Accurate Products, Inaccurate Inverse: Emulated FP64 BLAS in Blocked Triangular Inversion

Erzheng Zhang
Department of Computer Science, University of Toronto
Toronto, Canada
erwin@cs.toronto.edu

***Abstract*—FP64 emulation on integer tensor units now ships in cuBLAS and GEMMul8, and each emulated matrix product can be checked against an accuracy grade. We ask if a routine built from such products inherits their accuracy, and show that blocked triangular inversion, the core of LAPACK's inverse of a symmetric positive definite matrix, does not. Under fixed-precision emulation every product passes the grade-B error bound of Demmel et al., yet the inverse errs at 7.1×10^{-2}, against 2.0×10^{-15} natively. The matrices are well conditioned after diagonal scaling, but the scaling spans 64 binades. The cause is one product, $T = X_{22}L_{21}$, whose summed index carries both the diagonal scaling and its inverse, which a grade-B bound does not absorb. Computing those products natively restores native accuracy; leaving any one of them emulated does not. The pattern holds on four GPUs, and on real covariance matrices the fixed-precision routes lose up to about four orders of accuracy. Equilibration, a solve-based association and cuBLAS's automatic dynamic precision avoid the loss, and an exact error recurrence audits a completed run. On an RTX 4090 the fastest accurate emulated inversion takes 0.40 of cuSOLVER's time. Where each product's scaling falls decides if per-product accuracy carries over to a composite routine, and that is what builders of emulated libraries need to check.**

I. Introduction

Since CUDA 13, cuBLAS can run DGEMM, TRMM and TRSM on integer tensor units by emulating FP64 arithmetic [1], [2]. A guard, automatic dynamic precision (ADP), estimates per call how many bits a product needs. GEMMul8 offers fixed-precision emulation [3], [4], and emulation has been proposed as a default algorithm for a wide range of applications [5]. Demmel et al. grade each emulated product by the error bound it attains [6]. LAPACK-level libraries such as cuSOLVER, MAGMA and SLATE will call these products, so their accuracy is the accuracy that matters.

A composite routine combines many products. Demmel et al. proved that grade-B products keep tiled Cholesky componentwise accurate. LAPACK's inverse of a symmetric positive definite matrix has a second half: triangular inversion (TRTRI), then the product of the inverse factor with its transpose (LAUUM). Whether it inherits the products' accuracy has not been asked.

It does not. On an H100 at $n = 4096$, every product of the blocked inversion meets the row-and-column-scaled (B-type) error test with room to spare; the largest ratio, 0.12 of the allowance, is a native product's. Yet through fixed-precision emulation the inverse is wrong in its second digit: relative error 7.1×10^{-2}, against 2.0×10^{-15} for native FP64 on the same well-conditioned input. The cause is not a defective product but which index of a product the diagonal scaling falls on (Fig. 1).

Users meet such matrices whenever variables carry different units, as covariances do. They use explicit inverses where many inverse entries are needed, as in kriging variances [7]. A covariance with unit factors from 10^{-6} to 10^{6} loses accuracy the same way (14 moduli, RTX 4090, $n = 8192$: 1.1×10^{-4} against 3.4×10^{-15}, Appendix D).

The contributions are four.

- *Mechanism* (Section III): a three-by-three example proving that grade B permits the loss, an exact error recurrence with an a posteriori enclosure, and a provably scale-equivariant association.
- *Diagnosis* (Sections V-A, V-B and V-E): on four GPUs and two libraries, every product passes while the inverse fails; a causal test needs every inner-index product.
- *Repairs and cost* (Sections V-C–V-G): three repairs that keep every product emulated, an enclosure that certifies every accurate run, and complete device-resident timings.
- *Application* (Section V-H): the loss reaches cokriging predictions on census data, within a certified bound at the model's nugget.

II. Background and related work

A. FP64 emulation and its precision demand

The Ozaki scheme [8], [9] computes an FP64 product $C = AB$ on integer units. It scales each row of A and each column of B by a power of two, so that the largest entry of each has a fixed exponent. It splits the scaled entries into INT8 slices and multiplies slice pairs exactly with integer accumulation on tensor cores. The partial products are summed in FP64. Ozaki-II replaces the slices by residues modulo q coprime moduli and reconstructs the product by the Chinese remainder theorem [3]. More slices or moduli buy more bits at more cost.

In both, the bits available to c_{ij} are set relative to $r_i s_j$, with r_i and s_j the largest entries of row i of $|A|$ and column j of $|B|$ [4], [9], [10]. An entry much smaller than $r_i s_j$ loses about $\log_2(r_i s_j/|c_{ij}|)$ bits, its *scale excess*, so a fixed bit

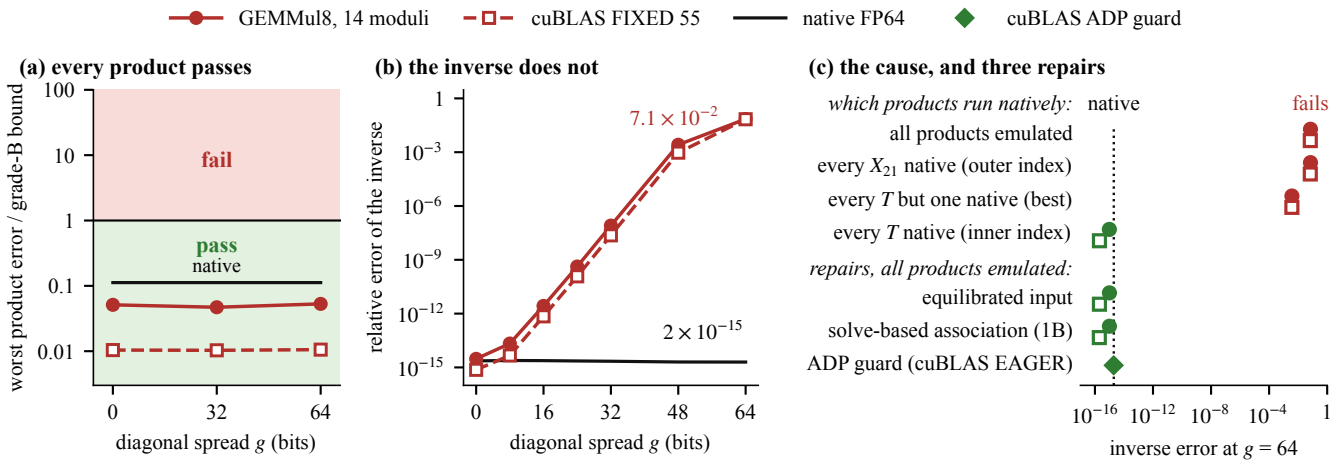

Fig. 1. Every product passes; the inverse does not (H100, $n = 4096$). (a) Worst TRTRI product error over its grade-B bound; (b) the inverse's error; (c) at $g = 64$, native T restores it, native X_{21} does not, and three repairs restore it with every product emulated.

count suffices only when every entry's excess is small. Fasi, Mary and Mikaitis bound the fixed-point width needed for a target precision and show that ADP's bit estimate can be very pessimistic [11]. cuBLAS runs either ADP or FIXED, a set number of bits [2]. ADP estimates the bits a call needs, then emulates with that many bits or dispatches to native DGEMM; its strategy, DEFAULT or EAGER, sets how readily emulation is chosen. GEMMul8 takes the number of moduli q. All of this is stated for one product; this paper asks what happens when a routine combines many.

B. Accuracy grades

Demmel et al. [6] grade a GEMM by the bound it attains:

- grade A, $|\widehat{C} - C| \le f(k)\,u\,|A||B|$ componentwise, the bound of classical multiplication;
- grade B, $|\widehat{C} - C| \le f(k)\,u\,\mathrm{rowmax}(|A|)\,\mathrm{colmax}(|B|)$, the natural model of row- and column-scaled emulation;
- grade C, the normwise bound that suffices for LU and QR [12], [13].

Grade B is invariant under row scaling of A and column scaling of B, not under scaling of the summed index. This paper uses one normalization, $f(k) = k$: with r_X the largest ratio of a product's error to u times the grade-X right-hand side, the product passes the *A-type* or *B-type test* when $r_X/k \le 1$. Since $(|A||B|)_{ij} \le k\,r_i s_j$, the two tests are not nested. The test $r_B/k \le 1$ is stricter than Demmel et al.'s own benchmark threshold $f_B(k) < k^2$. A quantity is *certified* when a bound with every rounding enclosed proves it.

C. Blocked triangular inversion and LAUUM

LAPACK inverts a symmetric positive definite A by Cholesky (`potrf`, $A = LL^T$), triangular inversion (`trtri`, $X = L^{-1}$) and LAUUM ($A^{-1} = X^T X$), together `potri`. Recursive blocked TRTRI splits $L = \left(\begin{smallmatrix} L_{11} & \\ L_{21} & L_{22} \end{smallmatrix}\right)$, inverts the diagonal blocks, $X_{ii} = L_{ii}^{-1}$, and forms

$$T = X_{22}L_{21}, \qquad X_{21} = -TX_{11}. \tag{1}$$

We call T the *inner-index* product and X_{21} the *outer-index* product. In Du Croz and Higham's terms [14], (1) is Method 2B. Method 2C forms T and then solves with L_{11} (the reference `dtrtri`); Method 1B forms $W = L_{21}X_{11}$ and solves with L_{22}, $X_{21} = -L_{22}^{-1}W$. In ordinary arithmetic 1B and 2C keep a componentwise residual bound and 2B does not.

D. Related work

Mixed-precision solvers are judged by solver-level outcomes, often with iterative refinement as the remedy [15]–[17]; LU and HPL on INT8 emulation fail the residual check at too few splits [18]. Block algorithms on fast level-3 BLAS stay normwise backward stable but may lose componentwise accuracy and invariance under power-of-radix scaling [12]. Carson and Ma analyse block LU with an Ozaki trailing update and a row- and column-norm bound [19]; they do not treat inversion or diagonal scaling. Diagonal scaling of a single product's operands is studied for fast multiplication [20], and equilibration is classical [21], [22]. Known, then, are that operand scaling consumes a fixed-precision product's bits [9], [11], that equilibration helps, and that the association of a block inversion matters in ordinary arithmetic [14]. To Demmel et al., whose theorem concerns Cholesky, this paper adds a composite in which grade-B products do not suffice. To Du Croz and Higham it adds an exposure of $X_{22}L_{21}$ that a diagonal scaling creates and native FP64 does not see. To the single-product evaluations of ADP [1] and the width bounds of Fasi et al. [11] it adds the behaviour inside a complete shipped routine and the cost of each repair.

III. Why per-product checks do not compose

a) Where the scaling falls: Let $A = DMD$ with $D = \mathrm{diag}(2^{k_i})$, $k_i \in \{0, \dots, g\}$, and $M = L_M L_M^T$. Then $L = DL_M$ and $L^{-1} = L_M^{-1}D^{-1}$. Fig. 2 shows which scaling each product's row, summed and column index carries. In $T = X_{22}L_{21}$ the summed index k carries d_k^{-1} on the columns

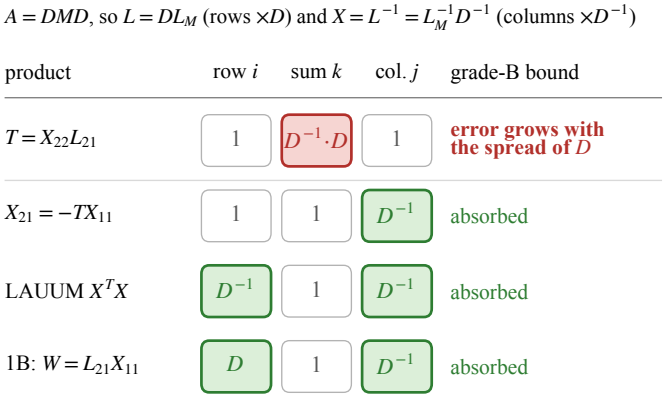

Fig. 2. Only T sums over a scaled index (red); grade B absorbs the scaling on row i and column j (green).

of X_{22} and d_k on the rows of L_{21}. $T = L_{M,22}^{-1}L_{M,21}$ itself does not depend on D, but a row of X_{22} or a column of L_{21} spans as many binades as the exponents in the contracted block. The span is g only if both extremes fall in one block, so it depends on the order of the k_i. X_{21} and LAUUM's $X^TX = D^{-1}L_M^{-T}L_M^{-1}D^{-1}$ carry D on outer indices only; Method 1B forms no product that sums over a scaled index. Native FP64 is unaffected: native inversion by (1) is *bit-for-bit scale equivariant* in every test here, that is, its computed inverse of DMD equals D^{-1} times its computed inverse of M times D^{-1} exactly. The loss needs a product bound that sees the inner scaling.

b) A grade-B bound does not secure the inverse: An *admissible evaluation* of the recursion returns, for each product, any value whose error meets the stated bound, and evaluates every other operation exactly.

