Error-Controlled Mixed-Precision Cholesky Factorization with MX Microscaling Formats

ABDURAUF ABDURAKHMANOV, King Abdullah University of Science and Technology, Saudi Arabia
THEO MARY, Sorbonne Université, CNRS, LIP6, France
HATEM LTAIEF, King Abdullah University of Science and Technology, Saudi Arabia
DAVID E. KEYES, King Abdullah University of Science and Technology, Saudi Arabia
SUHAIB A. FAHMY, King Abdullah University of Science and Technology, Saudi Arabia

Existing tile-centric mixed-precision Cholesky factorization, as used in recent Gordon Bell campaigns, is based on a heuristic that bounds only rounding contribution while ignoring underflow when classifying tiles for different IEEE floating point precisions. Direct application to new low-precision formats can produce numerically unstable outcomes.

We introduce an error-controlled ladder strategy for tile-centric precision assignment that addresses the underflow challenge, extending mixed-precision Cholesky factorization with Open Compute Project (OCP) standardized Microscaling (MX) block-scaled formats. Proven numerically robust, our ladder bounds rounding and underflow errors at each candidate precision and selects the smallest one satisfying a target tolerance.

On Matérn covariance matrices, existing approaches silently overshoot ε by more than an order of magnitude at loose tolerances where FP8 underflow dominates error, whereas our ladder approach better matches user-specified ε by relying on hardware-supported low-precision formats (MXFP8/MXFP4). Furthermore, with our proposed MXFP16 extension, our approach delivers up to 1.79× speedup over the same approach with just IEEE FP formats on NVIDIA GB200 GPU, while reducing memory by roughly 30% and preserving accuracy bounds violated by previous approaches. To our knowledge, this is the first study to evaluate MX formats (down to MXFP4) as working storage-and-compute precisions in dense matrix factorization.

ACM Reference Format:
Abdurauf Abdurakhmanov, Theo Mary, Hatem Ltaief, David E. Keyes, and Suhaib A. Fahmy. 2026. Error-Controlled Mixed-Precision Cholesky Factorization with MX Microscaling Formats. *ACM Trans. Math. Softw.* 52, 4 (November 2026), 19 pages. https://doi.org/10.1145/3857919

1 Introduction

Mixed-precision computation has gained significant traction in machine learning due to its potential to reduce computation time, memory footprint, and data movement. Block-scaled formats, which share an exponent across a group of elements, have emerged as an effective low-precision representation for machine learning, supported in modern hardware accelerators. The Microscaling (MX) specification [19]—now supported on NVIDIA's B200, AMD's MI300, and Microsoft's Maia 100—standardizes one such design point with a fixed 32-element block and an 8-bit shared scale across a family of sub-byte element formats (FP8, FP6, FP4), making block-scaled arithmetic broadly portable across vendors. Scientific computing applications have been slower to adopt these techniques, often relying on double-precision arithmetic to preserve numerical stability. Showcased in recent Gordon Bell campaigns [3, 13], tile-centric mixed-precision Cholesky factorization [17] mitigates the cost of full double precision by classifying matrix tiles into precision buckets (e.g., FP32, FP16, FP8) with

Authors' Contact Information: Abdurauf Abdurakhmanov, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, abdurauf.abdurakhmanov@kaust.edu.sa; Theo Mary, Sorbonne Université, CNRS, LIP6, Paris, France, theo.mary@lip6.fr; Hatem Ltaief, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, hatem.ltaief@kaust.edu.sa; David E. Keyes, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, david.keyes@kaust.edu.sa; Suhaib A. Fahmy, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, suhaib.fahmy@kaust.edu.sa.

2026. ACM 1557-7295/2026/11-ART
https://doi.org/10.1145/3857919

a norm-based rounding-error test. The classification rule is a single-threshold test: a tile is admitted to a bucket if its estimated rounding error lies below a user-specified tolerance ε. The assignment is made once, before factorization begins, and is not revisited during execution. This formulation is well suited to IEEE single precision, where the gap between the smallest normal number and typical factorization intermediates is large enough that rounding error dominates. The gap narrows at FP16 and shrinks by several orders of magnitude at FP8—the lowest-precision bucket the baseline admits—where a non-negligible fraction of tile entries may silently underflow. The resulting error is not proportional to the unit roundoff and is therefore invisible to the rounding-only rule: in practice, the baseline can deliver a relative factorization error more than an order of magnitude (up to $\sim 60\times$) greater than the user's requested ε while reporting that the assignment was successful, so the user-specified tolerance functions as a heuristic guideline rather than an enforced bound. Existing tile-centric mixed-precision frameworks also do not exploit block-scaled representations within tiles, even though such representations can substantially reduce the underflow component of the error and thereby admit lower-precision tiles than the IEEE-only candidate set allows.

We address both limitations together via an *error-controlled ladder* for tile-centric precision assignment that natively accommodates block-scaled formats as candidates. The choice to fold scaled representations into the ladder's candidate set rather than evaluate them in isolation is deliberate: applying a more principled assignment rule to the baseline's IEEE precision tiers tightens the error bound but loses throughput, because both FP8 and FP16 tiles are pushed up to FP32 once their underflow contributions become visible—FP32 becomes the only IEEE candidate that consistently passes the admissibility test for tiles the lower tiers can no longer carry. Scale-aware candidates close this gap—the shared scale enlarges the effective dynamic range at sub-byte and narrow-IEEE precision alike, letting the ladder admit tiles at the low tiers again with provable bounds on both rounding and underflow. The ladder and the scaled candidates are therefore complementary by design: the ladder is what makes scaled candidates safe to deploy, and scaled candidates are what make the ladder fast enough to be useful.

The ladder walks candidate precisions from cheapest to most expensive and, at each tier, bounds both the rounding error (via the classical norm-based inequality) and the underflow error (via an entrywise scan against the smallest representable value of the candidate format); the first precision whose combined bound satisfies the user tolerance is selected. The ladder is format-agnostic and natively accommodates per-block scaling factors, treating IEEE and block-scaled formats as peer candidates on the same admissibility test, with the standardized MX format (k=32, 8-bit shared scale) as a first-class member of the candidate set. We evaluate three instantiations to attribute observed gains to the admissibility rule, the on-hardware MX candidates, and the paper-proposed MXFP16 candidate: (i) *Ladder IEEE* (FP8, FP16, FP32, FP64) restricted to the baseline's precision tiers; (ii) *Ladder MX* (MXFP4, MXFP8, FP16, FP32, FP64) that adds the OCP-standardized sub-byte MX candidates supported by native Blackwell MMAs; and (iii) *Ladder MX + MXFP16* (MXFP4, MXFP8, MXFP16, FP32, FP64), where MXFP16 is our extension of the OCP MX block-scaled design to 16-bit elements, executed via TF32 tensor cores.

Experiments on symmetric positive definite Matérn covariance matrices with weak correlation, representative of scientific applications such as climate modeling, on a single NVIDIA GB200 GPU demonstrate that the ladder tracks the user-requested tolerance ε closely across all tested matrix sizes and tolerances, whereas the baseline rule of [17] can silently overshoot ε by more than an order of magnitude (up to $\sim 60\times$) where underflow dominates the actual error—turning ε from a heuristic guideline into a reliable target. The three instantiations differ in the cost of delivering this accuracy guarantee: Ladder IEEE pays for it in throughput, since both FP8 and FP16 tiles are promoted to FP32 once the underflow contribution becomes visible, eliminating the cheap tiers the baseline relied on; Ladder MX recovers the FP8 tiles via MXFP4 and MXFP8, but the originally-FP16 tiles still escalate

to FP32 since no mid-tier candidate sits between MXFP8 and FP32; Ladder MX + MXFP16 adds that mid-tier candidate to catch those FP16 tiles back, delivering up to 1.79× speedup over Ladder IEEE alone (155.6 vs 87.0 TFLOPS at N=120k, ε=10^{-5}) and reducing memory footprint by roughly 30% while matching the baseline's peak factorization throughput. Together, these comparisons highlight that error-controlled tile assignment and block-scaled representation are complementary: the ladder decides *which* precision a tile can tolerate, while block scaling decides *how accurately* that precision represents the data, and neither alone delivers both safety and speed. To our knowledge, this is the first study to evaluate MX formats (down to MXFP4) as working storage-and-compute precisions in dense matrix factorization. The implementation, extending the out-of-core tile Cholesky framework of [17], is available at https://github.com/accl-kaust/ooc-cholesky-mx.

The remainder of the paper is organized as follows. Section 2 reviews related work in mixed-precision linear algebra, tile-based precision assignment, and block-scaled formats. Section 3 presents the error-controlled ladder algorithm with its rounding-plus-underflow admissibility test, the MX quantization procedure, and the evaluation metrics. Section 4 reports experiments on a single NVIDIA GB200 GPU covering tile allocation, accuracy, memory footprint, and throughput. Section 5 concludes and outlines future directions.

