

Implied Volatility in One Straight Line: Machine Precision at Vector-Hardware Throughput

Wolfgang Schadner*

September 18, 2026

Abstract

We develop a Black–Scholes implied-volatility inverter that preserves machine-precision accuracy while reducing per-quote batch cost on vector hardware. At fixed scaled price, intrinsic coordinates expose the real and periodic obstructions that determine convergence. In these coordinates all nonalgebraic dependence is confined to a single one-dimensional function, and every remaining coefficient row is an exact rational polynomial. A conformal re-expansion of the same rows extends the representation into the deep wing. The resulting book kernel uses 63 tabulated coefficients, one intrinsic-region branch and no iteration, and answers 29,997 of 30,000 quotes in a tradeable 2024 S&P 500 feed. Four routed charts provide the wider service domain. Against an independent 40-digit oracle, the shipped entry point’s worst observed relative volatility error is 5.3×10^{-16} . On one CPU core, the kernel’s AVX-512 and AVX2 batch paths process the market feed at 11.6 and 8.3 times the rate of the public scalar release build of the Let’s Be Rational reference (LBR). On an NVIDIA H100 GPU, device-resident throughput is 0.061 ns per quote in file order and 0.031 ns after partitioning by intrinsic region. Thus the market-relevant inversion is reduced from branch-rich refinement to fixed straight-line arithmetic without relaxing the accuracy target.

Keywords. implied volatility; Black–Scholes inverse; machine precision; rational expansion; SIMD; GPU

*University of Liechtenstein, Fürst-Franz-Josef-Strasse, LI-9490 Vaduz. wolfgang.schadner@uni.li

1 Introduction

This paper shows that the market-relevant Black–Scholes inverse can be evaluated to machine precision by a fixed straight-line kernel rather than quote-by-quote refinement. On a tradeable 2024 S&P 500 feed, the kernel answers 29,997 of 30,000 normalized quotes without iteration, with a fitted table of only 63 double-precision coefficients plus exact rational rows stored as constants. The remaining quotes fall through to a routed inverter over the declared service domain. Against an independent arbitrary-precision oracle, the shipped entry point has a worst observed relative volatility error of 5.3×10^{-16} on the stated validation campaigns. On one CPU core, the kernel’s AVX-512 and AVX2 batch paths process the market feed at 11.6 and 8.3 times the rate of the public scalar release build of the reference method, Let’s Be Rational (LBR). On an NVIDIA H100 GPU, the same arithmetic runs at 0.061 ns per quote in file order and 0.031 ns per quote when the two intrinsic regions are maintained as separate device-resident partitions. These are batch-throughput measurements, not isolated-query latencies.

High scalar accuracy is already available. The remaining design question addressed here is how to retain it with lane-uniform arithmetic. Explicit approximations can be inexpensive and vectorizable, but they are usually designed for a bounded region or a weaker accuracy target [3, 5, 16, 17, 25, 27]. Machine-precision methods instead start from a strong approximation and refine it on an accurately reformulated Black equation. The reference used here is LBR [11], which uses a rational seed followed by two Householder corrections. Its scalar accuracy is excellent, but the runtime still contains the control flow and special-function evaluations of a refinement method. Precomputed approaches move work offline. Global Chebyshev interpolation [7] and the PDE table method [19] demonstrate that large parts of the inverse can be tabulated efficiently, while recent scalar constructions reduce the cost of iterative refinement further [13, 14]. A vectorized LBR port uses masked whole-array operations on CPUs and GPUs [22]. The relevant comparison is therefore not simply approximation versus iteration. It is whether the inverse admits a representation whose high-accuracy runtime is already lane-uniform.

The representation developed here follows from the analytic structure of the inverse. The normalized Black price has a generalized-inverse-Gaussian representation in total variance [23]. Its endpoint factorization identifies a logarithmic variable in which the deep wing is regular

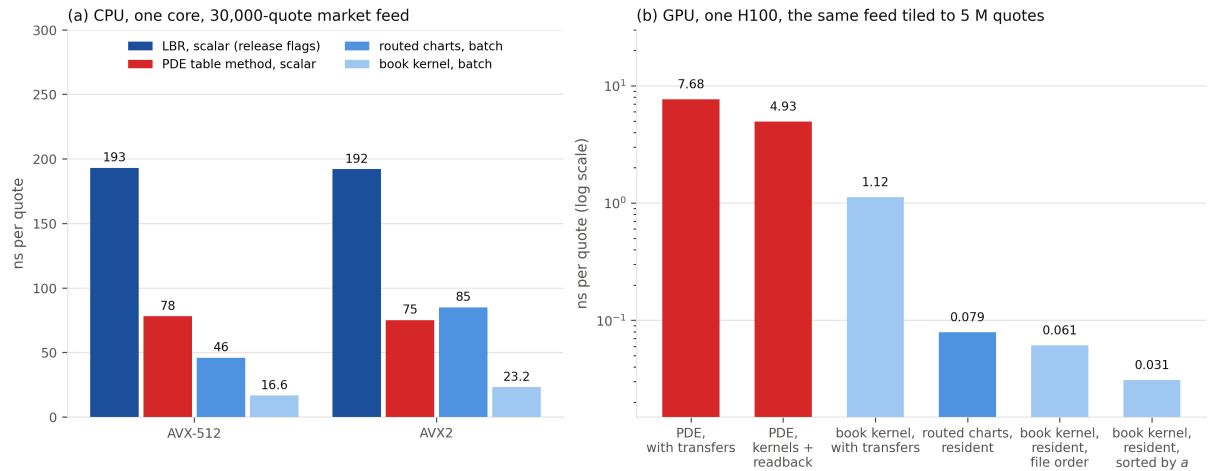


Figure 1: Cost per quote on the same market feed, with the methods at machine precision in blue and the less accurate method in red. CPU measurements use one core and a common timing loop, and GPU measurements use one NVIDIA H100 GPU. The book kernel maintains machine-precision accuracy on the feed while moving the dominant workload from branch-rich refinement to straight-line batch evaluation. Detailed timing conventions and all underlying rows are reported in the Online Appendix.

and a complementary-price variable in which the upper endpoint is regular. A second analytic result gives the exact radius of the fixed-scaled-price inverse branch in complex moneyness [24]. Together these facts determine the coordinates used below rather than merely motivating them.

The main mathematical contribution is an exact structure theorem for the fixed-scaled-price expansion. Let h be the absolute log-moneyness and v the total volatility, both defined in Section 2, let a denote the real obstruction scale of the inverse, let $\theta = h/a$, and write $V = v/\theta$. With $A_0(a)$ the at-the-money limit of h/v at fixed scaled price and $V_0(a) = a/A_0(a)$, we prove that

$$V(\theta^2, a) = \sum_{m \geq 0} V_m(a) \theta^{2m}, \quad V_m(a) = V_0(a)^{2m+1} P_m(A_0(a)^2),$$

where every P_m is a rational polynomial and the only nonalgebraic object is the one-dimensional function $A_0(a)$. The proof is constructive. After the exact price equation is expanded at fixed scaled price, the Gaussian density cancels from the formal inverse recursion and parity forces the polynomial degree structure. The theorem turns a multidimensional inverse problem, over the part of the domain occupied by a traded book, into one short one-dimensional table plus fixed rational rows. Above the periodic obstruction at $2\pi i$, a conformal re-expansion of the same rows restores rapid convergence without introducing a second fitted surface. Computationally, this shifts most of the nonlinear inversion work offline: at runtime the book kernel evaluates one one-dimensional approximation and a fixed sequence of polynomial rows rather than repeatedly evaluating the Black equation.

The numerical method has two layers. The *book kernel* applies the rational rows directly and covers essentially the entire observed S&P 500 feed. The *routed fallback* preserves the wider service domain with four lane-uniform charts, namely a matched small-moneyness expansion, a bivariate interior table, a logarithmic deep-wing chart, and a complement-based large-volatility chart. The last two use fixed numbers of refinement steps, three in the wing and one at the upper endpoint. There is no tolerance-driven loop. The distinction matters for interpretation. The claim is not that every point in the feasible strip has a closed-form inverse. It is that the empirically dominant part admits a fixed straight-line representation once the single one-dimensional function $A_0(a)$ has been tabulated, while the remainder can be served without destroying single-instruction multiple-data (SIMD) regularity.

Accuracy and performance are evaluated separately. Accuracy is scored against an independent 40-digit inverter of the *given double-precision price*, not against a round trip through a forward pricer. The worst cases are recomputed at 60 digits. On actual market prices LBR, the routed charts and the book kernel are all at machine precision, so the practical market-data advantage is throughput and deterministic arithmetic rather than a material accuracy difference. The accuracy separation appears only at two diagnostic endpoints, near the upper price bound and in the subnormal deep wing. Performance is measured from the same normalized inputs in common binaries on the CPU and by mirrored device kernels on the GPU. The main article reports the decisive accuracy and throughput results. The Online Appendix contains the proof, complete chart derivations, oracle-gating procedure, bit-identity tests, rejected approximation designs, detailed benchmark protocol, warm-start analysis and artifact map.

2 Analytic structure and the straight-line kernel

Let K be the strike, F the forward, T the time to expiry and σ the implied volatility, and write

$$h = |\log(K/F)|, \quad w = \sigma^2 T, \quad v = \sqrt{w} = \sigma \sqrt{T}$$

for the absolute log-moneyness h , the total variance w and the total volatility v . After the standard put-call parity reduction to the out-of-the-money twin and normalization by $\min(F, K)$,

calls and puts reduce to the same inversion problem. Given $h \geq 0$ and $c \in (0, 1)$, recover v from

$$c = C(h, v) = \Phi\left(\frac{v}{2} - \frac{h}{v}\right) - e^h \Phi\left(-\frac{v}{2} - \frac{h}{v}\right). \quad (1)$$

Here Φ is the standard normal distribution function, and φ below is its density. This normalization is convenient because it leaves the small price itself intact in the deep wing. The convenience wrapper used in the implementation performs the parity reduction internally. All analysis below starts from (h, c) .

An exact integral representation exposes the two endpoint structures:

$$C(h, v) = \frac{e^{h/2}}{\sqrt{2\pi}} \int_0^v e^{-t^2/8} e^{-h^2/(2t^2)} dt, \quad C(h, \infty) = 1. \quad (2)$$

The boundary-layer factor $e^{-h^2/(2t^2)}$ controls the small-price wing, while the Gaussian tail controls $c \rightarrow 1$. This decomposition explains why a single raw approximation is inefficient over the full strip because the two endpoints are regular in different variables.

For the small-price endpoint define

$$W = \frac{h^2}{2w} = \frac{h^2}{2v^2}. \quad (3)$$

Equation (2) can then be written exactly as

$$c = \gamma(h) e^{-W} W^{-3/2} H(W), \quad \gamma(h) = \frac{h e^{h/2}}{4\sqrt{\pi}},$$

where H is regular in the wing. This is the endpoint form of the generalized-inverse-Gaussian representation [23]. It also shows the appropriate conditioning. If $\nu = \partial C / \partial v$, the relative condition number for recovering v from c is $\kappa = c / (v\nu)$. Deep in the wing, $\log c \sim -W$, hence $\kappa \sim 1/(2W)$. Very small prices are not intrinsically ill conditioned when the inversion is performed on $\log c$.

The opposite endpoint must be read in the complement. As $v \rightarrow \infty$, $c \rightarrow 1$ and the two terms of (1) cancel, whereas $1 - c$ has a regular Gaussian-tail representation. For $c \geq 1/2$, the subtraction $1 - c$ is exact in binary floating point by Sterbenz's lemma [20]. The limitation is instead the information already lost when the caller rounded a price extremely close to one. Once $v \gtrsim 8$, that input-representation error can exceed 10^{-13} near the money and grows rapidly thereafter. The implementation therefore distinguishes its own inversion error from this unavoidable input floor.

The market-relevant coordinates arise from the small-moneyness layer. Define

$$\rho = \frac{c}{\exp m(h)}, \quad B(A) = \frac{\varphi(A)}{A} - \Phi(-A) = \int_A^\infty \varphi(u) u^{-2} du. \quad (4)$$

The function B is strictly decreasing, and $A_0 = B^{-1}(\rho)$ gives the at-the-money inverse at fixed scaled price. The same scaled price determines

$$a = \log\left(1 + \frac{1}{\rho}\right), \quad \theta = \frac{h}{a}. \quad (5)$$

Here a is the distance to the real moving-target obstruction. The fixed- ρ physical branch is holomorphic for

$$|h| < \min\{a, 2\pi\},$$

and this radius is exact [24]. The other obstruction is the periodic return at $h = \pm 2\pi i$. Thus a and 2π jointly determine the nearest singularity, while θ measures the quote relative to the real obstruction scale.

This geometry gives an exact convergence statement for the expansion used below. With $\tau = \theta^2$, the radius of the τ -series is

$$R_\tau(a) = \begin{cases} 1, & a < 2\pi, \\ (2\pi/a)^2, & a \geq 2\pi. \end{cases} \quad (6)$$

Below 2π , the real obstruction controls convergence and the raw τ -series is efficient on the market band. Above 2π , the periodic obstruction moves inside the raw τ disk and motivates the conformal continuation developed below. The coordinates are therefore not fitted feature engineering. They are fixed by the singularity geometry of the inverse.

Throughout the paper, *machine precision* means an observed relative error in σ below 10^{-15} on the stated validation sets, corresponding to at most a few units in the last place (ULP) in double precision. It does not mean correct rounding or a certified supremum over the continuum. The validated service domain is $h \leq 16.2$, $v \leq 8$, with a rescue band to $h \leq 16.5$ for $W \lesssim 40$. Points beyond the deeper validated corner return a flagged non-result. Points with $v > 8$ return a flagged best-effort value because the double representation of $1 - c$ is then the limiting information source. Detailed conditioning bounds are given in Online Appendix A, and the complete status contract in Online Appendix C.

The singularity geometry now yields the representation used by the fast path. Set

$$\tau = \theta^2, \quad V(\tau, a) = \frac{v}{\theta}, \quad A_0(a) = B^{-1}(\rho), \quad V_0(a) = \frac{a}{A_0(a)}. \quad (7)$$

At fixed ρ , the inverse branch is odd in h , so V is even in θ . The central structural result is that, once the single function $A_0(a)$ is known, the entire expansion is algebraic.

Proposition 1 (Rational rows). *Fix $\rho \in (0, \infty)$ and let $A_0 = B^{-1}(\rho)$ and $V_0 = a/A_0$. The physical branch has the formal expansion*

$$V(\tau, a) = \sum_{m \geq 0} V_m(a) \tau^m, \quad V_m(a) = V_0(a)^{2m+1} P_m(A_0(a)^2),$$

where $P_0 = 1$ and, for every $m \geq 1$, P_m is a polynomial with rational coefficients of degree at most $m - 1$.

The mechanism is visible before the full proof, and it is a single function. Write $\xi_\pm = \pm v/2 - h/v$ for the two Gaussian arguments in (1), so that $\varphi(\xi_+) = e^h \varphi(\xi_-)$, and put

$$M_\rho(\xi) = \frac{\Phi(\xi) + \rho}{\varphi(\xi)}, \quad M'_\rho(\xi) = 1 + \xi M_\rho(\xi). \quad (8)$$

Dividing (1) by $\varphi(\xi_+)$ shows that the price equation at scaled price ρ is exactly

$$M_\rho(\xi_+) = M_\rho(\xi_-). \quad (9)$$

The inverse at fixed ρ is therefore the involution $\xi_+ \leftrightarrow \xi_-$ on a level set of one function, and ρ enters only through the initial condition of the linear equation in (8). As $h \rightarrow 0$ the two points merge at the fixed point $\xi_\pm \rightarrow -A_0$, and (9) degenerates into $B(A_0) = \rho$. Differentiating (9) along the branch and using $M'_\rho = 1 + \xi M_\rho$ gives

$$v'(h) = M_\rho(\xi_+) = M_\rho(\xi_-), \quad (10)$$

with $v'(0) = M_\rho(-A_0) = 1/A_0$. Every further h -derivative of v follows from (10) by the chain rule and $M'_\rho = 1 + \xi M_\rho$, so it is a polynomial in h , v , $1/v$ and the single value M_ρ , and at $h = 0$ a polynomial in $A_0^{\pm 1}$ with rational coefficients. No Gaussian density or distribution function survives beyond A_0 itself. Parity in h and a count of the degrees then give the form in

Proposition 1. Online Appendix A carries this out as an explicit formal-series recursion, which is what the coefficient generator runs.

The first rows, written in $y = A_0^2$, are

$$P_0 = 1, \quad P_1 = \frac{1}{24}, \quad P_2 = \frac{21 - 2y}{5760}, \quad P_3 = \frac{16y^2 - 174y + 1143}{2903040}. \quad (11)$$

Rows through order 22 were generated in exact rational arithmetic and checked term by term against the arbitrary-precision solve. The theorem is useful computationally because the rows do not need to be fitted, stored as floating-point surfaces or regenerated quote by quote.

For runtime evaluation put $x = h^2/A_0^2$ and $y = A_0^2$. Proposition 1 becomes

$$v = \frac{h}{A_0} S(x, y), \quad S(x, y) = \sum_{m=0}^M x^m P_m(y). \quad (12)$$

with M the number of rows carried. Since $P_0 = 1$, the implementation sums the rows from $m = 1$ and forms $v = t + t x S_1$ with $t = h/A_0$ in one fused multiply-add, so the leading term h/A_0 is not rounded a second time. The only fitted object is therefore

$$G(a) = \frac{1}{V_0(a)} = \frac{A_0(a)}{a}.$$

The book kernel stores this one-dimensional information in two cells. On $a \in [0, 2\pi]$, G uses 23 Chebyshev coefficients. On $a \in [2\pi, 700]$, the table uses 40 coefficients for the flattened large- a combination $V_0/\sqrt{a}$ in $u = 1/\sqrt{a}$, from which $1/A_0$ follows directly. The complete table is therefore 63 doubles. No bivariate table is needed by the book kernel.

Equation (6) explains why two row evaluations share this table. For $a < 2\pi$, the raw series in τ has unit radius. The kernel uses sixteen rows for $\theta \leq 0.35$. For $a > 2\pi$, the periodic singularities at $h = \pm 2\pi i$ rather than the real pair limit the raw series. Let

$$z = \frac{h^2}{4\pi^2}, \quad q_c = \frac{z}{(\sqrt{1+z}+1)^2}. \quad (13)$$

This conformal variable maps the corresponding cut to the unit circle while keeping real h at small q_c . Re-expanding the *same* rational rows in q_c again gives a triangular polynomial with fixed coefficients. The kernel carries sixteen conformal rows, reaching $h = 4$ for $a \geq 20$ and a linearly shorter range as $a \downarrow 2\pi$, where the real and periodic obstructions are equidistant.

The exact at-the-money line $h = 0$ is handled before the kernel, avoiding the division implicit in $\rho = c/\text{expm1}(h)$. For $h > 0$, the fast path is as follows.

1. Form $E = \text{expm1}(h)$, $\rho = c/E$ and $a = \log(1 + E/c)$.
2. Recover A_0 (or $1/A_0$) from the appropriate one-dimensional table cell.
3. If $a < 2\pi$ and $h \leq 0.35a$, evaluate the raw rational rows in (12).
4. If $a \geq 2\pi$ and $h \leq \min(4, 0.16a + 0.85)$, evaluate the conformal rows in q_c .
5. Otherwise, pass the quote to the routed inverter of Section 3.1.

