Davis Math Lab Project Report, Spring 2026
Mixed-Precision Iterative Refinement for Symmetric Eigenpair Computation via Bordered Newton Iteration

Mark Soliman
Zhaojun Bai (Faculty Mentor)

June 4, 2026

Contents

1 Introduction

Modern computing hardware runs lower precisions much faster than higher ones. On current data-center accelerators such as NVIDIA's H100 and AMD's MI300X, single-precision (FP32) arithmetic runs at approximately twice the throughput of double-precision (FP64), and half-precision formats (FP16, BF16) on dedicated matrix-multiplication units run at factors ranging from roughly 16× to 60× over FP64 [1, 8]. This asymmetry has revived interest in *mixed-precision* algorithms, which perform the bulk of arithmetic at lower precision while preserving the accuracy of an FP64 result.

The foundational technique is iterative refinement (IR), introduced by Wilkinson [9] for the linear system $Ax = b$. The scheme is to compute an approximate x_0 cheaply, measure the residual $r = b - Ax_0$ accurately, solve $A\Delta = r$ cheaply, and update $x_1 = x_0 + \Delta$. Provided the cheap solver is not too inaccurate, each step reduces the error multiplicatively, and the iterates converge to an answer at working precision. Higham and Mary [5] survey the modern descendants of this idea, including the three-precision variant of Carson and Higham [3] that splits residual computation, factorization, and update into three separately-controllable precisions. Kelley [7] extended the framework to Newton's method for nonlinear systems, where the linear solve at each Newton step is performed in low precision, and observed empirically that the textbook backward-error bounds substantially overestimate the actual loss of accuracy.

In this paper we adapt the mixed-precision Newton framework to a different problem: refinement of approximate eigenpairs of a symmetric matrix $A \in \mathbb{R}^{n\times n}$. The vehicle is bordered Newton iteration [6], which reformulates the eigenvalue equation $Ax = \lambda x$ as a square nonlinear system in $(x, \lambda) \in \mathbb{R}^{n+1}$ and applies Newton's method. The inner linear solve at each step is a saddle-point system whose factorization

dominates the cost; we perform this solve in FP32 while keeping the residual computation and update in FP64.

Our contribution is empirical. We characterize, on a battery of symmetric test matrices spanning $\kappa(A) \in [10^1, 10^{13}]$, the convergence boundary of mixed-precision bordered Newton against an FP64-only baseline. We find that the standard convergence threshold $\kappa(A) \cdot u_{\text{factor}} \lesssim 1$, which follows from the general analysis of iterative refinement for linear systems, is overly pessimistic by approximately six orders of magnitude for the symmetric eigenvalue problem. Mixed precision matches FP64 up to $\kappa(A) \approx 10^{10}$ before failing sharply at $\kappa(A) \approx 10^{13}$. We argue, with reference to the Bauer-Fike theorem [2], that the relevant conditioning here is per-eigenpair, governed by eigenvalue separation, rather than the global $\kappa(A)$.

We additionally demonstrate the complete pipeline: starting from an FP32 eigendecomposition oracle, mixed-precision Newton refines each eigenpair to FP64 accuracy, yielding a roughly 10^8-fold improvement over the oracle while performing all heavy-lifting linear algebra in FP32. The pipeline is a working illustration of the central pitch of mixed-precision numerical linear algebra: FP32 cost, FP64 accuracy.

2 Progress

2.1 Bordered Newton iteration and mixed-precision variant

2.1.1 Bordered formulation

Given a symmetric $A \in \mathbb{R}^{n\times n}$, the eigenvalue problem

$$Ax = \lambda x, \quad x \neq 0, \tag{1}$$

has more unknowns than equations: there are n equations in the $n+1$ unknowns (x, λ). The extra equation needed to close the system is a normalization on x. Following Keller [6], we impose $\|x\|_2 = 1$ in the form

$$\tfrac{1}{2}(1 - x^T x) = 0. \tag{2}$$

The factor of $\frac{1}{2}$ is chosen so that the resulting Jacobian is symmetric. The augmented residual function $f : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is

$$f(x, \lambda) = \begin{bmatrix} Ax - \lambda x \\ \frac{1}{2}(1 - x^T x) \end{bmatrix}, \tag{3}$$

and the eigenvalue problem reduces to finding a root of f.