2 Background and Baseline Framework

2.1 Mixed Precision in Dense Linear Algebra

Mixed-precision algorithms have long been studied in high-performance computing (HPC) as a mechanism to reduce memory footprint and increase arithmetic throughput while preserving numerical accuracy. Two recent surveys frame the field: Abdelfattah et al. [1] review numerical linear algebra methods utilizing mixed-precision arithmetic across sparse and dense workloads, while Higham and Mary [11] provide a unified theoretical treatment of mixed-precision algorithms with an emphasis on rounding-error analysis and the conditions under which low-precision storage and computation are numerically safe. Building on this framework, Mary and Mikaitis [14] extend the analysis to narrow-range low-precision formats (FP8 and below, down to FP4), where the small dynamic range makes underflow a first-order concern alongside rounding; their disjoint rounding-plus-underflow decomposition forms the basis of our tile-centric admissibility criterion in Section 3.1.

Several works have investigated mixed precision in core linear algebra kernels. Zhang et al. [22] study GEMM operations using double and single precision in hardware-aware settings. Reliability aspects of mixed precision across HPC workloads are analyzed by Netti et al. [15], who note that half precision remains relatively underutilized in scientific applications compared to single and double precision. GPU-oriented optimizations for linear solvers using mixed precision, including iterative refinement strategies with FP16, are explored by Haidar et al. [10]. The use of Tensor Cores for accelerating Cholesky factorization in half precision is investigated by Abdelfattah et al. [2], demonstrating performance gains while maintaining acceptable numerical stability through refinement techniques.

2.2 Tile-Based Precision Assignment in Cholesky

Tile-based decompositions enable fine-grained precision assignment, allowing different tiles to be stored and computed at varying bit widths based on conditioning or error sensitivity. Two prior approaches are the most directly relevant comparators for this work.

Closest in spirit, Carrica et al. [6] combine recursive Cholesky on NVIDIA and AMD matrix engines with a per-block scalar rescaling around each FP16 GEMM. This anticipates block-scaled

execution, but the scaling is an external scalar wrapping an IEEE FP16 block rather than a hardware-portable format with a shared exponent applied at fixed-size element blocks, and precision is assigned by recursion-tree position rather than a tile norm-based rounding-error test.

Ren et al. [17] extend mixed-precision Cholesky factorization to FP8 in the out-of-core setting and demonstrate that certain applications exhibit tolerance to aggressive precision reduction. We take this work as our baseline framework. Its tile-classification rule applies a single-threshold test on rounding error against a user tolerance ε and is invariant to the underflow contribution—a fragility that can let the actual factorization error exceed the requested ε by more than an order of magnitude at loose tolerances where FP8 underflow is non-negligible. The error-controlled ladder is designed to close this gap.

While prior mixed-precision approaches rely on IEEE formats (FP16, FP32, FP64)—and where scaling appears, it is a single external scalar wrapping an IEEE block—none combine a per-tile rounding-plus-underflow admissibility rule with hardware-portable block-scaled candidate formats in dense scientific factorizations. This is the gap that our work addresses.

2.3 Block-Scaled Formats

Sharing a floating-point exponent across a group of values is a long-standing idea [21], with modern instantiations including Microsoft's MSFP [8] and Block Minifloat [9]. These schemes reduce storage and hardware complexity, but a single shared exponent is sensitive to outliers: every element in a block must conform to it without per-element adjustment. This sensitivity intensifies as element bit-widths shrink—from FP16's tens of binades of dynamic range, to FP8's handful, to FP4's mere few—making block-scaled construction not merely a memory optimization but a representational necessity at sub-byte widths.

The Microscaling (MX) specification, an Open Compute Project (OCP) standard [18, 19], refines this design by combining a per-element microscale exponent with a shared 8-bit scale over fixed 32-element blocks, reducing clipping risk and improving robustness to intra-block dynamic-range variation. Initial machine learning evaluations show that sub-byte MX formats can replace FP32 with minimal accuracy loss.

Hardware support now spans commercial accelerators (NVIDIA Blackwell, AMD MI300) as well as FPGA datapath designs for MX [5, 20] and ASIC precision-scalable MACs supporting the full MX family [7].

While MX has been extensively evaluated in deep learning, its implications for structured dense scientific computations remain largely unexplored. This work integrates MX as a first-class candidate in the precision-assignment ladder of Section 3.1.

3 Methodology

This section describes the per-tile precision-assignment methodology developed in this paper. We begin with the error-controlled ladder itself (Section 3.1), the algorithmic centerpiece. Once a precision is selected for a tile, the tile is quantized to the chosen format as described in Section 3.2. Section 3.3 defines the metrics used for evaluation.

Throughout this section, the error analysis concerns the *representation* of tiles in a target format—the perturbation incurred when data is stored—and not the errors introduced by subsequent arithmetic on those tiles. The admissibility criterion, the ladder, and the scaling discussion below are all statements about stored values.

3.1 Error-Controlled Ladder Assignment

This subsection presents the per-tile precision-assignment rule that replaces the rounding-only test of prior tile-centric mixed-precision frameworks with a combined rounding-plus-underflow

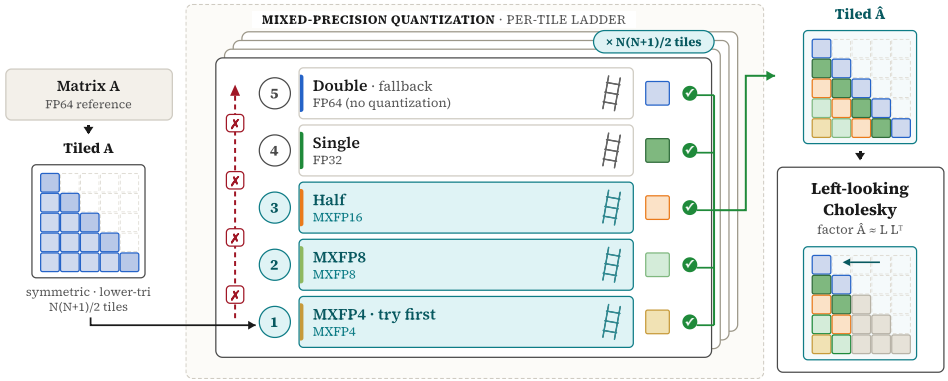


Fig. 1. Methodology pipeline (Ladder MX + MXFP16 instantiation shown). Input matrices are passed through the per-tile precision-assignment ladder (Algorithm 1), quantized to the selected precision, and factorized by a left-looking Cholesky; the result is evaluated against an FP64 reference. Other ladder instantiations use the same workflow with different candidate sets.

admissibility test. The development follows the error model in [14], adapted here to the tile-centric mixed-precision factorization setting. Figure 1 situates the ladder within the surrounding pipeline.

3.1.1 Error Model. Let A denote the global matrix and $B \in \mathbb{R}^{m\times n}$ one of its tiles. We consider a target storage format with unit roundoff u and exponent range $[e_{\min}, e_{\max}]$. Let $f_{\min} = 2^{e_{\min}}$ and $f_{\max} = 2^{e_{\max}}(2-2u)$. Any element with magnitude in $[f_{\min}, f_{\max}]$ is representable in the target format with relative accuracy at least u; elements with magnitude above $f_{\max}$ overflow and elements with magnitude below $f_{\min}$ underflow. The error analysis that follows treats overflow as inadmissible (the ladder of Section 3.1.4 unconditionally rejects any candidate at which a scaled magnitude exceeds $f_{\max}$) and bounds underflow as a contribution to the per-tile error. Table 1 lists these parameters for the formats considered in this paper. Two underflow modes are in common use:

- **Flush-to-zero (FZ):** any x with $|x| < f_{\min}$ is replaced by 0 or $\pm f_{\min}$, introducing an absolute error of at most $\min(|x|, f_{\min}/2)$.
- **Gradual underflow (GU):** any x with $|x| < f_{\min}$ is replaced by the nearest subnormal; subnormals are uniformly spaced by $2u\, f_{\min}$, so the absolute error is at most $\min(|x|, u\, f_{\min})$.

For a unified statement we define

$$g_{\min} = \begin{cases} f_{\min}/2 & \text{(FZ)} \\ u\, f_{\min} & \text{(GU)} \end{cases} \tag{1}$$

so that the underflow error on any single element is bounded by $\min(|x|, g_{\min})$. All experimental results in Section 4 use GU, which is the mode implemented by current hardware (including NVIDIA Blackwell) for the formats we evaluate; the framework analysis below applies to either mode.

For a scalar x whose floating-point representation $\mathrm{fl}(x)$ does not overflow, the rounding and underflow errors satisfy [14]

$$\mathrm{fl}(x) = x(1+\delta) + \eta, \qquad |\delta| \le u, \qquad |\eta| \le \min(|x|, g_{\min}), \qquad \delta\eta = 0, \tag{2}$$

where the disjointness $\delta\eta = 0$ encodes the fact that an element either rounds or underflows but not both. Applying (2) elementwise to a tile B yields

$$\mathrm{fl}(B) = B + E + F, \qquad |E| \le u|B|, \qquad |F| \le \min(|B|, g_{\min} J), \tag{3}$$

with the inequalities and the minimum applied componentwise, and $J \in \mathbb{R}^{m\times n}$ the matrix of ones.