Proposition 1: Let L_M be the 3×3 lower-triangular matrix of ones, $D = \mathrm{diag}(1, 1, 2^g)$ and $L = DL_M$, split after the first row and column. For every δ with $|\delta| \le f_B(2)\, u\, 2^g$ there is an admissible evaluation of the blocked inverse under grade B in which $\|\widehat{X} - L^{-1}\|_F/\|L^{-1}\|_F = |\delta|/(4+4^{-g})^{1/2}$, while $\kappa_2(M) \approx 16.39$ for every g.

Proof: $X_{11} = 1$, $X_{22} = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 2^{-g}\end{smallmatrix}\right)$ and $L_{21} = (1, 2^g)^T$, so $T = (1, 0)^T$, whose second entry is the cancellation $-1 + 2^{-g}2^g$. For it the row maximum of $|X_{22}|$ is 1, the column maximum of $|L_{21}|$ is 2^g and $k = 2$, so $\widehat{T}_2 = \delta$ meets grade B. With every other operation exact, $\widehat{X}$ differs from L^{-1} only in entry $(3, 1)$, by $-\delta$, and $\|L^{-1}\|_F^2 = 4 + 4^{-g}$. ∎

With $f(k) = k$ the bound admits a relative error of about 2^{g-53}, 2,048 at $g = 64$, while grade A admits only $O(u)$ on the same entry (Appendix B). The example applies verbatim to Method 2C, since solving with $L_{11} = 1$ changes nothing. Tiled Cholesky escapes this because the right-hand sides of its grade-B TRSM and SYRK are bounded through the factor's row norms, at most $(a_{ii}a_{jj})^{1/2}$, which carry the scaling on outer indices [6]. The inversion product T has no such protection: its grade-B allowance grows with the spread of the contracted block.

c) How product errors reach the inverse: L is the stored factor; $X = L^{-1}$ and $Y = X^TX = (LL^T)^{-1}$ are exact, and the factorization's error is outside the statement. At a split with column block ρ_1 and row block ρ_2, write the computed products as $\widehat{T} = \widehat{X}_{22}L_{21} + E_s$ and $\widehat{X}_{21} = -(\widehat{T}\widehat{X}_{11} + F_s)$, and let $\Phi_s = -(E_s\widehat{X}_{11} + F_s)$, $\Delta = \widehat{X} - X$ and Δ_ℓ the error of leaf ℓ's inverse. LAUUM returns $\widehat{Y} = \mathrm{fl}(\mathrm{sym}(\widehat{X}^T\widehat{X} + G))$ with output error G and $\mathrm{sym}(C) = (C + C^T)/2$, one rounded addition per entry.

Theorem 1: Let the negations and block copies be exact and every addition and halving of the symmetrization finite (round to nearest; underflow allowed).

(a) At every split

$$\Delta_{21} = \Phi_s - \Delta_{22}L_{21}\widehat{X}_{11} - (\widehat{T} - E_s)\Delta_{11} + \Delta_{22}L_{21}\Delta_{11}, \quad (2)$$

and $\Delta = \sum_\nu \Lambda_\nu\Psi_\nu\Gamma_\nu$ over the splits and leaves ν, with $\Lambda_\nu = X_{:,\rho_2}L_{22}$, $\Gamma_\nu = L_{11}X_{\rho_1,:}$, $\Psi_s = \Phi_s - \Delta_{22}L_{21}\Delta_{11}$ and $\Psi_\ell = \Delta_\ell$ (a leaf's block is both ρ_1 and ρ_2).

(b) If $|E_s| \le \mathcal{E}_s$, $|F_s| \le \mathcal{F}_s$, $|\Delta_\ell| \le B_\ell$ and $|G| \le \mathcal{G}$, then $|\Delta| \le B$ for B assembled from the leaves upward by

$$B_{21} = \mathcal{E}_s|\widehat{X}_{11}| + \mathcal{F}_s + B_{22}|L_{21}\widehat{X}_{11}| + (|\widehat{T}| + \mathcal{E}_s)B_{11} + B_{22}|L_{21}|B_{11}, \quad (3)$$

and $|\widehat{Y} - Y| \le |\widehat{X}|^TB + B^T|\widehat{X}| + B^TB + \mathrm{sym}(\mathcal{G}) + 2u|\widehat{Y}| + 2^{-1074}\mathbf{1}\mathbf{1}^T$.

(c) Let $A = DMD$, $L = DL_M$, and write $B' = BD$, $\widehat{X}' = \widehat{X}D$ (M's frame). If every T meets grade A with $\mathcal{E}_s = c\,u|\widehat{X}_{22}||L_{21}|$, the other products meet grade B, and c and $B_\ell D_\ell$ do not depend on D, then (3) for B' and the resulting bound on $D|\widehat{Y} - Y|D$ are one function of the run's $\widehat{X}'$, $\widehat{T}$, $D\widehat{Y}D$, L_M and c for every power-of-two D, except that the absolute term becomes $2^{-1074}dd^T$ (d the diagonal of D). If T meets grade B only, $\mathcal{E}_s|\widehat{X}_{11}|$ carries at most the spread $\rho_s = \max_{k\in\rho_2} d_k/\min_{k\in\rho_2} d_k$ of D over the contracted block.

No error is assumed small, and no exact or high-precision inverse enters: the enclosure is computed from L, the computed $\widehat{X}$ and $\widehat{T}$, and per-product allowances. Part (c) is the mechanism in one line: grade A on T keeps the bound free of D, grade B lets it grow with ρ_s, and Proposition 1 attains that growth. The proof is in Appendix B.

d) An association that keeps the scaling outside: Method 1B forms $W = L_{21}X_{11} = D_2L_{M,21}L_{M,11}^{-1}D_1^{-1}$ and solves $X_{21} = -L_{22}^{-1}W$ recursively down to native TRSM leaves, so no product contracts a scale; per split of order $2h$ it spends $3h^3$ flops against the product form's $4h^3$. Call a GEMM *outer-equivariant* if $G(D_1A, BD_2) = D_1G(A, B)D_2$ exactly for power-of-two diagonal D_1, D_2.

Proposition 2: Under hypotheses (i) to (v) of Appendix B, the solve-based inverse of DMD equals D^{-1} times that of M times D^{-1}, bit for bit.

Equivariance transports the error with the result; it does not imply accuracy, which is measured. The product form has no such property, and neither has LAPACK's Method 2C on a

lower factor, which forms the same $X_{22}L_{21}$: on a transparent CPU emulator (8 slices, $n = 2048$) with only that product emulated, its error grows from 6.0×10^{-14} at $g = 0$ to 8.9×10^{-2} at $g = 64$, as the product form's does (Appendix D).

IV. Experimental setup

Table I lists the design. Two implementations run the same split rule and call structure, form every product with $\beta = 0$ and perform each update as a separate operation. The numerical harness takes LAPACK's host factor, inverts the leaves on the host and sends every GEMM to the device; all accuracy results and grades use it. Grades cover every output entry of every product, against a long-double reference product whose own error is bounded entry by entry (Appendix C). The device-resident implementation factors with cuSOLVER and solves the leaves with `cublasDtrsm`; all times use it. Each timed route runs from the uploaded matrix to the inverse, including `potrf` and any equilibration, 10 times interleaved with the other routes in a rotating order, in several sessions per GPU. Its products are not graded; an audit regrades the products of the timed routes on the RTX 4090 and ties the two implementations together (Appendix A). `CUBLAS_EMULATION_STRATEGY` was unset, so DEFAULT is the library's own default.

The reference for DMD is $D^{-1}X_M D^{-1}$, with X_M LAPACK's inverse of M, checked against high-precision inverses (Appendix A). The error of an inverse is $\epsilon_X = \|\widehat{X} - X_{\mathrm{ref}}\|_F / \|X_{\mathrm{ref}}\|_F$; for the real matrices it is measured in the equilibrated frame ($\widehat{X}$ scaled by $\mathrm{diag}(A)^{1/2}$ on both sides), where the normwise error cannot hide the small-diagonal block. The *slope* is the number of bits of ϵ_X lost per unit of g. Routes are named by their GEMM back end (Table I); "eq." marks the power-of-two equilibrated input, and unless a row says otherwise every GEMM of the blocked inversion runs on that back end. For a group of matrices a table gives the worst ϵ_X in the group.

TABLE I
Experimental design.

factor	values
synthetic inputs	DMD, $M = W^T W/n + I$, W standard normal ($\kappa(M) \approx 6$); k_i uniform on $\{0, \ldots, g\}$, $g \in \{0, 8, 16, 24, 32, 48, 64\}$; $n \in \{2048, 4096, 8192\}$, breadth runs at 16384; seeds 0–4
controls	exponents sorted or in contiguous chunks; $\kappa = 10^8$, 10^{14} with uniform diagonal; Gaussian-process kernel; unit factors 10^{-6} to 10^6 (diagonal span 2^{80})
real matrices	16 SuiteSparse SPD matrices and 8 raw-unit covariances (census, air quality), n 1074 to 8190, diagonal spans 2^{20} to 2^{44}
application	cokriging, California housing, $n \in \{4608, 9216\}$, nugget $\tau \in \{10^{-1}, 10^{-3}, 10^{-6}\}$
GPUs and libraries	FP64-scarce: RTX 4090, L40S; FP64-rich: H100, A100. cuBLAS 13.8.1, cuSOLVER 12.3.5, GEMMul8 v3.5.4; ten releases, cuBLAS 13.0 to cuBLAS 13.10, on H100, L40S and A100
GEMM back ends	*native* FP64 DGEMM; ADP with DEFAULT or EAGER; F55, F79, F111: FIXED with that many bits; q = 14, 16, 20: GEMMul8 moduli
routes	cuSOLVER `potri`; blocked product form; hybrids with every T or every X_{21} native; solve-based (Method 1B); each also equilibrated; LAUUM as one GEMM or blocked; leaves of width 256

TABLE II
ϵ_X at $g = 64$, $n = 8192$ (real: L40S, worst per group). Fixed routes fail; equilibration repairs all but bcsstk23.

				equilibrated	
input	native	q=14	F55	q=14	F55
H100	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.9×10^{-16}
A100	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.7×10^{-16}
RTX 4090	2.8×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.7×10^{-16}
L40S	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.9×10^{-16}
census	2.9×10^{-15}	1.5×10^{-11}	3.7×10^{-12}	8.4×10^{-15}	2.3×10^{-15}
SuiteSparse	3.6×10^{-14}	1.7×10^{-12}	3.2×10^{-13}	2.7×10^{-13}	3.4×10^{-14}

V. Results

A. Does the inverse fail although every product passes?

Grading every output entry of every TRTRI product of ten routes at three spreads answers this. Every product of every route passes the B-type test (Fig. 1(a)), yet the inverses of the fixed-precision routes fail (Fig. 1(b)). A product-level check therefore cannot tell these routes from accurate ones. A normwise audit and a reference-free residual agree, so the failure is not an artifact of the grading or the reference (Appendix D).

B. Is the inner index the cause?

If the mechanism is right, the inner-index products carry the loss and the outer-index products do not. Rerunning 14 moduli and FIXED 55 at $g = 64$ with chosen TRTRI products computed by native DGEMM tests this (Fig. 1(c)). Computing every T natively restores native accuracy, at a cost in speed (Appendix F); computing every X_{21} natively changes nothing, and leaving a single T emulated still leaves at least 3.7×10^{-3}. Over every replacement plan tested (Appendix D), the 64 that leave seven or more T emulated err at 1.6×10^{-2} or more, and the 10 that compute every T natively at 2.0×10^{-15} or less. Each T contributes through its own contracted block, so the repair needs every one of them.

C. What repairs it?

Anything that moves the scaling off the summed index should repair the inverse. Power-of-two equilibration does: inverting SAS, with S the power-of-two diagonal that brings every diagonal entry into $[1, 4)$, and scaling the result by S again removes D up to a factor of two per entry, exactly. Every equilibrated fixed route keeps its $g = 0$ accuracy at every g (Table II). It needs only the diagonal, which every SPD inversion has, and no change to the library.

The solve-based association repairs the inverse without touching the input. It is bit-for-bit scale equivariant in every tested case, as Proposition 2 predicts, and its error on the raw input never exceeds 5.5×10^{-15} (Appendix D).

ADP avoids the loss with no change to the input or the algorithm: every guarded route stays within a factor 1.05 of

TABLE III
ONLY THE FIXED ROUTES MISS THE A-TYPE TEST ON T BY A FACTOR THAT GROWS WITH g (H100, $n = 4096$).

	r_A/k on T		
route	$g = 0$	$g = 32$	$g = 64$
native	1.5×10^{-2}	1.5×10^{-2}	1.5×10^{-2}
EAGER	7.2×10^{-3}	9.1×10^{-3}	1.3×10^{-2}
q=14	2.8×10^{1}	1.1×10^{9}	3.5×10^{13}
F55	4.3×10^{0}	5.6×10^{8}	3.5×10^{13}
q=14, eq.	2.8×10^{1}	2.8×10^{1}	2.8×10^{1}
F55, eq.	4.3×10^{0}	4.3×10^{0}	4.3×10^{0}

native over every GPU's synthetic grid. Under DEFAULT the choice depends on the GPU: inside cuSOLVER's `potri` it emulates no call on the FP64-rich GPUs and some on the FP64-scarce ones (Appendix E).