Differentiating gives the Jacobian

$$J_f(x, \lambda) = \begin{bmatrix} A - \lambda I & -x \\ -x^T & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \tag{4}$$

a symmetric *saddle-point matrix*. The upper-left block $A-\lambda I$ becomes singular as λ approaches an eigenvalue of A, but the full bordered matrix J_f stays nonsingular at any simple eigenpair (x^*, λ^*) [6]. This is the key advantage of the bordered formulation: it lets us apply Newton's method even though $A - \lambda I$ is singular at the target.

2.1.2 Newton iteration

Newton's method applied to $f(x, \lambda) = 0$ produces the iteration: given (x_k, λ_k), solve

$$J_f(x_k, \lambda_k)\, \Delta_k = -f(x_k, \lambda_k), \tag{5}$$

and update

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \Delta_k. \tag{6}$$

Provided (x_0, λ_0) lies in a sufficiently small neighborhood of a simple eigenpair, the iteration converges quadratically [6].

The dominant cost per iteration is the saddle-point solve (5), whose factorization requires $O(n^3)$ flops. The residual and update each cost only $O(n^2)$. This asymmetry is what makes the iteration a natural target for mixed precision: if the cubic-cost factorization can be done in a cheaper precision without spoiling the result, the savings dominate.

2.1.3 Mixed-precision variant

We adopt the two-precision Newton-IR scheme analyzed by Kelley [7]. Let u_w denote the working precision and u_f the factorization precision, with $u_f > u_w$. In our experiments, u_w corresponds to FP64 and u_f to FP32:

$$u_w = 2^{-52} \approx 1.1 \times 10^{-16}, \quad u_f = 2^{-23} \approx 5.96 \times 10^{-8}. \tag{7}$$

At each Newton step we (i) compute the residual $f(x_k, \lambda_k)$ in working precision, (ii) form J_f in working precision, cast it to factorization precision, and solve for Δ_k in factorization precision, then (iii) promote Δ_k back to working precision and apply the update. Algorithm 1 states the full procedure.

Algorithm 1 Mixed-precision bordered Newton

Require: symmetric $A \in \mathbb{R}^{n\times n}$, initial (x_0, λ_0), tolerance τ, max iterations K
Ensure: refined eigenpair (x, λ)

for $k = 0, 1, \ldots, K-1$ **do**
  $r_k \leftarrow \begin{bmatrix} Ax_k - \lambda_k x_k \\ \frac{1}{2}(1 - x_k^T x_k) \end{bmatrix}$ ▷ precision u_w
  if $\|r_k\| < \tau$ **then**
    break
  end if
  $J_k \leftarrow \begin{bmatrix} A - \lambda_k I & -x_k \\ -x_k^T & 0 \end{bmatrix}$ ▷ precision u_w
  Solve $\mathrm{cast}_{u_f}(J_k)\, \Delta_k^{(s)} = -\mathrm{cast}_{u_f}(r_k)$ ▷ precision u_f
  $\Delta_k \leftarrow \mathrm{cast}_{u_w}(\Delta_k^{(s)})$
  $(x_{k+1}, \lambda_{k+1}) \leftarrow (x_k, \lambda_k) + \Delta_k$ ▷ precision u_w
end for

Because J_f is symmetric indefinite, we use an LDL^T factorization for the inner solve. In MATLAB this is obtained via the backslash operator, which auto-detects symmetry and selects LDL^T for symmetric indefinite operands. Casting both J_k and r_k to FP32 forces the factorization and triangular solves to be performed entirely at the factor precision.

The classical iterative-refinement intuition applies. The FP32 solve produces a correction $\Delta_k^{(s)}$ contaminated by backward error of order $u_f \|J_k\|$. Because the next residual is computed in FP64, this error does not accumulate across iterations; each step starts from a faithful measurement of where the current iterate stands. The convergence rate of the loop depends on the condition number of J_f at the target, a topic we revisit in Section 2.2.

2.2 Error analysis and conditioning

We use the standard floating-point error model [4, 9]. For any two floating-point numbers x and y and any basic arithmetic operation $\mathrm{op} \in \{+, -, \times, \div\}$,

$$\mathrm{fl}(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)(1 + \delta), \qquad |\delta| \le u, \tag{8}$$

where u is the unit roundoff of the precision in question. For products and sums of n floating-point numbers we use the standard shorthand

$$\gamma_n := \frac{nu}{1 - nu}, \tag{9}$$

which bounds the cumulative effect of n rounding errors [4]. Both u and γ_n carry a subscript denoting the precision, so $u_{32} \approx 5.96 \times 10^{-8}$ and $u_{64} \approx 1.1 \times 10^{-16}$.