3.1.2 Admissibility Criterion. Following [11], we say a tile B is admissible in the target format at relative tolerance ε if

$$\|E + F\| \;\le\; \varepsilon\,\|A\|, \tag{4}$$

where the norm is the Frobenius norm in our implementation (the analysis carries over to any monotone matrix norm with adjusted constants). The triangle inequality gives the sufficient condition $\|E\| + \|F\| \le \varepsilon\|A\|$, which we adopt as the working criterion. The criterion is normwise by design: it measures the per-tile perturbation against the global norm $\|A\|$, following the relative-tolerance formulation of [11], which is what makes a single tolerance ε meaningful across tiles of widely differing norms.

The rounding term admits the standard a priori bound $\|E\| \le u\,\|B\|$. When $\|F\| = 0$—i.e., when no elements underflow—this collapses to the classical criterion $u\,\|B\| \le \varepsilon\,\|A\|$ used by the baseline [17]. The underflow term, however, is not dominated by $u\,\|B\|$ in general. A uniform a priori bound is $\|F\| \le \sqrt{mn}\,g_{\min}$, but this is pessimistic: it charges $g_{\min}$ to every entry of the tile regardless of whether it actually underflows. For wide-range formats (FP32 and FP64) and for GU-mode arithmetic, the rounding-only criterion is typically sufficient because $\sqrt{mn}\,g_{\min} \ll u\,\|B\|$. For narrow-range formats (FP8 and below), and especially in FZ mode, the inequality reverses—$\sqrt{mn}\,g_{\min}$ can substantially exceed $u\,\|B\|$—and the rounding-only test admits tiles that the combined criterion would reject.

To recover sharpness while keeping the cost manageable, we replace the uniform bound on $\|F\|$ with an entrywise scan: an entry (i, j) contributes to F only if its scaled magnitude falls below $f_{\min}$, and the contribution is bounded by $\min(|B_{ij}|, g_{\min})$. This is tighter than $\sqrt{mn}\,g_{\min}$ by exactly the factor needed to ignore non-underflowing entries.

3.1.3 Effect of Scaling. Let $X = \{x_1, \ldots, x_k\}$ denote a 32-element MX block sharing a common scaling factor. The optimal choice of scale factor is [14]

$$s^\star = \frac{f_{\max}}{x_{\max}}, \qquad x_{\max} = \max_{i=1,\ldots,k} |x_i|, \tag{5}$$

which prevents overflow and minimizes underflow. With this scale, the underflow error on any element x_i becomes

$$|\eta_i| \le \min\!\left(|x_i|,\ \frac{g_{\min}}{s^\star}\right) = \min\!\left(|x_i|,\ \frac{g_{\min}\,x_{\max}}{f_{\max}}\right). \tag{6}$$

Comparing (6) with the unscaled bound in (2), scaling reduces the underflow contribution by the factor $x_{\max}/f_{\max} < 1$. Two consequences follow. First, scaling reduces both the number of underflowing elements and the absolute error per underflow. Second, the gain is largest when $x_{\max}$ is small, which means that grouping elements of similar magnitude under a shared scale is preferable to grouping arbitrarily—motivating MX's fixed 32-element block size. In fact, in an ideal scenario where all the elements have about the same magnitude $|x_i| \approx x_{\max}$, (6) yields a *relative* error of order $g_{\min}/f_{\max}$, which is negligible compared with the rounding errors of order u for all floating-point formats of interest.

Equation (5) gives the unconstrained optimum. The OCP MX specification restricts the shared scale to the E8M0 encoding—an 8-bit biased exponent with no mantissa—so the applied scale is the largest power of two that aligns the block maximum with the exponent of the format's largest normal:

$$\sigma = 2^{\,e^{\text{elem}}_{\max} - \lfloor \log_2 x_{\max} \rfloor}, \qquad e^{\text{elem}}_{\max} = \lfloor \log_2 f_{\max} \rfloor, \tag{7}$$

applied multiplicatively as $\tilde{x} = \sigma x$ and stored as an unsigned 8-bit exponent with bias 127. Since σ is an exact power of two, applying and removing it shifts only the exponent field and contributes

Algorithm 1 Error-Controlled ladder for per-tile precision assignment.

Require: Tile $B \in \mathbb{R}^{m\times n}$; relative tolerance ε; global norm $\|A\|$; ordered candidate list $\mathcal{P}$ (cheapest first). For each candidate p, the scale map $s_p(\cdot,\cdot)$ of its quantizer (Section 3.2): $s_p \equiv 1$ for unscaled IEEE candidates, and $s_p(i,j) = \sigma$ of (7) for MX candidates, constant within each k=32 block.

Ensure: Selected (precision, scaling) pair $p^\star$.

nrmB $\leftarrow \|B\|$
$\tau \leftarrow \varepsilon \cdot \|A\|$ ▷ absolute target on $\|E+F\|$
for all $p \in \mathcal{P}$ in ascending order of storage cost **do**
  $u \leftarrow$ unit roundoff of p
  $f_{\min}, f_{\max} \leftarrow$ smallest and largest normal of p
  $g_{\min} \leftarrow$ as in Equation (1) (FZ or GU as appropriate)
  nrmE $\leftarrow u \cdot$ nrmB ▷ rounding bound, $\|E\| \le u\|B\|$
  $F \leftarrow 0_{m\times n}$; ovf $\leftarrow$ FALSE
  for $(i,j) = 1{:}m,\ 1{:}n$ **do** ▷ entrywise scan; $s_p \equiv 1$ if p unscaled
    if $s_p(i,j)\cdot|B_{ij}| > f_{\max}$ **then**
      ovf $\leftarrow$ TRUE ▷ overflow: candidate inadmissible
    else if $s_p(i,j)\cdot|B_{ij}| < f_{\min}$ **then**
      $F_{ij} \leftarrow \min\big(|B_{ij}|,\ g_{\min}/s_p(i,j)\big)$
    end if
  end for
  nrmF $\leftarrow \|F\|$
  if $\neg$ ovf **and** nrmE + nrmF $\le \tau$ **then**
    return p ▷ first admissible tier wins
  end if
end for
return highest-cost candidate in $\mathcal{P}$ ▷ safeguard fall-through

no rounding error of its own; the bound of (6) therefore holds with $s^\star$ replaced by σ, degraded by at most a factor of two—the in-standard cost of a shared exponent carrying no mantissa. We note that the optimal scale (5) is defined with respect to representation alone; scaling a matrix so that subsequent products do not overflow is a related but distinct problem [12].

In the ladder algorithm below, we extend (6) to allow the scale to vary across the tile, writing $s_p(i,j)$ for the scale that candidate p applies to entry (i,j). For MX candidates, $s_p(i,j) = \sigma$ of (7), constant within each 32-element block; for unscaled IEEE candidates, $s_p \equiv 1$ and the bound reduces to (2). The admissibility test inherits this generality.

3.1.4 The Ladder Algorithm. Given a tile B, a relative tolerance ε, the global norm $\|A\|$, and an ordered candidate list $\mathcal{P}$ of (precision, scaling) pairs sorted from cheapest to most expensive in storage cost, we walk $\mathcal{P}$ from the bottom and return the first candidate satisfying the admissibility criterion. Pseudocode is given in Algorithm 1.

We evaluate three instantiations of the ladder, each defined by its candidate list. We refer to them throughout the paper by the short names used in figures:

- **Ladder IEEE**: $\mathcal{P}$ = (FP8 E4M3, FP16, FP32, FP64). Restricted to the same precision tiers as the baseline [17], this isolates the contribution of the rounding-plus-underflow admissibility test from the contribution of the expanded MX candidate set.

Table 1. Parameters of the formats used in this paper. t is the significand precision including the implicit bit, $u = 2^{-t}$ the unit roundoff, $f_{\min}$ the smallest positive normal, $f_{\max}$ the largest normal, and $e_{\max}^{\text{elem}} = \lfloor \log_2 f_{\max} \rfloor$ the exponent used by the applied scale of (7). The MX variants share the element parameters of their unscaled counterparts and differ only in the shared block scale, which enlarges the effective dynamic range without changing u.

Format	t	u	$f_{\min}$	$f_{\max}$	$e_{\max}^{\text{elem}}$	Scale	bits/elt
FP64	53	1.11×10^{-16}	2.23×10^{-308}	1.80×10^{308}	1023	—	64
FP32	24	5.96×10^{-8}	1.18×10^{-38}	3.40×10^{38}	127	—	32
FP16	11	4.88×10^{-4}	6.10×10^{-5}	6.55×10^{4}	15	—	16
FP8 E4M3[1]	4	6.25×10^{-2}	1.56×10^{-2}	4.48×10^{2}	8	—	8
MXFP16[2]	11	4.88×10^{-4}	6.10×10^{-5}	6.55×10^{4}	15	E8M0, k=32	16.25
MXFP8 E4M3[1]	4	6.25×10^{-2}	1.56×10^{-2}	4.48×10^{2}	8	E8M0, k=32	8.25
MXFP4 E2M1	2	2.50×10^{-1}	1.00	6.00	2	E8M0, k=32	4.25

[1] E4M3 reserves its largest mantissa code for NaN, so $f_{\max} = 448$ rather than the $2^{e_{\max}}(2-2u) = 480$ given by the general expression in Section 3.1.1.
[2] Paper-proposed extension of the OCP MX family to 16-bit elements (Section 3.1.4).