D. Why do the repairs work?

Theorem 1(c) names the links: the size of T's miss of grade A, the spread ρ_s behind it, and the bits T is given. Four measurements test them.

a) The miss on T sets the error: At $g = 64$ every T has a 99th-percentile scale excess e_{99} of 69.5 to 71.5 bits, against at most 8.7 for X_{21}, so a fixed budget leaves T short and nothing else. The A-type test sees this (Table III): only the fixed routes miss it on T, by a factor that grows with g, and where that factor is large the inverse's error follows u times it to within about one order of magnitude (Appendix D). The size of the miss, not the miss itself, decides: at $g = 0$ the fixed routes miss grade A and still invert accurately.

b) The spread that matters is per block: Reordering the same exponents changes ρ_s and nothing else. With the exponents sorted, so that each contracted block spans few binades, 16 moduli give 5.5×10^{-11} at $n = 8192$, $g = 64$, against 6.7×10^{-2} in random order (contiguous chunks: Appendix D).

c) Equilibration removes the spread: After equilibration the diagonal spans less than a factor of four at every g, so no block carries a spread that grows with g. The equilibrated routes' miss on T at $g = 64$ stays at its $g = 0$ size (Table III), and ADP's estimates on the equilibrated input stay at most 75 bits.

d) ADP gives T the bits it consumes: ADP's estimate tracks $53 + e_{99}$ (least-squares slope 1.00, Fig. 3). Single products with a dialled-in inner-index span show what follows (Fig. 4): every output size switches to an FP64 path at about the same estimate, 4096×4096 outputs between 103 and 107 bits. That path is not free: it still computes the estimate, and on the H100 it is an FP64 GEMM of the emulation library, so it takes longer than native DGEMM, most on small outputs (Appendix E). Each T thus gets the bits it needs or runs in FP64, while the outer-index products keep emulating.

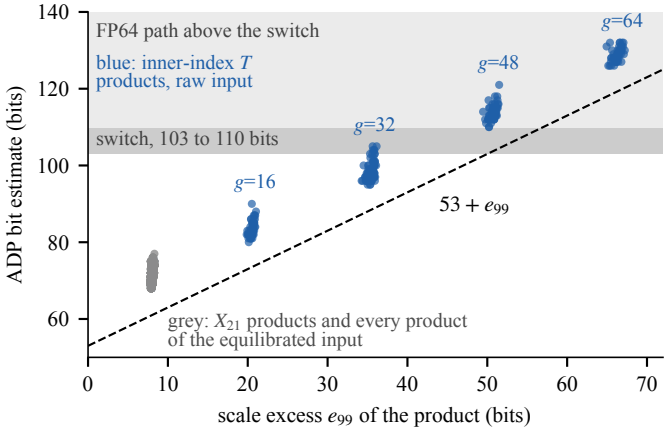

Fig. 3. ADP's estimate tracks $53 + e_{99}$, so it asks for the bits the inner-index scaling consumes; above the switch the call runs in FP64.

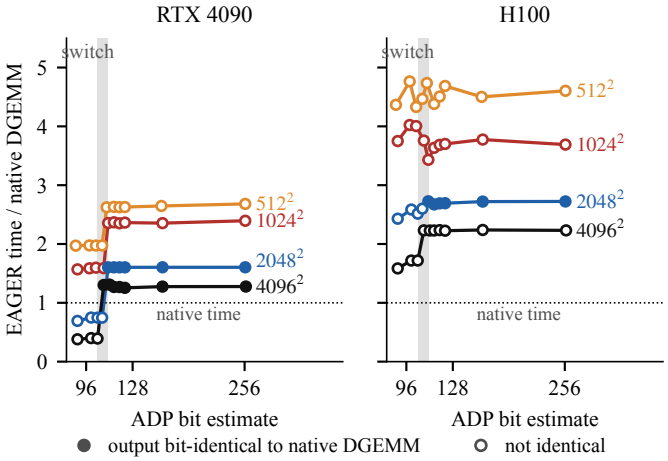

Fig. 4. Every output size of a single EAGER product switches path at the same reported estimate (shaded).

E. Does it hold on other GPUs and on real matrices?

The same grid ran on all four GPUs at $n = 8192$, and every route on 24 real SPD matrices (Table II shows two of the three groups). Every route reaches the same error at $g = 64$ on every GPU, to two digits, and loses bits at the same rate in g: the loss is a property of the arithmetic, not of one chip. It persists at $n = 16384$ and over seeds. At $g = 64$, 14 and 16 moduli and FIXED 55 all reach about 6.7×10^{-2}, so their error there is saturated; 20 moduli and FIXED 79 lower it by three to four orders, and FIXED 111 to within a small factor of native (Appendix D).

On the real matrices the raw fixed routes lose most on the census covariances (Table II). ADP brings every matrix within a factor 1.8 of native, and equilibrated 14 moduli every matrix but bcsstk23 within 10. On bcsstk23 the dynamic range within rows survives diagonal scaling ($\kappa = 3.9 \times 10^{7}$ after equilibration), equilibrated 14 moduli stay 108 times native, and the guarded and solve-based routes are the repairs to use (Appendix D).

TABLE IV
THE ENCLOSURE CERTIFIES EVERY ACCURATE PAIR (ROUTE–SPREAD PAIRS CERTIFIED, OF 30; –: NOT RUN).

GPU	$n = 2048$	$n = 4096$	$n = 8192$
H100	27	26	–
A100	27	26	26
RTX 4090	27	26	26
L40S	27	26	26

In every one of the 9 tested cuBLAS releases with FP64 emulation (cuBLAS 13.0 has none), the guarded routes stay at native level and the fixed routes fail. cuSOLVER's own `potri`, set to FIXED 55 through its emulation control, fails the same way, 1.2×10^{-2} at $g = 64$, and in the pinned release equilibration repairs it, so the shipped routine shows the loss under the same setting.

F. What does the enclosure certify?

Can a run be audited without a reference inverse? Theorem 1 is evaluated on each run's own operands, with every product's error bounded by its graded ratios and every rounding enclosed. It gives a certified bound on the error relative to $(LL^T)^{-1}$ for the stored L, and a run is certified when that bound is below 1. On every GPU and size run it certifies at least 26 of the 30 route–spread pairs (Table IV); on the H100 at $n = 4096$ these are all 21 accurate pairs and 5 moderately failing ones, with no bound only for the runs with the largest errors. The bound exceeds the measured error by one to five orders of magnitude, more at larger n, as worst-case bounds do. The enclosure is therefore a usable audit of a completed run, needing neither a reference inverse nor more precision than the run itself (Appendix C).

The evaluator itself was checked in exact arithmetic, against an independent second evaluator and by rebuilding stored runs on a CPU (Appendix C).

G. What does accuracy cost?

Is an accurate emulated inversion faster than native? Comparing complete device-resident routes that differ in one factor answers this (Table V, blocked LAUUM throughout; rows 2 and 3 multiply to row 4). Equilibration is nearly free, which makes it the cheapest repair. Where FP64 is scarce, emulation pays: 16 moduli take 0.30 of native's time on the RTX 4090, more than recovering the blocked algorithm's deficit against cuSOLVER, so the emulated route takes 0.40 of cuSOLVER's time. On the H100 the emulated route's gain over cuSOLVER (0.64) comes from the blocked algorithm, not from emulation: native blocked inversion alone takes 0.44 of cuSOLVER's time, and native FP64 is fastest. Absolute times, transfers, workspace and the cost of the other repairs are in Appendix F.

H. Does the loss reach an application?

Does a loss in the inverse reach a quantity a user reads? Cokriging predicts one spatial variable from several correlated ones and uses the inverse covariance for every prediction

TABLE V
TIME RATIOS ($n = 8192$, $g = 64$): EMULATION WINS WHERE FP64 IS SCARCE, THE BLOCKED ALGORITHM WHERE IT IS PLENTIFUL.

	FP64-rich		FP64-scarce	
time ratio	H100	A100	RTX 4090	L40S
native: equilibrated vs raw	1.06	1.05	1.01	1.01
native, eq., vs cuSOLVER	0.44	0.69	1.31	1.29
q=16 vs native, both eq.	1.46	1.26	0.30	0.30
q=16, eq., vs cuSOLVER	0.64	0.88	0.40	0.39

TABLE VI
RAW FIXED ROUTES MOVE COKRIGING PREDICTIONS; EQUILIBRATION AND ADP DO NOT (H100, $n = 4608$).

route	$\tau = 10^{-1}$	$\tau = 10^{-3}$	$\tau = 10^{-6}$
native, cuSOLVER	1.0×10^{-13}	2.4×10^{-11}	6.5×10^{-9}
EAGER, cuSOLVER	1.7×10^{-13}	2.6×10^{-11}	6.4×10^{-9}
q=14	1.2×10^{-9}	4.2×10^{-8}	1.9×10^{-5}
F55	2.7×10^{-10}	3.3×10^{-8}	1.0×10^{-5}
q=14, eq.	4.7×10^{-13}	2.2×10^{-11}	3.5×10^{-8}
F55, eq.	1.3×10^{-13}	2.3×10^{-11}	1.9×10^{-8}
certified bound, worst fixed	7.9×10^{-7}	7.4×10^{-3}	—

and its variance [7]. Five variables of the California housing data [23], in the units of the file, give a covariance $B_{\text{cov}} \otimes (C + \tau I)$ whose diagonal spans many binades, with C a squared-exponential kernel on location. The nugget $\tau = 10^{-1}$ is the model, 10^{-3} a sensitivity case and 10^{-6} a stress case. The measure is the RMS distance of held-out predictions from a certified long-double reference, as a fraction of the predictor's own held-out RMSE (Appendix G).

The weakest fixed routes move predictions 120 to 33,000 times as far from the reference as native cuSOLVER does, over both sizes and all three nuggets (Table VI shows $n = 4608$). ADP stays at native level, and equilibration stays within a small factor of it. The loss is damped, because each prediction sees the inverse's error only through a projection onto its covariance vector, and the realised share of the worst case falls as the inverse's error grows (Appendix G). At the model's nugget a certified bound keeps the loss below 7.9×10^{-7} of the prediction error. An application with a smaller nugget, or one that reads the inverse's entries directly, as kriging variances do, sees more of it.

VI. LESSONS AND LIMITS

a) For library builders: Three rules follow. Check, for each product of a composite routine, which index its operand scaling falls on, not only how large its error is. Prefer associations that keep the scaling on outer indices, such as Method 1B, or equilibrate before emulated products. LAPACK's reference `dtrtri` already does so for an upper factor: its upper branch on $R = R_M D$ is the transpose of Method 1B on L, and in the emulator model of Section III it is bit-for-bit scale equivariant at every g (5 of 5), with errors at most 6.0×10^{-14}, while its lower branch is Method 2C. Rely on a per-call guard such as ADP only where its estimate sees

the scale excess, as it does here. A guard that looked only at the outer scaling of each product, or a fixed bit budget tuned on well-scaled inputs, would pass every per-product test and still fail here.

b) Limitations: All four GPUs are NVIDIA's; Blackwell GPUs were not available to us, and cuBLAS's ADP is the only shipped emulation guard we know of. Equilibration repairs FIXED 55 to native level only from cuBLAS 13.7 (13.8 on the H100). The closed libraries do not document Proposition 2's hypotheses; they are checked on outputs only. A precision selector built on the enclosure is future work. The application uses one dataset and one covariance family.

VII. Conclusion

Per-product accuracy grades do not compose for blocked triangular inversion, because its product form contracts the diagonal scaling along the inner index of one product. Moving the scaling off that index, or giving the product the bits it consumes, restores accuracy, and where FP64 is scarce the accurate emulated inversion still beats cuSOLVER.

Data availability

An anonymized artifact holds the harness, the retained run records and the scripts behind each number, table and figure, and `supplementary_tables.pdf` with every table in full and the verification record: https://figshare.com/s/bb220965206030883d9e (sha256 `c68e11946d3fe33f51c8006806922fb05c084ea3b758ac3517e47bb4f53fc76f`; the separate witness archives are listed in its README).

Use of Generative AI

Claude Code (Anthropic; Claude Opus 5.5) assisted with writing experiment code and drafting and revising the text; the authors reviewed and revised all of it, and every number in the paper is generated from the result files by a script. ChatGPT (OpenAI; GPT-5.6 Sol) was used for critique of drafts.

References

[1] A. Schwarz, A. Anders, C. Brower, H. Bayraktar, J. Gunnels, K. Clark, R. G. Xu, S. Rodriguez, S. Cayrols, P. Tabaszewski, and V. Podlozhnyuk, "Guaranteed DGEMM accuracy while using reduced precision tensor cores through extensions of the Ozaki scheme," in *Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region (SCA/HPCAsia)*, 2026.