2.2.1 Iterative refinement for linear systems

Consider the linear system $Ax = b$ solved by computing an LU factorization $A \approx \widehat{L}\widehat{U}$ at precision u_f, performing the triangular solves at precision u_f, computing the residual $r = b - A\widehat{x}$ at precision u_w, and

updating $\widehat{x} \leftarrow \widehat{x} + \widehat{\Delta}$ at precision u_w, where $\widehat{\Delta}$ comes from another precision-u_f triangular solve. The classical analysis [4, 9] shows that the iteration converges as long as

$$\rho := c_n \, \kappa(A) \, u_f \;\ll\; 1, \tag{10}$$

where $\kappa(A) := \|A\| \cdot \|A^{-1}\|$ is the condition number of A and c_n is a modest polynomial factor in n. When this condition is satisfied, the iterates converge linearly at rate ρ, and the limit error is bounded by the working precision u_w rather than u_f. The cheap (low-precision) factorization is, in effect, traded for the cost of a few additional residual-and-update cycles, with no loss of final accuracy.

Carson and Higham [3] generalized this to three precisions; Kelley [7] extended the framework from linear systems to Newton's method for general nonlinear systems. In the nonlinear setting, the condition number of A in (10) is replaced by the condition number of the Jacobian J at the root being sought.

2.2.2 Application to bordered Newton

For our problem, the relevant operator is the saddle-point Jacobian $J_f(x^*, \lambda^*)$ at the target eigenpair. The naive prediction from (10) is that mixed-precision bordered Newton converges to working precision when $\kappa(J_f) \cdot u_f \ll 1$ and fails when $\kappa(J_f) \cdot u_f \gtrsim 1$.

The relationship between $\kappa(J_f)$ and the more familiar $\kappa(A)$ is worth explaining. The condition number $\kappa(J_f)$ depends on x^*, λ^*, and the gap between λ^* and the rest of the spectrum of A, not on $\kappa(A)$ directly. For a simple eigenvalue λ^* with spectral gap $g := \min_{\mu \neq \lambda^*} |\mu - \lambda^*|$, one has $\|J_f^{-1}\| = O(1/g)$, so

$$\kappa(J_f) = O\!\left(\frac{\|A\|}{g}\right). \tag{11}$$

For matrices like hilb(n), whose eigenvalues are tightly clustered near zero, g shrinks rapidly with n and $\kappa(J_f)$ tracks $\kappa(A)$ closely. For matrices with well-separated spectra, $\kappa(J_f)$ can remain modest even when $\kappa(A)$ is large.

2.2.3 Conditioning of symmetric eigenvalues

Symmetry gives us an additional advantage that does not appear in the general analysis. The Bauer-Fike theorem [2] states that for any symmetric A and any eigenvalue λ^* of A, the perturbed eigenvalue $\widetilde{\lambda}^*$ of $A + E$ satisfies

$$|\widetilde{\lambda}^* - \lambda^*| \leq \|E\|_2. \tag{12}$$

Each eigenvalue has condition number one with respect to additive perturbations of A, regardless of how ill-conditioned the matrix as a whole may be. A pure FP64 computation of any eigenpair therefore incurs forward error bounded by $\|A\| \cdot u_{64}$, independent of $\kappa(A)$.

The mixed-precision case is more delicate. The FP32 inner solve perturbs the Newton correction direction, not the matrix A itself, so (12) does not apply directly to the iteration as a whole. The convergence threshold of the outer iteration is still governed, in principle, by the IR bound $\kappa(J_f) \cdot u_f \lesssim 1$. However, this bound has been observed empirically [7] to be substantially pessimistic in practice; in his survey, Kelley notes that "the usual textbook estimates are very pessimistic and even the state-of-the-art estimates using probabilistic rounding analysis do not fully conform to experiments" [7, p. 192].

The empirical question we address in Section 2.3 is the location of the actual convergence boundary for our problem class. Mixed-precision bordered Newton matches FP64 accuracy up to $\kappa(A) \approx 10^{10}$ and fails sharply at $\kappa(A) \approx 10^{13}$, where the naive theory $\kappa(A) \cdot u_{32} \lesssim 1$ would predict failure already at $\kappa(A) \approx 1.7 \times 10^7$. We argue in Section 2.4 that this six-order-of-magnitude gap reflects both the Newton-specific contraction noted by Kelley and the symmetric-eigenvalue structure described above.

2.3 Numerical experiments

2.3.1 Setup

We implement Algorithm 1 in MATLAB R2026a. The working precision u_w corresponds to the native `double` type. The factor precision u_f is obtained by casting the saddle-point system to `single` before each

Table 1: Battery sweep results for FP64 versus mixed-precision (MP) bordered Newton. Errors are the maximum strict error across all n eigenpairs; iteration counts are means across eigenpairs; MP tgt is the fraction of eigenpairs for which MP converged to the correct eigenpair (eigenvector overlap above 0.95).