- **Ladder MX**: $\mathcal{P}$ = (MXFP4, MXFP8 E4M3, FP16, FP32, FP64). Adds the OCP-standardized sub-byte MX candidates (MXFP4, MXFP8 E4M3) that are supported by native block-scaled MMAs on Blackwell. Used to characterize performance under fully on-hardware execution.
- **Ladder MX + MXFP16**: $\mathcal{P}$ = (MXFP4, MXFP8 E4M3, MXFP16, FP32, FP64). Further admits MXFP16, a paper-proposed extension of the OCP MX specification beyond its sub-byte formats: a 16-bit element with a shared 8-bit block scale over $k = 32$ contiguous elements.

We initially considered MXFP8 E5M2 and MXFP6 (E2M3 and E3M2 variants) as additional candidates in Ladder MX + MXFP16, exploring different exponent allocations within the MX format family. The admissibility test rarely selected these tiers for the Cholesky tiles studied here: MXFP6 sat at intermediate accuracy below MXFP8 E4M3, and MXFP8 E5M2 was dominated by MXFP8 E4M3 under the dynamic ranges we encountered. We omit them from the final candidate list and report all results using the three instantiations above.

IEEE and scaled variants appear as peer candidates rather than as a primary/fallback pair: a tile may be admitted at MXFP8 even if it would also be admitted at FP16, provided MXFP8 sits earlier in $\mathcal{P}$ and its admissibility test passes. The ordering within $\mathcal{P}$ encodes the storage-cost preference.

One might ask why the ladder does not scale the IEEE candidates directly to avoid overflow. Scaling an IEEE tile does not leave it an IEEE tile: the scale must be stored alongside the data and reapplied on every read, which is precisely what the MX candidates already provide, with native hardware support.

To compare the per-tile precision assignments produced by the ladder against those of the baseline rule of [17], we evaluate both at matched relative tolerance ε. The ladder may promote tiles that the rounding-only rule admitted to FP8 because the underflow contribution at FP8—hidden from the baseline test—becomes visible and pushes $\|E\| + \|F\|$ above the threshold. The resulting allocations, factorization error, memory footprints, and throughputs across the three instantiations are reported and analyzed in Section 4.

3.1.5 Properties.

Safety. By construction, the returned $p^\star$ satisfies $\|E\| + \|F\| \le \varepsilon\|A\|$, hence $\|\mathrm{fl}_{p^\star}(B) - B\| \le \varepsilon\|A\|$ by the triangle inequality applied to (3). The ladder is therefore conservative with respect to the relative-tolerance criterion of [11]. We emphasize that this bound governs the *per-tile representation* error, whereas the metric reported in Section 3.3 and used to evaluate accuracy in Section 4 is the relative *factorization* error $\|A - LL^T\|_\infty / \|A\|_\infty$—a global quantity that depends on how per-tile errors propagate through the factorization. The theorem above does not bound the factorization error directly. Empirically, however, the two quantities track closely on the workloads we study: changing the candidate set (Ladder IEEE vs. Ladder MX + MXFP16) also moves the measured factorization error in the same direction, indicating that ε functions as an operationally useful target for the factorization error even though only the representation error is provably bounded by it.

Toward a factorization-level bound. The bound above governs the per-tile representation error. A bound on the factorization error would additionally have to account for how those perturbations propagate, and for the arithmetic itself, which is likewise mixed: each update is carried out at a precision determined by the tiles involved rather than at a single working precision. This would require a backward-error result for tile Cholesky with per-tile unit roundoff, control of the growth of intermediate quantities—the ladder assigns precisions once, from the norms of the input tiles—and a choice of dimensional constant, where the deterministic worst case grows linearly in the accumulation length while the probabilistic model of [11] grows with its square root, at the cost of a high-probability rather than deterministic guarantee. Beyond that constant, the factorization error depends on the conditioning of A and on the distribution of tile norms. Cholesky is the favourable case, being unconditionally stable and pivot-free, but a rigorous bound in the mixed-precision-arithmetic setting remains open and we do not claim one here.

Recovery of the baseline rule. When $\sqrt{mn}\, g_{\min} \ll u\,\|B\|$ at all candidate precisions—typical for FP32 and FP64 —the entrywise scan returns $\|F\| \approx 0$ and the admissibility test reduces to $u\,\|B\| \le \varepsilon\|A\|$. In this regime the ladder selects the same tile precision as the rounding-only rule of [17]. The two procedures diverge only when underflow becomes a non-negligible fraction of the total error, which by construction is when divergence is numerically justified.

Conservatism of the rounding term. The disjointness $\delta\eta = 0$ in (2) means an entry that underflows incurs no rounding error. Algorithm 1 nonetheless evaluates the rounding bound as $u\,\|B\|$ over the whole tile, including entries counted in F, and is therefore conservative: the sharper bound restricts the norm to the non-underflowing entries. Because those entries are by construction the smallest in the tile—each bounded in magnitude by $f_{\min}/s_p(i,j)$—their contribution to $\|B\|$ is negligible relative to the entries that dominate it, and the refinement would tighten the admissibility test by a correspondingly negligible margin. We retain the simpler form, noting that the resulting bound errs on the side of safety.

Cost relative to the baseline. The baseline rule applies a single rounding test per tier. Because $\|B\|$ is precision-independent, it is computed once per tile and reused across tiers, giving an overall pre-analysis cost of $O(mn)$ per tile. The ladder performs additional work at each tier. For every candidate, an entrywise underflow scan visits all mn entries to evaluate the $\|F\|$ contribution under the candidate's $f_{\min}$ and $g_{\min}$. For IEEE candidates, an overflow check additionally compares the tile's maximum magnitude against the candidate's $f_{\max}$, which is $O(1)$ per tier once $\|B\|_\infty$ is precomputed. For scaled candidates, overflow is structurally prevented by the applied scale of (7); the per-region maxima $x_{\max}$ underlying the scale are precision-independent and computed once per tile in $O(mn)$. The pre-analysis cost therefore rises from $O(mn)$ to $O(|\mathcal{P}| \cdot mn)$ per tile, a factor-$|\mathcal{P}|$ asymptotic increase in the dominant term. The constant factors are non-trivial in practice: scaled

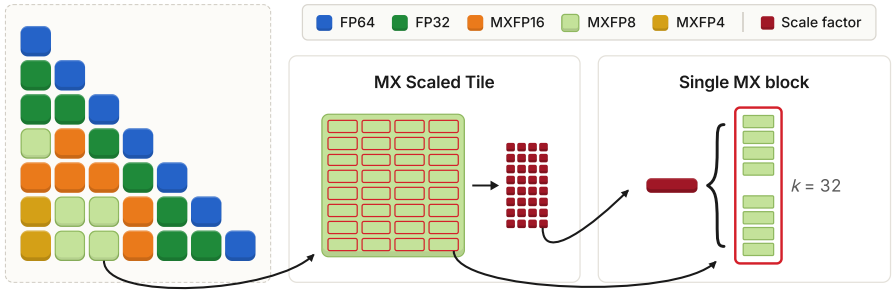


Fig. 2. Visualization of mixed-precision Cholesky factorization with MX scaling. **Left**: lower-triangular SPD matrix with per-tile precision assignment from the ladder of Section 3.1; tiles are encoded at FP64, FP32, MXFP16, MXFP8, or MXFP4 (color-coded). **Middle**: an MX-scaled tile is partitioned into fixed blocks of k=32 contiguous elements, with the corresponding shared E8M0 scale factors (red)—an 8-bit biased exponent per block– stored as a separate metadata array. **Right**: detail of a single MX block—k=32 elements with one shared scale, following the OCP MX specification [19].

candidates require per-region scale derivation and per-block quantization that IEEE-only flows do not, and the per-entry scan involves conditional logic at each tier. Aggregated over the matrix, the pre-analysis is $O(|\mathcal{P}|\, N^2)$ against $O(N^3)$ factorization, so the relative overhead shrinks with N.

3.2 Quantization

Once the ladder of Section 3.1 has selected a precision for a tile, the tile is quantized as follows. For each 32-element MX block (Figure 2), the shared scale is σ of (7), computed from the block maximum $x_{\max} = \max_{x\in\mathcal{R}} |x|$, where $\mathcal{R}$ denotes the set of elements in the block. Each element $x \in \mathcal{R}$ is normalized as $\tilde{x} = \sigma x$ and quantized to the target floating-point format by round-to-nearest, saturating at the format's maximum normal magnitude.

The scale is stored as an unsigned 8-bit exponent with bias 127 in a separate metadata array, following the OCP E8M0 encoding, and reapplied on reconstruction. Because σ is an exact power of two, applying and removing it shifts only the exponent field and introduces no rounding error of its own: the quantization error is entirely that of the element format. Unscaled IEEE candidates are quantized directly, with no scale ($s_p \equiv 1$ in Algorithm 1). The admissibility scan of Algorithm 1 uses the same scale rule as the quantizer, so every entry is tested against the scale it will actually receive.