[2] *cuBLAS Library User Guide, Release 13*, NVIDIA Corporation, 2026, sections on floating-point emulation; https://docs.nvidia.com/cuda/cublas/.

[3] K. Ozaki, Y. Uchino, and T. Imamura, "Ozaki scheme II: A GEMM-oriented emulation of floating-point matrix multiplication using an integer modular technique," *The International Journal of High Performance Computing Applications*, 2026.

[4] Y. Uchino, K. Ozaki, and T. Imamura, "High-performance and power-efficient emulation of matrix multiplication using INT8 matrix engines," in *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2025.

[5] Y. Uchino, Q. Ma, T. Imamura, K. Ozaki, and P. L. Gutsche, "Emulation of complex matrix multiplication based on the Chinese remainder theorem," in *ISC High Performance 2026 Research Paper Proceedings (41st International Conference)*. IEEE, 2026, pp. 1–12.

[6] J. Demmel, G. Henry, I. Kozachenko, J. Langou, X. S. Li, J. Riedy, and J. Vanover, "How to grade the accuracy of the BLAS," *ACM Transactions on Mathematical Software*, 2026, published online 12 Nov 2026.

[7] H. Wackernagel, *Multivariate Geostatistics*, 3rd ed. Springer, 2003.

[8] K. Ozaki, T. Ogita, S. Oishi, and S. M. Rump, "Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications," *Numerical Algorithms*, vol. 59, 2012.

[9] H. Ootomo, K. Ozaki, and R. Yokota, "DGEMM on integer matrix multiplication unit," *The International Journal of High Performance Computing Applications*, 2024.

[10] Y. Uchino, K. Ozaki, and T. Imamura, "Performance enhancement of the Ozaki scheme on integer matrix multiplication unit," *The International Journal of High Performance Computing Applications*, 2025.

[11] M. Fasi, T. Mary, and M. Mikaitis, "Sharper bounds on the required fixed-point precision for emulating floating-point matrix multiplication," *ACM Transactions on Mathematical Software*, 2026, published online 16 October 2026, article 3858628; no volume or pages assigned yet.

[12] J. W. Demmel and N. J. Higham, "Stability of block algorithms with fast level-3 BLAS," *ACM Transactions on Mathematical Software*, vol. 18, no. 3, 1992.

[13] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numerische Mathematik*, vol. 108, no. 1, 2007.

[14] J. J. Du Croz and N. J. Higham, "Stability of methods for matrix inversion," *IMA Journal of Numerical Analysis*, vol. 12, no. 1, 1992.

[15] E. Carson and N. J. Higham, "Accelerating the solution of linear systems by iterative refinement in three precisions," *SIAM Journal on Scientific Computing*, vol. 40, no. 2, 2018.

[16] A. Haidar, S. Tomov, J. Dongarra, and N. J. Higham, "Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018.

[17] N. J. Higham and T. Mary, "Mixed precision algorithms in numerical linear algebra," *Acta Numerica*, vol. 31, 2022.

[18] P. Luszczek, V. Gadepally, L. Anderson *et al.*, "Performance and numerical aspects of decompositional factorizations with FP64 floating-point emulation in INT8," in *IEEE High Performance Extreme Computing Conference (HPEC)*, 2025.

[19] E. Carson and Y. Ma, "Adaptive accuracy-driven splitting in Ozaki matrix multiplication and application to block LU factorization," arXiv:2611.03158, 2026.

[20] G. Ballard, A. R. Benson, A. Druinsky, B. Lipshitz, and O. Schwartz, "Improving the numerical stability of fast matrix multiplication," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 4, 2016.

[21] A. van der Sluis, "Condition numbers and equilibration of matrices," *Numerische Mathematik*, vol. 14, 1969.

[22] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. SIAM, 1999.

[23] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, 1997.

[24] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. SIAM, 2002.

[25] IEEE, "IEEE standard for floating-point arithmetic," IEEE Std 754-2019, 2019.

[26] S. M. Rump, "Verification methods: Rigorous results using floating-point arithmetic," *Acta Numerica*, vol. 19, 2010.

Appendix A
Setup and evidence

a) Platforms and libraries: All four GPUs run the same pinned Python wheels (nvidia-cublas 13.8.1.7, nvidia-cusolver 12.3.5.3, cudart 13.4.92); the H100 also runs a CUDA 13.2 module's libraries. Drivers, clocks, power limits and NumPy and SciPy versions are in the supplement (`supplementary_tables.pdf` in the artifact). GEMMul8 v3.5.4 uses its INT8 back end with accurate scaling for $q = 14, 16, 20$ moduli. Host LAPACK is OpenBLAS from

the PyPI wheels, which a harness test checks to be bit-for-bit scale equivariant on every job.

b) Routes: cuSOLVER `potri` runs on LAPACK's factor natively, under ADP with DEFAULT and with EAGER, and EAGER on the equilibrated factor. A blocked TRTRI and LAUUM route every GEMM through one back end. Recursion halves down to 256-wide leaves. LAUUM is either one X^TX GEMM, whose operands are both triangular, or blocked by the same recursion, which leaves no product with zero triangles on both operands; every route uses the single GEMM unless marked "blocked LAUUM". Hybrid routes compute every T, or every X_{21}, by native DGEMM. The solve-based routes use the association of Proposition 2. Both implementations use the lower triangle, as do the cuSOLVER routes, and perform each update and negation as a separate operation (NumPy; `cublasDgeam`).

c) Inputs: The scaled class is DMD with $M = W^TW/n + I$ for $W \in \mathbb{R}^{n\times n}$ with independent standard normal entries and k_i uniform on $\{0,\dots,g\}$. Seed 0 covers the grid of Table I at $n \in \{2048, 4096, 8192\}$; seeds 1 to 4 repeat $g \in \{0, 16, 32, 64\}$ at $n = 4096$ and $g = 64$ at $n = 8192$; the breadth runs add $n = 16384$. Ordering controls reorder the same exponents. The mixed-units class scales the rows and columns of M by unit factors between 10^{-6} and 10^6, not powers of two; its diagonal spans 2^{80}, about $g = 40$.

d) Reference and metrics: The reference $D^{-1}X_MD^{-1}$ is scale-consistent with the input. Against inverses computed with 60 decimal digits for $n \le 512$ and $g \in \{0, 32, 64\}$, it is within 1.2×10^{-15} normwise and 1.9×10^{-11} componentwise, as is LAPACK's own inverse of A. Elsewhere the reference is LAPACK's inverse of A. $\widehat{X}$ is the symmetric computed inverse. A reference-free check is the scaled inverse residual: with $\widehat{Y} = D\widehat{X}D$, exact for power-of-two D, it is $\|I - \widehat{Y}M\|_F/(\|M\|_F\|\widehat{Y}\|_F)$, formed by native DGEMM on the device for every full route.

e) Timing: Full routes run on the device from the uploaded matrix to the inverse: cuSOLVER's potrf (native), any power-of-two equilibration by `cublasDdgmm`, the inversion and any rescaling. Each route is also timed with the upload of A and the download of the inverse from pageable host memory inside the clock. Device memory is sampled while each route runs once in a fresh process. The H100 ran 3 independent sessions (separate jobs, $n = 8192$, $g \in \{0, 64\}$), and an earlier session gives a further one for the routes it shares; the RTX 4090 ran 3 sessions at $g = 64$ only. The bit-count pointer is not installed in timing runs.

f) The implementation audit: Table VII states what each kind of evidence covers. On the RTX 4090 ($n = 4096$, $g \in \{0, 64\}$) every TRTRI product of 11 device-resident routes was copied back after its call, which leaves L^{-1} bit-identical, and graded with the harness's code: the largest r_B^{up}/k is 0.11 on the device (harness: 0.12), and at $g = 64$ the largest r_A/k on T for 14 moduli is 3.5×10^{13} (harness: 3.5×10^{13}). The TRSM leaves run natively: on handles carrying DEFAULT, EAGER or FIXED 55 a leaf TRSM issues its internal products with the native compute type (4 of 4 configurations) and launches exactly native's kernels (3 of 3). The operations Proposition 2 assumes were tested through their observable input/output equivariance, bit for bit, at $n = 4096$, $g = 64$. Hypothesis (i) held on every product of the solve-based route for every library (224 of 224), while inner scaling changed every emulated product (0 of 160 unchanged).

TABLE VII
EVIDENCE COVERAGE. ROUTES: TIMED ROUTES WHOSE STRONGEST LINK IT IS (H100 / RTX 4090).

evidence	executions and coverage	routes
product grades	harness, $n = 4096$, H100: every output entry of every TRTRI product; LAUUM on one sampled block per run	
full-output grades and enclosure	harness, $n = 4096$, both GPUs, 30 route–spread pairs and 4 hybrids: every output entry of every TRTRI and LAUUM product; certificate from those grades	7 / 4
other inverse errors	harness, all GPUs and sizes: ϵ_X only	
device audit	device-resident, RTX 4090: TRTRI products of 11 routes at $n = 4096$, $g \in \{0, 64\}$	0 / 5
captured timed execution	device-resident, $n = 8192$, $g = 64$: every product of the timed execution copied back and graded; decisive routes also traced call by call	3 / 4
timing	device-resident, $n = 8192$, all GPUs: time, ϵ_X and residual; products not graded	9 / 7

A paired audit ties a failing timed route to its repair on each GPU ($n = 8192$, $g = 64$): FIXED 55 and equilibrated FIXED 55 (one-GEMM LAUUM) on the H100, 14 moduli and equilibrated 14 moduli on the RTX 4090, with every product of the timed execution captured and graded on its full output. LAUUM's product meets $r_A/k \le 520$ on all four routes, and the failing routes' T products miss grade A by up to 3.6×10^{13} (H100) and 3.6×10^{13} (RTX 4090), with inverse errors 6.8×10^{-2} and 7.1×10^{-2} against 5.7×10^{-16} and 2.1×10^{-15} repaired: the loss sits in the T products of the timed execution, as in the harness.

The four decisive routes on each GPU were rerun in separate jobs with every kernel of every GEMM call traced: in 5610 of 5610 traced calls on the RTX 4090 and 3740 of 3740 on the H100 the kernel sequence equals that of the captured execution, and every returned inverse of a route is equal across untraced, timed, traced and captured runs. The supplement gives the full audit and the per-route evidence matrix.

g) A leftover setting: cuBLAS keeps the maximum mantissa bit count of a FIXED configuration on the handle after the setting returns to ADP, and the harness, which uses one handle per process, did not restore it. A dynamic route that followed a FIXED route in the same process therefore ran under a cap of 55 or 79 bits: EAGER at $n = 4096$, $g = 32$ on the H100 has an enclosure of 6.0×10^{-14} in a fresh process and 1.4×10^{-13} after FIXED 55. The harness now restores and asserts the default at every emulated call and records the effective cap per route. Every set from which a printed number of a dynamic route comes was rerun, and the artifact's provenance check fails if a printed number comes from a capped row. The reruns brought EAGER's grades at $g = 32$ and 64 to their level at $g = 0$ and moved errors in the last printed digit at most; the

superseded records and the rerun list are in the artifact.

APPENDIX B
THEORY IN FULL

a) Proposition 1: what the bounds admit: With this paper's $f(k) = k$ the bound admits $|\delta|$ up to 2^{g-52}; at $g = 64$ already $\delta = 0.07$ gives 0.035. Grade A admits at most $f_A(2)\,u\,p$ with $p = 2$ on the same entry, and $(|L^{-1}||L||L^{-1}|)_{31}$, which bounds classical triangular inversion [14], [24], is 4 for every g; a script checks all of this in exact rationals for g from 0 to 64. The proposition concerns what the bound permits, not any library, and the error of L^{-1}; the measurements below concern $Y = X^T X$, which the perturbation reaches at the same order.

b) Proof of Theorem 1: The remainder of (a) beyond first order is $R = -\sum_s \Lambda_s \Delta_{22} L_{21} \Delta_{11} \Gamma_s$. Part (c) claims nothing about how the frame quantities vary with D for a given implementation.