Matrix	n	$\kappa(A)$	FP64 err	MP err	FP64 it.	MP it.	MP tgt
`lehmer`	6	2.90×10^{1}	1.3×10^{-15}	5.7×10^{-14}	4.2	4.2	100%
`lehmer`	20	3.72×10^{2}	1.7×10^{-13}	5.6×10^{-13}	4.7	4.7	100%
`lehmer`	50	2.49×10^{3}	1.7×10^{-14}	7.1×10^{-15}	5.5	5.5	100%
`hilb`	4	1.55×10^{4}	6.2×10^{-17}	1.8×10^{-15}	4.5	4.5	100%
`hilb`	6	1.50×10^{7}	4.4×10^{-16}	3.7×10^{-15}	5.0	5.0	100%
`hilb`	8	1.53×10^{10}	1.3×10^{-15}	2.1×10^{-14}	5.0	5.8	100%
`hilb`	10	1.60×10^{13}	2.4×10^{-15}	4.7×10^{-6}	5.4	28.0	70%

backslash solve, which forces MATLAB to perform the LDL^T factorization and triangular solves entirely in FP32. The FP64 baseline is the same algorithm with all operations in `double`.

To probe the convergence boundary, we run the methods on a battery of seven symmetric test matrices from MATLAB's `gallery` collection, spanning condition numbers from $\kappa(A) \approx 30$ to $\kappa(A) \approx 1.6 \times 10^{13}$:

- `gallery('lehmer', n)` for $n \in \{6, 20, 50\}$: symmetric positive definite, condition number grows polynomially with n.
- `hilb(n)` for $n \in \{4, 6, 8, 10\}$: the Hilbert matrix, with condition number growing exponentially in n.

For each matrix and each reference eigenpair (x_j^*, λ_j^*) computed from MATLAB's `eig`, we form a starting guess by perturbing the reference. The eigenvector is perturbed additively, $x_0 = x_j^* + 0.01\,\xi$ with ξ a standard normal vector, then renormalized. The eigenvalue is perturbed multiplicatively, $\lambda_0 = \lambda_j^*(1 + 0.01\,\eta)$ with η standard normal, so the relative perturbation is constant across the spectrum. We chose the multiplicative form because some matrices (especially `hilb` at large n) have eigenvalues spanning many orders of magnitude, and an additive perturbation of size 0.01 would swamp the smallest eigenvalues entirely.

The convergence tolerance on the bordered residual is $\tau = 10^{-12}$ and the maximum iteration count is $K = 100$. To assess accuracy we report the strict error

$$e_j := |\lambda_j^* - \widehat{\lambda}_j|, \tag{13}$$

where $\widehat{\lambda}_j$ is the eigenvalue returned by Newton from the perturbed starting point near λ_j^*. We compare to the specific target λ_j^*, not the nearest reference. If Newton drifts to a different eigenpair than the target, this counts as a failure, not as accidental success. We also track the eigenvector overlap $|x_j^{*T} \widehat{x}_j|$ as a diagnostic for whether the iteration converged to the intended eigenpair.

2.3.2 The convergence boundary

Table 1 reports the per-matrix results, and Figure 1 visualizes $\max_j e_j$, the maximum strict error across eigenpairs, as a function of $\kappa(A)$. Three observations stand out.

First, the FP64 curve stays close to the working precision floor u_{64} across the entire range of $\kappa(A)$ tested, including the most ill-conditioned case $\kappa(A) \approx 10^{13}$. This sits roughly six orders of magnitude below the general textbook prediction $\kappa(A) \cdot u_{64}$, which would have predicted errors as large as 10^{-3} for `hilb`(10). The gap is explained by the Bauer-Fike result of Section 2.2: each individual eigenvalue of a symmetric matrix is conditioned by $\|A\|$, not by $\kappa(A)$, so the per-eigenvalue forward error stays bounded by $\|A\| \cdot u_{64}$ regardless of how ill-conditioned the matrix is as a whole.

Second, the MP curve tracks the FP64 curve closely from $\kappa(A) \approx 30$ all the way to $\kappa(A) \approx 10^{10}$. Over this range, which spans nine orders of magnitude, mixed-precision Newton produces eigenvalues at full working precision despite performing every factorization and triangular solve in FP32. The naive convergence threshold $\kappa(A) \cdot u_{32} \lesssim 1$ would have predicted failure beginning at $\kappa(A) \approx 1.7 \times 10^{7}$, just past `hilb`(6) in our battery. Mixed precision actually keeps up an additional three orders of magnitude beyond this prediction.