We use two execution modes, each serving a distinct purpose. For *accuracy analysis*, all scaled formats are evaluated via software emulation: MXFP16, MXFP8, and MXFP4 are quantized by value under the same scale rule (7) and k=32 block partition as the native path, with the quantized values carried in FP32 containers rather than packed low-precision words. Quantized values propagate through the factorization exactly as they would on hardware: arithmetic and accumulation are carried out in FP32 under both modes, updates accumulate in FP32 within an operation, and tiles are returned to their assigned format once each panel operation completes. The two modes therefore differ in the physical representation of the operands and in the order of accumulation within each product, not in the precision at which values are held or summed. The memory footprints reported in Section 4.4 are computed analytically from the per-tile format assignment using the cost model of Section 3.3, and therefore reflect native packed storage rather than the emulation containers.

For *performance evaluation*, we execute on real hardware. The MX formats with native Blackwell MMA support—MXFP4 and MXFP8 E4M3—run on the GB200 tensor cores directly. MXFP16, the paper-proposed extension of OCP MX beyond its standardized sub-byte formats (Section 2.3), runs

via TF32 tensor cores, which provide higher throughput than FP32 while preserving the dynamic range needed to absorb the MXFP16 block scale. FP16, FP32, and FP64 use their standard cuBLAS code paths.

Operations between tiles of different formats. A tile's format is fixed by the ladder assignment and does not change during the factorization, so an update may combine operands stored at different precisions. Following the baseline framework [17], mixed operands are promoted to the higher of the two formats before the operation, and the result is returned to the destination tile's assigned format once the panel operation completes. For MX operands, promotion dequantizes each block by its shared scale, so the promoted values carry the precision of the format they were stored in rather than that of the compute type. The native block-scaled tensor-core path (Section 4.1) is taken when both operands share an MX format.

3.3 Evaluation Metrics

We evaluate all configurations along three axes: factorization accuracy, storage cost, and performance.

Factorization error. The factorization accuracy of a configuration is measured by the relative factorization error in the matrix infinity norm:

$$\frac{\|A - LL^T\|_\infty}{\|A\|_\infty},$$

where L is the Cholesky factor produced by the mixed-precision factorization and A is the FP64 reference matrix.

Memory footprint. Because the input matrices are symmetric positive definite, we account only for the lower-triangular portion of the factorization—the upper triangle is implicit. Diagonal tiles are always assigned to FP64 in our setup, both as a numerical stability measure for the in-place POTRF operation and to ensure the diagonal contribution to the factorization error is exact; block quantization considerations therefore apply only to off-diagonal tiles. Storage cost is reported at each tile's native format bit-width. For an off-diagonal tile t of size $n_b \times n_b$ assigned to a format with b_t bits per element, divided into g_t scale regions of b_s bits each, the per-tile cost is

$$n_b^2 \cdot b_t + g_t \cdot b_s \quad \text{bits},$$

and the total memory across all T tiles in the lower triangle (off-diagonal at the selected ladder precision, diagonal at FP64) is

$$M_{\text{bits}} = \sum_{t=1}^{T} \left(n_b^2 \cdot b_t + g_t \cdot b_s \right), \qquad M_{\text{GB}} = \frac{M_{\text{bits}}}{8 \times 10^9}.$$

Performance. We report factorization throughput in TFLOPS on a single NVIDIA GB200 GPU; experimental setup, including the Baseline reference point we measure against, is detailed in Section 4.

4 Results

This section presents the empirical evaluation of the error-controlled ladder against the rounding-only baseline of [17]. We begin with the setup (Section 4.1), then characterize the ladder's tile-allocation behavior (Section 4.2) as the mechanism that underlies every subsequent claim. The allocation determines accuracy (Section 4.3), memory footprint (Section 4.4), and throughput (Section 4.5), reported in that order.

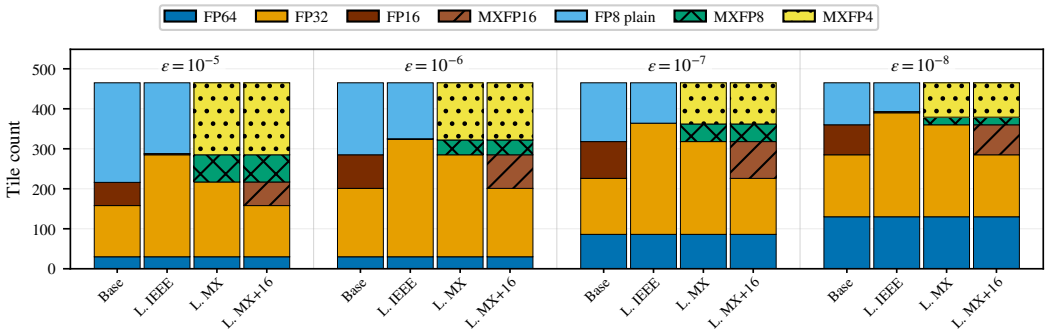


Fig. 3. Per-tile precision allocation at $N = 120\text{k}$ ($n_b = 4096$) for four target tolerances. Each cluster of four bars shows, left to right: Baseline, Ladder IEEE, Ladder MX, Ladder MX + MXFP16. Stack contents (bottom-to-top): FP64, FP32, FP16, MXFP16, FP8 plain, MXFP8, MXFP4. This is a representative example; the same qualitative trends hold across the tested range $N \in \{20\text{k}, \ldots, 120\text{k}\}$.

4.1 Experimental Setup

Hardware. Performance results in this section are obtained on a single NVIDIA GB200 superchip GPU. The ladder's candidate set spans formats with diverse hardware paths: MXFP4 and MXFP8 E4M3 execute on Blackwell's native block-scaled tensor cores, MXFP16 runs via the TF32-based GEMM, and FP16 and FP32 use their standard cuBLAS code paths. FP64 GEMM updates use the Ozaki scheme [16] since Blackwell omits native FP64 tensor cores; Ozaki applies uniformly across all configurations including the Baseline. Accuracy results are obtained from software emulation following the protocol of Section 3.2.

Workload, sizes, and tolerances. We generate symmetric positive definite Matérn covariance matrices using the ExaGeoStat framework [4] under weak correlation ($\beta = 0.02627$, smoothness $\nu = 0.5$), which exhibit heterogeneous tile norms appropriate for precision-aware tiling. We evaluate matrix sizes N from 20k to 120k with $n_b \in \{2048, 4096\}$ tuned for GB200 throughput rather than inherited from [17], both satisfying the OCP MX block-size requirement (k=32 | n_b), and four target tolerances $\varepsilon \in \{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$.

Configurations. We compare four configurations: *Baseline* (the rounding-only mixed-precision assignment of [17]: FP64 → FP32 → FP16 → FP8, reimplemented on GB200), *Ladder IEEE* (the ladder restricted to the baseline's precision tiers), *Ladder MX* (the ladder with the sub-byte MX candidates MXFP4 and MXFP8 E4M3), and *Ladder MX + MXFP16* (the full ladder with MXFP16 in the middle tier via TF32 compute). All configurations use gradual underflow (GU) by default; throughput measurements use 4 CUDA streams per device.

The two execution modes (Section 3.2) do not affect the comparison across configurations: the ladder produces the per-tile assignment before any arithmetic occurs, so the allocation measured for accuracy is exactly the one executed for throughput, and all four configurations are evaluated through the same path in each mode.

4.2 Tile Allocation: What the Ladder Does

We begin by characterizing the tile-allocation pattern produced by each configuration. The allocation is the mechanism behind every subsequent result—accuracy, memory, throughput—and is illustrated in Figure 3 at $N = 120\text{k}$ as a representative example; the same qualitative trends hold across the full $N \in \{20\text{k}, \ldots, 120\text{k}\}$ range.

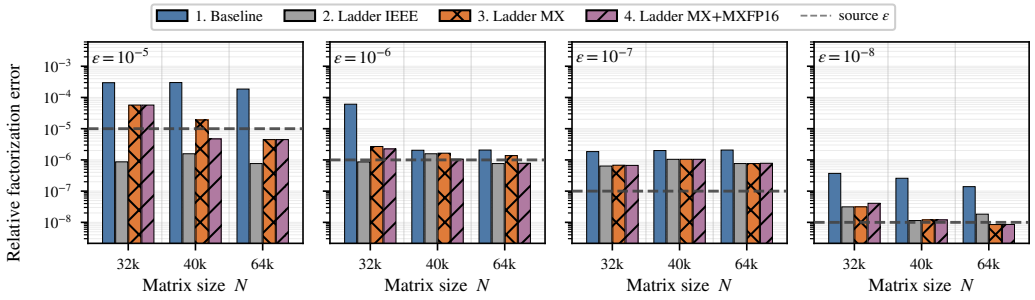


Fig. 4. Relative factorization error $\|A - LL^T\|_\infty / \|A\|_\infty$ (log scale) for the four configurations at matrix sizes $N \in \{32k, 40k, 64k\}$ and tolerances $\varepsilon \in \{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$. Dashed gray line marks the requested ε.