Proof: (a) $X_{21} = -X_{22} L_{21} X_{11}$ and $\widehat{X}_{21} = -\widehat{X}_{22} L_{21} \widehat{X}_{11} + \Phi_s$; substituting $X_{ii} = \widehat{X}_{ii} - \Delta_{ii}$ and $\widehat{X}_{22} L_{21} = \widehat{I} - E_s$ gives (2). Trivially $\Delta = X(L\Delta L)X$, and by (2) and $L_\nu X_\nu = I$ on each node the block (ρ_2, ρ_1) of $L\Delta L$ is $L_{22}\Psi_s L_{11}$ (the identity below). (b) Take absolute values in (2), by induction from the leaves, and in $\widehat{Y} - Y = \widehat{X}^T\Delta + \Delta^T\widehat{X} - \Delta^T\Delta + \mathrm{sym}(G) + \varrho$, where $|\mathrm{sym}(G)| \le \mathrm{sym}(\mathcal{G})$ (not $\mathcal{G}$, unless it is symmetric) and the rounding ϱ of the symmetrization satisfies $|\varrho| \le 2u|\widehat{Y}| + 2^{-1074}$ entrywise: the addition errs by at most a relative u and is exact if its result is subnormal, and the halving errs by at most 2^{-1075}. (c) Substitute $L = DL_M$ and $\widehat{X} = \widehat{X}' D^{-1}$: every D cancels, for instance $B_{22}|L_{21}\widehat{X}_{11}|D_1 = B'_{22}|L_{M,21}\widehat{X}'_{11}|$ and $|\widehat{X}_{22}||L_{21}| = |\widehat{X}'_{22}||L_{M,21}|$. Under grade B on T the allowance is $c\,u\,r_i s_j$ with $r_i = \max_k |\hat{x}'_{ik}|/d_k$ and $s_j = \max_k d_k |l_{M,kj}|$ over $k \in \rho_2$, at most ρ_s times its value at $D = I$. ∎

(2) is the forward-error counterpart of Du Croz and Higham's residual analysis of Method 2B, kept exact and with arbitrary product errors. Appendix C evaluates a sharper form of (b) that keeps Λ_ν and Γ_ν. Three limits of (c) matter. What is free of D is the enclosure of the frame error $D|\widehat{Y} - Y|D$ as a function of the frame's computed quantities; these lie within B' and $\mathcal{E}_s$ of the D-free L_M^{-1} and $L_{M,22}^{-1} L_{M,21}$, so the enclosure is independent of D to first order in the allowances. The unscaled normwise ϵ_X weights the entries by D^{-1} and has no such property. And the condition is sufficient, not necessary; Proposition 1 attains the ρ_s growth, so under grade B alone D cannot be removed.

The identity. Fix a node of the recursion with contiguous index set $\nu = \rho_1 \cup \rho_2$. L and Δ are lower triangular, so the block (ρ_2, ρ_1) of $L\Delta L$ involves only indices in ν and equals $L_{21}\Delta_{11}L_{11} + L_{22}\Delta_{21}L_{11} + L_{22}\Delta_{22}L_{21} = L_{22}(\Delta_{21} + X_{22}L_{21}\Delta_{11} + \Delta_{22}L_{21}X_{11})L_{11}$. Expanding $X_{22}L_{21}X_{11}$ with $X_{ii} = \widehat{X}_{ii} - \Delta_{ii}$ gives $\Delta_{21} = \Phi_s - \widehat{X}_{22}L_{21}\Delta_{11} - \Delta_{22}L_{21}\widehat{X}_{11} + \Delta_{22}L_{21}\Delta_{11}$, which is (2) and, with $\widehat{X} = X + \Delta$ once more, $\Delta_{21} + X_{22}L_{21}\Delta_{11} + \Delta_{22}L_{21}X_{11} = \Phi_s - \Delta_{22}L_{21}\Delta_{11} = \Psi_s$. Every strictly lower entry lies in the block (ρ_2, ρ_1) of exactly one split and every other entry of the lower triangle in one leaf, where the block of $L\Delta L$ is $L_\ell \Delta_\ell L_\ell$. Hence $\Delta = X(L\Delta L)X = \sum_\nu X_{:,\rho_2} L_{22} \Psi_\nu L_{11} X_{\rho_1,:}$. Both identities are polynomial; a script checks (2) symbolically and both as exact rational equalities on 66 random cases.

The bound that is evaluated. $\widehat{Y} - Y = X^T\Delta + \Delta^T X + \Delta^T\Delta + \mathrm{sym}(G)$ and $X^T\Delta = \sum_\nu (X^T\Lambda_\nu)\Psi_\nu\Gamma_\nu$. With $X = \widehat{X} - \Delta$ and $|\Delta| \le B$ from (3),

$$
\begin{aligned}
|X^T\Lambda_\nu| &\le |\widehat{X}^T\widehat{X}_{:,\rho_2}L_{22}| + |\widehat{X}|^T B_{:,\rho_2}|L_{22}| \\
&\quad + B^T\big(|\widehat{X}_{:,\rho_2}L_{22}| + B_{:,\rho_2}|L_{22}|\big) =: \mathcal{Y}_\nu, \\
|\Gamma_\nu| &\le |L_{11}\widehat{X}_{\rho_1,:}| + |L_{11}|B_{\rho_1,:} =: \bar{\Gamma}_\nu, \\
|\Psi_s| &\le \mathcal{E}_s|\widehat{X}_{11}| + \mathcal{F}_s + B_{22}|L_{21}|B_{11} =: \bar{\Psi}_s,
\end{aligned}
$$

and $\bar{\Psi}_\ell = B_\ell$, so that

$$
\begin{aligned}
|\widehat{Y} - Y| \le{}& \textstyle\sum_\nu \big(\mathcal{Y}_\nu \bar{\Psi}_\nu \bar{\Gamma}_\nu + (\cdot)^T\big) + B^T B \\
&+ \mathrm{sym}(\mathcal{G}) + 2u|\widehat{Y}| + 2^{-1074}\mathbf{1}\mathbf{1}^T,
\end{aligned}
$$

the last two terms covering the rounding of the symmetrization, as in Theorem 1(b). Without the absolute term the bound is false: for $\widehat{X} = X = I_2$, $B = 0$ and an output error G with off-diagonal entries 2α and α, $\alpha = 2^{-1074}$, the halving rounds $3\alpha/2$ to 2α, which exceeds $\mathrm{sym}(\mathcal{G})_{12} + 2u|\widehat{Y}_{12}|$; the self-tests check this example. The evaluator enters the absolute term raised to its floor $\tau = 2^{-500}$, a larger and therefore valid allowance. The symmetrization of the allowance matters: for $\widehat{X} = I_2$ and $G = e_1 e_2^T$, $\mathrm{sym}(G)$ has the entry $1/2$ in position $(2,1)$, where the envelope $|G|$ is zero. The allowance used here, $\min(r_A^{\mathrm{up}} u|\widehat{X}|^T|\widehat{X}|, r_B^{\mathrm{up}} u\, cc^T)$ with c the column maxima of $|\widehat{X}|$ and one pair of ratios for the whole product, is symmetric in exact arithmetic; the code nevertheless replaces whatever allowance it is given by $\max(\mathcal{G}, \mathcal{G}^T) \ge \mathrm{sym}(\mathcal{G})$ and asserts the result symmetric (largest relative asymmetry of the computed allowance before that step: 0). The terms of first order in the allowances keep $X^T\Lambda_\nu$ and Γ_ν formed before the absolute value, with the run's $\widehat{X}$ in place of the exact inverse; B enters at second order only.

Leaves. With $N \ge |\widehat{X}_\ell L_\ell - I|$ and $\|N\|_\infty = \nu_\ell < 1$, $\Delta_\ell = (\widehat{X}_\ell L_\ell - I)(\widehat{X}_\ell - \Delta_\ell)$ gives $|\Delta_\ell| \le (I - N)^{-1} N|\widehat{X}_\ell| \le N|\widehat{X}_\ell| + \frac{\nu_\ell}{1-\nu_\ell}(Ne)(e^T|\widehat{X}_\ell|) =: B_\ell$. The residual is formed in long double and N includes its rounding, $\gamma^{ld}_{n_b+1}(|\widehat{X}_\ell||L_\ell| + I)$; in the runs below $\nu_\ell \le 4.6 \times 10^{-16}$.

c) Proposition 2 in full: In Method 1B each update forms $P = L_b Z_1$ by GEMM with $\beta = 0$ and then subtracts, $W_2 - P$, as a separate operation; the leaves are native TRSM of width n_b. The full statement: Run the solve-based recursion on L_M and on DL_M, D a power-of-two diagonal, with the same partition and the same sequence of operations, in one binary floating-point format and rounding mode [25] (binary64, round to nearest, in the tested libraries) with no trap or flag altering control flow, the entries of D and DL_M being exact. Assume in both runs that (i) every product is computed by an outer-equivariant G; (ii) each update and each negation is one IEEE operation per entry, neither fused or contracted with the product it follows nor reassociated; (iii) the leaves are equivariant: the inversion returns $\widehat{X}(L_a)D_a^{-1}$ for $D_a L_a$, and the solve returns $\widehat{Z}(L_a, W)C$ for $(D_a L_a, D_a W C)$, D_a, C

power-of-two diagonal; (iv) every routine is deterministic and takes the same computational path in both runs, with the same order of operations inside every reduction, and no operation, intermediate ones and the symmetrization included, overflows, underflows, returns a subnormal or raises another exception. Then $\widehat{X}(DL_M) = \widehat{X}(L_M)D^{-1}$ bit for bit, and LAUUM on G with symmetrization returns $D^{-1}\widehat{Y}(L_M)D^{-1}$. If also (v) the computed Cholesky factor of DMD is D times that of M, the computed inverse of DMD is D^{-1} times that of M times D^{-1}, bit for bit (± 0 equal).

Proof: By induction on the recursion; the leaves are (iii). If $\widehat{X}_{ii} = \widehat{X}_{M,ii}D_i^{-1}$, then $W = G(D_2L_{M,21}, \widehat{X}_{M,11}D_1^{-1}) = D_2W_MD_1^{-1}$ by (i). In the solve, a leaf maps $(D_aL_{M,a}, D_aW_{M,a}D_1^{-1})$ to $\widehat{Z}_{M,a}D_1^{-1}$ by (iii); an update forms $G(D_bL_{M,b}, \widehat{Z}_{M,a}D_1^{-1}) = D_bP_MD_1^{-1}$, and a correctly rounded subtraction or negation of operands sharing power-of-two factors returns the factors times the unscaled result under (ii) and (iv), so $\widehat{X}_{21} = \widehat{X}_{M,21}D_1^{-1}$. LAUUM's $G(\widehat{X}^T, \widehat{X})$ carries D^{-1} on outer indices only, and $(Y + Y^T)/2$ commutes with the scaling. (v) closes the pipeline. ∎

(i), (iii), (iv) and (v) are properties of library routines that no documentation guarantees; Appendix A checks them on outputs, bit for bit, which is evidence for the tested versions, configurations and inputs and not a guarantee for others. Under equivariance $\widehat{Y}(DMD) - (DMD)^{-1} = D^{-1}(\widehat{Y}(M) - M^{-1})D^{-1}$, so every entry's relative error, and the error measured in M's frame, equal those of M's inversion, whatever D; the normwise ϵ_X need not be constant: for $M = I$, an error $\eta e_1 e_1^T$ in M's inverse and $D = \mathrm{diag}(1, 2^g, 2^g)$, it rises from $|\eta|/\sqrt{3}$ to nearly $|\eta|$. The product form has no such property, because outer equivariance says nothing about a scaling shared along the contracted index. In the emulator run of Section III the solve with L_{11} is native, and with native products both Method 2C and the product form are bit-for-bit scale equivariant and accurate (10 of 10 runs).

d) The triangle: LAPACK here means the reference `dtrtri`, characterized by its source-level association; nothing is claimed about vendor implementations, which were not inspected. For both values of `uplo`, the reference `dtrtri` multiplies the already inverted block into the triangle's off-diagonal block (`DTRMM`) and then solves with the diagonal block (`DTRSM`), contracting the triangle's row index; it never calls `DGEMM`, so it is exposed only where those two routines are themselves emulated. The lower factor $L = DL_M$ carries D on its rows, the upper factor $R = R_MD$ on its columns. Section VI states the upper branch's result in the emulator model of Section III, with the triangular product emulated and the solve and the leaves native; it is a statement about the reference association in that model. It does not make `uplo = 'U'` a repair: the contracted index also depends on how a library forms its triangular products and solves and how it inverts its diagonal blocks, and a left triangular solve against the identity, for instance, contracts the triangle's column index, which is the scaled one for R. Compiled LAPACK and cuSOLVER with the upper triangle were not measured in this paper.

e) A first-order counterfactual: Before the enclosure, the first-order terms were evaluated with LAPACK's binary64 inverse in place of the exact one and the H100's graded ratios (LAUUM's from its sampled block). That value is not a bound. It exceeds the measured ϵ_X in 30 of 30 pairs, by 23 to 9,880 times. Assigning native DGEMM's graded ratios to the products that a replacement plan computes natively, without rerunning, it is, over 38 plans, at most 1.2×10^{-13} for every plan measured at native level and at least 8.7×10^{0} for every other; for 14 moduli at $g = 64$ it falls from 8.0×10^{1} to 1.2×10^{-13} with every T native and stays at 8.0×10^{1} with every X_{21} native. This orders replacement plans without running them; it certifies nothing.