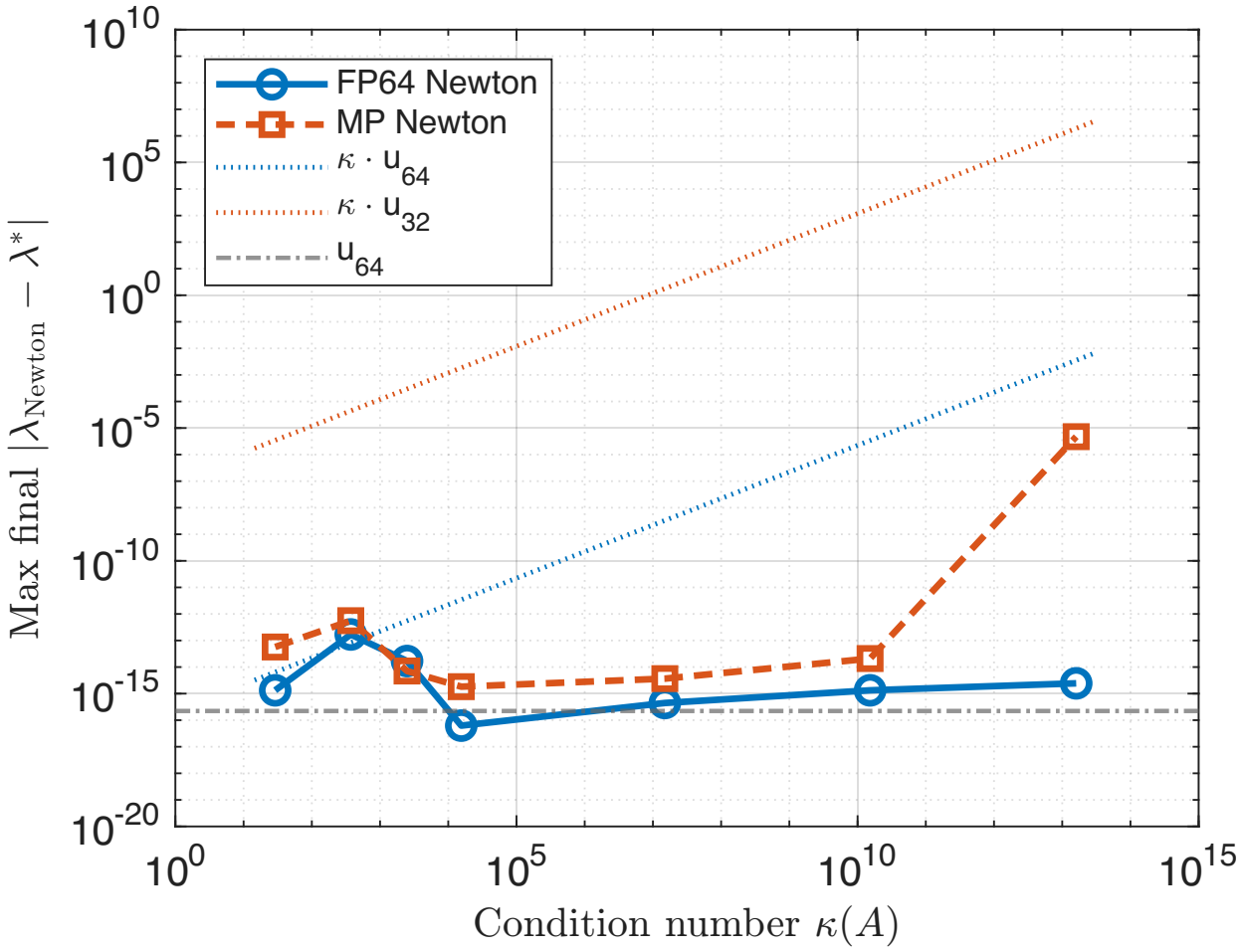

Figure 1: Maximum final eigenvalue error across all eigenpairs as a function of $\kappa(A)$, for FP64 (blue circles) and mixed-precision (red squares) bordered Newton. Reference lines show the textbook predictions $\kappa(A) \cdot u_{64}$ (blue dotted), $\kappa(A) \cdot u_{32}$ (red dotted), and the working precision floor u_{64} (gray dash-dot). FP64 stays at the floor across the entire range. MP matches FP64 up to $\kappa(A) \approx 10^{10}$, then fails sharply at $\kappa(A) \approx 10^{13}$.