Within either accepted region the arithmetic is fixed. The only intrinsic-region branch is at $a = 2\pi$. The entry-point reach tests merely decide whether this kernel is applicable or whether the fallback should answer instead.

This is the computational payoff of Proposition 1. Once A_0 has been recovered from the one-dimensional table, the shipped whole-book path executes about two hundred and fifty explicitly ordered fused multiply-adds, three or four divisions and two square roots, with the remaining elementary transforms supplied by fixed kernels rather than transcendental library

Chart	Region	Runtime construction
NEAR	$h < 0.3$, below the upper seam	Matched even expansion in $s = h/B^{-1}(\rho)$ with no iteration.
FAR	Interior between the seams	Bivariate Chebyshev table for $W = h^2/(2w)$ in $(h, \log c)$ with no iteration.
WING	$W \gtrsim 3.8$	Peeled log-price equation for W , table-seeded and followed by exactly three Newton steps.
UPPER	Above the $v = 1.85$ seam	Complement form, a 13-coefficient endpoint seed and exactly one Householder step on the exact equation.

Table 1: The routed fallback. The two iterative finishers use fixed step counts and branchless price kernels. There is no tolerance-driven loop. Detailed formulas, fitting domains and oracle gates are in the Online Appendix.

calls. There is no root-finding loop and no data-dependent polynomial degree. The reference solver, by contrast, obtains its last-bit accuracy by correcting an initial approximation on the Black equation. The representation therefore changes the runtime unit of work. Accuracy that would otherwise require repeated high-accuracy forward evaluations is carried by precomputed one-dimensional coefficients and exact rational rows.

On the 30,000-quote market feed used in Section 4.2, the kernel accepts 29,997 quotes, with 91.1% through the raw rows and 8.9% through the conformal rows. This split is by the kernel’s two intrinsic regions and differs from the routed-chart mix of the same feed reported in Section 4.2. The three declined quotes are extreme-wing points beyond the conformal reach. On a separate 1,200-point 40-digit truth set spanning both intrinsic regions, the worst errors are 1.69 ULP in the raw region and 1.99 ULP in the conformal region. A broader boundary-concentrated campaign is reported in Section 4.1. The computational consequence is that the singularity geometry, coefficient structure and market occupancy align. The high-probability workload is exactly the part of the inverse for which one short table and exact rows suffice.

3 Full-domain algorithm and vector implementation

3.1 Full-domain completion

The book kernel is deliberately specialized to the part of the inverse map occupied by a traded book. The public entry point retains the declared service domain. A quote outside the kernel’s reach is sent to one of four routed charts. Their purpose is coverage, not to compete with the book kernel on its natural workload. Each chart resolves one part of the endpoint structure of (2), and after routing each executes a fixed operation count.

The small-moneyness fallback uses the same scaled price ρ as Section 2. With $A = B^{-1}(\rho)$ and $s = h/A$, it evaluates

$$v = v_0\left(\frac{s}{\sqrt{2\pi}}\right) + h^2 V_2(s) + h^4 V_4(s) + h^6 R(s, h), \quad v_0(t) = 2\sqrt{2} \operatorname{erf}^{-1}(t), \quad (14)$$

where erf^{-1} is the inverse error function. The functions B^{-1} , V_2 , V_4 and the residual R are frozen Chebyshev surrogates gated against independent arbitrary-precision values. This chart remains useful outside the narrower raw-row reach of the book kernel.

The interior chart tabulates W rather than w or σ ,

$$W \approx \sum_{i=0}^{d_h} \sum_{j=0}^{d_\lambda} c_{ij} T_i(x_h) T_j(x_\lambda), \quad \lambda = \log c, \quad (15)$$

where T_i are Chebyshev polynomials, x_h and x_λ are the cell-normalized coordinates of h and λ on $[-1, 1]$, d_h and d_λ the degrees and c_{ij} the frozen coefficients, and recovers $w = h^2/(2W)$. The

shipped table has 239 cells and 27,211 double coefficients (about 220 KB), with price-direction degree at most 16 and moneyness degree at most 12. Cell sizes follow the branch-point geometry of the inverse rather than a uniform grid. The derivation and the observed degree law are given in the Online Appendix.

For smaller prices the wing chart takes logarithms of the endpoint factorization and solves

$$W = \tilde{L} - \frac{3}{2} \log W + \log H(W). \quad (16)$$

where $\tilde{L} = \log(\gamma(h)/c)$ collects the known price and moneyness terms of the endpoint form below (3) and H is its regular wing factor. The fitted object is a peeled residual of $\log H$, and a low-degree table supplies the starting value. Exactly three Newton steps then reach the machine-precision gate on the validated wing domain, including prices down to 10^{-320} . For $h > 16.2$ a rare scalar rescue branch extends coverage while $W \lesssim 40$. Still deeper points return a flagged non-result rather than a plausible extrapolation.

At the opposite endpoint the implementation works with $1 - c$. The complement equation has the exact root $x_0 = -\Phi^{-1}(\frac{1}{2}e^{-a_U/2})$ on the line $h = 0$, where $a_U = h - 2\log(1 - c)$ and $x = v/2$, and x_0 is a function of a_U alone for every h . The chart stores this one-dimensional function as a short Chebyshev series and corrects it by a 13-coefficient surface in two bounded coordinates, which places the seed within 4.9×10^{-5} of the root over the whole chart. One fourth-order Householder step on the exact complement equation then suffices, because the step's error constant is bounded by 217/18 on the chart, so any seed within 9.5×10^{-5} reaches 10^{-15} . The derivative is cancellation-free and its higher derivative ratios are elementary, and the residual needs one exponential and two scaled complementary error functions. The full derivation is in Online Appendix B.

The seams are fixed before runtime. The wing boundary is

$$c_w(h) = C\left(h, \frac{h}{\sqrt{7.6}}\right),$$

the level set $W = 3.8$, and the upper boundary is $c_{\text{top}}(h) = C(h, 1.85)$. Both are represented by frozen one-dimensional polynomials. In the intrinsic variables these boundaries are coordinate lines, and neighboring charts overlap at machine precision on gated overlap bands (Online Appendix B), so a one-ULP seam-fit perturbation cannot expose an unvalidated region. Figure 2 shows the resulting partition and the empirical concentration of the S&P 500 feed.

3.2 Vector implementation and deterministic arithmetic

The analytic construction matters for throughput only if its fixed algebraic structure survives the software path. The scalar entry and the batch implementations therefore share the same kernel code. Polynomial evaluations are explicitly ordered fused multiply-add chains. Divisions, square roots and the few elementary transforms appear at fixed locations. The AVX-512 path advances eight quotes per register and the AVX2 path four. Routing decisions use the same integer tests and frozen seam polynomials as the scalar entry, so vectorization changes scheduling rather than the mathematical map being evaluated.

A mixed book is highly unbalanced across charts. On the market population used in Section 4.2, roughly 92% of quotes belong to the small-moneyness routed chart. The batch driver therefore evaluates that dominant chart speculatively on each SIMD block, stores the lanes that belong there in input order, and compresses only the minority into a deferred buffer. When a sampled probe indicates a different route mix, the driver instead counting-sorts by chart. The book kernel is simpler still. It is attempted before routing and, on the traded feed, hands only three quotes in thirty thousand to the fallback. Its batch driver works through each tile in passes with short loop bodies. The first pass forms a for every register. The second evaluates

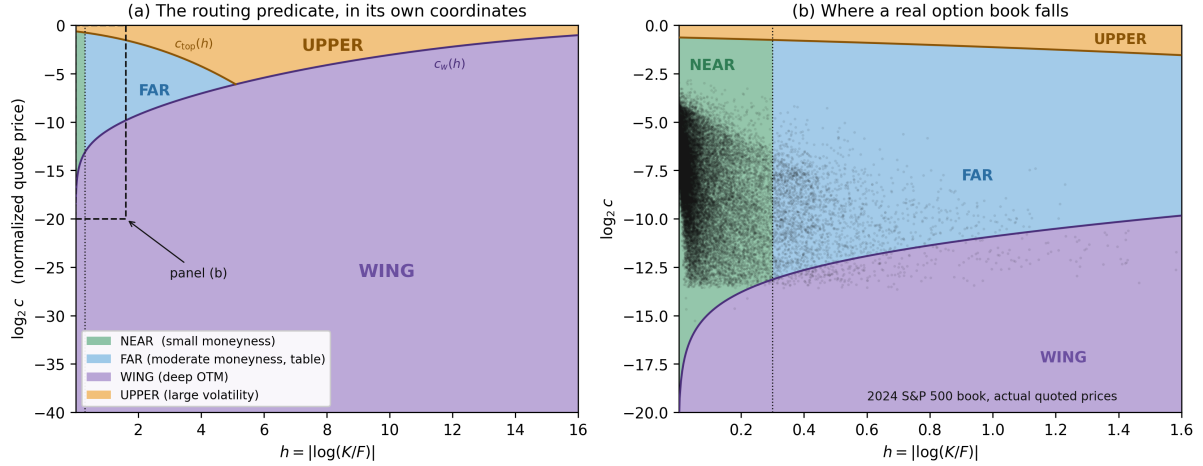


Figure 2: Routing in normalized price and absolute log-moneyness. The routed charts tile the declared service domain, while the 2024 S&P 500 quotes concentrate overwhelmingly in the part subsequently covered by the straight-line book kernel.

registers that lie wholly in the raw-row region in place and collects all other lanes into per-tile lists, and the conformal-row list is evaluated as a table pass followed by a row pass. A single loop over the whole chain carries one dependency chain per register that is longer than the processor’s reorder window, so consecutive registers cannot overlap. The split leaves the per-lane arithmetic unchanged and is worth about a fifth of the feed time (Online Appendix C). Quotes that the kernel declines are classified with the vectorized seam polynomials, and those routed to UPPER are evaluated in the same two-pass manner.

Bit identity is treated as a correctness invariant. With GCC’s default floating-point contraction, inlining context can change which multiply-add pairs are fused and thereby change last bits. The reported builds therefore use `-ffp-contract=off`, with every intended fusion explicit in the source. The CUDA port uses `-fmad=false` with the same explicit operation order. On the stated validation feeds, scalar, AVX2 and AVX-512 builds under GCC, the AVX2 and AVX-512 builds under Clang, and the CUDA port return the same double for every quote tested. The Online Appendix gives the poison test used to verify that the vector kernels genuinely execute and the cross-compiler byte comparisons.

The checked API has three outcomes. It returns a validated result on the stated service domain, a flagged best-effort result in the near-saturation regime $v > 8$ where input representation is the accuracy limit, or a quiet NaN with an explicit status for invalid or unsupported inputs. A convenience wrapper accepts forward, strike, undiscounted price, expiry and call/put flag and performs the parity reduction to (1) internally.

4 Numerical results

The numerical evidence addresses two separate questions: whether the inverter reaches the stated accuracy target and whether the resulting fixed-operation representation lowers batch cost on vector hardware. We establish accuracy first with independent arbitrary-precision oracles, then measure throughput on the market workload used for the principal comparisons.

4.1 Accuracy

Accuracy is measured against an independent `mpmath` inverter that solves the *given double-precision price* at 40 significant digits. We do not score against a volatility used to generate a price, because the forward map itself loses digits near both endpoints. The worst points of

Region	Book kernel in front		Routed charts		LBR	
	max rel.	ULP	max rel.	ULP	max rel.	ULP
Near	3.9×10^{-16}	3	6.7×10^{-16}	4	5.5×10^{-16}	3
Far	4.3×10^{-16}	2	4.3×10^{-16}	2	4.2×10^{-16}	2
Wing	5.3×10^{-16}	3	5.3×10^{-16}	3	degrades at extreme prices	
Upper	3.8×10^{-16}	2	3.8×10^{-16}	2	5.1×10^{-14}	426
All	5.3×10^{-16}	3	6.7×10^{-16}	4	—	—

Table 2: Worst relative volatility error and worst ULP error on the 16,039-point regular grid. The book kernel answers 425 grid points and hands the remainder to the routed charts. LBR is the public C++ reference implementation called from the same normalized input pairs.

every campaign were recomputed at 60 digits and rounded to the same doubles. Relative error is $|\hat{\sigma} - \sigma_{\text{oracle}}|/\sigma_{\text{oracle}}$, with $\hat{\sigma}$ the returned volatility and σ_{oracle} the oracle value. ULP counts are taken relative to the oracle value rounded to double.

The regular test set is a 150×110 grid over $h \in [0.02, 16]$ and $v \in [0.02, 8]$, with 16,039 feasible points. A hand-built stressed set targets the deep wing, near-intrinsic prices and the small- h deep-OTM corner. A 3,511-point chart-stress feed targets every routing region and seam, and a separate 120,000-quote broad feed exercises coverage and scalar/SIMD identity through $h = 16.5$ and $v = 8$. The book-kernel campaigns additionally concentrate on $a = 2\pi$, $\theta = 0.35$, the conformal reach line and both table endpoints. Table 2 reports the regular grid. The “book kernel in front” column is the shipped entry point. The book kernel is attempted first and the routed charts answer all points outside its reach.

The regular grid is not the strongest test of the book kernel because only a small fraction of a uniform box lies in its market-oriented reach. A separate 20,000-point campaign concentrates on both sides of $a = 2\pi$, the $\theta = 0.35$ boundary, the conformal reach line, the $h = 4$ cap, $a \rightarrow 0$, $a \rightarrow 700$ and $h \rightarrow 10^{-6}$. The shipped entry point has worst relative error 5.3×10^{-16} (four ULP, RMS 1.4×10^{-16}) on that campaign. The routed charts alone reach 9.2×10^{-16} . The forty worst points were recomputed at 60 digits with unchanged rounded oracle values. Across the stated validation sets no point returned by either of our machine-precision paths exceeds 10^{-15} .

The two regimes in which LBR separates from our results are deliberately extreme. Near $c \rightarrow 1$, its normalized objective loses conditioning while the complement formulation remains stable. In the deep wing its $\sqrt{FK}$ -normalized input $\beta = ce^{-h/2}$ reaches the subnormal range before the forward-normalized c used here. These effects are diagnostic rather than economically typical. On 1,150 actual projected prices sampled from the 2024 S&P 500 book and scored with an independent 30-digit oracle, all three solvers are at machine precision, with worst relative errors of 6.2×10^{-16} for the book kernel, 8.5×10^{-16} for the routed charts and 5.2×10^{-16} for LBR. The practical market-data claim is therefore throughput and deterministic arithmetic, not superior accuracy on ordinary quotes.

The claims above are empirical validation claims, not a proof of correct rounding or of a continuum supremum. Polynomial errors can in principle peak between sampled points. The online material therefore records every validation population, the boundary-concentrated campaign, the 60-digit rechecks, the oracle code and the generation gates so that the distinction between theorem, numerical gate and observed error is explicit.

4.2 Throughput

The market workload is derived from one year of 2024 end-of-day S&P 500 option quotes. Starting from 1.85 million observations with positive volume and open interest, a tradeability screen requiring a live uncrossed market, a five-cent minimum mid, relative spread at most 25%, open interest at least 100, volume at least 10 and at least one week to expiry leaves 641,072

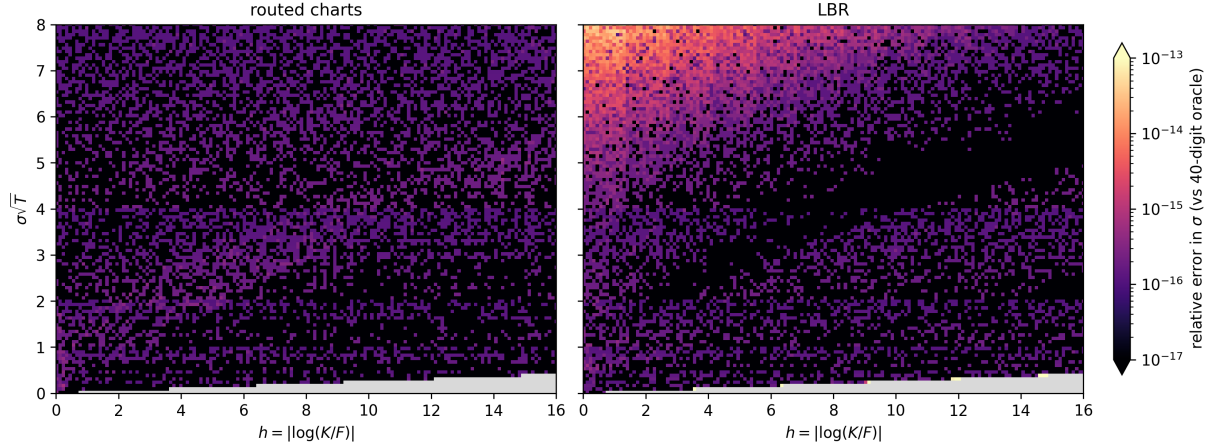


Figure 3: Relative volatility error against the 40-digit oracle on the regular grid for the routed charts (left) and LBR (right). Both are at machine precision through the market-relevant interior. The visible differences occur near the upper endpoint and where LBR’s reduced wing variable becomes subnormal. Grey points have prices that round to 0 or 1 in double precision.

quotes. Calls and puts are projected to the normalized OTM branch using the vendor forward and discount curve. A 30,000-quote resample preserves the route mix: 92.0% NEAR, 6.8% FAR, 1.2% WING and no UPPER quotes. The maximum observed h is 3.4. This feed is used for throughput because the broad validation box is about 70% wing and is not representative of a traded equity-index book.

CPU measurements use one pinned core of an Intel Core i5-1145G7 under Linux/WSL2 with GCC 11.4. Every method starts from the same raw (h, c) pair in the timed loop. Our builds use `-O3 -ffp-contract=off -fno-fast-math -funroll-loops`. LBR is compiled from the public sources at its release flags, and the PDE table method [19] at its published compile settings. Table 3 reports the quiet-host one-binary campaign. The full timing protocol, repeated runs, LTO cross-check and chart-pure measurements are in Online Appendix E.

Relative to the public scalar LBR release build, the book batch path therefore processes the feed at 11.6 times the rate on AVX-512 and 8.3 times on AVX2. Its scalar entry is also faster on this feed (115 versus 193 ns on the AVX-512-capable build), and it remains faster on every tile including the synthetic UPPER tile. The routed inverter is slower as a scalar entry but becomes competitive once its dominant chart is evaluated in SIMD. These rates reflect the change in the runtime unit of work established in Section 2. The book kernel advances through a fixed polynomial schedule rather than repeatedly evaluating an accurate Black residual. On a build without SIMD and without hardware fused multiply-add, the reference is faster. The speed claim is explicitly a vector-hardware claim.

GPU results. The four fallback charts and the book kernel were mirrored as CUDA device functions with the same explicit operation order. Table 4 reports an NVIDIA H100 PCIe under CUDA 12.4, with the feed tiled to about five million quotes. The 0.031 ns figure is a throughput measurement on a device-resident book partitioned by the side of $a = 2\pi$. It is not the latency of an isolated query. In file order the same kernel takes 0.061 ns per quote, which is the unconditional figure when no partition is maintained.

