners vs analytic truth (A11)
test_mcmc_primitives	4	CPU	stochastic primitives vs their definitions
test_operator_matvec_parity	1	CPU	CSR vs matrix-free compression-contract parity

file	#	tier	certifies
test_operator_serialize	4	CPU	save/load: byte-identity, stale-matrix guard, fresh-process serve (A10)
test_payoff_smoothing	7	CPU	the 6th-order contract loading (moment-preserving)
test_slv_hw_api	6	CPU	3-factor SLV+HW route guard rails (A12): sub-cell, memory, $\xi\sqrt{dt}$, identity key
test_smoke	12	GPU	product-layer reproducibility smoke
test_stepdown_instrument	9	CPU	step-down autocall term-sheet mechanics + validation
test_termsheet	3	CPU	term-sheet spec $\rightarrow$ instrument; unknown terms refuse (A8)
test_tracker	6	CPU	OperatorTracker: recovery, exact estimator equivalence, trust region, checkpoint, tangent economics, shock channel
test_transport_asian	2	CPU	path-dependent split via composed solver outputs
test_two_layer_separation	1	CPU	the numerical layer imports with no finance on the path
test_validation_harness	6	GPU	parametrized cert harness over the engines
test_weight_routing	3	GPU	invariant-weight plumbing + Lanczos auto-routing
test_worst_of_adapter	2	CPU	finance worst-of adapter $\equiv$ validated engine

B.3 Campaign rungs banked with artifacts

Beyond the always-on gates: the 2D leverage-calibration referee (a coupled two-level common-random-number Monte-Carlo with per-path Richardson bias cancellation; base and stressed configurations 78/78 quote checks each, seed-validated with an analytic pure-local-vol control); the kernel-vs-ADI long-end crossover measurement (published as a null: no wall-clock crossover — the hybrid claim rests on the operator artifact, not marching speed); the three-factor SLV + Hull–White existence proof of Section 7 with its discriminator campaign (referee validated by a CRN two-level dt -ladder before any attribution; every accuracy knob measured); the real-data assimilation chain on exchange (Deribit) options — quiet-window null, historical-episode shock attribution, and a live collector armed for the next volatile session; and the productized tracker’s GPU equivalence rung (six gates, machine-precision agreement with the research-grade result). Each is banked with logs and arrays in the repository and summarised in the corresponding methodology documents.

References

- [1] Risk.net, “Review of 2020: chaos on a roll,” December 2020; “Natixis’s €260m hit blamed on big books and Kospì3 product,” 2018.
- [2] KED Global, “Korean investors doomed to suffer losses from HSCEI ELS,” January 2024; The Korea Herald, “Top finance firms put aside ₩1.7tr for ELS compensation,” 2024.
- [3] Bloomberg, “Why structured notes are booming again,” September 2025; SPi global issuance data, 2025.
- [4] Risk.net, “Equity derivatives house of the year: JP Morgan,” Risk Awards 2026; Structured Retail Products, “Harbor files for autocallable ETFs with worst-of QIS structure,” 2026.
- [5] Zanders, “To hedge or not to hedge: navigating the catch-22 of FRTB’s PLA test”; ActiveViam, “Basel endgame 2026: the Fed’s FRTB recalibration,” 2026.
- [6] L. Abbas-Turki, S. Crépey et al., “XVA principles, nested Monte Carlo strategies, and GPU optimizations,” *IJTAF* 21(6), 2018.
- [7] H.-O. Kreiss, V. Thomée, O. Widlund, “Smoothing of initial data and rates of convergence for parabolic difference equations,” *CPAM* 23, 1970.
- [8] R. Stulz, “Options on the minimum or the maximum of two risky assets,” *JFE* 10, 1982.
- [9] The engine’s validation repository: one-command gated benchmark report, model-validation dossier, and the falsification-program benchmarks (CN/ADI head-to-heads, hedging A/B, grid-noise ladder). Access on request.