Third, MP fails sharply at the rightmost point, `hilb`(10) with $\kappa(A) \approx 1.6 \times 10^{13}$. The MP max error jumps to 4.7×10^{-6}, the target-hit rate drops to 70%, and the mean iteration count rises from roughly 5 to 28. The iteration count tells us that for the eigenpairs that do converge, MP requires far more steps than usual to compensate for the noisier FP32 corrections; the target-hit rate tells us that for the remaining 30%, the corrections are too noisy to keep Newton in the correct basin of attraction at all.

Taken together, mixed-precision bordered Newton enjoys two separate sources of robustness compared to general mixed-precision iterative refinement: the empirical contraction noted by Kelley [7], that the Newton outer loop tolerates much more FP32 backward error than the worst-case bound suggests, and the Bauer-Fike effect, that individual eigenvalue forward errors are governed by $\|A\|$ rather than $\kappa(A)$. Both factors push the empirical failure boundary far past the naive prediction.

2.3.3 End-to-end pipeline

The matrix-battery experiment compares FP64 and MP on a level playing field, both starting from the same near-truth initial guess. To assess the full mixed-precision workflow, we additionally run an end-to-end demonstration on `lehmer`(20), well within the convergence regime established above.

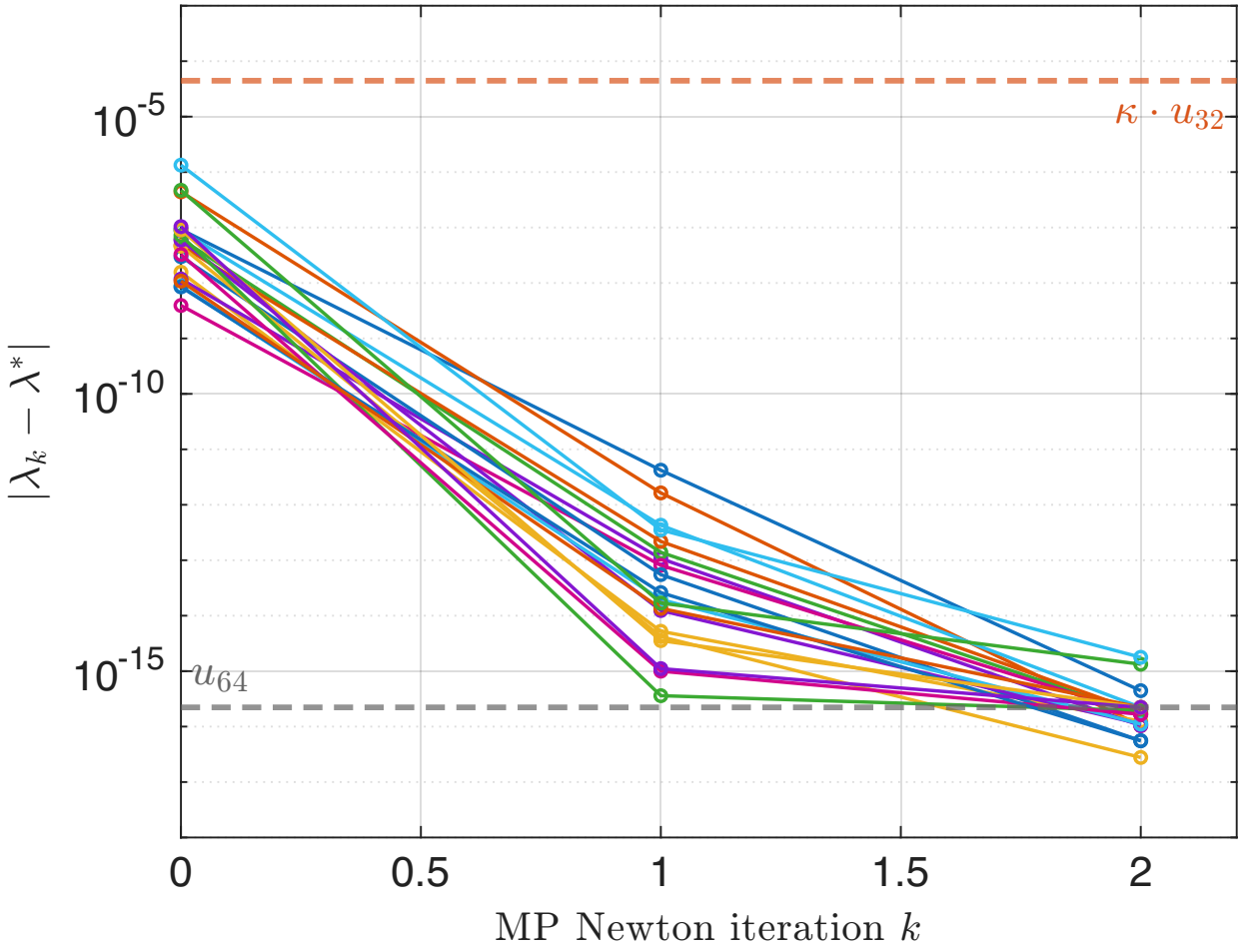

Figure 2: Per-eigenpair refinement trajectories for the end-to-end pipeline on `lehmer`(20). Each of the twenty curves plots the eigenvalue error $|\lambda_k - \lambda_j^*|$ across MP Newton iterations, starting from an FP32 oracle initial guess. The upper red dashed line marks the theoretical FP32 worst-case bound $\kappa(A) \cdot u_{32}$; the lower gray dashed line marks the FP64 working precision floor u_{64}. All twenty eigenpairs refine from oracle accuracy to working precision in two MP Newton iterations.

The pipeline has two steps. First, we obtain an initial eigendecomposition by calling `eig(single(A))`, an FP32 eigendecomposition oracle. Casting A to `single` forces the entire LAPACK eigensolver underneath `eig` to run in FP32. Second, for each of the 20 eigenpairs returned by the oracle, we promote (x_0, λ_0) back to FP64 and feed them to mixed-precision bordered Newton as the starting guess. The result is twenty refined eigenpairs at FP64 accuracy.

Figure 2 shows the per-eigenpair refinement trajectories. Each curve plots $|\lambda_k - \lambda_j^*|$ across MP Newton iterations $k = 0, 1, 2$ for a different eigenpair of `lehmer`(20). The horizontal reference lines mark the FP32 worst-case bound $\kappa(A) \cdot u_{32} \approx 4.4 \times 10^{-5}$ and the FP64 working precision floor $u_{64} \approx 2.2 \times 10^{-16}$.