The partition is not perfectly persistent. Across 596,566 consecutive-day observations of 43,485 book nodes, the side of $a = 2\pi$ changes on 2.2% of node-days, so a deployment that keeps the previous partition leaves a small fraction of lanes in the other region. The cost of repartitioning every snapshot was not measured. We therefore report both 0.061 ns in file order and 0.031 ns for the partitioned regime rather than presenting the latter as unconditional.

Build	LBR	PDE table	Routed batch	Book scalar	Book batch
AVX-512	193	78	46	115	16.6
AVX2	192	75	85	116	23.2
AVX-512, NEAR tile	186	74	25	113	15.6
AVX-512, FAR tile	245	66	224	129	16.4
AVX-512, WING tile	249	66	264	163	20.7
AVX-512, UPPER* tile	187	109	55	161	33.9

Table 3: Median nanoseconds per quote on the 30,000-quote market feed, one CPU thread, medians of 31 passes of eight sweeps. All methods are in the same binary and timed by the same loop. The book batch path is bit-identical to its scalar entry. The lower block times region-filtered tiles of the same feed (27,574, 2,084 and 342 quotes, each cycled to 30,000). The NEAR tile is served almost entirely by the raw rows and the FAR and WING tiles by the conformal rows, so the tiles approximate the kernel’s two intrinsic regions. The routed batch driver speculates on the NEAR chart, so the pure FAR and WING tiles are its worst case rather than a chart cost. *The feed contains no UPPER quotes, so this tile is synthetic. It keeps the feed’s moneyness and draws v uniformly from $[1.9, 6]$. The kernel declines every quote of that tile, so its book columns measure the declined path, the attempt followed by the routed UPPER chart.

The nearest published table method [19] targets about 10^{-7} accuracy and ships a coefficient file of roughly 46 MB, compared with the 63-coefficient book table and the 220 KB full-domain interior table here. On the same H100 and feed its public OpenCL double-precision path takes 7.68 ns per quote under its transfer-inclusive convention, versus 1.12 for the book kernel under the same convention. Its worst relative volatility error on the traded feed is 1.2×10^{-5} , versus 6.2×10^{-16} for the book kernel. The comparison shows that moving from a lower-accuracy table to machine precision does not require giving up batch throughput when the approximation is organized around the analytic geometry of the inverse.

Recent iterative work provides useful context but not a like-for-like benchmark. The trimmed Householder constructions of [13, 14] report 137 ns per call in Java on their own grid for a trimmed Householder construction, with the accuracy figures stated in their papers. The vectorized LBR port [22], run on our feed and host through its Numba backend, takes 411 ns per quote at one thread in a loaded session and has worst relative error 3.3×10^{-14} against the book-kernel reference values. Because that run was not taken under the quiet protocol, we treat it only as an order-of-magnitude check. Its author reports a resident Triton figure of 0.56 ns per quote on an H100 NVL. We have not run that kernel ourselves.

4.3 Scope and reproducibility

The performance claims have several explicit boundaries. The route mix is one year of one equity-index underlying and need not transfer to FX, single-name, commodity or intraday books. Absolute CPU timings are hardware-specific, and the largest gains occur on AVX-512. AVX2 remains faster than the public scalar reference on the market feed, whereas a machine without SIMD or hardware FMA does not realize the intended arithmetic model. The machine-precision validation box is finite, and for $v > 8$ the double representation of $1 - c$ becomes the dominant accuracy floor. The H100 results are datacenter-FP64 measurements. Consumer GPUs have very different double-precision throughput. Finally, the fastest GPU comparison to the vectorized LBR port is an author-reported number on an H100 NVL, not a measurement from our campaign.

The implementation, frozen tables, exact-row generator, arbitrary-precision oracle, golden vectors, accuracy harnesses, bit-identity tests, CPU benchmarks and CUDA kernels form the reproducibility package for version v0.3.1. The package is public at <https://github>.

GPU workload	ns per quote
Routed charts, bucket-ordered full book	0.079
Book kernel, full feed, file order	0.061
Book kernel, full feed, partitioned by $a < 2\pi$	0.031
Book kernel, NEAR tile, resident	0.060
Book kernel, FAR tile, resident	0.070
Book kernel, WING tile, resident	0.041
Book kernel with UPPER fallback, UPPER* tile, resident	0.054
Book kernel, partitioned, with host transfers	1.12
PDE table method, kernels and readback	4.93
PDE table method, with host transfers	7.68

Table 4: Median H100 throughput. Resident rows exclude host transfers. The tile rows run one region-filtered subset of the feed at a time and approximate the kernel’s two intrinsic regions as in Table 3. The UPPER* tile is the synthetic tile of Table 3. The last three rows include host transfers, the last one under the PDE method’s own transfer-inclusive convention. These three rows are bound by the host link and not by the kernel, and they come from one earlier session on the same GPU model in which both methods ran on the same card. Every reported device result of our method is bit-identical to the CPU scalar entry on the tested quotes.

`com/wol-fi/volfi`. The raw outputs behind the reported campaigns are stored with the corresponding harnesses. Three external inputs are not redistributed: the licensed market feed, the LBR sources and the PDE method’s source/table package. The artifact documentation explains how to supply each of them, while the accuracy suite and the exact-row generator run without the licensed feed. The Online Appendix records the full generation protocol, rejected approximation designs, detailed benchmark settings and artifact map.

5 Conclusion

The Black–Scholes inverse has a market-relevant representation in which one tabulated function and exact rational-polynomial rows replace quote-by-quote refinement. The representation is dictated by the inverse’s singularity geometry. The raw rows serve the real-obstruction region, and a conformal re-expansion of the same rows serves the periodic-obstruction region. Four routed charts retain the wider service domain. On the stated validation sets the resulting entry point remains below 10^{-15} relative error, while the book kernel substantially lowers per-quote batch cost on AVX-512 and AVX2. On the H100 its file-order rate is about a fifth better than that of the routed charts, and partitioning by intrinsic region halves the cost again. The main computational gain therefore comes from changing the representation of the inverse, not from weakening its accuracy target.

References

- [1] G. A. Baker and P. Graves-Morris. *Padé Approximants*. Cambridge University Press, Cambridge, 2nd edition, 1996. doi: 10.1017/CBO9780511530074.
- [2] O. Barndorff-Nielsen and C. Halgreen. Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 38(4):309–311, 1977. doi: 10.1007/BF00533162.
- [3] M. Brenner and M. G. Subrahmanyam. A simple formula to compute the implied standard deviation. *Financial Analysts Journal*, 44(5):80–83, 1988. doi: 10.2469/faj.v44.n5.80.
- [4] W. J. Cody. Rational Chebyshev approximations for the error function. *Mathematics of Computation*, 23(107):631–637, 1969. doi: 10.1090/S0025-5718-1969-0247736-4.
- [5] C. J. Corrado and T. W. Miller. A note on a simple, accurate formula to compute implied standard deviations. *Journal of Banking & Finance*, 20(3):595–603, 1996. doi: 10.1016/0378-4266(95)00014-3.
- [6] J. Demmel and H. D. Nguyen. Parallel reproducible summation. *IEEE Transactions on Computers*, 64(7):2060–2070, 2015. doi: 10.1109/TC.2014.2345391.
- [7] K. Glau, P. Herold, D. B. Madan, and C. Pötz. The Chebyshev method for the implied volatility. *Journal of Computational Finance*, 23(3):1–31, 2019. doi: 10.21314/JCF.2019.375. arXiv:1710.01797.
- [8] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991. doi: 10.1145/103162.103163.
- [9] P. S. Hagan, D. Kumar, A. S. Lesniewski, and D. E. Woodward. Managing smile risk. *Wilmott Magazine*, pages 84–108, 2002. September issue.
- [10] IEEE. IEEE standard for floating-point arithmetic, 2019. IEEE Std 754-2019, doi: 10.1109/IEEESTD.2019.8766229.
- [11] P. Jäckel. Let’s be rational. *Wilmott*, 2015(75):40–53, 2015. doi: 10.1002/wilm.10395.
- [12] B. Jørgensen. *Statistical Properties of the Generalized Inverse Gaussian Distribution*, volume 9 of *Lecture Notes in Statistics*. Springer, New York, 1982.
- [13] F. Le Floc’h. Faster monotone implied volatility solver. arXiv:2605.22427, 2026.
- [14] F. Le Floc’h and J. Healy. Implying volatility: How fast can we go? arXiv:2605.29102, 2026.
- [15] R. W. Lee. The moment formula for implied volatility at extreme strikes. *Mathematical Finance*, 14(3):469–480, 2004. doi: 10.1111/j.0960-1627.2004.00200.x.
- [16] M. Li. Approximate inversion of the Black–Scholes formula using rational functions. *European Journal of Operational Research*, 185(2):743–759, 2008. doi: 10.1016/j.ejor.2006.12.028.
- [17] S. Manaster and G. Koehler. The calculation of implied variances from the Black–Scholes model: A note. *The Journal of Finance*, 37(1):227–230, 1982. doi: 10.1111/j.1540-6261.1982.tb01105.x.
- [18] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. Chapman & Hall/CRC, Boca Raton, FL, 2003.

- [19] I. Matić, R. Radoičić, and D. Stefanica. A PDE method for estimation of implied volatility. *Quantitative Finance*, 20(3):393–408, 2020. doi: 10.1080/14697688.2019.1675898. SSRN 3264356 (2018).
- [20] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser, 2nd edition, 2018. ISBN 9783319765259. doi: 10.1007/978-3-319-76526-6.
- [21] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, 2010.
- [22] R. Saqur. Fast-vollib: A fast implied volatility library for Python with PyTorch, JAX, and CUDA fused-kernel backends. arXiv:2604.27210; <https://github.com/raeidsaqur/fast-vollib>, 2026.
- [23] W. Schadner. Implied volatility as a quantile. *Quantitative Finance*, 2026. doi: 10.1080/14697688.2026.2721562. forthcoming.
- [24] W. Schadner. Moneyness as a global projective modulus of the Black–Scholes inverse. SSRN working paper 7462078, 2026. doi: 10.2139/ssrn.7462078.
- [25] D. Stefanica and R. Radoičić. An explicit implied volatility formula. *International Journal of Theoretical and Applied Finance*, 20(7):1750048, 2017. doi: 10.1142/S0219024917500480.
- [26] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, Philadelphia, extended edition, 2019. doi: 10.1137/1.9781611975949.
- [27] Y. Xia and Z. Cui. An exact and explicit implied volatility inversion formula. *International Journal of Financial Engineering*, 5(3):1850032, 2018. doi: 10.1142/S2424786318500329.

Online Appendix

This appendix holds the material the main article points to. Appendix A proves Proposition 1 and records the analytic facts behind it, Appendix B derives the four routed charts, Appendix C gives the kernels, drivers and complete hot path of the implementation, Appendix D documents the offline construction, oracle gates and extended validation, Appendix E gives the benchmark protocol and extended performance results, and Appendix F is the artifact map. Equations, figures and tables of the main article are cited by their main-text numbers and are not repeated.

A Analytic foundations and proof of the rational-row theorem

A.1 Reduction, conditioning and the input floor

The price equation (1) is written for the out-of-the-money call ($K \geq F$). An in-the-money option first maps to its out-of-the-money twin by put–call parity, an out-of-the-money put ($K < F$) is the dual call with F and K exchanged, and the surviving premium is normalized by $\min(F, K)$. The wrapper of Appendix C implements exactly this map. It is the reduction the reference solver uses [11], save that the normalizer is $\min(F, K)$ rather than $\sqrt{FK}$, a choice that matters in the deep wing (Appendix D.3).

In the endpoint form $c = \gamma(h)e^{-W}W^{-3/2}H(W)$ below (3), H is an elementary, error-function-free convergent integral with $H(W) \sim 1 - (a + \beta)/W + \dots$ for the fixed pair $(a, \beta) = (3/2, h^2/16)$, and Appendix B.5 places it in a two-parameter family. The map (1) is analytic and strictly monotone in v but treacherous at both ends. As $v \rightarrow 0$ the price collapses like $c \sim \varphi(h/v)v/h$, so a plain Newton iteration on $c - C(h, v)$ has a vanishing derivative, and as $v \rightarrow \infty$ the two terms of (1) cancel. Every formula below therefore works with whichever of c and $1 - c$ is small, and no formula amplifies the rounded complement through an ill-conditioned intermediate.

The conditioning statement of the main text can be quantified. A relative perturbation ε of the input price moves the exact root by $\kappa\varepsilon$ with $\kappa = c/(v\nu)$. On the validation grid of Appendix D.3 ($h \leq 16$, $v \leq 8$) the relative condition number ranges from 7×10^{-4} deep in the wing to 925 at $v = 8$ near the money, so the rounding of the input alone moves the root by between 7×10^{-20} and 1.0×10^{-13} relative. The propagated floor exceeds 10^{-15} only above $v \approx 4.8$ near the money and $v \approx 6$ at $h = 3$, and it grows to 10^{-3} near $v \approx 16.6$, where $1 - c$ reaches machine epsilon. On the box a traded book occupies ($h \leq 3.4$, $v \leq 1.5$) the condition number never exceeds 1.21 and the floor is at most 1.3×10^{-16} , below one rounding of the output. The errors reported in Appendix D.3 are the inverter’s own, scored on the exact double it was given, so the total error of a computed volatility is this propagated rounding plus the few units in the last place reported there. Scoring against a round trip through a forward pricer would instead conflate forward-pricer cancellation near both endpoints with inverter error, and on our feeds that construction error reaches tens to hundreds of ULP in the wings, identically for every method.

A.2 Origin of the intrinsic coordinates

Two objects of Section 2 of the main text come directly from the integral (2). Substituting $t = h/u$ turns the layer factor $e^{-h^2/(2t^2)}$ into the Gaussian $\varphi(u)$, and after the price is scaled by $\text{expm1}(h)$ the $h \rightarrow 0$ limit of the integral is the function B of (4), so the at-the-money inverse $A_0 = B^{-1}(\rho)$ is dictated by the representation rather than chosen. The radius a_ρ of (5) is the value of $|h|$ at which the moving target $\rho(e^h - 1)$ collides with the asymptotic value 1 of the direct map, and the cap at 2π is the first nonzero solution of $e^h = 1$. The radius is exact rather than a bound because these collisions are the only obstructions. The direct map has no finite critical points, its v -derivative being the Gaussian $\varphi(d_1)$ [24].

A.3 Proof of Proposition 1

In the intrinsic coordinates (7) the τ -series of V has radius exactly 1 for $a < 2\pi$ and $(2\pi/a)^2$ above, because the only obstructions of the fixed- ρ branch are the collisions at $k = \pm a + 2\pi in$ and $k = 2\pi in$ recorded above. The rows of that series are exact, which is Proposition 1 of the main text, restated here in the form the proof uses.

Proposition 2. *Fix $\rho \in (0, \infty)$, let $A_0 = B^{-1}(\rho)$ and $V_0 = a/A_0$. The branch of the price equation with $V(0, a) = V_0$ has the formal expansion $V(\tau, a) = \sum_{m \geq 0} V_m(a) \tau^m$ with $V_m = V_0^{2m+1} P_m(A_0^2)$, where $P_0 = 1$ and, for every $m \geq 1$, P_m is a polynomial with rational coefficients of degree at most $m - 1$.*

Proof. Write k for the log-moneyness h of (1), $v = k/A$, and $q = k^2 = a^2 \tau$.

Step 1, the price equation in closed form. By (1), $c = \Phi(v/2 - A) - e^k \Phi(-v/2 - A)$, and since $\varphi(A \mp s) = \varphi(A) e^{\pm As - s^2/2}$, $\Phi(v/2 - A) = \Phi(-A) + \varphi(A) I_+$ and $\Phi(-v/2 - A) = \Phi(-A) - \varphi(A) I_-$ with $I_{\pm} = \int_0^{v/2} e^{\pm As - s^2/2} ds$. Hence $\rho = c/(e^k - 1) = -\Phi(-A) + \varphi(A) (I_+ + e^k I_-)/(e^k - 1)$. Expanding $e^{-s^2/2}$ and substituting $s = t/A$, $I_{\pm} = \sum_{n \geq 0} \frac{(-1)^n}{2^n n!} A^{-2n-1} \int_0^{k/2} t^{2n} e^{\pm t} dt$. The $n = 0$ term of $I_+ + e^k I_-$ is $(e^k - 1)/A$, which together with $-\Phi(-A)$ is exactly $B(A)$; the remaining terms give

$$\rho = B(A) + \varphi(A) \sum_{n \geq 1} \frac{(-1)^n}{2^n n!} A^{-2n-1} G_n(k), \quad G_n(k) = \frac{\int_0^{k/2} t^{2n} \cosh(k/2 - t) dt}{\sinh(k/2)}. \quad (17)$$

Substituting $t = (k/2)(1 - u)$ gives $G_n(k) = (k/2)^{2n} \int_0^1 (1 - u)^{2n} \cosh(ku/2) du / [\sinh(k/2)/(k/2)]$. The Beta integrals of the cosh series and the Bernoulli series of $x/\sinh x$ are rational, so $G_n(k) = k^{2n} g_n(k^2)$ with $g_n \in \mathbb{Q}[[q]]$ and $g_n(0) = 4^{-n}/(2n + 1)$. Collecting powers of q ,

$$\rho = B(A) + \varphi(A) S(A, q), \quad S(A, q) = \sum_{m \geq 1} q^m S_m(A^{-1}), \quad (18)$$

where each S_m is an odd polynomial in A^{-1} with rational coefficients and degree at most $2m + 1$.

Step 2, formal inversion. From $B'(A) = -\varphi(A)/A^2$ and $\varphi' = -A\varphi$, $B^{(j)} = \varphi b_j$ with $b_1 = -A^{-2}$ and $b_{j+1} = b'_j - Ab_j$; likewise $(\varphi S_m)^{(j)} = \varphi \sigma_{m,j}$ with $\sigma_{m,0} = S_m$ and $\sigma_{m,j+1} = \sigma'_{m,j} - A\sigma_{m,j}$. All b_j and $\sigma_{m,j}$ are Laurent polynomials in A with rational coefficients, of parity $(-1)^{j+1}$ under $A \mapsto -A$, the powers of A^{-1} lying in $[2 - j, j + 1]$ for b_j and in $[3 - j, 2m + 1 + j]$ for $\sigma_{m,j}$. Write $A(q) = A_0 + \delta$ with $\delta = \sum_{m \geq 1} c_m q^m$. Expanding (18) about A_0 and dividing by $\varphi(A_0) \neq 0$,

$$\sum_{j \geq 1} \frac{b_j(A_0)}{j!} \delta^j + \sum_{j \geq 0} \sum_{i \geq 1} \frac{\sigma_{i,j}(A_0)}{j!} q^i \delta^j = 0, \quad (19)$$

in which no exponential, error function or density survives. The coefficient of q^m in the $j = 1$ term of the first sum is $-c_m/A_0^2$, and every other contribution to q^m involves only $c_1, \dots, c_{m-1}$, so with $[\cdot]_m$ the coefficient of q^m ,

$$c_m = A_0^2 \left(\sum_{j \geq 2} \frac{b_j(A_0)}{j!} [\delta^j]_m + \sum_{j \geq 0} \sum_{i=1}^m \frac{\sigma_{i,j}(A_0)}{j!} [\delta^j]_{m-i} \right), \quad (20)$$

which is the recursion the generator evaluates in exact rational arithmetic. By induction on m , c_m is an odd polynomial in A_0^{-1} with rational coefficients and degree between 1 and $2m - 1$. A product of j of the c_i with indices summing to m has parity $(-1)^j$ and degree in $[j, 2m - j]$, so every term inside the brackets is odd with degree in $[2, 2m + 1]$, oddness excludes degree 2, and the factor A_0^2 lowers the range by two.