APPENDIX C
GRADING AND CERTIFICATION

a) Grading: Table VIII grades the blocked inversion's products at $n = 4096$: all 900 TRTRI products over all routes and g, on every output entry, and LAUUM's $n \times n$ product on one randomly sampled 512×512 output block per run (30 blocks). The reference is a sequential x87 long-double product, whose error the next paragraph bounds by $\gamma_k^{ld} p_{ij}$. The grading therefore bounds the true ratio entry by entry, $r_B^{\mathrm{up}} = \max_{ij} (|\hat{c}_{ij} - c_{ij}^{ld}| + \gamma_k^{ld} p_{ij})/(u r_i s_j)$, and r_A from above and below; an independent outward-rounded script recomputes these bounds to within 8.7×10^{-16}. No classification depends on the reference's own error (0 grade-B and 0 grade-A verdicts change), nor on doubling the allowance to γ_{2k}^{ld} (0 and 0). Maximized over all ten routes, all three g and every graded entry, r_B^{up}/k is at most 0.052 on the 450 products T, 0.12 on the 450 products X_{21} and 0.019 on the 30 sampled LAUUM blocks. A second run grades every output entry of every product, LAUUM's $n \times n$ product included, with the doubled allowance: r_B^{up}/k is at most 0.053, 0.12 and 0.033 on the H100 and 0.05, 0.11 and 0.032 on the RTX 4090 (0 and 0 products fail). The ratios are rounded outward at every step in long double, the conversion to binary64 included, so a stored ratio bounds the true one. The ratios are taken over every graded output entry, zeros included; an entry whose right-hand side is zero counts as 0 if its error is zero and fails otherwise.

b) The reference's rounding model: The reference is NumPy's long-double matrix product, which has no BLAS behind it: a compiled loop that, per entry, adds the k products one by one to a sum that starts at zero. On x86-64 the type is the 80-bit x87 format, whose multiply and add each round once to a 64-bit significand, and x87 has no fused multiply-add, so no contraction is possible. A term's relative perturbation is therefore a product of at most k factors $1 + \delta$, $|\delta| \le u_{ld}$ (one for its product, at most $k - 1$ for the additions after it), and the error is at most $\gamma_k^{ld} p_{ij}$ [24, Section 3.1], negligible in r_A/k units (at most 4.9×10^{-4}) but not, for generic operands, in r_B/k units at large k: the standard bound for rounded products followed by rounded additions is already γ_k, in any order of summation, and a fused accumulation has

the same bound. A script compares the library product bit for bit with that explicit sequence in long-double scalars and with a correctly rounded fused accumulation built in rationals: 540 of 540 entries equal the former, and the two models differ on 219 of them; the largest error is 0.37 of $\gamma_k^{ld} p_{ij}$. The model depends on the platform and the build: the type must be the x87 80-bit format, the loop must not be vectorized, reassociated or evaluated in a wider or narrower precision, and the x87 precision control must be at its 64-bit default. The grading code stops unless the type has a 64-bit significand, and each full-output job first runs the bitwise comparison on its own host and build and stops, grading nothing, if a single entry differs (18 jobs on 8 hosts, 9,720 of 9,720 entries equal). On a platform where long double is binary64 or binary128, grading therefore fails closed.

c) The componentwise error: The componentwise error is $\max |\hat{x}_{ij} - x_{ij}|/|x_{ij}|$ over the entries whose scaled value $|d_i d_j x_{ij}|$ exceeds 10^{-8} of the largest, so that no entry below roundoff in its own scale, and no zero, enters. On a transparent CPU Ozaki-I emulator (7-bit slices, exact integer slice products, $n = 2048$) the largest TRTRI e_{99} rises from 8 at $g = 0$ to 67 at $g = 64$, the scaling term carrying the rise.

d) The certificate: Each product's error is bounded entrywise by $\min(r_A^{\mathrm{up}} u|A||B|, r_B^{\mathrm{up}} u\, rs^T)$ with its own graded ratios, so the bound of Appendix B can be evaluated on a run's own operands: the factor L the route was given and its computed leaves, products and inverse, with the leaves' errors enclosed from their residuals. The evaluator runs in binary64, round to nearest. It returns a certificate only if every operation was evaluated under the assumptions of the error bound used for it; otherwise it refuses and returns no number.

- *Invariant.* Every nonnegative matrix it forms is checked and normalized before further use: it refuses on an entry that is NaN, infinite, negative or above 2^{500}, and raises nonzero entries below $\tau = 2^{-500}$ to τ, which is sound because every bound is monotone in each nonnegative operand. Signed inputs must lie in $[2^{-480}, 2^{480}]$, so no multiplication inside a matrix product underflows and nothing overflows.
- *Enlarged operations.* Every product and sum of nonnegative operands is multiplied by a factor that covers its rounding for any summation order, with or without fused operations [24]; an absolute product $|AB|$ is enclosed by $|\mathrm{fl}(AB)|$ plus $\gamma_k |A||B|$; the allowance $\min(r_A^{\mathrm{up}} u|A||B|, r_B^{\mathrm{up}} u\, rs^T)$ carries a margin of $1 + 8u$.
- *Steps.* (1) leaf bounds from long-double residuals, refused unless $\nu_\ell < 1/2$; (2) splits, children first, by (3); (3) the sum over splits and leaves of the sharper form above; (4) LAUUM's allowance, symmetrized, with the rounding of the symmetrization; (5) the two Frobenius norms, $\beta \geq \|W\mathcal{B}W\|_F$ and $\eta \leq \|W\widehat{Y}W\|_F$ in the route's output scaling W, each formed as an exact sum of exact squares with a directed square root; (6) $\beta/(\eta - \beta)$ in exact rationals.

The status is *certified* when that quotient is below 1, *no relative bound* when $\eta \leq \beta$, and *refused* on any failed check. By the invariant every enlargement is applied to an operation that satisfies the standard model, so by induction every normalized matrix dominates the exact quantity it stands for; steps 5 and 6 add no rounding. The claim rests on IEEE binary64 arithmetic in the BLAS and NumPy with conventional products, on the x87 model of the grading reference, and on the route forming its products from the operands the harness passes it; the residual argument of step 1 is the standard one of verification methods [26]. The supplement gives the evaluator operation by operation, a reference procedure and test vectors.

e) How the evaluator was checked: In exact rationals, on 66 random triangular systems with diagonal spreads up to 2^{40} and product errors up to 0.2 of the grade-B right-hand side, the evaluated bounds dominate $|\Delta|$ and $|\widehat{Y} - Y|$ entrywise in every case (66 of 66). On a CPU Ozaki-I emulator (7 cases) the bounds dominate in every case and exceed the actual error by 40 to 402 times. Adversarial tests inject a NaN, an infinity or a negative value into each of the 150 intermediates (444 injections, every one refused), and both evaluators pass every one of 94 test vectors with exact expected outcomes (0 misses). An independent second evaluator, sharing no code, agrees in status and bound on 34 of 34 pairs on the H100 and 34 of 34 on the RTX 4090 (largest relative difference 3.9×10^{-9}). At $n = 64$ the operands of 42 runs on the H100 and 42 on the RTX 4090 were stored and every bound recomputed in exact integer arithmetic: the exact bound dominates the true error entrywise, and the evaluator's status equals the exact one (42 and 42 runs without a discrepancy). Five $n = 4096$ runs and six drawn by a preselected rule were rerun with every GPU output stored and rebuilt on a CPU from those outputs alone: 6 of 6 drawn runs and all five witnesses reproduced their status and bound bit for bit, and on an independent x86-64 host 5 of 5 witness statuses reproduced. A preflight on the evaluating host (6 checks) gates every certificate. The archives, their sha256 values and the selection rules are in the artifact's README and the supplement.

f) Results: Table VIII lists the enclosure for every H100 pair. No pair was refused and none has a finite relative bound of 1 or more: the 4 pairs without a certificate all fail the denominator test, with β/η from 1.3 to 3.6×10^{17} on the H100. In M's frame the native route's enclosure is $6.5{\times}10^{-13}$, $6.5{\times}10^{-13}$ and $6.5{\times}10^{-13}$ at $g = 0$, 32 and 64. The first-order part alone, with no leaves, no B and no rounding terms, is within 1.8% of the enclosure wherever the enclosure is below 1.0×10^{-6}; the leaves contribute at most 2.7×10^{-15}. Above that threshold the matrix B, which takes absolute values at every level of the recursion, makes the second-order terms dominate: FIXED 79 at $g = 64$ is certified at 272% above its first-order part, and 20 moduli at $g = 64$, whose first-order part is below 1, has no relative bound.

Appendix D
Results in full

a) Grades and enclosure: Table VIII gives all ten routes at three spreads; grading every entry of LAUUM too gives a B-type ratio of at most 0.12; native and ADP products meet

$r_A/k \le 8.7 \times 10^{-2}$ at every g. Over the 9 route–spread pairs whose worst A-type ratio over the TRTRI products exceeds 100, the inverse's error is 0.11 to 18 times u times that ratio. With every T native the enclosure is at most 1.5×10^{-13}; with every X_{21} native there is no relative bound.

b) Four GPUs: Table IX gives every route on the four GPUs and at $n = 16384$. The bits lost per unit of g range from 0.76 for 14 moduli to 0.96 for FIXED 79, and the onset is consistent with the single-product analysis [11]; more bits lower the error, FIXED 111 to within a small factor of native at $g = 64$. The slopes on the H100 differ from the RTX 4090's by at most 9.4×10^{-5} bits per unit of g, and over seeds 0 to 4 at $n = 4096$ by at most 0.01 (H100) and 0.01 (RTX 4090). Seeds 1 to 4 at $n = 8192$, $g = 64$ give 7.1×10^{-2} to 7.6×10^{-2} for 14 moduli and 2.7×10^{-16} to 2.9×10^{-16} for equilibrated FIXED 55 on the H100; the other GPUs are alike (supplement). In contiguous chunks of exponents 16 moduli give 9.9×10^{-4}, and EAGER stays at native accuracy sorted and in chunks (4.9×10^{-15}, 3.5×10^{-15}).

Two further checks agree. A normwise audit of grade-C type passes on every TRTRI product at $n = 4096$, $g = 64$ ($\|C - C_h\|_F \le \beta u \|A\|_F \|B\|_F$ with $\beta \le 0.55$ against the host's DGEMM, whose bound is about $\beta = k$). The scaled residual, which needs no reference, separates the routes exactly as ϵ_X does: at most 5.9×10^{-17} for every accurate full route and at least 1.9×10^{-3} for every failing one on the H100 (5.9×10^{-17} and 1.9×10^{-3} on the RTX 4090). Bit-for-bit scale equivariance is a reference-free symptom: native routes passed all 30 checks on each GPU; every raw fixed route with $g > 0$ failed on the RTX 4090, and all but 1 (route, input) pair on the H100. EAGER routes failed too (all 15 cases with $g > 0$) while accurate, so a failure flags a changed path, not an error.

Where the difficulty is conditioning rather than scaling, emulation tracks native. With $\kappa = 10^8$, potrf followed by potri differs from LAPACK by 5.3×10^{-9} natively and 5.3×10^{-9} under EAGER on the RTX 4090 (5.4×10^{-9} and 5.4×10^{-9} on the H100), a factor-level difference of order κu; the Gaussian-process kernel gives 3.1×10^{-7} both ways. On the H100's second version, cuBLAS 13.3.0 and cuSOLVER 12.1.0, every cuSOLVER error is within 1% of the pinned version's.

c) Replacement and re-association: Table X extends Fig. 1(c) with $n = 8192$, LAUUM and the 16 products of largest and smallest excess. Leaving any single T emulated, with all others native, still gives 1.9×10^{-3} or worse at $n = 8192$. On a transparent CPU emulator ($n = 2048$, 8 slices) the solve-based inverse of DMD is bit-for-bit D^{-1} times its inverse of M times D^{-1} at every tested g (5 of 5), with errors from 4.2×10^{-14} to 6.0×10^{-14} where the product form reaches 8.9×10^{-2}. Through the harness at $n \in \{4096, 8192\}$ and g from 16 to 64 it is bit-for-bit scale equivariant in 8 of 8 cases with 14 moduli and 8 of 8 with FIXED 55 on the H100, likewise on the RTX 4090 (8 and 8 of 8), and so is every complete device pipeline from cuSOLVER's potrf (5 of 5; product form: 0 of 2). On the H100 its error on the raw input never exceeds 5.5×10^{-15} and 9.8×10^{-16}, where the product form reaches 7.1×10^{-2}; on mixed units it stays at most 2.7×10^{-15}.

d) Real matrices and library versions: The 16 SuiteSparse matrices are those with $1024 \le n \le 8192$ and diagonal span at least 2^{20} that the NIST Matrix Market mirrors, chosen by a rule fixed before any matrix was seen; the covariances use the 1990 California census (seven variables) and the UCI air-quality data (twelve), in raw units. Table XI gives the group maxima. Equilibrated FIXED 55 stays within a factor 2.2 of native on every matrix. Equilibrated 16 moduli stay within a factor 1.0 of native on every matrix; equilibrated 14 moduli stay 138 times native on bcsstk23 on the A100 and at most 10 times on every other matrix (9 on the A100). Over the 9 releases the guarded routes' largest factor over native is 1.01 on the L40S, 1.00 on the A100 and 1.00 on the H100. Equilibrated FIXED 55 reaches native level from cuBLAS 13.7 (2.9×10^{-16} on the L40S) and stops at 1.3×10^{-14} before it; on the H100 the change arrives in 13.8 (1.3×10^{-14} under 13.7, 2.9×10^{-16} under 13.8). cuSOLVER's `potri` under its own FIXED control gives 1.2×10^{-2}, 1.7×10^{-7} and 9.8×10^{-16} with 55, 79 and 111 bits on the L40S and 1.2×10^{-2} with 55 on the RTX 4090; equilibrated, 6.7×10^{-16}. The supplement gives every matrix and every release.