At iteration 0, the trajectories cluster between $\sim 10^{-7}$ and $\sim 10^{-6}$. This is roughly one to two orders of magnitude below the theoretical $\kappa(A) \cdot u_{32}$ bound at the top of the plot, another instance of the symmetric eigenvalue advantage from Section 2.2: even the FP32 eigensolver itself does better than its general worst-case bound on this problem class. By iteration $k = 2$, all twenty trajectories sit within a factor of a few of u_{64}.

Quantitatively, the mean initial error across the 20 eigenpairs is 1.52×10^{-7}, and the mean final error is 2.67×10^{-16}. The improvement factor is approximately 5.7×10^{8}. The entire pipeline performs its expensive

$O(n^3)$ work, both the eigendecomposition oracle and the inner saddle-point factorizations at each Newton step, in FP32 arithmetic. The only FP64 work is the bordered residual and the state update, both $O(n^2)$ per iteration. The pipeline achieves FP64 accuracy at FP32 cost.

2.4 Summary of findings

We posed a single empirical question: at what value of $\kappa(A)$ does mixed-precision bordered Newton stop matching its FP64 counterpart? The naive answer, from the general convergence theory of iterative refinement applied to the saddle-point system, is $\kappa(A) \cdot u_{32} \lesssim 1$, predicting failure at $\kappa(A) \approx 1.7 \times 10^7$. The actual answer, from the matrix battery sweep of Section 2.3, is $\kappa(A) \approx 10^{13}$. The naive prediction is too pessimistic by approximately six orders of magnitude on the symmetric eigenvalue problem.

Two effects compound to push the empirical failure boundary so far past the naive prediction.

The first is specific to Newton iterations and was observed in general by Kelley [7]. When the inner linear solve at a Newton step is performed in low precision, the resulting correction Δ_k is contaminated by backward error of order $u_f \|J_k\|$, but the next residual is still computed in full working precision. The outer Newton loop therefore behaves like an iterative refinement of the eigenpair itself: each step starts from a faithful measurement of where the current iterate stands, regardless of how the previous correction was computed. The naive bound $\kappa \cdot u_f \lesssim 1$ assumes the worst-case backward error is realized on every step, but in practice the Newton contraction absorbs most of the slack.