Step 3, the rows. $V = a/A = V_0 (1 + \sum_m d_m q^m)^{-1}$ with $d_m = c_m/A_0$, even in A_0^{-1} with degree in $[2, 2m]$; hence $V = V_0 \sum_m e_m q^m$ with $e_0 = 1$ and, for $m \geq 1$, e_m a polynomial

in $d_1, \dots, d_m$ of total order m , even with degree in $[2, 2m]$. With $q = a^2\tau$ and $a = V_0 A_0$, $V_m = V_0 a^{2m} e_m = V_0^{2m+1} A_0^{2m} e_m(A_0)$, and $A_0^{2m} e_m$ is a polynomial in A_0^2 of degree at most $m - 1$ with rational coefficients. $\square$

The first step of (20) returns $c_1 = -1/(24A_0)$ and $P_1 = 1/24$. The rows to order 22 that the kernel stores are (20) carried out exactly, checked term by term against the 40-digit solve, and their leading coefficients carry the Bernoulli numbers of the $x/\sinh x$ series of Step 1. Whether the P_m obey a closed recurrence in m alone, which would generate them at runtime, is open, so the kernel stores them.

The runtime consequence is (12). Once A_0 is known, the chart is a fixed polynomial sweep of exact rational constants, and the only fitted object is $G(a) = A_0(a)/a$, a smooth function of one variable with $G(0) = 1/\sqrt{2\pi}$ whose Chebyshev series decays at the geometric rate set by the poles of V_0 at $a = \pm 2\pi i$. The error budget of the chart therefore reduces to a one-dimensional Bernstein statement for G plus the explicit first dropped row.

A.4 Book-kernel construction

Nothing in Proposition 1 knows about the small-moneyness band. Read in h , the radius statement says the series converges for $|h| < \min(a, 2\pi)$, which in the wing is never less than 2π , so the same rows reach across the interior into the deep wing. Two things make that a kernel. The first is a table that runs to large a , where $V_0 \sim \sqrt{2a}$, with one cell for G on $[0, 2\pi]$ and one for the flat function $V_0/\sqrt{a}$ in the variable $u = 1/\sqrt{a}$ on $[2\pi, 700]$, which covers prices down to 10^{-300} at a cost of 23 and 40 coefficients. Tabulating $1/V_0$ itself on the second cell, where it falls to 0.05, would let the Clenshaw sum's absolute rounding become a relative error of several units in the last place, whereas $u V_0/\sqrt{a}$ is $1/A_0$ directly, so the wing side needs no division for h/A_0 . The second is the conformal variable (13). Above $a = 2\pi$ the raw series in h^2 converges slowly toward the periodic obstruction, and re-expanding in q_c restores a fast rate. In q_c the series is again a triangular polynomial with fixed constants, combinations of the P_m with binomials and powers of $4\pi^2$, evaluated by the same Horner sweep in $1/A_0^2$. Twelve conformal rows reach $h = 3$ and sixteen reach $h = 4$ at any a above 2π , where the raw series needed twenty rows for $h = 3$ and could not reach 4. The kernel carries sixteen, valid to $h = 4$ for $a \geq 20$ and to a reach that shortens linearly toward $a = 2\pi$, where the real and periodic obstructions are equidistant. Below 2π the raw rows serve, sixteen of them for $\theta \leq 0.35$. The two regions meet at $a = 2\pi$, which is also the cell boundary of the table, so the kernel has a single branch, three or four divisions and two square roots, and no two-dimensional table anywhere. The region above $\theta = 0.35$ below 2π , which the feed never visits, stays with the large-volatility chart of Appendix B.4.

B Full-domain completion

The book kernel answers the traded book, not the whole feasible domain. Its reach ends at $\theta = 0.35$ below $a = 2\pi$ and at $h = 4$ in the wing, and the feasible strip extends to prices near the smallest positive doubles and to total volatilities at which $1 - c$ is a few roundings. The remainder is the routed inverter of Section 3.1 of the main text, whose four charts are derived here. Their arrangement mirrors the base integral (2). The boundary layer at $t = 0$ is resolved by a matched coordinate, the regular Gaussian tail by an elementary price transform, and the middle, where neither simplification applies, by a table. Figure 2 of the main text shows the resulting partition.

B.1 The small-moneyness chart: NEAR

For h below the threshold $h < 0.3$ the geometry collapses toward the explicit at-the-money line $c = \text{erf}(\sigma/2\sqrt{2})$. The level sets of c crowd against an exactly known curve, and direct

two-dimensional approximation of $w(h, c)$ degenerates precisely where the answer is simplest. The cure is the matched coordinate $s = h/A$ with $A = B^{-1}(\rho)$, which reduces to v as $h \rightarrow 0$. In it the volatility has the even, regular expansion (14) of the main text, whose leading term $v_0(\cdot) = 2\sqrt{2} \operatorname{erf}^{-1}(\cdot)$ is the exact at-the-money kernel and whose V_2, V_4 are elementary functions of s alone. Evenness in h is the put–call symmetry $h \mapsto -h$ of the underlying density, and it is what makes the expansion efficient, each retained order gaining a uniform factor h^2 . This was verified directly, by halving h at fixed v and observing the residual error of the one-, two- and three-term truncations divide by 4, 16 and 64. The leading term alone is uniformly second-order accurate through the deep-OTM corner (small c jointly with small h) that defeats naive tabulation. The frozen residual R and the Chebyshev replacements of V_2, V_4 and B^{-1} are recorded in Appendix D.1.

B.2 The tabulated chart: FAR

Over the interior of the domain the inversion is resolved by direct evaluation of a precomputed table, with no iteration at all. The rationale is a cost observation before it is an accuracy one. A machine-precision iterative inverter spends essentially all of its time in its accurate forward evaluations. A seed plus two Householder corrections costs two evaluations of a carefully written Black function, each dominated by error-function calls. A table evaluation replaces all of that by one polynomial sweep whose operation count is fixed, known at compile time, and free of data-dependent branches, which is also exactly the shape of computation that vectorizes. The questions that decide whether the approach reaches machine precision are which variable to tabulate, in which coordinate, and at what polynomial degree. All three answers come from the analytic structure of the inverse map.

A prior question is whether tabulation is needed at all, or whether some change of variable turns the interior inversion into a classical quantile that could simply be evaluated. It cannot. Read as a quantile identity, the Black–Scholes variance law meets the generalized inverse Gaussian family only at the orders $p = \pm \frac{1}{2}$, realized on the variance and on the inverse Gaussian side respectively, and the only monotone re-coordinatizations that preserve that family’s exponent are scaling, which moves the tilt parameters, and reciprocation, which flips the sign of the order [2, 12, 23]. Neither removes the inversion. Scaling leaves it in place, and reciprocation exchanges the two orders, which is the $c \leftrightarrow 1 - c$ pairing the two endpoint charts already exploit. Representations outside this elementary orbit require a genuine change of variable, carrying special functions of its own. Tabulating the interior is therefore forced rather than merely convenient, and the three questions above are the right ones to ask of it.

The approximation target is the transformed variable W of (3) rather than σ or w themselves. W is the exponential scale variable of the wing asymptotics, it is nearly linear in $\log c$ across each cell, and fitting σ directly was measured to cost one to two additional Chebyshev degrees per cell for no benefit. The abscissa is $\lambda = \log c$. The choice of λ rests on a design principle about where the inverse map ceases to be analytic, which we state and then say precisely how far we have verified it. Viewed as a function of the price c at fixed h , the inverse $w(h, \cdot)$ has branch points at the two endpoints $c = 0$ and $c = 1$, and the distances from an interior price c to these branch points are exactly c and $1 - c$, *independently of h* . The convergence of a Chebyshev approximation on a cell is governed by the Bernstein ellipse through the nearest *complex* singularity [18, 26]. At the time the table was built we had not proved that no complex singularity lies nearer than the two real endpoints, so we treated the endpoint distances as a predictive design principle and tested it against the fitted degrees of all 239 cells, where a nearer singularity would announce itself as degrees systematically above the prediction. That principle is now partly a theorem, as recorded at the end of this subsection. The observed degrees match the prediction throughout (Figure 4). In the raw mantissa (binade) coordinate the $c = 0$ branch point sits one octave to the left of every octave, in a self-similar way, which pins the Bernstein parameter at $\rho = 3 + 2\sqrt{2}$ and predicts a floor of roughly twenty-one degrees

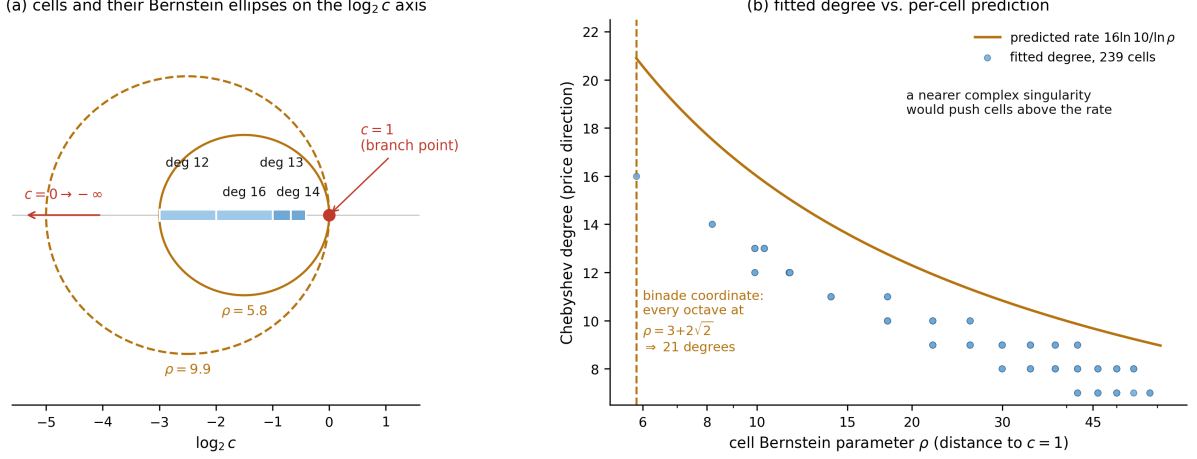


Figure 4: The branch-point geometry that sizes the tabulated chart. **(a)** The cells nearest the money in one h -band of the table on the $\log_2 c$ axis, with the Bernstein ellipses of two cells drawn in place. Each cell’s largest singularity-free ellipse passes through the $c = 1$ branch point, so ρ grows as cells recede from the money, and sub-splitting the octaves adjacent to the money (dark cells) keeps $\rho \geq 3 + 2\sqrt{2}$ for every cell. **(b)** The fitted price-direction degrees of all 239 cells against ρ , with the predicted rate $16 \ln 10 / \ln \rho$ overlaid. The degrees track the rate from below at a uniform offset (Appendix D.1), and no cell sits systematically above it, as a nearer complex singularity would force.

per octave everywhere (confirmed by fits), so a transcendental-free table in that coordinate is a dead end, elegant as it looks. A conformal coordinate adapted to the annulus fares even worse in practice (fitted degrees near one hundred), because it compresses the real interval against the boundary. In $\lambda = \log c$ the distance to the branch point grows with the logarithmic cell width, and the fitted degrees fall to a median of nine, never exceeding sixteen. These are measured degrees from the table generator, not estimates.

On each cell the table stores the bivariate Chebyshev surrogate (15) of the main text, with $x_h, x_\lambda \in [-1, 1]$ affine images of the local h -band and $\log c$ interval. The degrees, the cell geometry in numbers, the bit-field routing to a cell, the extent of the bands and the oracle-gated generation are collected in Appendix D.1, together with the logarithmic prefactor that explains the offset in Figure 4(b).

B.3 The deep wing: WING

For prices below the table’s reach the inversion works on the logarithm of the endpoint factorization, with the dominant asymptotic behaviour peeled off analytically. This is the extreme-strike regime whose leading-order asymptotics are classical (the moment formula [15]), but the chart must recover the map itself to full accuracy, not its leading order. Taking logarithms of $c = \gamma(h)e^{-W}W^{-3/2}H(W)$ gives the implicit equation (16) of the main text. Because it lives on the logarithm of the price, its conditioning is the benign $1/(2W)$ of Appendix A, and the inverter is validated on prices down to 10^{-320} , across the entire subnormal range of double precision. The shared logarithm kernel of Appendix C is extended by an exact power-of-two rescaling so that subnormal prices lose no accuracy on entry.

What makes the chart vectorizable is the evaluation of $\log H$. A runtime quadrature, the natural first implementation, has neither a fixed operation count in the useful sense (forty exponentials per evaluation) nor competitive speed, so it is retired to the offline oracle role and replaced by a frozen two-piece bivariate Chebyshev fit. The fitted quantity is deliberately not

$\log H$ itself but the peeled, weighted residual

$$S(u, \beta) = \frac{\log H}{u} + \left(\beta + \frac{3}{2}\right), \quad u = \frac{1}{W}. \quad (21)$$

Peeling the leading asymptotic term $-(\beta + 3/2)/W$ removes the dominant variation, and dividing by u makes the fit's equioscillating error automatically proportional to the σ -tolerance, which grows like W . Fitting the raw residual instead was measured to demand degrees beyond thirty-six where the weighted form needs twenty-six. Two pieces with a generous overlap cover $W \in [2.5, 260]$ (668 coefficients, 5.2 KB), the eight-term asymptotic series takes over beyond, and the piece-or-series regime is selected *once* per quote from $\tilde{L}$, so no per-step branching enters the kernel. The derivative $\partial_W \log H$ that drives the Newton steps is computed from the *same* coefficients by the Chebyshev derivative recurrence, so the value and slope seen by the iteration are consistent by construction. The solve is a fixed count of three Newton steps from a tabulated seed. Started from the cheap logarithmic seed $W_0 = \max(\tilde{L} - \frac{3}{2} \log \tilde{L}, 2.6)$ alone, the iteration needs six steps, and that count is delicate. Five passed a sparse de-risk grid, yet a dense sweep exposed them as one step short at the high- β , low- W corner, where convergence is slowest (Appendix D.2 records the incident). Seeding instead with W_0 plus a small peeled-residual correction, one bivariate Chebyshev surface of degree (8, 8) in $(\tilde{L}, \beta)$ per piece (81 coefficients each, 1.3 KB in total), starts the iteration inside the quadratic basin and cuts the fixed count from six to three, halving the chart's dominant cost. A direct table for W itself, dispensing with Newton entirely, fails on degree, because the small- W edge is genuinely asymptotic, but as a *seed* the low-degree surface is exactly enough. The reduction is gated the way the step-count incident taught. Against the oracle over 6,325 points spanning the full service domain, including the slowest-converging corner and the series regime, the three seeded steps reach a worst volatility error of 8.0×10^{-16} , marginally better than the six bare-seeded steps (8.4×10^{-16}), with no point above 10^{-15} . The chart shares the fixed-operation structure of the other charts, so the batch drivers evaluate it eight lanes at a time like the rest.

One boundary of the fitted table must be handled rather than clamped. The fit covers $\beta = h^2/16 \leq 16.4$, i.e. $h \leq 16.2$, corresponding to strikes more than 10^7 times the forward. Benchmark grids nevertheless reach past it, and a clamped evaluation there would return a plausible-looking volatility wrong in the third digit, the worst kind of failure. Quotes beyond the fitted β therefore take a rare scalar branch. A seed obtained by dropping the second Black term and solving $\Phi(v/2 - h/v) = c$ in closed form (then restoring the dropped term by three fixed-point passes, worst seed error 0.23% over $h \in [16.2, 30]$), finished by the same three Householder steps as the warm-start entry. The branch is validated for $W \lesssim 40$. The still-deeper corner, unreachable by any market data, returns a quiet NaN with an explicit out-of-domain status rather than a number (Section C). Both the scalar router and the batch drivers call the identical branch function, so the bit-identity invariant is preserved structurally.

B.4 The large-volatility chart: UPPER

At the upper price endpoint, $v > 1.85$ and $c \rightarrow 1$, the complement of (2) is

$$1 - c = k e^{h/2} \int_v^\infty e^{-t^2/8} e^{-h^2/(2t^2)} dt, \quad (22)$$

and the structural point that makes this endpoint fundamentally easier than the left one is that the layer factor is *regular* there, with $e^{-h^2/(2t^2)} \rightarrow 1$ as $t \rightarrow \infty$, with an expansion in h^2/t^2 that converges absolutely for every $t > 0$. There is no layer to match, and termwise integration (justified by dominated convergence) yields an exact, absolutely convergent series rather than an asymptotic one. Introducing the ladder $B_n(A) = \int_A^\infty \varphi(u) u^{-2n} du$, which satisfies the elementary recursion $B_{n+1}(A) = [\varphi(A)A^{-(2n+1)} - B_n(A)]/(2n+1)$ with $B_0(A) = \Phi(-A)$ and $B_1 = B$, one

obtains with $x = v/2$

$$\frac{(1-c)e^{-h/2}}{2} = \Phi(-x) + \sum_{n \geq 1} \frac{(-h^2/8)^n}{n!} B_n(x). \quad (23)$$

Every identity in this subsection was checked against 30-digit arithmetic across the strip, with exact agreement to working precision.

Two consequences follow. First, since $B'_n(x) = -\varphi(x)x^{-2n}$, the series for the derivative of the right-hand side of (23) resums in closed form,

$$g'(x) = -\varphi(x) e^{-h^2/(8x^2)}. \quad (24)$$

The derivative is cancellation-free and free of error functions. Because $\log |g'|$ is the elementary expression $\text{const} - x^2/2 - h^2/(8x^2)$, the higher derivative ratios needed by a third-order Householder step are closed-form rationals, $g''/g' = -x + h^2/(4x^3)$ and $g'''/g' = (g''/g')^2 - 1 - 3h^2/(4x^4)$, so an order-four step costs the same one price evaluation as a Newton step. This is the same economics that motivates Householder refinement in the reference solver [11].