APPENDIX E
THE GUARD: ESTIMATES, SWITCH AND DEFAULT

a) An explanatory proxy: For $C = AB$ with $c_{ij} \neq 0$ and r_i, s_j, p_{ij} as in Section II, the excess $e_{ij} = \log_2(r_i s_j / |c_{ij}|)$ is the sum of $\log_2(r_i s_j / p_{ij})$, the ratio of the grade-B to the grade-A right-hand side, and $\log_2(p_{ij}/|c_{ij}|)$, ordinary cancellation. The harness records its 99th percentile e_{99} and maximum per product. It needs the product, so it explains a library's choice after the fact and is not an online estimate. The fit of Fig. 3 covers all 636 TRTRI products of the scaled grid on both GPUs; at $n = 8192$ they report 68 to 75 bits at $g = 0$. The figure plots every TRTRI product under EAGER at $n \in \{2048, 4096, 8192\}$, $g \in \{0, 16, 32, 48, 64\}$, raw and equilibrated, on the RTX 4090; the H100's estimates are identical. Componentwise, no ADP route's error exceeds its native counterpart's by more than a factor of 1.95 (RTX 4090) or 1.56 (H100) over the grid.

b) Where a product switches path: cuBLAS writes its bit estimate to an output pointer the caller provides. The pointer holds the estimate whether or not the product emulates (cuBLASLt documents -1 when emulation is not used), so dispatch has to be read from outputs, time and kernels. The probes of Fig. 4 compare each output bit for bit with native DGEMM. The estimate is 62 bits at zero span and 257 at a span of 200 bits on both GPUs. On the RTX 4090 with cuBLAS 13.8.1, outputs of 1024×1024 or smaller step between 107 and 109 bits (for example from 5.59 to 6.51 times native), 2048×2048 outputs between 107 and 110 bits (from at most 0.98 to at least 1.60 times native). Above the step, outputs of 2048×2048 and larger are bit-identical to native DGEMM on the RTX 4090; smaller ones are not, so the path above the step is not always the default DGEMM kernel. On the

TABLE VIII

GRADES OF THE TRTRI PRODUCTS, THE INVERSE'S ERROR AND THE CERTIFIED ENCLOSURE OF $\|\widehat{Y} - (LL^T)^{-1}\|_F / \|(LL^T)^{-1}\|_F$ (H100, $n = 4096$). ON X_{21}, $r_A/k \leq 1.2 \times 10^1$ THROUGHOUT. N.R.B.: NO RELATIVE BOUND.

		$g = 0$			$g = 32$			$g = 64$		
route	r_B/k, all g	r_A/k on T	ϵ_X	enclosure	r_A/k on T	ϵ_X	enclosure	r_A/k on T	ϵ_X	enclosure
native	1.1×10^{-1}	1.5×10^{-2}	2.3×10^{-15}	6.5×10^{-13}	1.5×10^{-2}	2.2×10^{-15}	1.5×10^{-13}	1.5×10^{-2}	2.0×10^{-15}	1.1×10^{-13}
DEFAULT	1.1×10^{-1}	1.5×10^{-2}	2.3×10^{-15}	6.5×10^{-13}	1.5×10^{-2}	2.2×10^{-15}	1.5×10^{-13}	1.5×10^{-2}	2.0×10^{-15}	1.1×10^{-13}
EAGER	7.1×10^{-3}	7.2×10^{-3}	2.1×10^{-15}	2.3×10^{-13}	9.1×10^{-3}	2.0×10^{-15}	6.0×10^{-14}	1.3×10^{-2}	2.0×10^{-15}	6.6×10^{-14}
q=14	5.3×10^{-2}	2.8×10^{1}	3.0×10^{-15}	2.9×10^{-12}	1.1×10^{9}	8.1×10^{-8}	3.1×10^{-5}	3.5×10^{13}	7.1×10^{-2}	n.r.b.
q=16	3.9×10^{-3}	1.0×10^{-1}	3.1×10^{-16}	3.0×10^{-13}	3.0×10^{6}	3.8×10^{-10}	2.0×10^{-7}	3.5×10^{13}	6.7×10^{-2}	n.r.b.
q=20	3.9×10^{-3}	3.9×10^{-3}	3.1×10^{-16}	1.5×10^{-13}	1.5×10^{2}	1.2×10^{-14}	4.8×10^{-11}	1.7×10^{12}	2.3×10^{-5}	n.r.b.
F55	1.1×10^{-2}	4.3×10^{0}	7.3×10^{-16}	5.0×10^{-13}	5.6×10^{8}	2.3×10^{-8}	8.5×10^{-6}	3.5×10^{13}	6.9×10^{-2}	n.r.b.
F79	9.1×10^{-3}	8.7×10^{-3}	3.3×10^{-16}	2.2×10^{-13}	3.0×10^{1}	1.4×10^{-15}	1.6×10^{-11}	2.5×10^{11}	3.1×10^{-6}	2.6×10^{-2}
q=14, eq.	5.1×10^{-2}	2.8×10^{1}	3.0×10^{-15}	2.9×10^{-12}	2.8×10^{1}	1.1×10^{-15}	6.5×10^{-13}	2.8×10^{1}	1.0×10^{-15}	5.0×10^{-13}
F55, eq.	1.0×10^{-2}	4.3×10^{0}	7.3×10^{-16}	5.0×10^{-13}	4.3×10^{0}	2.5×10^{-16}	1.1×10^{-13}	4.3×10^{0}	2.0×10^{-16}	8.6×10^{-14}
q=14, T native									9.9×10^{-16}	1.5×10^{-13}
q=14, X_{21} native									7.1×10^{-2}	n.r.b.
F55, T native									2.0×10^{-16}	7.5×10^{-14}
F55, X_{21} native									6.9×10^{-2}	n.r.b.

TABLE IX

ϵ_X AT $g = 64$, $n = 8192$, ON FOUR GPUS, AND AT $n = 16384$ (H100). SLOPE: H100; UNITS: MIXED UNITS (RTX 4090); ADP: WORST GUARDED / NATIVE RATIO OVER EACH GPU'S GRID.

route	slope	H100	A100	RTX 4090	L40S	n=16384	units
native	0.00	2.9×10^{-15}	2.9×10^{-15}	2.8×10^{-15}	2.9×10^{-15}	4.8×10^{-15}	3.4×10^{-15}
q=14	0.76	7.1×10^{-2}	7.1×10^{-2}	7.1×10^{-2}	7.1×10^{-2}	7.4×10^{-2}	1.1×10^{-4}
q=16	0.87	6.7×10^{-2}	6.7×10^{-2}	6.7×10^{-2}	6.7×10^{-2}	7.0×10^{-2}	6.1×10^{-7}
q=20	0.96	4.7×10^{-5}	4.7×10^{-5}	4.7×10^{-5}	4.7×10^{-5}	8.3×10^{-5}	1.9×10^{-11}
F55	0.81	6.8×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	7.1×10^{-2}	2.3×10^{-5}
F79	0.96	4.1×10^{-6}	4.1×10^{-6}	4.1×10^{-6}	4.1×10^{-6}	5.9×10^{-6}	1.4×10^{-12}
F111	0.00	3.3×10^{-14}	3.3×10^{-14}	3.3×10^{-14}	3.3×10^{-14}	–	2.6×10^{-16}
q=14, eq.	0.00	2.0×10^{-15}	2.0×10^{-15}	2.0×10^{-15}	2.0×10^{-15}	3.9×10^{-15}	2.8×10^{-15}
q=16, eq.	0.00	2.3×10^{-16}	2.2×10^{-16}	2.1×10^{-16}	2.3×10^{-16}	–	2.5×10^{-16}
F55, eq.	0.00	2.9×10^{-16}	2.7×10^{-16}	2.7×10^{-16}	2.9×10^{-16}	3.5×10^{-16}	4.5×10^{-16}
F79, eq.	0.00	2.4×10^{-16}	2.3×10^{-16}	2.3×10^{-16}	2.4×10^{-16}	–	2.7×10^{-16}
ADP		1.05×	1.03×	1.02×	1.02×		

TABLE X

ϵ_X WITH CHOSEN PRODUCTS NATIVE (H100, $g = 64$). TOP/BOTTOM 16: LARGEST/SMALLEST SCALE EXCESS.

	GEMMul8 $q = 14$		FIXED 55	
computed natively	n=4096	n=8192	n=4096	n=8192
none	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
LAUUM	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
every X_{21}	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
bottom 16	7.0×10^{-2}	7.1×10^{-2}	6.8×10^{-2}	6.8×10^{-2}
top 16	9.9×10^{-16}	1.6×10^{-2}	2.0×10^{-16}	1.6×10^{-2}
every T but one	$\geq 3.7 \times 10^{-3}$	$\geq 1.9 \times 10^{-3}$	$\geq 3.7 \times 10^{-3}$	$\geq 1.9 \times 10^{-3}$
every T	9.9×10^{-16}	2.0×10^{-15}	2.0×10^{-16}	2.9×10^{-16}

H100, Nsight Systems traces show the same kernels launched on both sides of the switch, with the durations moving: for the 4096^3 product the INT8 GEMMs take 2,350 μs per call at 103 bits and 47.2 μs at 107. Above the switch a CUTLASS FP64 tensor-core GEMM of the emulation library does the product, not cuBLAS's default FP64 kernel, which accounts for over-cap outputs that are accurate but not bit-identical to native DGEMM. The 4096^3 product takes 1.69 times native at 103 bits and 2.17 at 107, which places the H100's switch where the RTX 4090's is. Exact zeros on both operands, as in a single-GEMM LAUUM, also change the path: the 8192^3 product returns native's output at 1.16 times native time on the RTX 4090.

c) Inside cuSOLVER and under DEFAULT: cuSOLVER 12.3.5 imports eleven FP64 level-3 entry points from cuBLAS, and an `LD_PRELOAD` interposer wraps all of them, logging each internal call's operation, shape, strategy and estimate. With all eleven wrapped, potri at $n = 4096$ and $g = 64$ makes 108 internal FP64 level-3 calls in every mode; cuBLAS's own API log of the same call, taken without the interposer, records 108, 108 and 108 under native, DEFAULT and EAGER, the interposer's count operation for operation. At $n = 8192$, $g = 64$, the internal TRSM calls report 66 to 129 bits under EAGER and TRMM up to 125, identically on both GPUs; none of the 63 SYRK calls is emulated, as documented below compute capability 10.0. DEFAULT, the strategy a caller gets without asking for EAGER, decides for itself. On the RTX 4090 it writes an estimate on 500 of the 1090 GEMMs issued by the blocked routes: 421 at or below 103 bits, emulated at every tested output size, 59 at or above 110 bits, native

TABLE XI
REAL SPD MATRICES: LARGEST ERROR PER GROUP IN THE EQUILIBRATED FRAME (L40S; LAST COLUMN A100). SPAN: $\log_2$ OF THE DIAGONAL SPAN. ADP: CUSOLVER EAGER.

						equilibrated			
matrices	span	native	ADP	q=14	F55	q=14	q=16	F55	q=14, A100
census (4)	32	2.9×10^{-15}	1.7×10^{-15}	1.5×10^{-11}	3.7×10^{-12}	8.4×10^{-15}	1.2×10^{-15}	2.3×10^{-15}	8.3×10^{-15}
air (4)	20	1.4×10^{-15}	9.9×10^{-16}	1.6×10^{-13}	6.3×10^{-14}	3.0×10^{-15}	5.9×10^{-16}	7.9×10^{-16}	3.0×10^{-15}
SuiteSparse (16)	23–44	3.6×10^{-14}	1.2×10^{-14}	1.7×10^{-12}	3.2×10^{-13}	2.7×10^{-13}	2.9×10^{-14}	3.4×10^{-14}	2.6×10^{-13}
bcsstk23	35	2.5×10^{-15}	3.1×10^{-15}	1.7×10^{-12}	5.1×10^{-14}	2.7×10^{-13}	2.5×10^{-15}	5.4×10^{-15}	2.6×10^{-13}

at every size, and 20 in between, unclassified; the path is inferred from the estimate against the switch. One complete RTX 4090 DEFAULT execution, traced kernel by kernel with CUPTI, took the inferred path on 63 of its 63 calls. On the H100 DEFAULT writes no estimate on any blocked GEMM (0 of 1090) or internal cuSOLVER call, and kernel traces of potri show emulation kernels on 0 of 830 calls. At $n = 8192$, $g = 64$, DEFAULT emulates none of 220 internal potri calls on the H100 and A100 but 53 on the L40S and RTX 4090, and EAGER 155 on all four. A caller who enables emulation through cuSOLVER's strategy setting thus gets a different numerical path on each GPU.

d) The bit-count pointer on cuBLAS 13.3: On the H100's second version, directing the bit-count output to page-locked host memory, as the harness does, slows single EAGER products to 16 to 148 times native time. A controlled experiment places the added time in the emulation library's epilogue kernel, launched with the same configuration and running 620 times longer with the pointer; cuSOLVER does not enable the pointer on its internal calls. Instrumenting an opaque library can thus perturb what is measured (supplement).