The second effect is specific to the symmetric eigenvalue problem. By the Bauer-Fike theorem, each eigenvalue of a symmetric matrix has condition number one with respect to additive perturbations of A: $|\widetilde{\lambda} - \lambda| \leq \|E\|$ for a perturbation $A + E$. The forward error in any computed eigenvalue is therefore bounded by $\|A\| \cdot u$ rather than $\kappa(A) \cdot u$, regardless of how ill-conditioned the matrix is as a whole. This advantage shows up at every level of our pipeline, in the FP32 eigensolver oracle (which beats $\kappa(A) \cdot u_{32}$ by one to two orders of magnitude), in the FP64 baseline (which stays at u_{64} across the full κ range), and in the mixed-precision iteration (which keeps up with the baseline until $\kappa(A) \cdot u_{32}$ becomes large enough to overwhelm the Newton contraction itself).

We emphasize that $\kappa(A)$ is the wrong axis in principle. The conditioning of bordered Newton at the target eigenpair depends on the spectral gap of λ^*, not on the matrix condition number directly. Hilbert matrices are useful here precisely because their spectral gaps shrink rapidly with n, so $\kappa(A)$ tracks the bordered-Jacobian conditioning closely. For matrices whose ill-conditioning comes from a different source, for instance a single tiny eigenvalue but otherwise well-separated spectrum, $\kappa(A)$ would overstate the difficulty of refining the well-separated eigenpairs and understate the difficulty of refining the tiny one. The proxy is useful here, but it is a proxy.

Mixed-precision iterative refinement is one of the central tools of modern numerical linear algebra, and its application to nonlinear problems through Newton's method, recently surveyed by Kelley, is an active area of research. We have shown that for the symmetric eigenvalue problem, the combination of the Newton contraction and the Bauer-Fike conditioning of symmetric eigenvalues pushes the empirical convergence boundary roughly six orders of magnitude past the naive prediction. Mixed-precision bordered Newton, used as the refinement step in a pipeline initialized by an FP32 eigendecomposition oracle, delivers FP64-accurate eigenpairs at FP32 computational cost across a range of conditioning that covers most matrices arising in practice. The result is a clean instance of the broader thesis that structured problems can do far better than worst-case error analysis predicts when given the chance.

3 Future Plans

Several directions follow naturally from the present study.

The most direct extension is a three-precision variant of Algorithm 1, following Carson and Higham [3]: factor in FP16, compute residuals in FP64, work in FP32. MATLAB's Fixed-Point Designer toolbox provides a `half` type that would make a prototype implementation straightforward, though on CPUs without native FP16 arithmetic the type acts only as storage and computations fall back to FP32. The additional speedup from FP16 matrix-multiplication units on the H100 and MI300X would require a GPU-targeted implementation. The interesting question is whether the structural advantages identified here persist when the factor precision drops by another nine bits.

A second direction is generalized eigenvalue problems $Ax = \lambda Bx$ with symmetric positive definite B. The bordered formulation extends cleanly, and the Bauer-Fike argument generalizes through a B-weighted inner product. The mixed-precision behavior on this family is not, to our knowledge, characterized in the literature.

A third direction concerns other structured nonlinear problems where the Jacobian has favorable conditioning at the solution. Newton-based eigenpair refinement is one example; iteratively reweighted least squares, generalized linear model fitting, and multinomial logistic regression are others. The block structure of the softmax Hessian in particular suggests an interesting question: can the block-diagonal pattern of within-class curvature be exploited to design a mixed-precision variant whose convergence depends only on the per-block conditioning rather than on the global Hessian condition number?

4 References

[1] Advanced Micro Devices, Inc. AMD Instinct MI300X Accelerator Data Sheet. `https://www.amd.com/content/dam/amd/en/documents/instinct-tech-docs/data-sheets/amd-instinct-mi300x-data-sheet.pdf`, 2023. Accessed May 2026.

[2] F. L. Bauer and C. T. Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2:137–141, 1960.

[3] Erin Carson and Nicholas J. Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018. doi: 10.1137/17M1140819.

[4] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, 2nd edition, 2002.

[5] Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022. doi: 10.1017/S0962492922000022.

[6] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P. H. Rabinowitz, editor, *Applications of Bifurcation Theory*, pages 359–384. Academic Press, New York, 1977.

[7] C. T. Kelley. Newton's method in mixed precision. *SIAM Review*, 64(1):191–211, 2022. doi: 10.1137/20M1342902.

[8] NVIDIA Corporation. NVIDIA H100 Tensor Core GPU Datasheet. `https://www.nvidia.com/en-us/data-center/h100/`, 2023. Accessed May 2026.

[9] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice-Hall, Englewood Cliffs, NJ, 1963.