Second, setting $h = 0$ in (23) gives $\Phi(-x) = \bar{g}$ with $\bar{g} = \frac{1}{2}(1-c)e^{-h/2}$. Define the endpoint coordinate

$$a_U = h - 2 \log(1-c), \quad \text{so that} \quad \bar{g} = \frac{1}{2}e^{-a_U/2} \quad \text{and} \quad x_0(a_U) = -\Phi^{-1}(\frac{1}{2}e^{-a_U/2}). \quad (25)$$

The function x_0 is the exact root on the line $h = 0$, and it depends on the quote through a_U alone. It is smooth in $\sqrt{a_U}$, and the chart stores it as a Chebyshev series of 12 coefficients in that variable with relative error 2.3×10^{-8} . No quantile kernel and no Mills ratio are evaluated at runtime. At fixed a_U the feasible moneyness is $0 \leq h < a_U$, with $h \rightarrow a_U$ as $v \rightarrow 0$, so $q_U = h/a_U \in [0, 1)$ is a bounded coordinate and the endpoint $q_U = 1$ is the nearest obstruction. The map

$$\zeta = \left(\frac{q_U}{1 + \sqrt{1 - q_U^2}} \right)^2 \quad (26)$$

sends the q_U^2 -plane cut along $[1, \infty)$ to the unit disk and keeps the chart at small ζ . The seed is

$$x_{\text{seed}} = x_0(1 + \zeta R(\sigma_U, \zeta)), \quad \sigma_U = \frac{x_0^2}{a_U}, \quad R = \sum_{j=0}^3 \sigma_U^j r_j(\zeta), \quad (27)$$

with polynomial rows r_j of degrees 5, 3, 1 and 0, thirteen coefficients in all, fitted in the uniform norm by a linear programme on top of the frozen series for x_0 . The factor ζ enforces exactness on $h = 0$. A scan of the compiled seed over 5.2 million points, a dense grid with both seams and the limit $h \rightarrow 0$ oversampled, gives a worst relative error of 4.9×10^{-5} .

One Householder step of order three then finishes the chart. Writing δ for the relative error of the seed, the step leaves the relative error $K_4\delta^4 + O(\delta^5)$ with

$$K_4 = W + \frac{W^2}{2} - \frac{Wx^2}{6} - \frac{x^4}{24}, \quad W = \frac{h^2}{2v^2}, \quad (28)$$

and $|K_4| \leq 217/18$ on the chart ($W \leq 3.8$, $x \leq 4$), the maximum being attained at $x = 4$ and $W = 5/3$. A seed within 9.5×10^{-5} therefore reaches 10^{-15} in one step, and the measured seed leaves a truncation error below 6×10^{-17} , a factor of two in the seed and a factor of sixteen in the result inside the threshold. The accuracy of the chart is then set by the rounding of the residual and not by the iteration. A three-term truncation of the ladder (23) is a much weaker seed, with relative error up to 0.27 near the wing seam, and needs exactly three Householder steps. The present seed removes two of the three residual evaluations together with the quantile and Mills-ratio kernels.

The residual is evaluated with one exponential. With $y = x - h/(2x)$ and $r = x + h/(2x)$ one has $r^2 - y^2 = 2h$, hence $e^h \varphi(r) = \varphi(y)$, and with $p = e^{-y^2/2}$

$$C = \frac{p}{2} \left[\operatorname{erfcx}\left(\frac{-y}{\sqrt{2}}\right) - \operatorname{erfcx}\left(\frac{r}{\sqrt{2}}\right) \right] \quad (c < \tfrac{1}{2}), \quad 1 - C = \frac{p}{2} \left[\operatorname{erfcx}\left(\frac{y}{\sqrt{2}}\right) + \operatorname{erfcx}\left(\frac{r}{\sqrt{2}}\right) \right] \quad (c \geq \tfrac{1}{2}), \quad (29)$$

while $\partial C/\partial x = 2\varphi(y) = \sqrt{2/\pi} p$ reuses the same p , so the Newton quotient needs no further transcendental. The square y^2 is formed with a compensated product. On the chart the negative erfcx argument of the direct branch stays in $[-0.40, 0]$ and is served by one additional frozen piece of degree 14. The step count is fixed and there is no early exit, which keeps the vector path lane-uniform. Prices above one half are handled throughout in the complement form. The subtraction $1 - c$ is exact by Sterbenz's lemma, and the point of the complement form is that no subsequent formula amplifies that rounded complement. This is why the chart retains full precision as $c \rightarrow 1$ where a $\sqrt{FK}$ -normalized formulation measurably does not (Section D.3). On a 2,465-point 40-digit oracle set concentrated on the wing seam, the $v = 1.85$ boundary, the h cap, $h \rightarrow 0$, $v \rightarrow 8$ and $c = \frac{1}{2}$, the worst error is 1.62 ULP with no point above 10^{-15} , against 2.15 ULP for the three-step ladder construction on the same points.

B.5 One family, analytically bounded, and derived seams

A single identity ties the charts together. With

$$H(W; a, \beta) = \int_0^\infty e^{-r} \left(1 + \frac{r}{W}\right)^{-a} e^{-\beta/(W+r)} dr, \quad (30)$$

the substitution $u^2 = A^2 + 2r$ gives, exactly,

$$B_n(A) = \varphi(A) A^{-(2n+1)} H\left(\frac{A^2}{2}; n + \frac{1}{2}, 0\right), \quad (31)$$

and the same family evaluated at $(a, \beta) = (3/2, h^2/16)$ and $(1/2, h^2/16)$ furnishes the two deep wings. Every special function appearing anywhere in the architecture is therefore one two-parameter family, and only two genuinely one-dimensional, h -independent inverses (erf^{-1} and B^{-1}) ever appear. In matched coordinates, all the h -dependence is a smooth prefactor times a regular even perturbation. For $\beta = 0$ the family is Stieltjes,

$$H(W; a, 0) = \frac{1}{\Gamma(a)} \int_0^\infty \frac{t^{a-1} e^{-t}}{1 + t/W} dt, \quad (32)$$

so its continued-fraction convergents bound it from alternating sides [1], and B , every B_n , and by monotone inversion B^{-1} admit rigorous two-sided enclosures at any order (the one place the word *certified* applies in its strict sense), and these enclosures supply the truth values against which the endpoint kernels were gated. For $\beta > 0$ the Stieltjes property fails, and the mechanism is an identity rather than a numerical observation. The same Stieltjes kernel represents the family with a *signed* density,

$$H\left(W; \frac{3}{2}, \beta\right) = \frac{1}{\Gamma(3/2)} \int_0^\infty t^{1/2} e^{-t} \frac{\sin(2\sqrt{Xt})}{2\sqrt{Xt}} \frac{dt}{1 + t/W}, \quad X = \frac{\beta}{W} = \frac{w}{8}, \quad (33)$$

because the inverse Laplace transform of $z^{-3/2} e^{-X/z}$ is $\sin(2\sqrt{Xt})/\sqrt{\pi X}$. The oscillating factor is the sign change, and it degenerates to unity at $\beta = 0$. One consequence closes the circle with the forward evaluation discussed at the end of this section. Writing $a = -d_1/\sqrt{2}$ and $b = -d_2/\sqrt{2}$, the two arguments the shared price kernel already forms, and using the exact relation $b^2 = a^2 + h$,

$$H\left(W; \frac{3}{2}, \frac{h^2}{16}\right) = \sqrt{\pi} W e^{-X} \frac{\operatorname{erfcx}(a) - \operatorname{erfcx}(b)}{b - a}, \quad (34)$$

so the wing member of the family is a divided difference of the very `erfcx` the refiners evaluate. Both identities were checked against adaptive quadrature to forty digits. The exact identity $H(W; a, \beta) = H(W; a, 0) - \int_0^\infty \sqrt{\beta/\tau} J_1(2\sqrt{\beta\tau}) e^{-W\tau} H(W(1+\tau); a, 0)/(1+\tau) d\tau$, obtained by writing $1 - e^{-\beta s}$ as the classical J_1 Laplace transform [21, Ch. 10] and applying it under the integral sign, expresses $H(\cdot; a, \beta)$ through the $\beta = 0$ family (we exercise it numerically to 10^{-21} in the generation pipeline), so two-sided enclosures extend to the wing evaluators as well.

The seams between the charts are derived, not tuned, and they are more than that, since each is a level set of the exact analyticity geometry of the inverse. In the intrinsic coordinates (θ, a_ρ) of Section A, with a_ρ the radius (5), both shipped seams are coordinate lines. The ceiling is the curve $\theta = \theta^*$ with $\theta^* = \text{erf}(1.85/(2\sqrt{2})) = 0.645034092553535$ exactly, this being the $k \rightarrow 0$ limit of the $v = 1.85$ contour, and freezing θ^* across the whole small-moneyness band moves the seam only within $v \in [1.85000, 1.85178]$. The wing seam is the line $\rho = B(\sqrt{7.6})$, which is simultaneously a level set of a_ρ and of $A = B^{-1}(\rho)$ because all three are functions of ρ alone, and freezing it moves the seam only within $W \in [3.79625, 3.8]$. Both ranges are extrema located by optimization at fifty digits rather than grid samples. That the seams are this flat in the intrinsic coordinates while the corresponding prices sweep eight decades is precisely why their price-space description needs polynomials at all. Independently of the geometry, the truncation error envelopes of the two endpoint expansions follow the scaling laws (asymptotically motivated, confirmed by high-precision measurement of the envelopes) $E_{\text{near}} \propto v^2 h^{2N+2}$ and $E_{\text{upper}} \propto (h^2/2v^3)^{N+1}$, and equating envelopes across the wide overlap bands, where two charts are simultaneously accurate, yields thresholds that are constants in the matched coordinates, in agreement with the geometric placement. The ceiling sits at 1.85 rather than 2 so that the exact-equation Householder refinement of Section B.4, rather than an extrapolated finisher, owns the corner where the charts meet. The large-volatility chart is machine-precise from $v = 1.55$ upward, whereas the small-moneyness chart’s matched expansion begins to extrapolate as $v \rightarrow 2$, and 1.85 is the highest level at which both tabulated charts remain validated. Freezing the seams as polynomials matters beyond tidiness. Routing then costs two short polynomial sweeps rather than any runtime price evaluation, which is what allows the classification itself to vectorize (Section C). A fitted seam differs from the exact boundary only within its fit error ($\sim 10^{-16}$ relative), reclassifying at most quotes within a ulp of the seam—inside the overlap bands, where both neighbouring charts are machine-precise, so nothing delicate happens at a crossing. The dense seam sampling reported in Section D.3 confirms it.

The resulting coverage map is this. The deep wing owns $W \gtrsim 3.8$, including the feasibility pinch at very large h , which is a wing region in disguise since the feasible price there is necessarily small. The small-moneyness chart owns $h < 0.3$ up to the $v = 1.85$ ceiling. The large-volatility chart owns everything above its seam, including the whole strip $h > H_{\text{box}} \approx 6.65$ beyond the last table band (Appendix D.1) that is not wing. The tabulated chart owns the remainder. The four regimes tile the domain exactly, and on a broad random feed no feasible quote fails to route.

The accurate forward. As in the reference solver [11], the precision floor of the whole method is the accuracy of the forward evaluation used inside the places that refine, the large-volatility Householder finisher and the warm-start entry of Section C. We evaluate the normal functions there through a scaled complementary error function `erfcx` built on frozen Chebyshev pieces in the spirit of [4], three degree-22 pieces covering $[0, 7.2]$, selected per lane by blends off the critical path, which halves the sequential Clenshaw chain relative to a single degree-36 fit and extends the domain to the arguments the warm-start basin requires, and an exponential reduced by a Cody–Waite splitting of $\log 2$ followed by a Chebyshev polynomial on $[-\frac{1}{2} \log 2, \frac{1}{2} \log 2]$ and an exact power-of-two scaling. Both Black terms of a price residual are produced by one shared kernel that evaluates the $\pm$ pair from a single `erfcx` core and one shared normal tail, with a compensated square killing the rounding of the squared argument, so each warm-start refinement step costs two `erfcx` and four exponential evaluations rather than the naive six, and

the single step of the large-volatility chart costs two `erfcx` and one exponential through (29). The refiners therefore perform no library `erf`, `erfc` or `exp` call, and their rounding behaviour is fully determined by the frozen coefficients.

C Implementation and deterministic arithmetic

C.1 Shared kernels, routing and vector execution

Every kernel is a branchless chain of explicitly ordered fused multiply-adds shared verbatim by the scalar and SIMD paths, so that “batch equals scalar” is the single statement that validates every vector driver (Section 3.2 of the main text). This appendix records the shared kernels, the drivers, the build discipline behind the invariant and the complete hot path (Appendix C.3). The offline generation is in Appendix D.2.

Three kernels are shared. A branchless logarithm proxy evaluates $\log_2 m$ for a mantissa $m \in [1, 2)$ through the reduction $t = (m - 1)/(m + 1)$ and a Chebyshev series in t^2 , with absolute error a few units in the last place, comparable to the library function it replaces. A full natural logarithm, needed by the universal inverse, the quantile seed and the wing, is then assembled by reuse as $\log x = \log 2 (e + \log_2 m)$ from the exponent e and mantissa of x , introducing no new approximation and agreeing with the library logarithm to three units in the last place over the domains that consume it. A subnormal-safe variant rescales arguments below the normal range by an exact power of two before the bit extraction, so the wing’s prices near 10^{-320} enter with full accuracy. A degree-seventeen branchless kernel evaluates the at-the-money base $v_0 = 2\sqrt{2} \operatorname{erf}^{-1}$. And a bivariate Chebyshev evaluator computes (15) by an inner accumulation in x_λ followed by an outer Clenshaw sweep in x_h . One design lesson from gating these kernels deserves record, because it is counterintuitive. For a kernel that feeds a downstream polynomial fit, the smoothness of its error function matters as much as its magnitude. An early candidate for the logarithm kernel, a plain minimax polynomial of the *same* worst-case accuracy as the reduction above, raised the Chebyshev degrees demanded by the table cells by eleven, because its equioscillating error injected a rough component into the quantity being fitted. The smooth-error reduction adds no degree at all. Kernel acceptance was therefore gated on the downstream fitted degrees, not only on the kernel’s own error.

Two more kernels serve the intrinsic chart of Section A.3, and both were designed for the vector path. Its coordinate $a = \log(1 + q)$, $q = E/c$, needs `log1p` to well under a unit in the last place because the chart’s sensitivity to a is 1.34. The classical route, a logarithm of the rounded $1 + q$ plus the Sterbenz correction $(q - d)/u$, costs a division and, with the library-grade `log` we first tried, 1.19 ULP against 80-bit truth. Reducing $1 + q$ into $[\frac{3}{4}2^k, \frac{3}{2}2^k)$ instead lets the reduced argument $f = (q - (2^k - 1))2^{-k}$ be formed from q itself, exactly by Sterbenz’s lemma for every $k \geq 1$ and trivially for $k = 0$, so the correction and its division disappear. The wider window needs an eight-term fit of the classical odd-argument polynomial, and the kernel measures 0.87 ULP worst against 80-bit truth over four million points, below the library’s own 0.81-ULP worst by a comparable margin. The whole-book kernel also needs `expm1` to a unit in the last place up to $h = 16.5$, which the small-moneyness Taylor kernel does not provide. Writing $h = k \log 2 + r$ with $k = \lfloor h / \log 2 \rfloor$ and $r \in [0, \log 2)$, the Cody–Waite split is exact, a twenty-term Taylor series of `expm1(r)/r` has only positive terms, and $E = 2^k \operatorname{expm1}(r) + (2^k - 1)$ adds two non-negative numbers, so the kernel is division-free and measures 1.00 ULP worst against the library over the whole range. Each kernel has an eight-wide AVX-512 twin and a four-wide AVX2 twin that reproduce the scalar fused-multiply order lane for lane. Masked selections use blends whose argument order is mirrored from the scalar ternaries, and per-lane routing decisions are the same integer tests as the scalar router. The only intrinsic without a direct AVX2 analogue is the exponent scaling `scalef` used by the shared exponential. We reproduce it, including its behaviour in the subnormal underflow range, by an exact two-factor

power-of-two construction (splitting the integer exponent so that each factor remains a normal number), so the AVX2 path matches both the scalar `ldexp` and the AVX-512 `scalef` across the whole finite range. The equivalence was verified bit-for-bit over seven hundred thousand arguments down to the underflow threshold.

The batch driver is shaped by the route mix of a real book, about 90% small-moneyness quotes on the out-of-the-money branch (Appendix E). As the main text describes, it streams the (h, c) vector eight (or four) lanes at a time, classifies every lane, evaluates the small-moneyness kernel on every lane speculatively, masked-stores the lanes that belong there in input order, and compresses the minority into a deferred buffer for a counting-sort-by-cell pass over the remaining charts. Making the classification vectorizable required removing the last library calls from the router. The two seam prices consumed on the hot path, $C(h, 1.85)$ and $\text{expm1}(h)$, are replaced by frozen low-degree Chebyshev fits, and the wing seam and the ceiling are moved onto the shared exp kernel. The scalar entry adopts the identical fits, so the classification is a handful of branchless polynomials that the scalar and SIMD routers evaluate to the same bits. Speculation costs the wasted small-moneyness evaluations on the deferred minority, far less than the sorting it removes on the majority, and a strided sample of the batch’s moneyness picks the speculative driver only when the small-moneyness share is actually high, falling back otherwise to the plain counting-sort driver. Both return bit-identical results, so the dispatch changes only speed.

One further property of the speculative driver is worth recording, because it is invisible in an operation count and was measured only after the fact. Vectorizing eight quotes per register removes the routing branch but not the *serial* structure inside each chart. The small-moneyness path is a sequence of dependent Chebyshev recurrences (two seam polynomials, expm1 , the universal inverse’s logarithm and regime fit, the at-the-money kernel, V_2 , V_4), each a fused-multiply-add ladder whose latency one eight-lane group cannot hide. The machine is therefore idle at the arithmetic ports while the register file still has room, since the small-moneyness kernel occupies twenty of the thirty-two vector registers with no spill. Running several *independent* eight-lane groups through those prefix chains at once fills the gap. The subtlety is that this must be done to the prefix only. Interleaving the whole kernel, including the finisher whose bivariate Clenshaw holds a seventeen-term Chebyshev basis, exceeds the register file and buys nothing measurable (a percent, inside the noise). Interleaving the prefix while the finisher stays one group at a time is worth about twenty percent on the market-realistic feed. Nothing in the per-quote arithmetic changes—only mutually independent groups are co-scheduled—so the bit-identity invariant is preserved by construction and was verified unchanged. The four-wide instruction set is deliberately excluded. With sixteen registers it already spills at one group, so the same restructuring there would only add stack traffic.

The book kernel’s batch driver meets the same obstacle in a different place and removes it differently. Its chain per register runs through expm1 , a division, $\log 1p$, the one-dimensional table and the rows, several hundred dependent operations, which is longer than the processor’s reorder window, so a single loop over that chain cannot overlap consecutive registers. Timed separately, the pieces of the loop sum to roughly seventy percent of the single-loop time. Running several registers in lockstep, the remedy above, does not help here. The raw-row kernel alone is not chain-bound and becomes slower under lockstep evaluation, and only the conformal kernel, whose 40-coefficient table recurrence precedes its rows, gains about a quarter. The driver therefore splits each tile of 4096 quotes into passes with short bodies. The first pass forms a for every register and stores it. The second pass evaluates a register in place when all of its lanes lie in the raw-row region and are accepted, and appends every other accepted lane to one of two per-tile lists with a compressing store. The raw-row list is then evaluated as contiguous registers, and the conformal list in two further passes, the table recurrence for $1/A_0$ first and the rows second. Lanes that the reach tests decline are not evaluated by the kernel at all, because the driver repeats the scalar reach test in vector form with the same multiply, fused multiply-add and comparisons. Every lane still executes the scalar operation sequence, so the bit-identity invariant

holds by construction and was verified unchanged on 393,806 quotes on both instruction sets. On the quiet host the split lowers the AVX-512 feed figure from 20.7 to 16.6 ns per quote and the AVX2 figure from 33.3 to 23.2, and the conformal tiles gain the most, the wing tile falling from 29.7 to 20.7.