APPENDIX F
COST

The times below describe the recorded sessions; they do not bound what another device, driver, clock state or session would give, and the conclusions rest on the order of the routes and on ratios far from 1. Table XII lists every route. All 7 matched comparisons on the H100 (3 sessions on 2 nodes) and all 7 on the RTX 4090 are resolved by 97.9% order-statistic intervals under a rule fixed before computing; the closest to 1 is equilibration's cost on the H100, session medians 1.036 to 1.043. Blocking LAUUM saves time on every GPU (blocked / one GEMM, native: 0.70 on the H100, 0.57 on the RTX 4090), and equilibrated FIXED 55 takes 1.13 of the equilibrated native route's time on the H100 with blocked LAUUM and 0.92 with one-GEMM LAUUM (0.33 and 0.27 on the RTX 4090).

On the RTX 4090 the fastest accurate route, equilibrated GEMMul8 with 16 moduli and blocked LAUUM, inverts an 8192×8192 matrix in 237 to 244 ms over sessions, against 595 to 615 ms for cuSOLVER. On the H100 the fastest accurate route is native, the blocked inversion with blocked LAUUM, and the fastest accurate emulated route, equilibrated FIXED 55 with blocked LAUUM, takes 1.20 of its time. With upload and download from pageable memory inside the clock the margins shrink: the RTX 4090's fastest accurate route takes 0.52 of cuSOLVER's time, and on the H100 the copies dominate (native cuSOLVER 236 to 273 ms). Emulation also needs workspace: the sampled peak of device memory reaches 3,652 MiB for 14 moduli and 5,474 MiB for FIXED 55 on the H100, against 2 MiB native. The T products are half of TRTRI's GEMM work, so the T-native hybrid gives back part of emulation's gain where FP64 is slow: 1.54 times the equilibrated route's time on the RTX 4090, and the solve-based route 1.07. Inside cuSOLVER's potri on an existing factor, the spread raises DEFAULT on the RTX 4090 from 0.737 of native time at $g = 0$ to 0.918 at $g = 64$, and equilibration recovers it (EAGER 1.050 on the equilibrated input).

APPENDIX G
APPLICATION

a) Design: With B_{cov} the covariance of the five variables (median age, total rooms, population, median income and median house value) and C a squared-exponential kernel on location, the model's covariance is $B_{\text{cov}} \otimes (C + \tau I)$. If house value is observed at every training site, the model's secondary weights are exactly zero (runs of that kind are in the artifact as a negative control). The design is therefore heterotopic: the four secondary variables are observed at all m training sites and house value at a predetermined half of them, so the covariance has order $n = 4.5m$, $n \in \{4608, 9216\}$, in its raw units. B_{cov} and the means come from every block group except the 256 held-out ones, so the covariance model is fixed across designs. The length scales are 0.1 and 0.3 degrees (about 11 and 33 km); 148 training sites share coordinates with another, and the nugget keeps K positive definite. At the model's nugget the secondary observations contribute 14,629 to 33,421 dollars RMS to the prediction, and cokriging changes the held-out RMSE by -2.3 to 5.1%. Power-of-two equilibration reduces $\kappa_2(K)$ from 2.2×10^{12}–1.7×10^{13} to at most 1.4×10^{5}, and the TRTRI products' 99th-percentile scale excess reaches 127 bits ($n = 4608$). At $n = 4608$ (Table VI) the reference predictor's held-out RMSE is 85,729, 101,684 and 525,730 dollars at the three nuggets. At $\tau = 10^{-6}$ the binary64 eigensolver returns a negative smallest eigenvalue estimate for 5 of the 8 matrices, which are positive definite in exact arithmetic; that nugget is a numerical stress case only, and nothing is certified there.

b) Results: Every ADP route stays at native's level against the reference (at most 1.9×10^{-7} of the held-out RMSE for EAGER, 7.2×10^{-8} for native). Over both sizes the weakest fixed routes reach 1.2×10^{-8} of the held-out RMSE

TABLE XII
FULL ROUTES, DEVICE-RESIDENT, $n = 8192$, $g = 64$: MEDIAN TIME IN MS [INTERQUARTILE RANGE], RATIO TO NATIVE CUSOLVER AND ϵ_X. ALL COLUMNS, ROWS AND SESSIONS: SUPPLEMENT.

	RTX 4090			H100		
route	ms [IQR]	ratio	ϵ_X	ms [IQR]	ratio	ϵ_X
cuSOLVER, native	595 [0.74]	1.00	8.6×10^{-16}	58.4 [0.27]	1.00	8.7×10^{-16}
cuSOLVER, DEFAULT	548 [0.8]	0.92	8.7×10^{-16}	58.4 [0.34]	1.00	8.7×10^{-16}
cuSOLVER, EAGER	912 [1.1]	1.53	8.7×10^{-16}	186 [0.7]	3.18	8.7×10^{-16}
blocked, native, blocked LAUUM	775 [0.2]	1.30	8.8×10^{-16}	24.3 [0.14]	0.42	8.9×10^{-16}
q=14	241 [0.37]	0.40	7.1×10^{-2}	37.5 [0.075]	0.64	7.1×10^{-2}
q=14, T native	377 [0.51]	0.63	2.1×10^{-15}	34.9 [0.14]	0.60	2.1×10^{-15}
q=14, eq.	245 [0.38]	0.41	2.1×10^{-15}	38.9 [0.099]	0.67	2.1×10^{-15}
q=14, solve-based	263 [0.17]	0.44	2.1×10^{-15}	46.6 [0.17]	0.80	2.1×10^{-15}
q=16, eq.	252 [0.43]	0.42	5.4×10^{-16}	41.5 [0.099]	0.71	5.5×10^{-16}
q=16, eq., blocked LAUUM	237 [0.11]	0.40	8.4×10^{-16}	37.5 [0.11]	0.64	8.5×10^{-16}
F55	360 [1.2]	0.61	6.8×10^{-2}	31.9 [0.18]	0.55	6.8×10^{-2}
F55, T native	497 [0.62]	0.84	5.6×10^{-16}	30.7 [0.22]	0.53	5.7×10^{-16}
F55, eq., blocked LAUUM	261 [0.15]	0.44	8.4×10^{-16}	29.2 [0.23]	0.50	8.5×10^{-16}
F55, solve-based	381 [1.1]	0.64	5.6×10^{-16}	39 [0.32]	0.67	5.7×10^{-16}

at the model's nugget, 6.0×10^{-7} at $\tau = 10^{-3}$ and 7.4×10^{-5} in the stress case; equilibration brings every fixed route to at most 9.0×10^{-8}. Recalibrating B_{cov} and the means on the observed house values only (8 cases) changes the held-out RMSE from 85,729 to 85,725 dollars at $\tau = 10^{-1}$ and leaves these findings in place: the worst raw fixed route moves from 1.5×10^{-9} to 1.3×10^{-9} and equilibrated routes stay at or below 3.0×10^{-11}.

c) The reference certificate: The reference prediction is computed by binary64 Cholesky of K with iterative refinement, residuals and iterates in x86 long double; for all 12 systems ($n = 4608$) the componentwise backward error falls to at most 4.8×10^{-19} within 4 steps. *Exact* below means exact arithmetic on the stored binary64 arrays K, b, k_j: $x = K^{-1}b$ and $s_j = k_j^T x$. Let $\hat{s}_j$ be the computed prediction, $r = b - K\hat{x}$ the exact residual of the computed solution, S the power-of-two equilibration and $0 < \ell \le \lambda_{\min}(SKS)$. Since $\hat{x} - x = -S(SKS)^{-1}Sr$,

$$|\hat{s}_j - s_j| \le \|Sk_j\|_2 \|Sr\|_2/\ell + \eta_j, \tag{4}$$

where $\eta_j = \gamma_n |k_j|^T |\hat{x}| + u|\hat{s}_j|$ encloses the rounding of the long-double dot product and its conversion. With $K_* = P[(C_* + \tau I) \otimes B_{\mathrm{cov}}]P^T$ the exact model matrix, $S_v B_{\mathrm{cov}} S_v \succeq \mu I$ proved by an exact-rational LDL^T, and $\Delta \ge \|S(K - K_*)S\|_F$ bounding the stored-versus-model rounding, Weyl's inequality gives

$$\lambda_{\min}(SKS) \ge \tau\mu - \Delta =: \ell. \tag{5}$$

With B_{cert} an upper bound on the RMS over the held-out points of the right-hand side of (4), R a lower bound on the reference's held-out RMSE and d an upper bound on the RMS discrepancy of a route's predictions from the reference,

$$\frac{\mathrm{RMS}(\hat{s}_{\mathrm{route}} - s)}{\mathrm{RMSE}_{\mathrm{exact}}} \le \frac{d + B_{\mathrm{cert}}}{R - B_{\mathrm{cert}}}, \quad R > B_{\mathrm{cert}}. \tag{6}$$

Here $\mu \ge 0.142$, $\Delta \le 2.7 \times 10^{-13}$ and $\ell/\tau \ge 0.142$ in all 24 cases. $B_{\mathrm{cert}}/(R - B_{\mathrm{cert}})$ is at most 7.8×10^{-7} at $\tau = 10^{-1}$ and 7.4×10^{-3} at $\tau = 10^{-3}$; at $\tau = 10^{-6}$, $B_{\mathrm{cert}} \ge R$ in all 8 cases. The certified bound is set by B_{cert}: the measured discrepancies d are two orders below it, so ratios between discrepancies compare computed solutions only.

d) Scope: The certificate assumes IEEE binary64 and x87 long-double arithmetic with no overflow, underflow or invalid operation in its own evaluation, which a script checks per case. It trusts no library exponential: its own long-double routine, whose argument reduction is checked on every run of representable inputs (supplement), errs by less than $167u_{ld}$. The route measurements and the certificate refer to the same reference vector, by its SHA-256 digest (16 of 16 cases). A verifier that shares no code with the certificate, trusting the platform's exponential to one ulp, passes all 24 cases and gives $B_{\mathrm{cert}}/(R - B_{\mathrm{cert}})$ at most 3.2×10^{-7} at $\tau = 10^{-1}$. Enclosed: ℓ, the pointwise bounds, B_{cert}, R and the ratio (6) at $\tau = 10^{-1}$ and 10^{-3}. Not enclosed: the predictive variances, ratios between routes, and the statistical adequacy of the covariance model.

e) The projection of the inverse error: Why does an inverse error large in its own terms stay small in the predictions? Let $E = \widehat{X} - K^{-1}$ be the error against the exact inverse. Each output sees it only through a projection: the mean through $k_j^T Eb$, the variance $v_j = c_0 - k_j^T K^{-1} k_j$ through $-k_j^T E k_j$. Relative to the variance, that error is bounded by $\epsilon_X \, \alpha_j \, \kappa_j$, with $\epsilon_X = \|E\|_2/\|K^{-1}\|_2$, $\alpha_j = \|K^{-1}\|_2 \|k_j\|_2^2/(k_j^T K^{-1} k_j)$ the alignment the bound allows and $\kappa_j = k_j^T K^{-1} k_j / v_j$ the cancellation in v_j. The measurements cannot use E itself. They use $E_{\mathrm{ref}} = \widehat{X} - X_{\mathrm{LAPACK}}$, the difference from LAPACK's computed inverse, with binary64 eigenvalue estimates for the norms, and the variance discrepancies are taken against the variance from the refined solves, which the certificate does not cover. They describe how computed results differ; they bound the exact-inverse quantities only up to those references' own errors, which are not certified here. Measured on the H100 ($n = 4608$, location-major, 0.3 degrees), GEMMul8

with 14 moduli raises ϵ_X (with E_{ref}) from 4.0×10^{-15} (native, blocked) to 5.7×10^{-11} at $\tau=10^{-1}$, but the realised fraction of the worst case, $|k_j^T E_{\mathrm{ref}} k_j|/(\|E_{\mathrm{ref}}\|_2\|k_j\|_2^2)$, falls from 1.3×10^{-11} to 4.0×10^{-14} (medians). The measured quadratic projections are small relative to the spectral-norm bound on the tested cross-covariance vectors; since E_{ref} can be indefinite, the experiment does not distinguish weak alignment from cancellation within those projections. The largest relative variance discrepancy moves only from 4.1×10^{-12} to 2.1×10^{-10}, and at $\tau=10^{-3}$ it is no larger than native's (1.6×10^{-8} against 8.6×10^{-8}). The cancellation factor κ_j reaches 9.8, 9.7×10^{2} and 9.5×10^{5} at the three nuggets, and every route's variance discrepancy grows with it; at $\tau=10^{-6}$ every route, native ones included, shows large variance discrepancies (6.0×10^{-2} for cuSOLVER).