The Baseline follows the rounding-only rule of [17]: tiles whose rounding bound $u\|B\|$ fits within $\varepsilon\|A\|$ are admitted to the cheapest precision passing this test. At $\varepsilon = 10^{-5}$ this admits a substantial FP8 population (249 tiles), with a smaller share at FP16 (58) and FP32 (128), and a fixed FP64 contribution from diagonal tiles (30). As ε tightens, the cheapest tier yields to higher precisions: by $\varepsilon = 10^{-8}$ most tiles are at FP32 or above.

Switching to Ladder IEEE on the same precision tiers exposes what the rounding-only test had been hiding. The entrywise underflow scan reveals that many tiles incur a non-negligible underflow contribution, and the admissibility test pushes them upward. At $\varepsilon = 10^{-5}$ the FP8 count drops from 249 to 177 and FP16 collapses from 58 to 3; both populations escalate to FP32, whose count grows from 128 to 255—a single-tier absorption of 127 additional tiles. The cost of enforcing the bound manifests as a visible mass shift upward, which Section 4.4 and Section 4.5 quantify as the corresponding memory and throughput cost.

Introducing MX scaling at the lowest tiers (Ladder MX) reverses this upward shift. MXFP4 and MXFP8 share an 8-bit scale across each 32-element block, absorbing the local dynamic range that caused plain FP8 to fail the admissibility test. The lower-tier population recovers at sub-byte storage—at $\varepsilon = 10^{-5}$, the combined MXFP4+MXFP8 count is 248, approximately matching the size of the Baseline's FP8 population at the same (N, ε). The crucial difference is that the scaled formats satisfy the admissibility test that plain FP8 fails: the lower tier carries the same volume of tiles, but the shared scale makes each one numerically safe.

A vacancy emerges at the mid tier. Ladder MX places few tiles at plain FP16: when FP16's underflow contribution exceeds the per-tile budget, the ladder lacks a scaled mid-tier candidate and must escalate these tiles to FP32. At $\varepsilon = 10^{-5}$, Ladder MX's FP32 count grows from 128 (Baseline) to 187, absorbing tiles that Ladder MX could not safely admit at plain FP16. Extending the candidate set with MXFP16 (Ladder MX + MXFP16, fourth bar) fills this gap: MXFP16's per-block scale shrinks the underflow contribution at half precision, letting these tiles fit within the budget at scaled half precision. The mid-tier population recovers, tracking Baseline's FP16 distribution closely across the tested tolerances.

The progression Baseline → Ladder IEEE → Ladder MX → Ladder MX + MXFP16 thus traces a single arc: enforcing the rounding-plus-underflow bound first pushes tiles up the precision tiers (Ladder IEEE), and then scaled candidates at the same admissibility test pull them back down, ultimately reproducing the Baseline's tile-count distribution at every tier (Ladder MX + MXFP16) while replacing plain FP8 with MXFP4/MXFP8 and plain FP16 with MXFP16. The accuracy, memory, and throughput consequences of this trajectory are reported in the next three subsections.

4.3 Accuracy: ε as a Reliable Target

The relative factorization error across the four configurations and four tolerances is reported in Figure 4. Given the allocation patterns of Section 4.2, the accuracy outcomes follow directly.

Baseline overshoots its own tolerance. At $\varepsilon = 10^{-5}$, the baseline delivers a factorization error of order 10^{-4} across all three tested matrix sizes—2.97×10^{-4} at $N = 32$k, 3.00×10^{-4} at $N = 40$k, and 1.86×10^{-4} at $N = 64$k—overshooting the requested tolerance by a factor of roughly 30×. The mechanism is the FP8 band of Section 4.2: the baseline admits tiles to FP8 by the rounding-only test, but at FP8 a non-negligible fraction of entries underflows, and the actual error is dominated by this underflow contribution. The overshoot persists across the tested grid, reaching $\sim 60\times$ at $\varepsilon = 10^{-6}$, and even at $\varepsilon = 10^{-8}$ the Baseline still exceeds the requested tolerance by more than an order of magnitude, since the rounding-only test continues to admit some FP8 tiles whose underflow contribution dominates the error.

The ladder configurations track ε closely. Ladder IEEE delivers errors at or modestly above ε across every (N, ε) pair because it eliminates the FP8 band (Section 4.2) and pays the cost in higher-tier tiles. Ladder MX and Ladder MX + MXFP16 also track ε closely, despite admitting cheaper sub-byte tiers the baseline cannot reach safely: the shared scale absorbs the local dynamic range so MXFP4 and MXFP8 tiles satisfy the admissibility test on regions of the matrix where plain FP8 fails. At $\varepsilon = 10^{-5}$ Ladder MX + MXFP16 reaches 4.47×10^{-6} at $N = 64$k—a 41.6× reduction relative to the Baseline at the same (N, ε), achieved by changing only the format in the low-precision slot. The largest overshoot for Ladder occurs at $\varepsilon = 10^{-7}$ (factors of 6.7 to 10.5 across the tested matrix sizes), where FP32's unit roundoff $u_{\mathrm{FP32}} \approx 1.19 \times 10^{-7}$ sits essentially at the target tolerance: FP32 tiles narrowly pass admissibility rather than being pushed to FP64, and their accumulated edge-of-budget contribution drives the overshoot. This behaves like a step function across u_{FP32}—looser tolerances have slack, tighter ones ($\varepsilon < u_{\mathrm{FP32}}$) force FP64 promotion.

From heuristic guideline to reliable target. Under the baseline rule, ε functions as a heuristic guideline: the assignment satisfies a rounding-only test, but the actual error may substantially exceed ε—by more than an order of magnitude at loose tolerances—when underflow dominates. Under any of the three ladder configurations, ε functions as a reliable target: the per-tile representation error is provably bounded by $\varepsilon\|A\|$ (Section 3.1.5), and the factorization error tracks ε closely, exceeding it by at most a small constant factor at the operating point where the per-tile budget is fully utilized. The ladder does not claim a strict factorization-error bound—the per-tile theorem of Section 3.1.5 does not propagate as one—but turns ε from a number the baseline can overshoot by an order of magnitude into one the ladder respects to within a small factor across every tested (N, ε) pair.

4.4 Memory Footprint

L-factor memory footprint by tile format is shown in Figure 5. The allocation patterns translate directly into memory: Ladder IEEE's promotion of FP8 and FP16 tiles to FP32 quadruples and doubles their per-element cost, respectively, while Ladder MX's MXFP4 substitution halves it.

Ladder IEEE costs memory; Ladder MX partially recovers; Ladder MX + MXFP16 saves. Ladder IEEE shifts mass upward: at $N = 120$k, $\varepsilon = 10^{-5}$, the FP32 count grows from 128 to 255 (+127), absorbing both promoted FP8 tiles (249 → 177) and most former FP16 tiles (58 → 3). The result is a memory footprint that grows substantially relative to the Baseline at the same tolerance. Ladder MX places tiles at sub-byte storage again via MXFP4 (4.25 bits per element including amortized scale metadata) and MXFP8 (8.25 bits), with MXFP4 absorbing a substantial fraction of what the Baseline placed at FP8—halving the per-element cost on those tiles—though FP32 promotion of former mid-tier tiles

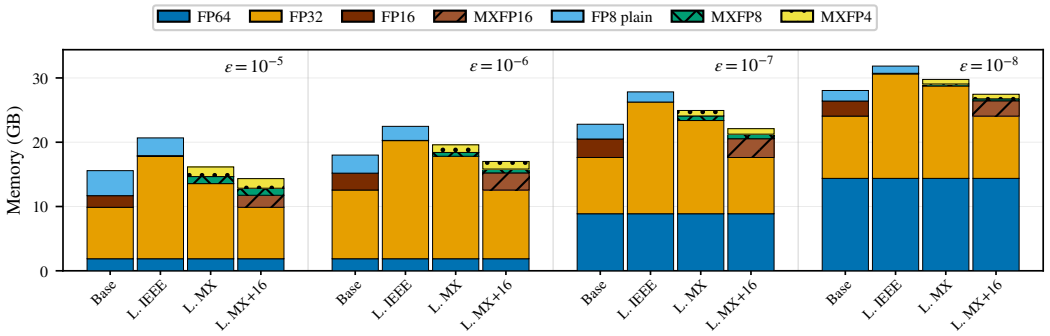


Fig. 5. Memory footprint (GB) of the lower-triangular factor at $N = 120$k ($n_b = 4096$) for the four configurations across tolerances ε. Stacked bars decompose memory by tile-format bucket (bottom-to-top: FP64, FP32, FP16, MXFP16, FP8 plain, MXFP8, MXFP4). Diagonal tiles are at FP64 in all configurations.

keeps total memory above the Baseline. Ladder MX + MXFP16 shifts some tiles previously at FP32 in Ladder MX down to MXFP16 storage, bringing memory below the Baseline while enabling the throughput recovery reported in Section 4.5.