Declined quotes are routed without the scalar router. The two seam polynomials are evaluated for whole registers by their vector twins, lanes above the ceiling and not below the wing seam are compacted into a list, and that list is evaluated in two passes, the seed (27) first and the Householder step second. All other lanes take the routed passes described above. For single-quote calls the same decision is made in the book coordinate. Since $c = E/(e^a - 1)$, the test $c > c_{\text{top}}(h)$ is $a < a_{\text{top}}(h)$ with $a_{\text{top}}(h) = \log(1 + E/c_{\text{top}}(h))$, and $c < c_w(h)$ is $a > a_w(h)$ likewise. Both are smooth functions of h alone and are stored per moneyiness band as Chebyshev fits of 6 to 14 coefficients together with a margin of four times the worst fit error. Outside the margin the decision is certain and agrees with the exact seam comparison, and inside it the exact seam polynomials are evaluated, so the routing decision is unchanged on every quote. A gate of 806,000 quotes placed on both seams and on the band edges at -2 to $+2$ units in the last place shows no contradiction and no difference between the batch and scalar results. On the synthetic UPPER tile this removes two seam polynomials of degree 18 and 32 and two exponentials from each single-quote call, and the routed and book scalar entries fall from 230 and 268 to 138 and 161 ns per quote against 187 for the reference.

Three further drivers complete the set. A fixed-moneyiness batch driver for the case in which all quotes share one h (so the h -derived constants are hoisted out of the loop entirely) is the throughput ceiling. Note that this is a fixed-moneyiness batch, not an expiry slice, since at one maturity h varies across strikes. A permuted-output variant returns results in bucket order with the permutation, saving the scattered store when the caller can consume sorted output. And a warm-start driver for streaming recalibration takes the previous variance of each (K, T) node as its seed, skips routing entirely, and applies three fixed Householder steps on the exact equation. Within its basin (previous $W \leq 40$, volatility moving up to a few percent) it returns the new root to within a few ULP rather than the last bit, because the half-ULP rounding of its working variables is amplified by the deep-wing cancellation ratio, so a caller needing full accuracy re-inverts cold—a documented API contract. For the live streaming regime, where snapshot-to-snapshot moves are sub-percent, a measured two-step variant reaches the same warm floor as three steps (verified for $|\Delta v/v| \leq 1\%$ at every $W \leq 40$. It degrades only from 2% moves at deep W) and is guarded by the size of its own last correction, so a move outside the two-step basin flags itself and falls back to a cold inversion rather than returning a degraded result silently. A thin book-mode wrapper packages the pattern. Exchange strike grids are standardized, so a live book’s (K, T) node set is persistent, the caller carries only static log-strikes and previous variances, and re-deriving h under a new forward is one subtraction per node before the whole book warm-refines with no routing at all. In every driver the deep-wing bucket is split by the wing’s three evaluation regimes so each sub-bucket is lane-uniform, and the wing kernel has both an AVX-512 and an AVX2 twin. Two routes remain scalar by design. The exact at-the-money line $h = 0$ and the rare large- β rescue branch of Section B.3. Both drop to the same non-inlined scalar entry the reference path uses, so the bit-identity statement covers them trivially.

The correctness of the vectorization is checked adversarially rather than assumed. Perturbing a single constant inside the vector kernels alone, by one part in 2^{40} , makes the batch result diverge from the scalar result on precisely the quotes routed through those kernels, while the unperturbed build shows zero divergences. This confirms the vector path genuinely executes rather than silently falling back to the scalar kernels and passing the identity check vacuously. The same check was run independently for the AVX-512 and AVX2 builds.

The invariant constrains the build, not only the source. Under GCC’s default contraction mode (`-ffp-contract=fast`) the compiler may fuse any multiply-add pair into one fused operation with one rounding [8, 10, 20], and it exercises that freedom during middle-end

optimization, after inlining, so which pairs fuse depends on the inlining context of the translation unit. We observed the same header, compiled at identical flags into two translation units of one program, produce two floating-point functions that disagree by one to five ULP on about seven percent of deep-wing quotes. A non-inlined entry point compiled to 4,427 instructions in the benchmark unit and 2,932 in a test unit, from identical source. The robust discipline is contraction off with every intended fusion an explicit `fma`. Since the accuracy-critical chains were written that way from the start, the flag costs nothing measurable and upgrades determinism from a per-build property to a portable one. Bitwise-reproducible floating point has mostly been pursued for reductions, where reassociation is the hazard [6], and the discipline here targets the complementary hazard, fusion. On every validation feed of Appendix D.3 the AVX-512, AVX2 and non-SIMD builds, compiled with either GCC 11 or Clang 14, return bitwise-identical doubles for every quote, scalar and batch paths alike, verified by byte comparison of the full output vectors.

C.2 Input and status contract

A reference implementation is defined as much by what it does with bad inputs as by its accuracy on good ones, because a live feed delivers crossed quotes, stale prices and convention violations as a matter of course, and the worst possible response is a plausible-looking number. The entry points therefore implement a *total* contract. Every representable pair (h, c) returns either a machine-precision result or a quiet NaN, and a checked variant reports a machine-readable status alongside, at or below intrinsic ($c \leq 0$), at or above the forward bound ($c \geq 1$), convention violation ($h < 0$ or NaN input), the out-of-domain deep corner of Section B.3, and a near-saturation advisory for $v > 8$, where the $1 - c$ representability bound of Section A limits what any double-precision inverter can resolve. In that last case the best-effort result is still returned. A convenience wrapper accepts the industry-convention inputs (forward, strike, undiscounted price, expiry, call/put flag), performs the in-the-money to out-of-the-money parity flip internally, and returns the annualized volatility with the same status, which removes the most common integration error, a wrong normalization, from the caller’s responsibility.

C.3 Complete hot path

The complete scalar evaluation, which the batch drivers reproduce bucket-wise, is the following. All constants are frozen doubles. `bits(·)` denotes IEEE-754 bit extraction, and $\log_2^*$ is the shared logarithm kernel of Section C.

Routing (given $h \geq 0$ and $c \in (0, 1)$):

1. if $h = 0$: solve the exact ATM relation $c = \text{erf}(\sigma/2\sqrt{2})$ (quantile seed, two guarded corrections). Return.
2. if $c < c_w(h)$ (one Chebyshev in h): WING.
3. if $h < 0.3$: NEAR if $c \leq c_{\text{top}}(h)$ (a frozen Chebyshev fit of $C(h, 1.85)$, so the test is a polynomial, not a price evaluation), else UPPER.
4. if $h \leq H_{\text{box}}$ and $c \leq c_{\text{top}}(h)$: FAR. Else UPPER.

Both seam polynomials are evaluated for every quote, whichever chart it reaches, and the order of the tests above is immaterial to cost. Each tabulated chart is a *band*, bounded below by c_w and above by c_{top} , so establishing membership requires both boundaries however the branches are arranged, and the comparisons are negligible beside the two Chebyshev sweeps that produce them. This is deliberate rather than incidental: the vectorized classifier evaluates both seams on every lane and selects by mask, since a branch that skipped work for some lanes would break the lockstep on which bit-identity rests (Section C). The one place the empirical route mix is

exploited is the choice of batch driver, not the branch order—when a sampled probe finds the small-moneyness share high, the speculative driver evaluates NEAR on all lanes unconditionally and defers the minority.

NEAR: $\rho = c / \text{expm1}(h)$; $A = B^{-1}(\rho)$ (three-regime Chebyshev); $s = h/A$; $v = v_0(s/\sqrt{2\pi}) + h^2 V_2(s) + h^4 V_4(s) + h^6 R(s, h)$. Return $w = v^2$.

NEAR, book-kernel form (the entry point as shipped, Section 2 of the main text): $E = \text{expm1}(h)$; $a = \log 1p(E/c)$ (Sterbenz-exact reduction). For $a < 2\pi$ and $h \leq 0.35 a$: $A_0 = a G(a)$ (23-term Clenshaw), $t = h/A_0$, $x = t^2$, $y = A_0^2$, $v = t + t x \sum_{1 \leq m \leq 15} x^{m-1} P_m(y)$ (sixteen exact rows, the last product and sum fused). For $a \geq 2\pi$ and $h \leq \min(4, 0.16 a + 0.85)$: $1/A_0$ from the second cell as $u F_1(u)$ with $u = 1/\sqrt{a}$, and $v = t(1 + \sum_{n \leq 16} q_c^n d_n(1/A_0^2))$ with q_c of (13). Return $w = v^2$. Outside both: the routed charts.

FAR (table): band from $\text{bits}(h)$. Octave and sub-cell from $\text{bits}(c)$; $x_\lambda = \alpha_{\text{cell}} \log_2^*(m) + \beta_{\text{cell}}$ with m the mantissa of c ; W by the bivariate Clenshaw sweep (15). Return $w = h^2/(2W)$.

WING: if $h^2/16 > 16.4$: rescue branch (Φ -inversion seed, three fixed-point corrections, three Householder steps. Quiet NaN beyond its validated band). Otherwise $\tilde{L}$ from $\log c$ (subnormal-safe $\log_2^*$) and h . Regime (piece A / piece B / series) once from $\tilde{L}$. Seed $W_0 = \max(\tilde{L} - \frac{3}{2} \log \tilde{L}, 2.6)$, plus the degree-(8, 8) peeled-residual surface in $(\tilde{L}, \beta)$ for the fitted pieces (bare W_0 in the series regime). Exactly three Newton steps on $W = \tilde{L} - \frac{3}{2} \log W + \log H(W)$ with $\log H$ and its analytic derivative from the fitted $S(u, \beta)$ (or the series). Return $w = h^2/(2W)$.

UPPER: $a_U = h - 2 \log(1 - c)$; x_0 from its Chebyshev series in $\sqrt{a_U}$; ζ from (26) and $\sigma_U = x_0^2/a_U$; seed (27) with thirteen coefficients. Then exactly one Householder-3 step on the exact equation, with the residual from (29), the quotient f/f' from the shared p , and the closed-form ratios $g''/g' = -x + h^2/(4x^3)$, $g'''/g' = (g''/g')^2 - 1 - 3h^2/(4x^4)$, prices evaluated in whichever of c , $1 - c$ is small. Return $w = (2x)^2$.

Batch drivers: a strided moneyness sample selects one of two paths. *Speculative* (small-moneyness-heavy feeds, the live case): stream eight/four-wide. Classify each lane by the frozen seam polynomials. Evaluate the small-moneyness kernel on every lane. Masked-store the small-moneyness lanes in input order. Compress the rest into a deferred buffer. *Sort-then-batch* (the deferred buffer, and minority-dominated feeds): classify, counting-sort by cell, chart and wing regime, and evaluate each bucket eight/four-wide. Both are bit-identical to the scalar entry, so the dispatch changes only speed. Remainders, the rescue branch and the ATM line go through the scalar entry unchanged.

D Offline construction and extended numerical validation

D.1 Matched-chart and table engineering

The two charts of Sections B.1 and B.2 that the book kernel of Section A.3 replaces on the traded book remain the inverter’s coverage outside the kernel’s reach, and their engineering is recorded here in the form the runtime executes.

The matched small-moneyness chart. The chart is completed by two implementation decisions. First, the residual R is small enough to be frozen once as a single low-order bivariate Chebyshev surface (degree two in the h^2 direction, sixteen in s). It absorbs everything beyond h^4 and brings the chart to machine precision across its whole strip. Second, the two functions V_2 and V_4 , whose closed forms contain exponentials and a series crossover, are replaced by one-dimensional Chebyshev expansions in s of degrees 28 and 26. This removes the last library call and the last data-dependent branch from the chart’s inner loop, which matters for the vectorized path of Section C. The universal inverse B^{-1} itself is evaluated by a three-regime

branchless Chebyshev approximation, one regime in the variable $u = \kappa/(\rho + \frac{1}{2})$ for moderate arguments and two in $L = \log(1/\rho)$ for the tail, totalling 456 bytes of coefficients with worst error 5.4×10^{-16} . The chain $c \mapsto \rho \mapsto A \mapsto s$ has relative condition number at most one, so no accuracy is lost in the transforms.

The tabulated chart: degrees, cells and bytes. The surrogate (15) is sized as follows. The degree never exceeds sixteen in the price direction and twelve in the moneyness direction, with medians of nine and ten. The full table comprises nineteen h -bands over $h \in [0.30, 6.65]$ and 239 cells holding 27,211 double-precision coefficients, about 220 KB in total, small enough to be resident in a single core’s L2 cache, so that in steady state the table costs no memory traffic. Cache-cold and cache-warm evaluation are indistinguishable after the first few hundred quotes of a batch, so residency is realized in practice. Recovery from the surrogate is the single map $w = h^2/(2W)$.

Why the fitted degrees sit below the geometric rate. Two things sharpen that account. The first concerns the uniform offset by which the fitted degrees sit below the predicted geometric rate in Figure 4(b). It is the order-one prefactor of the asymptotic rate, and it is not unknown. When the dominant singularity is a power of a logarithm, which is the end type of this inverse at both price endpoints, the Chebyshev coefficients behave as $a_n \sim C \varrho^{-n} n^{-1} (\log n)^{q-1}$ with $q = \frac{1}{2}$ at an outer end and $q = \frac{3}{2}$ at an inner one. Over the degree range the table actually uses, $n \leq 16$, a straight-line fit of $\log |a_n|$ absorbs that algebraic and logarithmic prefactor as a roughly constant vertical shift, which is what the panel shows, and the degree rule is correspondingly the fixed point of $N \approx [\log(1/\varepsilon) - \log N - q \log \log N] / \log \varrho$ rather than a pure geometric rate. The second concerns the design principle itself, which moves partway from assumption to theorem. In the variable $L = -\log \rho$, an affine reflection of λ , the physical price branch at fixed k is single-valued and holomorphic on the half-strip $\Re L > \log(e^k - 1)$, $|\Im L| < \pi$, and the finite set of prices at which the branch can fail is not unstructured but the discrete lattice $\{L_0 + 2\pi i n\} \cup \{(2n+1)\pi i\} \cup \{\ell_0 + (2n+1)\pi i\}$, with $L_0 = \log(e^k - 1)$ and $\ell_0 = L_0 - k$. Inside the strip the nearest obstruction is therefore the price endpoint, exactly as assumed, and that settles the cells nearest the money, whose Bernstein ellipses stay within it and which are also the cells setting the worst case $\rho = 3 + 2\sqrt{2}$ drawn in panel (a). For the wider cells, whose ellipses have semi-minor axis exceeding π , what remains open is no longer where the obstructions are, since the lattice locates them, but whether the physical sheet continues past them. The fitted degrees indicate that it does.

Routing to a cell. Routing to a cell is performed by integer operations on the IEEE-754 bit fields of the inputs, before any transcendental evaluation. The high bits of h (exponent plus two leading mantissa bits) select the h -band. The exponent bits of c select a price octave within the band, and the mantissa of c , forced into $[1, 2)$, becomes the evaluation abscissa through the shared logarithm kernel of Section C, so that the per-quote classification involves no division, no logarithm and no comparison ladder. The octaves adjacent to $c = 1$, where the near-at-the-money geometry would otherwise demand a special analytic branch, are instead refined by one or two additional mantissa bits into sub-cells, so that they become ordinary table cells. This sub-splitting is what removes the last analytic bypass from the interior.

The extent of the bands. The bands deliberately stop at $H_{\text{box}} \approx 6.65$. Beyond it the feasible strip is owned entirely by the endpoint charts of Sections B.1, B.3 and B.4. For such large h every feasible quote is either deep in the wing (small price) or above the $v = 1.85$ ceiling (large volatility), so extending the table would add cells that no quote needs. In fact the handover happens earlier still. The two seams cross at $h = 5.10$, beyond which the ceiling lies below the wing seam and no price can reach the table at all. The same reasoning caps the tabulated

maturity range from below at $h = 0.3$, where the matched small-moneyness chart is uniformly accurate. The table is therefore exactly as large as the region in which neither endpoint structure simplifies, and no larger.

D.2 Offline generation and oracle gates

The offline pipeline that produces the frozen constants is as important to the accuracy claim as the runtime, so we record it.

1. *Truth generation.* For each cell, Chebyshev nodes in (x_h, x_λ) are mapped to exact double-precision prices, and each price is inverted by an arbitrary-precision solver (no fewer than 25 significant digits, price residual driven below 10^{-18}). The transformed truth $W = h^2/(2w)$ is recorded.
2. *Fitting.* Tensor Chebyshev coefficients are computed per cell. Degrees are the smallest passing the gate below, capped at 13 in the moneyness and 16 in the price direction (the shipped table reaches 12 and 16).
3. *Gating.* An independent 40-digit oracle, sharing no code with the generator, re-inverts a dense sample of each cell including its boundaries and the sub-cell seams. The cell is accepted only if the relative σ error is below 10^{-15} at every sample. Failing cells are refit at higher degree or split. The same gate validates every auxiliary kernel ($\log_2^*$, v_0 , B^{-1} , V_2 , V_4 , erfcx , the exponential, the residual R), with the additional downstream-degree criterion of Section C.
4. *Freezing.* Accepted coefficients are emitted as exact hexadecimal doubles into a header. The runtime contains no fitted constant that did not pass the gate.

Certified enclosures for the endpoint functions are available independently of the gate through the Stieltjes representation of Section B.5, whose continued-fraction convergents bound B , B_n and B^{-1} from alternating sides.

For the benefit of implementers we also record the design alternatives that the gate *rejected*, since each looks attractive on paper. A transcendental-free table in the raw mantissa coordinate (routing and abscissa both from bit fields, no logarithm at all) fails on degree. The h -independent branch-point distances of Section B.2 pin the Bernstein parameter and roughly double the degree per cell, blowing the byte budget. A conformal coordinate adapted to the annulus fails the same way, more severely. Evaluating the cells in the monomial basis with Estrin’s scheme, instead of Chebyshev with Clenshaw’s, fails the precision gate outright at the required degrees. An equal-accuracy minimax replacement for the logarithm kernel fails through the downstream-degree effect described in Section C. Fitting the wing’s $\log H$ directly, or its unweighted peeled residual, demands degrees past thirty-six where the σ -weighted form needs twenty-six, and a $(1/W, \beta/W)$ chart for the same fit, natural-looking because $\beta/W = v^2/8$ is bounded, brings no degree reduction because the hard structure at the small- W edge is genuinely asymptotic rather than an artifact of coordinates. Finally, one accepted choice had to be revised by a denser gate. Five wing Newton steps passed a sparse step-count de-risk yet left six points of the dense regular grid above 10^{-15} (worst 1.2×10^{-14}), all at the high- β , low- W corner where convergence is slowest. With the bare logarithmic seed the safe fixed count is six, and the incident is a reminder that step-count de-risking must sample the corner of slowest convergence, not the typical point. The tabulated seed of Section B.3 later cut the count to three, and that reduction was gated on a dense sweep of the same corner plus the series regime, per the lesson. Each rejection was measured, not argued.

D.3 Extended numerical validation

Oracle and scoring. The reference is the independent `mpmath` inverter of Section 4.1 of the main text, a bounded log-space Newton iteration on the given double-precision price at 40 significant digits (not `findroot`, not a round trip). The ULP error is the number of representable doubles between the returned value and the oracle value rounded to double, and that rounding is correct unless the exact value lies within 10^{-40} relative of a rounding boundary. The worst points of every campaign were recomputed at 60 digits and returned the same double in every case. LBR [11] is the latest public C++ implementation, downloaded unmodified (Appendix F), compiled at the flags of its own Makefile and called through its normalized entry point with $\beta = ce^{-h/2}$. We verified on the extreme cases that this call agrees with the library’s direct price interface, so no artefact of the calling convention enters the comparison.

Accuracy. Table 2 of the main text reports the worst and ULP errors on the regular grid. The root-mean-square errors on the same grid are 8.0×10^{-17} with the book kernel in front and 7.9×10^{-17} for the routed charts alone, a median at the level of a single rounding. The book kernel answers 425 grid points, and on the NEAR and FAR rows, where it answers 156 of 585 points, it lowers the worst error from 6.7×10^{-16} to 4.3×10^{-16} . On the stressed set no point of either of our paths exceeds 10^{-15} (worst 3.5×10^{-16} with the kernel in front, 6.9×10^{-16} for the routed charts) against a 3.4×10^{-14} upper-endpoint tail for the reference. Two remarks sharpen the reference comparison. Forming its normalized price as $c/e^{h/2}$ instead of $ce^{-h/2}$, a difference of one rounding in β , doubles its worst error on the same points to 1.1×10^{-13} (868 ULP), so every figure here forms β as $ce^{-h/2}$. In the deep wing its loss reflects input conditioning rather than iteration, because β is smaller than the forward-normalized c by $e^{-h/2}$ and reaches the subnormal range several hundred orders of magnitude before c does (Figure 5).

The book kernel’s switches are boundaries where a polynomial approximation could peak between sampled points, so they were sampled on purpose. The 20,000-point campaign of the main text concentrates 2,500 points each on both sides of $a = 2\pi$ at every scale of distance from 10^{-6} to 10^{-1} , on the $\theta = 0.35$ edge, within two percent of the conformal reach line, on the $h = 4$ cap, at $a \rightarrow 0$, at $a \rightarrow 700$ and at $h \rightarrow 10^{-6}$, with a uniform fill, and no class exceeds 5.3×10^{-16} . Set against the propagated input rounding of Appendix A, at most 1.3×10^{-16} on the box a traded book occupies, these are the whole error budget of a market inversion, whereas at the $v = 8$ edge of the grid the propagated floor of 10^{-13} dwarfs both. The seam crossings between charts were sampled densely on both sides with no error excursion at any of them. This was not automatic. Dense seam verification exposed two seam placements whose corners violated the gate, and both were corrected (one by lowering the ceiling shared by the two tabulated charts, one by extending the wing crossover), the intended operation of an oracle-gated design.

Edge cases. Table 5 isolates individual extreme inputs, with the exact double-precision price each method was given and the relative σ error each returned. The routed inverter is exact or within four ULP on every row, including prices in the subnormal range near 10^{-321} , the formerly excluded small- h deep-OTM corner, the near-intrinsic upper endpoint, and the feasibility pinch at $h = 16$. The reference solver’s two visible losses have different characters and should be read differently. Near the intrinsic value its error (10^{-14} scale) is algorithmic, the conditioning of its normalized objective discussed in Section A. In the subnormal wing its input $\beta = ce^{-h/2}$ has already lost significance before the algorithm starts, so the error there (10^{-6} to 10^{-4}) reflects the conditioning of its chosen normalization rather than its iteration. The routed inverter is immune because its normalization keeps the small price intact and its wing chart works on the logarithm of the price. At the two most extreme prices of the regular grid, below 10^{-320} , the β input underflows to exactly zero and the reference necessarily returns zero volatility, while the routed inverter still recovers σ to the correctly rounded last bit. Prices this small are admittedly of

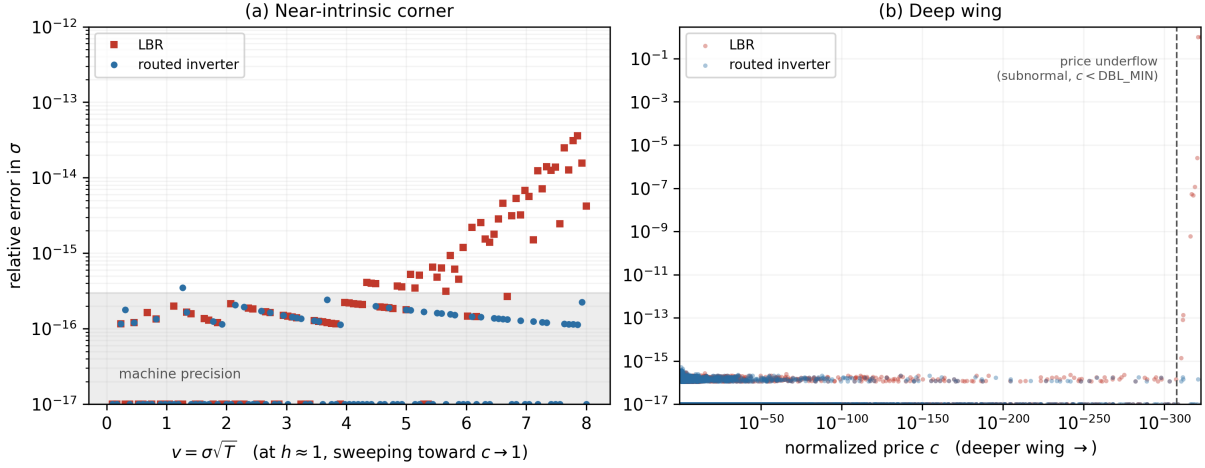


Figure 5: Relative error in σ in the two regimes where the reference degrades. **(a)** Near the intrinsic value, at $h \approx 1$ as $c \rightarrow 1$, LBR climbs to 3.6×10^{-14} from the conditioning of its normalized objective, while the routed inverter stays below 4×10^{-16} . **(b)** In the deep wing both hold machine precision across three hundred orders of magnitude of the price, until the reference’s reduced variable $\beta = c e^{-h/2}$ becomes subnormal and it loses precision, whereas the routed inverter, working in c , holds to the last representable price.

h	$\sigma\sqrt{T}$	price	regime	this method	LBR
14.8	0.386	$c = 9.9 \times 10^{-321}$	wing	0	1.4×10^{-4}
3.56	0.0932	$c = 2.2 \times 10^{-321}$	wing	0	3.9×10^{-6}
8	0.90	$c = 1.5 \times 10^{-18}$	wing	1.2×10^{-16}	0
0.29	0.02	$c = 9.6 \times 10^{-51}$	corner	0	0
10^{-4}	1.0	$c = 0.38$	left	0	0
0	6.4	$c = 0.9986$	ATM	5.6×10^{-16}	1.4×10^{-16}
0.5	7.9	$1 - c = 1.0 \times 10^{-4}$	right	1.1×10^{-16}	5.6×10^{-16}
1.2	7.93	$1 - c = 1.3 \times 10^{-4}$	right	1.1×10^{-16}	1.3×10^{-14}
16	7.5	$c = 0.93$	right	1.2×10^{-16}	1.2×10^{-16}

Table 5: Relative error in σ against the 40-digit oracle on hand-picked extreme inputs. An entry of 0 means the returned double equals the correctly rounded oracle value. The reference’s subnormal-wing losses reflect the conditioning of its $\sqrt{FK}$ normalization, and its near-intrinsic losses the conditioning of its objective near the upper price bound (text).

more diagnostic than trading interest, but they mark where each formulation’s validity actually ends.

Coverage and bit-identity. On a broad random feed of 120,000 quotes ($\sigma\sqrt{T}$ up to 8, h up to 16.5) every quote routes to a chart and returns a gated result, with no routing failures and no silent fallbacks. The continuum contract remains the graded one of Section A. Across the scalar, AVX2 and AVX-512 builds and all batch drivers (fixed- h batch, mixed- h grid, permuted-output, warm-start), the batch result equals the scalar result on every one of the 120,000 quotes, with zero mismatches. The same holds on the 16,039-point accuracy grid. Beyond within-build identity, the full output vectors of the three instruction-set builds under GCC and of the AVX-512 and AVX2 builds under Clang were compared byte for byte on the chart-stress feed. All five binaries return identical doubles for every quote, in both the scalar and the batched path. The accuracy of Table 2 is therefore simultaneously the accuracy of every batch path, every instruction set, and both compilers. The same identity extends to the CUDA port of Section E, which returns the CPU scalar entry’s double exactly on every quote of the market feed tiled to five million

and of four chart-pure surfaces, so the accuracy of this section is also the accuracy of the device. Before their device runs, the book kernel’s device functions compiled for the host reproduced the CPU result bit for bit on the 832,287 quotes of the 849,818-point price box that the kernel covers.

E Benchmark protocol and extended performance results

Timing protocol. The host, compiler and flags are those of Section 4.2 of the main text. The timing loop guards against dead-code elimination with a volatile accumulator. The figures of Table 7 come from the machine quiesced (mains power, no concurrent load, the process pinned to a fixed core), running each benchmark binary five times, each internal row itself a median over fifteen timing passes of eight repeats, and the table reports medians across the five executions, whose spread is a few percent. Loop unrolling is enabled because it duplicates loop bodies without altering any per-iteration floating-point operation, so it is bitwise-neutral (verified), and it recovers about ten percent by keeping the four highest-degree Chebyshev loops rolled. The floating-point environment is the IEEE-754 default, round-to-nearest-even, gradual underflow, hardware fma. Bit-identity is claimed for the tested configurations (two compilers by three instruction-set builds on this host), not for arbitrary hardware. Comparisons against the reference compile its four source files into the same binary as our headers, by the same compiler, at the flags of its own Makefile and the same instruction-set switch as the build under test, every method timed by the same loop over the same quotes in the same session, from the raw (h, c) pair. One asymmetry is inherent. Our kernels are header-only and inline into the timing loop, whereas the reference is called across a translation-unit boundary. A second build of every binary with link-time optimization on both sides moves the reference’s row by three percent and the ratios by less than it moves our own rows (Table 6), so the boundary is not what the tables measure. Two campaigns appear below. The earlier one (Table 7) compiled the reference at our flags, contraction off and assertions on, where it takes 227 ns per quote on the market feed. At its author’s release flags, the build of Table 6, it takes 194, and every ratio quoted outside Table 7 uses the release build. The GPU measurements were taken separately on one NVIDIA H100 PCIe under CUDA 12.4, in the CUDA development container of a cloud instance.

What a real book exercises. The tradeability screen of the main text leaves 641,072 of the year’s 1.85 million quotes. Forwards are the vendor’s own, matched to each quote by expiry and settlement convention (the file carries distinct AM- and PM-settlement forwards, differing by about half an index point), and discount factors come from the vendor zero curve. We do not rely on a parity regression, which we found can bias the implied forward by up to several tenths of a percent at individual expiries where deep in-the-money premiums dominate the fit. Every surviving quote is folded onto the out-of-the-money-call branch by put–call parity, exactly the reduction the inverter itself uses, and 15.8% of quotes are in-the-money and reach the branch through that projection. The wing, though barely one tradeable quote in eighty, is a permanent feature of this book. 97% of those quotes are downside puts, the crash-skew tail [9] firing the wing route on absolute cheapness, which is what motivated vectorizing it rather than leaving it scalar. The large-volatility chart never fires on end-of-day equity-index data and earns its place as coverage. The uniform broad box used for coverage stress is 70% wing, a distribution no traded book resembles, so throughput is reported on the 30,000-quote resample that reproduces the live route mix.

Throughput of the routed charts. Table 7 reports the earlier campaign for the routed inverter’s scalar entry and batch drivers on both instruction sets. The market-feed row is timed all-inside, every method starting from the raw (h, c) pair, so the reference pays its $\beta = c e^{-h/2}$ input transform and our scalar entry builds its per-quote context. The fixed-moneyness rows isolate the

kernels by hoisting the shared per- h constants, so they are chart-pure kernel benchmarks rather than expiry slices, since at one expiry h varies across strikes. On those batches the batch driver is $5.4\times$ faster than the reference on the small-moneyness chart, $9.9\times$ on the tabulated chart and, for the three-step ladder construction of the previous release, $2.8\times$ on the compute-heavy large-volatility endpoint (AVX2 3.3, 9.1 and $1.4\times$), the direct consequence of a fixed, branch-light operation count. On the market feed the speculative driver inverts the mixed book at 48 ns per quote against the reference’s 227 at our flags, a factor of 4.7 (86 against 225 on AVX2, a factor of 2.6), and against the release build of the later campaign the factors are 4.3 and 2.2. The deep wing is where the comparison has moved. Its evaluator carries more arithmetic than a Householder iteration, and on a wing-weighted grid the reference used to lead the batch driver by half again. Seeding the wing’s Newton solve from the tabulated surface of Appendix B.3 closed that gap. On the wing subset of the broad grid the vectorized driver inverts at 250 ns per quote against the reference’s 307, while the scalar entry remains far behind there (1,225 ns), the cost of a fixed-count kernel shared with its SIMD twin. Since only about one quote in eighty is a wing quote, neither figure moves the aggregate much.

One throughput property is invisible in medians but operationally significant. The cost per quote is *deterministic*. Every chart executes a fixed operation count (a fixed-degree polynomial sweep, or a fixed number of refinement steps, three seeded Newton in the wing, one Householder at the large-volatility endpoint), so the routed inverter has no iteration-count distribution and no data-dependent tail latency. The slowest quote of a batch costs the same as the median quote of its chart. The reference solver is also fixed-count (a seed and two Householder corrections), so its cost distribution is likewise narrow. The distinction here is not the absence of a tolerance loop but that the fixed count is executed by *branchless* kernels, which is what lets eight quotes advance in lockstep. Solvers that do iterate to a tolerance carry a tail, however short, and hedging pipelines that wait on the last quote of a batch inherit it.

The book kernel. Table 6 extends Table 3 of the main text with the small-batch, no-SIMD and link-time-optimized rows. Its cells are medians of 31 passes of eight sweeps. At 16 ns per quote a cell of seven passes of four sweeps lasts 13 ms, and a single disturbance then moves its median by several nanoseconds, which we observed as a bimodal feed figure until the cells were lengthened. At batches of 64 quotes, where the routed driver’s sort and drain do not amortize, the kernel’s advantage over the routed batch widens from 2.8 and 3.7 times on the full feed to 4.0 and 4.3, and its scalar entry takes 115 ns against the routed scalar entry’s 262. The routed charts’ scalar entry, whose routing and per-quote context were designed to amortize across a batch, is slower than the reference on isolated cold quotes, whereas the book kernel’s scalar entry, which routes nothing and iterates nowhere, is 1.7 times as fast as the reference on both vector-capable builds, so a caller inverting one quote at a time no longer gives anything up. On the build without SIMD and without hardware fused multiply-add every explicit fma of our kernels becomes a library call and the reference, which has none, is the faster scalar there. That row is the cost of the kernels on hardware the design does not target. On the small-moneyness quotes alone the eleven-row recurrence form of the kernel takes 16.6 ns on AVX-512 and 23.9 on AVX2 against the routed chart’s 23.4 and 62.4, both measured with the earlier single-pass driver. The shipped kernel’s sixteen rows and second table cell cost 15.6 on the same tile under the two-pass driver (Table 9), so covering the interior and the wing in the same straight line no longer costs anything on the small-moneyness quotes.

Streaming re-inversion. A live system rarely inverts cold twice. The persistent node set of Section C means every snapshot after the first starts from the previous one. On the market feed, seeding each node with its previous variance after a half-percent move, the guarded two-step warm batch completes in 0.76 times the cold speculative driver’s time on the same feed and host—37 ns per quote against the cold driver’s 48 and the reference’s 227. That $6.2\times$ margin

Build	LBR	PDE table method	Routed charts		Book kernel	
			scalar	batch	scalar	batch
AVX-512, full feed	193	78	262	46	115	16.6
AVX-512, batches of 64				75		18.9
AVX2, full feed	192	75	263	85	116	23.2
AVX2, batches of 64				109		25.1
no SIMD, full feed [§]	220	89	557	578	342	352
AVX-512, full feed, LTO on every side [§]	189	82	267	43	118	29
AVX2, full feed, LTO on every side [§]	188	85	269	83	119	42

Table 6: Nanoseconds per quote on the market feed (30,000 actual projected prices, every route), all methods in one binary, timed by the same loop in the same session on the quiet host, medians of 31 passes of eight sweeps with run-to-run spread of about one percent, every method starting from the raw (h, c) pair. The reference is compiled at the flags of its own Makefile, the PDE table method [19] at the flags of its compile script and timed single-threaded (its own convention uses every core), ours at the flags of Section E. The batch columns are the speculative driver of Section C and the kernel’s SIMD twin. On the no-SIMD build both degenerate to their scalar entries, and that build also lacks hardware fused multiply-add, so every explicit fma of ours is a library call there. The lower block adds link-time optimization on every side, which moves the reference by three percent and the PDE method by eight. Bold marks the book kernel’s batch path, bit-identical to its scalar entry on every quote. [§]These three rows were measured on the previous release (v0.3.0, seven passes of four sweeps, book kernel with seventeen and eighteen rows and the single-pass driver) and were not repeated for v0.3.1.

(5.2 against the release build of Table 6) is a deployment-mode number, not a method-to-method one—the warm driver is handed each node’s previous root, which the cold reference is not, and delivers the warm few-ULP contract—but it is the number a persistent-book system experiences. Its advantage over a cold re-inversion has narrowed as the cold driver improved. Warm was worth about twice the cold path before the driver was interleaved, and is worth roughly 1.3 times now. The streaming case for it rests less on raw margin than on the shape of the cost. Having no routing and no deferred drain, its cost is flat in batch size from one full register upward, whereas the cold driver’s sort-and-drain overhead only amortizes over batches of a few thousand. Below one register width the warm driver, like every batch driver here, falls to its scalar tail, so the step at eight quotes in the batch-size sweep is the register filling and not a measurement artefact: at one, two and four quotes the warm path costs 288 ns per quote against the reference’s 264, and at eight it costs 37. The warm contract can now be stated more sharply than a sweep allows. The update actually run is Householder’s method of order three applied to $F(v) = C(h, v) - c$. Written in $x = v/2$ on $g(x) = \frac{1}{2}e^{-h/2}(c - C(h, 2x))$, the constant prefactor and the affine change of variable both cancel out of the Newton quotient and of the two curvature ratios, so no translation is lost. Feeding the exact derivatives $F' = \phi(d_1)$, $F'' = -d_1\phi(d_1)(k/v^2 + \frac{1}{2})$ and their two further v -derivatives into a Newton–Kantorovich gate on a disk about the previous root, and replacing the Newton majorant by the degree-three one appropriate to an order-four iteration, gives a per-quote certificate that is elementary to evaluate and was validated against the exact iteration at ninety digits on 829 gated cases with no violation. Three consequences matter here. The two-step fast path is a theorem for $W \leq 4.33$ and remains sweep-measured beyond it, Kantorovich being sufficient rather than necessary. The certified move over the whole code box is a (h, W) -indexed surface rather than a single percentage, and reading a single number off it understates the shallow end by a factor of six and overstates the deep end by a factor of two and a half. And, most usefully, the bound at $W = 8$ under a half-percent move is 1.5×10^{-17} , some 160 times below the floor actually observed, which turns

Workload	LBR	Routed inverter, ns per quote		
		scalar	batch AVX-512	batch AVX2
NEAR batch, $h=0.2$	139	141	26	42
FAR batch, $h=1$	169	69	17	18
WING subset [†]	307	1225	250	347
UPPER batch, $h=1$ [§]	205	557	74	148
Market feed (2024 SPX, actual prices)	227	321	48	86
Market feed, warm 2-step (streaming)	227	—	37	79

Table 7: Median nanoseconds per quote (five repetitions, quiet host, GCC 11.4, the flags of Section E) on chart-pure fixed-moneyness batches of 4096 quotes (fixed h , varying price) and on the market feed of 30,000 actual projected prices (route mix 92/1/7/0 percent across near/wing/far/upper). The reference column is its author’s scalar implementation compiled at our flags, contraction off and assertions on. At its release flags it is seventeen percent faster (Table 6). The market row is timed all-inside, the fixed-moneyness rows hoist the per- h constants. [†]The wing row is the wing subset of the 120,000-point coverage grid under the mixed-grid driver, the only driver that vectorizes the wing, so its reference column is a different, wing-only population. The final row is the guarded two-step warm driver seeded from each node’s previous variance, under the few-ULP warm contract, listed against the reference for scale rather than like for like. Bold marks where the batch beats the reference. [§]This row is the three-step ladder construction of the previous release and was not repeated for the one-step chart of Appendix B.4.

this section’s attribution of that floor to the double-precision price representation, rather than to the iteration, from an inference into a consequence.