Quantitative comparison at large N. At $N = 120$k, $n_b = 4096$, Ladder MX + MXFP16 ranges from 14.34 GB at $\varepsilon = 10^{-5}$ to 27.46 GB at $\varepsilon = 10^{-8}$. Compared to the Baseline at $\varepsilon = 10^{-5}$ (15.58 GB), Ladder MX + MXFP16 saves roughly 8% of L-factor memory while delivering the accuracy guarantee the Baseline overshoots. The savings come from the MXFP4 substitution at the cheap tier: of the 249 tiles the Baseline places at FP8 plain (8 bits per element), 180 drop to MXFP4 (4.25 bits per element including amortized scale) under the ladder—halving the per-element cost on those tiles—while 68 move to MXFP8 (8.25 bits) at approximately the same cost as plain FP8. The savings shrink as ε tightens (5.5% at $\varepsilon = 10^{-6}$, 3.0% at $\varepsilon = 10^{-7}$, 2.1% at $\varepsilon = 10^{-8}$) because the MXFP4 population shrinks as more tiles are pushed up the ladder. The savings against Ladder IEEE are substantially larger: 30.6% at $\varepsilon = 10^{-5}$ (20.67 GB) and 13.8% at $\varepsilon = 10^{-8}$ (31.84 GB), reflecting Ladder IEEE's full memory cost of FP32 promotion. Holding the full L factor (lower triangle including diagonal) at FP64 would require 56.3 GB; at $\varepsilon = 10^{-7}$ Ladder MX + MXFP16 uses 22.11 GB, or 39% of this all-FP64 reference.

Effect of tolerance. As ε tightens, the precision distribution in every configuration shifts upward and memory grows. The shift is most pronounced at the transition from $\varepsilon = 10^{-5}$ to 10^{-6}, where the cheapest tiers yield to FP16/MXFP8/MXFP16. At $\varepsilon = 10^{-8}$ most of the matrix is at FP32 or above, and the gap between Baseline and Ladder MX + MXFP16 narrows; Ladder IEEE and Ladder MX remain noticeably higher because their candidate sets cannot capture all the precision savings available to the full ladder.

4.5 Throughput and the Accuracy–Throughput Trade-off

Factorization throughput is reported in Figure 6. The progression Baseline → Ladder IEEE → Ladder MX → Ladder MX + MXFP16 mirrors the allocation arc of Section 4.2, telling a three-step recovery story. At $N = 120$k, $\varepsilon = 10^{-5}$: the Baseline reaches its peak of ~158 TFLOPS (at the cost of the error overshoot reported in Section 4.3), falling monotonically to ~76 TFLOPS at $\varepsilon = 10^{-8}$; Ladder IEEE drops to ~87 TFLOPS (~55% of Baseline), as eliminating the cheap-tier and most mid-tier IEEE candidates shifts mass to FP32; Ladder MX recovers to ~115 TFLOPS (~73%) via MXFP4 and MXFP8, with the remaining gap reflecting the mid-tier vacancy of Section 4.2: when plain FP16 fails the admissibility test, Ladder MX has no scaled mid-tier candidate and must escalate

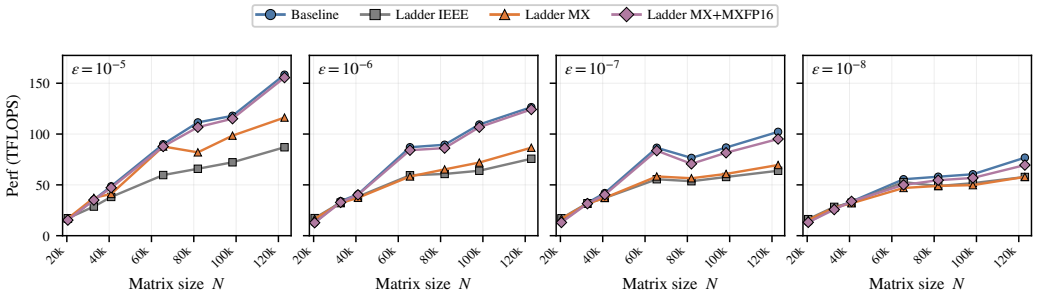


Fig. 6. Factorization throughput (TFLOPS) vs. matrix size N for the four configurations, with one panel per tolerance ε. Throughput obtained on a single NVIDIA GB200 GPU using 4 CUDA streams per device.

to FP32; Ladder MX + MXFP16 reaches ∼156 TFLOPS (within ∼2% of Baseline), filling the mid tier via MXFP16's TF32 path while delivering the accuracy of the other ladders and the memory savings of Section 4.4. With cheap-tier throughput restored, the factorization bottleneck shifts from low-precision GEMMs onto residual FP64 panel operations.

The accuracy–throughput trade-off. Together, Figure 6 and the accuracy results in Figure 4 characterize the trade-off each configuration offers. The Baseline reaches the highest throughput but at a factorization error that exceeds the requested ε by up to $\sim 60\times$ across the tested grid and still by more than an order of magnitude at $\varepsilon = 10^{-8}$; the throughput peak comes with no guarantee on the actual accuracy delivered. Ladder IEEE enforces the bound but pays a substantial throughput cost, since the only IEEE candidates available below FP32 (FP8 and FP16) cannot pass the admissibility test once underflow becomes visible, so most of the matrix falls onto the FP32 path. Ladder MX recovers cheap-tier throughput by admitting MXFP4 and MXFP8 at the same admissibility bound, but the mid-tier vacancy keeps it short of the Baseline's throughput level. Ladder MX + MXFP16 closes both gaps simultaneously: it matches the Baseline's throughput within ∼2% at the loose-tolerance operating point while delivering the same accuracy guarantee as the other two ladder configurations. At intermediate tolerances ($\varepsilon = 10^{-6}, 10^{-7}$) the three configurations approach parity in measured factorization error, and Ladder MX + MXFP16's advantage is its provable enforcement of the per-tile bound (Section 3.1.5) rather than an empirical accuracy gain. We note that the ladder's pre-analysis is more expensive than the Baseline's rounding-only rule due to the per-candidate entrywise scan (Section 3.1.5). Its relative cost falls as N grows, since the scan is $O(|\mathcal{P}|N^2)$ against $O(N^3)$ factorization, and it is paid once per matrix rather than once per operation. The scan is also independent across tiles and across candidates within a tile, so it parallelizes in both dimensions; because the left-looking traversal consumes tiles in a known order, the analysis of later panels can in principle overlap with the factorization of earlier ones. A characterization of an optimized implementation is left to future work.

5 Conclusions and Future Work

This paper presented an error-controlled ladder for per-tile precision assignment in tile-centric mixed-precision Cholesky factorization, exploiting microscaling formats. The ladder decomposes the per-tile error into rounding and underflow contributions, walks an ordered candidate list from cheapest to most expensive in storage cost, and admits the first candidate that satisfies a unified admissibility test against the user-requested tolerance ε. Integrating block-scaled MX formats (MXFP4 and MXFP8) and MXFP16 as ladder candidates extends the framework beyond IEEE storage

classes: the shared per-block scale absorbs the local dynamic range that causes plain low-precision formats to fail the test, restoring throughput at the cheap and mid tiers without sacrificing the accuracy bound.

Experimental evaluation on a single NVIDIA GB200 GPU shows that the ladder turns ε from a heuristic guideline—which the baseline rule of [17] can overshoot up to $\sim 60\times$ when underflow dominates the actual error—into a reliable target tracked closely across all tested operating points. Among the three ladder instantiations, Ladder MX + MXFP16 delivers up to 1.79× speedup over Ladder IEEE alone (155.6 vs 87.0 TFLOPS at N=120k, ε=10^{-5}), reduces L-factor memory footprint by roughly 30% over Ladder IEEE, and matches the Baseline's peak throughput within ∼2% while enforcing the same accuracy guarantee as the other ladder configurations. To our knowledge, this is the first study to evaluate MX formats (down to MXFP4) as working storage-and-compute precisions in dense matrix factorization.

There are several directions for extending this work. The present evaluation focuses on Cholesky factorization of weak-correlation 2D Matérn covariance matrices. The combination of error-controlled tile assignment and block-scaled formats is not specific to Cholesky—it operates on any tile-centric workload where rounding and underflow can be bounded per tile—so testing the ladder on other dense matrix algorithms (LU, QR, eigensolvers, and tile low-rank variants) would establish whether the recovery story observed here generalizes across the broader linear algebra landscape. Likewise, extending the analysis to medium and strong correlation regimes and to 3D Matérn kernels for volumetric geospatial data would test the ladder under different tile-norm structures, with relevance to atmospheric and oceanographic modeling.

The framework itself opens up a design space that this paper only partially explores. The scaling granularity is fixed at the OCP MX block (k=32); a systematic comparison across granularities (tile-level, intermediate, and MX) would characterize the accuracy–metadata trade-off. Similarly, the error model treats FZ and GU uniformly (Section 3.1.1) but the experimental evaluation uses GU exclusively; a head-to-head FZ-vs-GU comparison would quantify whether the simpler FZ mode suffices in practice.