The same algorithm on a GPU. Kernels whose entire rounding behaviour is a frozen, branchless fused-multiply-add sequence port to a GPU with their semantics intact. We mirrored all four charts and the book kernel as CUDA device functions, operation for operation (compiled with `-fmad=false`, the device analogue of `-ffp-contract=off`), and scheduled them the way a persistent book invites. The book is stored bucket-ordered, partitioned by chart, each chart runs as one kernel over a contiguous slice, and no sort, gather or scatter touches the device in the timed passes, so the figures of Table 8 are resident, pre-partitioned kernel throughput. The host performs only the per- h hoists the CPU index pass already performs, and the rare scalar-only branches (the rescue wedge and the analytic edges, zero quotes on this feed) stay host-side, counted and excluded. On the market feed tiled to 5.01 million quotes every device result equals the CPU scalar entry’s double exactly, with zero mismatches, on every chart in every run, and the same holds on four chart-pure fixed-moneyness surfaces of 5.24 million quotes each, which is also how the large-volatility chart is exercised at all. The figures are fp64 throughput on datacenter-class hardware. Consumer GPUs execute double precision at 1/32 to 1/64 rate and will not reproduce them.

A number of that size invites disbelief—a tenth of a nanosecond is roughly the time light needs to cross the die—so it is worth showing that it is unremarkable. It is a throughput, not a latency. No individual quote is inverted in that time, and a single isolated query costs microseconds dominated by launch overhead. But of order 10^5 threads are resident at once, so completions emerge at that rate. The card carries 114 multiprocessors of 64 double-precision lanes each, 7,296 in all, at about 1.7 GHz, against the development laptop’s single 512-bit fused-multiply-add port, eight lanes, at about 3.2 GHz. The ratio of double-precision execution resources is therefore of order 400 to 500, the range reflecting how far the laptop’s clock throttles under sustained wide-vector load. The measured ratio between the two market-feed figures is about 620, modestly above that range, and the excess is informative rather than anomalous. The CPU figure is a *cold* driver that routes, sorts and drains on every snapshot, whereas the

device schedule stores the book bucket-ordered once and never re-partitions it, so the comparison charges the host for work the device has been designed out of. Set against the CPU’s warm streaming driver, which likewise does no routing, the ratio is about 470 and falls inside the range. Both implementations thus attain broadly the same efficiency per execution unit, which is the substantive point. The device figure is the hardware’s arithmetic rather than an artifact of the harness, and the routed method carries to a machine three orders of magnitude wider without losing efficiency on the way. The whole-book kernel tells the same story with one addition. Its CPU batch at 16.6 ns against its device figure in file order, 0.061, is a ratio of about 270, below the range, as it should be, because the CPU driver partitions mixed registers by region and overlaps its passes, whereas a mixed warp on the device still pays for both branches. Sorting the book by *a* removes that cost on the device, where lanes are grouped by warp. The sorted 0.031 is a ratio of about 540, just above the range.

One device-side lesson generalizes beyond this solver, and it is the mirror image of the register argument that shaped the CPU driver. The wing initially ran twenty-four times slower per quote than the small-moneyness chart, which no operation count explains. The cause was storage class, not arithmetic. Its $\log H$ fit accumulated a row of Chebyshev coefficients and a row of derivative coefficients into two arrays indexed by a *runtime* loop bound, and an array a compiler cannot resolve to registers is placed in device *local* memory—off-chip. The compiler’s own report made this visible without running anything, showing a 432-byte stack frame, exactly the two arrays, and 128 registers per thread, which caps occupancy near a quarter and removes precisely the warp-level parallelism that hides latency. The small-moneyness chart, whose polynomial degrees are compile-time constants and therefore fully unrolled into registers, reported a zero-byte frame and thirty-two registers. The fix is a restructuring, not new mathematics. Because both outer Clenshaw recurrences and the derivative recurrence run in the same direction, each row can be computed and consumed in a single backward sweep, holding a constant number of values instead of two arrays. The frame falls to zero bytes, and in the campaign where the change was made the wing evaluated in 0.83 instead of 1.28 nanoseconds on the feed’s wing slice, taking the full book from 0.12 to 0.10. Every lane still executes the identical sequence of fused multiply-adds, so the transformation is bitwise neutral, and it was verified as such against the array form over the full feed and a dense wing sweep. The wing is no longer the device’s cost centre. Its per-quote cost in situ has since risen to 1.17 ns, because the seam of Section B now hands all but the genuinely extreme quotes to the tabulated charts and what remains sits deeper, but it carries only 1.1% of the book and so accounts for about a sixth of the total. Two thirds is now the small-moneyness chart, 92% of the book at 0.057 ns, and that is where the next lever had to act. The kernels of Section A.3, below, are that lever.

A second layout lesson, opposite in sign to the first, shaped Section A.3. A first device port of an oracle-gated two-dimensional table in the intrinsic coordinates, twelve cells with a graded row profile stored as ragged rows behind three offset arrays, measured 0.062 ns per quote on the same H100 against the small-moneyness kernel’s 0.037 while executing *less* arithmetic, about sixty-four fused multiply-adds against two hundred and fifty. Launching one kernel per cell over bucket-ordered slices, the schedule this paper recommends for the shipped charts, made it slower still, 0.077, because twelve launches cost more than the divergence they removed. The loss lay in the table’s layout. Its loop bounds are data-dependent, so the compiler cannot unroll them, and its coefficients are reached through an index rather than present as constant-bank operands, whereas the shipped kernel is straight-line code whose every coefficient is a compile-time array index. Emitting the row profile as template constants restored that shape and, with it, register parity with the shipped kernel (36 registers, no stack frame). The exact rows of Section A.3 then removed most of the table altogether. The compiled recurrence kernel is 512 machine instructions with 134 double-precision fused multiply-adds, against 696 and 190 for the template-emitted table it replaced, in straight-line code. The whole-book kernel compiles to 48 registers with no stack frame. Both reproduce the CPU reference bit for bit when their device functions are

compiled for the host over the full price box, the port check the shipped kernels passed before their device runs, and on the device itself (Table 8, lower block). Their timings confirm the layout lesson and add one of their own. The recurrence kernel inverts the small-moneyness feed tile at 0.021 ns per quote against the shipped small-moneyness kernel’s 0.037, the same in file order and sorted, since it has no branch to diverge on. The template-emitted table kernel takes 0.036, and the ragged twelve-cell table, on the same tile in file order, 1.50, twenty-four times slower than its bucketed 0.062, which is the divergence a data-dependent layout costs when its cells are not warp-uniform. The whole-book kernel carries one branch, at $a = 2\pi$, and region B’s two square roots. In file order, where nine percent of the feed is region B and every warp mixes the two, it inverts the full feed at 0.061 ns per quote, a fifth under the shipped bucket-ordered book. On the same full feed sorted once by a it takes 0.031, about half, and below the shipped small-moneyness kernel alone, with both device runs agreeing to three digits. The shipped 0.079 is itself a sorted schedule, the book partitioned once by chart and kept so, and a book kept partitioned by the side of $a = 2\pi$ is the corresponding deployment of the whole-book kernel. The persistence of that partition across snapshots is reported in the main text. Branch by branch on region-filtered tiles (Table 9) the kernel costs 0.060 on the small-moneyness tile, 0.070 on the interior tile, where region A and B quotes share warps, and 0.041 in the wing, where the conformal rows run without a branch. On the synthetic UPPER tile of Table 3, where the kernel declines every quote and the one-step chart answers in the same thread, the cost is 0.054. The one-step chart alone takes 0.045 on that tile against 0.173 for the three-step ladder construction, measured on the same card in the same session. With the inputs uploaded from pageable host memory and the results read back on every pass, the same kernel took 1.12 ns per quote in the earlier session in which the table method was also run. The difference is the 120 MB of host traffic per pass, and it is the convention under which the table method [19] reports and under which Appendix E compares the two.

Relation to other methods. The explicit formulas [16, 25, 27] are the fastest possible single evaluations and vectorize trivially, but they hold on bounded regions with accuracy far from the last bit. In the present architecture their natural role is as seeds. The Chebyshev interpolation of [7] treats the implied-volatility surface as a single global object, whereas partitioning the domain at the inverse map’s branch points is what lets a low, fixed degree hold the last bit out to the boundary of feasibility. The PDE table method [19] precomputes a polynomial table over moneyness and price, as we do, but targets 10^{-7} , and the gap in accuracy and size (46 MB against 220 KB) has one source. Removing the singular endpoint regimes from the two-dimensional fit lets the tabulated chart follow the branch-point principle instead of refining toward the boundaries. Their published 9.25 ms per million prices on a Tesla V100 against the 7.68 ns per quote we measure on the H100 under their transfer-inclusive convention shows their figure is bound by host transfers rather than arithmetic, as ours is under the same convention (4.93 ns with the uploads removed). Their published CPU convention uses every core under OpenMP (11 ns per quote on a twenty-eight-core server), where every CPU figure here is one thread. Table 9 gives both methods branch by branch on identical region-filtered tiles, so that neither the route mix nor the tile favours either side. Their cost is nearly flat across branches, as a single table’s must be, the book kernel’s batch path is 3.2 to 4.7 times their scalar rate on one core and about seven times it on the H100 under their convention on every branch, and their scalar entry is faster than ours on every branch, by a third on the small-moneyness tile and by more than half in the wing.

On the scalar iterative side, [13, 14] keep the reference’s unit of work and optimize inside it, with a distribution-aware seed, one fast and two exact third-order Householder steps on an erfcx log-price objective and no convergence loop, reaching 137 ns per call in Java against 243 for a Java port of the reference on their grid, at a worst error of 130 units in the last place of total volatility (23 with an optional final Newton polish). That scalar latency is of the same order as

Workload	quotes	ns per quote
NEAR surface, $h=0.2$	5,242,880	0.038
FAR surface, $h=1$	5,242,880	0.141
WING surface, $h=1$	5,242,880	0.671
UPPER surface, $h=1$	5,242,880	0.043
Full book (2024 SPX market mix)	5,010,000	0.079
recurrence kernel, NEAR feed tile, file order or sorted	4,990,713	0.021
table kernel of Section A.3, NEAR feed tile	4,990,713	0.036
whole-book kernel, full feed, file order	4,980,000	0.061
whole-book kernel, full feed, sorted by a	4,980,000	0.031
whole-book kernel with UPPER fallback, synthetic UPPER tile	4,980,000	0.054
whole-book kernel, sorted, with transfers [‡]	4,980,000	1.12
PDE table method [19], with transfers [‡]	4,979,968	7.68

Table 8: GPU throughput on one NVIDIA H100 (PCIe, CUDA 12.4, `-fmad=false`), median nanoseconds per quote over 300 passes, two independent runs agreeing to three digits on every row but the tabulated surface (two digits). Every result is bitwise identical to the CPU scalar entry. The four chart rows are chart-pure fixed-moneyness surfaces route-filtered through the classifier, which is also what makes the large-volatility row measurable at all. The full-book row is the market feed at its live route mix, where the three charts cost 0.057, 0.153 and 1.18, the wing sitting deeper in situ than on its surface. The lower block is the intrinsic-kernel session on the same host class, the feed tiled to about five million quotes and every kernel bit-identical to the CPU entry (4,979,502 covered quotes, 498 handed to the routed charts). Its file-order and sorted figures differ by a factor two because the kernel’s one branch, at $a = 2\pi$, diverges inside mixed warps. [‡]Uploads h and c from pageable host memory and reads w back on every pass, the PDE method’s published convention. These two rows are bound by the host link and come from one earlier session on the same GPU model in which both methods ran on the same card. That row is its OpenCL double-precision path from the authors’ sources with their 46 MB table.

the book kernel’s scalar entry, 115 ns on our host. The difference is the 16.6 ns per quote once eight quotes advance in lockstep, which a Householder chain, however trimmed, does not offer. The vectorized port of the reference [22] re-implements the seed, its normalized Black function and its two Householder corrections as whole-array operations with masked branch selection, in NumPy with Numba on the CPU and as one fused Triton kernel on the GPU. On our feed and host its Numba path takes 411 ns per quote at one thread in a loaded session (350 with all eight threads), with a worst relative error of 3.3×10^{-14} against the book-kernel reference values, and its default Halley path 2,160 ns at 6.8×10^{-5} , so on the CPU the masking cost of running a four-branch iteration on every lane exceeds what the width returns. Its author reports 116 ns per quote for the same path on a 10^5 -point grid of his own. On the GPU its author reports the Triton kernel at 0.56 ns per quote on an H100 NVL, CUDA-event compute time on a resident batch, at a worst error of 9×10^{-15} against the reference’s own C implementation. That is the author’s figure on a faster variant of the card, not our measurement. Taken at face value it sits an order of magnitude behind the book kernel’s resident figures and seven times behind the routed charts’ bucket-ordered book, which is the expected order, because a fused two-step Householder iteration on the GPU pays its four-branch masking and its error-function calls per lane, and the book kernel pays neither.

Region	feed quotes	PDE method	PDE method, CPU		Book kernel, CPU		PDE, H100	Book kernel, H100	
		max rel err	scalar	OpenMP	scalar	batch	with transfers	resident /	with transfers
NEAR	27,574	1.2×10^{-5}	74	—	113	15.6	7.09	0.060	/ 1.14
FAR	2,084	1.7×10^{-8}	66	—	129	16.4	6.77	0.070	/ 1.16
WING	342	5.3×10^{-9}	66	—	163	20.7	6.71	0.041	/ 1.12
whole feed	30,000	1.2×10^{-5}	78	11*	115	16.6	7.68	0.031	/ 1.12

Table 9: The PDE table method [19] and the book kernel on identical region-filtered tiles of the market feed, ns per quote unless stated. Accuracy is the worst relative error in σ against the book kernel, itself within 3 units in the last place of the 40-digit oracle on this feed. CPU columns are one thread on the quiet host, AVX-512 build (*their OpenMP column is their published convention, all 28 cores of the H100 host). The device tiles cycle each region’s quotes to 4,980,000. “With transfers” uploads the inputs and reads the results back on every pass, their published convention and their wrapper’s only mode. The book kernel’s resident figure is the kernel alone on a device-resident book, sorted by a for the whole-feed row and in file order for the region tiles.

F Reproducibility and artifacts

F.1 Reproducing the reported results

A golden-vector verifier (oracle-checked $(h, c) \rightarrow w$ triples spanning every chart and seam) lets an adopter confirm, in one command after building, that their compiler and hardware reproduce both the accuracy gate and the bit-identity invariant. What each number needs differs. Every accuracy figure of Appendix D.3 and every bit-identity check regenerates from the repository alone. The CPU rows of Appendix E need, in addition, the reference’s sources and the PDE method’s sources and table, obtained from their authors, and a market feed regenerated from a licensed copy by the recipe the repository documents. The GPU rows need a datacenter card, and the persistence figures need the licensed yearly file. The raw outputs of every run reported here are checked in beside the harnesses that produced them.

F.2 Artifact inventory and external inputs

The implementation and every harness behind the tables form the version v0.3.1 reproducibility package (BSD 3-Clause). The package is public at <https://github.com/wol-fi/volfi>. It contains the header-only C++17 implementation (the book kernel with its 63-term table and exact rows, its SIMD twins and the generator that derives the rows in exact rational arithmetic, and the routed charts’ runtime, frozen tables, wing evaluator and checked API). The 40-digit `mpmath` oracle, as a script that regenerates the regular set from its grid definition and re-inverts any shipped oracle file at a chosen precision, so that the hand-built stressed, edge and verification sets ship as data and remain checkable; the generators of the book kernel’s truth sets and of its boundary-concentrated campaign, with the 60-digit recheck of its worst points; and the node-persistence measurement over the year’s consecutive snapshots. The golden-vector verifier and its oracle files (the one-command build check of Section E). The accuracy harness producing Table 2, Table 5 and Figure 3, and the one-binary harness producing Table 6, with the build lines that compile the reference at its author’s flags with and without link-time optimization. The bit-identity and poison-test harnesses. The throughput benchmark, including the build lines for the AVX-512, AVX2 and scalar targets used in Section E, and the raw outputs of the quiet-host campaign. The CUDA full-book port behind Table 8, comprising the device kernels mirrored from the scalar entry, the bucket-ordered driver with its per-chart ULP cross-check against the CPU reference compiled into the same binary, the mechanical device-constant generator, and the `nvcc` build line (`-arch=sm_90 -fmad=false`). The device driver of the intrinsic kernels of Section A.3, with its host self-check, its sorted and transfer-inclusive timings and its region tiles.

Finally, the harness that builds the table method [19] from its authors' published sources and times it on the same feed and tiles, on the CPU and through its OpenCL path (Table 9), with the container recipe that supplies the OpenCL loader the CUDA image lacks.

Three inputs are not redistributed, because their licences are not ours to pass on. The market feed derives from a licensed OptionMetrics file, so the repository carries the filter and projection recipe needed to regenerate the 30,000 normalized (h, c) pairs from a licensed copy, and every harness that consumes the feed accepts any file of such pairs; the verification suite and every accuracy table run without it. The reference implementation [11] (revision 1520, obtained from <http://www.jaeckel.org/>) is the version current when the benchmarks were run; the harnesses compile it unmodified from the author's sources at a path the caller supplies. The PDE table method is built from its authors' MIT-licensed sources and their 46 MB coefficient file, both obtained from their repository as its instructions describe. For independent audit, the closed forms of V_2 and V_4 , the ladder coefficients B_n , the frozen seam and wing fits, and the exact residual equations driving the Householder and Newton refiners appear in the shipped headers in the notation of the text. One portability note deserves record. A code-generation defect in one Windows GCC toolchain (misaligned vector spills in a heavily spilled function, affecting AVX-512 and AVX2 builds alike and unrelated to this code's logic) can crash the mixed-grid driver on that platform. The artifacts document the affected configuration, and all results here are from Linux, which is unaffected.