Acknowledgments

We would like to thank NVIDIA for providing remote access to the GB200 NVL4 GPU system. Theo Mary was partially supported by projects managed by the French National Research Agency (ANR): France 2030 NumPEx Exa-MA (ANR-22-EXNU-0002), MixHPC (ANR-23-CE46-0005), and FPT-4 (ANR-24-CE46-7572).

References

[1] Ahmad Abdelfattah, Hartwig Anzt, Erik G Boman, Erin Carson, Terry Cojean, Jack Dongarra, Alyson Fox, Mark Gates, Nicholas J Higham, Xiaoye S Li, Jennifer Loe, Piotr Luszczek, Srikara Pranesh, Siva Rajamanickam, Tobias Ribizel, Barry F Smith, Kasia Swirydowicz, Stephen Thomas, Stanimire Tomov, Yaohung M Tsai, and Ulrike Meier Yang. 2021. A Survey of Numerical Linear Algebra Methods Utilizing Mixed-Precision Arithmetic. *The International Journal of High Performance Computing Applications* 35, 4 (2021), 344–369. doi:10.1177/10943420211003313

[2] Ahmad Abdelfattah, Stan Tomov, and Jack Dongarra. 2020. Investigating the Benefit of FP16-Enabled Mixed-Precision Solvers for Symmetric Positive Definite Matrices Using GPUs. In *Computational Science – ICCS 2020*, Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira (Eds.). Vol. 12138. 237–250. doi:10.1007/978-3-030-50417-5_18

[3] Sameh Abdulah, Allison H. Baker, George Bosilca, Qinglei Cao, Stefano Castruccio, Marc G. Genton, David E. Keyes, Zubair Khalid, Hatem Ltaief, Yan Song, Georgiy L. Stenchikov, and Ying Sun. 2024. Boosting Earth System Model Outputs And Saving PetaBytes in Their Storage Using Exascale Climate Emulators. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (Atlanta, GA, USA) *(SC '24)*. IEEE Press, Article 2, 12 pages. doi:10.1109/SC41406.2024.00008

[4] Sameh Abdulah, Hatem Ltaief, Ying Sun, Marc G. Genton, and David E. Keyes. 2018. ExaGeoStat: A High Performance Unified Software for Geostatistics on Manycore Systems. *IEEE Transactions on Parallel and Distributed Systems* 29, 12 (2018), 2771–2784. doi:10.1109/TPDS.2018.2850749

[5] Abdurauf Abdurakhmanov and Suhaib A. Fahmy. 2025. Exploring Microscaling MX Minifloat Systolic Arrays on FPGAs. *ACM Transactions on Reconfigurable Technology and Systems* 18, 4 (2025), 1–23. doi:10.1145/3773041

[6] Victoria Carrica, Rabab Alomairy, Evelyne Ringoot, and Alan Edelman. 2026. Hierarchical Precision and Recursion for Accelerating Symmetric Linear Solves on MXUs. arXiv:2601.08082 [cs.DC] doi:10.48550/arXiv.2601.08082

[7] Stef Cuyckens, Xiaoling Yi, Nitish Satya Murthy, Chao Fang, and Marian Verhelst. 2025. Efficient Precision-Scalable Hardware for Microscaling (MX) Processing in Robotics Learning. In *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. 1–7. doi:10.1109/ISLPED65674.2025.11261796

[8] Bita Darvish Rouhani, Daniel Lo, Ritchie Zhao, Ming Liu, Jeremy Fowers, Kalin Ovtcharov, Anna Vinogradsky, Sarah Massengill, Lita Yang, Ray Bittner, Alessandro Forin, Haishan Zhu, Taesik Na, Prerak Patel, Shuai Che, Lok Chand Koppaka, XIA SONG, Subhojit Som, Kaustav Das, Saurabh T, Steve Reinhardt, Sitaram Lanka, Eric Chung, and Doug Burger. 2020. Pushing the Limits of Narrow Precision Inferencing at Cloud Scale with Microsoft Floating Point. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33.

[9] Sean Fox, Seyedramin Rasoulinezhad, Julian Faraone, Philip Leong, et al. 2020. A Block Minifloat Representation for Training Deep Neural Networks. In *International Conference on Learning Representations*.

[10] Azzam Haidar, Harun Bayraktar, Stanimire Tomov, Jack Dongarra, and Nicholas J. Higham. 2020. Mixed-Precision Iterative Refinement Using Tensor Cores on GPUs to Accelerate Solution of Linear Systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476, 2243 (2020), 20200110. doi:10.1098/rspa.2020.0110

[11] Nicholas J. Higham and Theo Mary. 2022. Mixed Precision Algorithms in Numerical Linear Algebra. *Acta Numerica* 31 (2022), 347–414. doi:10.1017/s0962492922000022

[12] Nicholas J. Higham, Srikara Pranesh, and Mawussi Zounon. 2019. Squeezing a Matrix into Half Precision, with an Application to Solving Linear Systems. *SIAM Journal on Scientific Computing* 41, 4 (2019), A2536–A2551. doi:10.1137/18M1229511

[13] Hatem Ltaief, Rabab Alomairy, Qinglei Cao, Jie Ren, Lotfi Slim, Thorsten Kurth, Benedikt Dorschner, Salim Bougouffa, Rached Abdelkhalak, and David E. Keyes. 2024. Toward Capturing Genetic Epistasis From Multivariate Genome-Wide Association Studies Using Mixed-Precision Kernel Ridge Regression. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (Atlanta, GA, USA) *(SC '24)*. IEEE Press, Article 6, 12 pages. doi:10.1109/SC41406.2024.00012

[14] Theo Mary and Mantas Mikaitis. 2025. Error Analysis of Matrix Multiplication with Narrow Range Floating-Point Arithmetic. *SIAM Journal on Scientific Computing* 47, 4 (2025), B785–B800. doi:10.1137/24M1685109

[15] Alessio Netti, Yang Peng, Patrik Omland, Michael Paulitsch, Jorge Parra, Gustavo Espinosa, Udit Agarwal, Abraham Chan, and Karthik Pattabiraman. 2023. Mixed Precision Support in HPC Applications: What about Reliability? *J. Parallel and Distrib. Comput.* 181 (2023), 104746. doi:10.1016/j.jpdc.2023.104746

[16] Katsuhisa Ozaki, Takeshi Ogita, Shin'ichi Oishi, and Siegfried M. Rump. 2012. Error-Free Transformations of Matrix Multiplication by Using Fast Routines of Matrix Multiplication and Its Applications. *Numerical Algorithms* 59, 1 (2012), 95–118. doi:10.1007/s11075-011-9478-1

[17] Jie Ren, Hatem Ltaief, Sameh Abdulah, and David E Keyes. 2025. Accelerating Mixed-Precision Out-of-Core Cholesky Factorization with Static Task Scheduling. In *ISC High Performance Computing (40th International Conference)*. Prometeus GmBH. doi:10.23919/ISC.2025.11017733

[18] Bita Darvish Rouhani, Nitin Garegrat, Tom Savell, Ankit More, Kyung-Nam Han, Ritchie Zhao, Mathew Hall, Jasmine Klar, Eric Chung, Yuan Yu, Michael Schulte, Ralph Wittig, Ian Bratt, Nigel Stephens, Jelena Milanovic, John Brothers, Pradeep Dubey, Marius Cornea, Alexander Heinecke, Andres Rodriguez, Martin Langhammer, Summer Deng, Maxim Naumov, Paulius Micikevicius, Michael Siu, and Colin Verrilli. 2023. OCP Microscaling Formats (MX) Specification v1.0. https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf

[19] Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Stosic Dusan, Venmugil Elango, Maximilian Golub, Alexander Heinecke, Phil James-Roxby, Dharmesh Jani, Gaurav Kolhe, Martin Langhammer, Ada Li, Levi Melnick, Maral Mesmakhosroshahi, Andres Rodriguez, Michael Schulte, Rasoul Shafipour, Lei Shao, Michael Siu, Pradeep Dubey, Paulius Micikevicius, Maxim Naumov, Colin Verrilli, Ralph Wittig, Doug Burger, and Eric Chung. 2023. Microscaling Data Formats for Deep Learning. arXiv preprint arXiv:2310.10537.

[20] Ebby Samson, Naveen Mellempudi, Wayne Luk, and George A. Constantinides. 2024. Exploring FPGA Designs for MX and Beyond. In *Proceedings of International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE. doi:10.1109/FPL64840.2024.00049

[21] James Hardy Wilkinson. 1963. *Rounding errors in algebraic processes.* SIAM.

[22] Qiao Zhang, Rabab Alomairy, Dali Wang, Zhuowei Gu, and Qinglei Cao. 2025. Leveraging Hardware-Aware Computation in Mixed-Precision Matrix Multiply: A Tile-Centric Approach. In *Asynchronous Many-Task Systems and Applications: Third International Workshop, WAMTA 2025, St. Louis, MO, USA, February 19–21, 2025, Proceedings* (St. Louis, MO, USA). Springer-Verlag, Berlin, Heidelberg, 174–185. doi:10.1007/978-3-031-97196-9_15