# Recent Advances in Mixed-Precision (Hybrid) Iterative Methods

*Eda Oktay, Erin Carson*

Abstract

Mixed-precision hardware has recently become commercially available, and more than 25% of the supercomputers in the TOP500 list now have mixed-precision capabilities. Using lower precision in algorithms can be beneficial in terms of reducing both computation and communication costs. According to the recently developed mixed-precision benchmark, HPL-MxP, multiple supercomputers today already exceed exascale ($10^{18}$ floating-point operations per second) performance through the use of mixed-precision computations. Many current efforts are focused on developing mixed-precision numerical linear algebra algorithms, which will lead to speedups in real applications. These new algorithms are increasingly being implemented in libraries, such as the MAGMA library.

Motivated by this, the aim of this talk is to discuss recent advances in developing and analyzing mixed-precision variants of iterative methods. Iterative methods for solving linear systems and least squares problems are useful in practice when the coefficient matrix is large and sparse or not explicitly stored and/or when accuracy less than machine precision is sufficient. An iterative method starts with an initial guess and then iteratively improves the solution to the desired accuracy. One can use stationary methods, Krylov subspace methods, or some hybrid approach, depending on the problem. We focus on *hybrid methods*, where we use a Krylov subspace method as an inner solver of a variant of Newton's approach (stationary method), such as RQI and iterative refinement.

Iterative methods can be used to improve the accuracy of solutions to least squares (LS) problems $min_x\|b - Ax\|_2$, where $A \in \mathbb{R}^{m\times n}$. Using the QR factorization $A = [Q_1 \quad Q_2][R \quad 0]^T$, the solution to the LS problem is given by $x = U^{-1}Q_1^T b$ and the residual by $r = \|b - Ax\|_2 = \|Q_2^T b\|_2$. The LS problem can also be solved via the normal equations, $A^TAx = A^Tb$, which are equivalent to the augmented system [1]

$$\underbrace{\begin{bmatrix} I^{m\times m} & A \\ A^T & 0 \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} r \\ x \end{bmatrix}}_{\tilde{x}} = \underbrace{\begin{bmatrix} b \\ 0 \end{bmatrix}}_{\tilde{b}} \quad \text{or} \quad \tilde{A}\tilde{x} = \tilde{b}.$$

If $m > n$, then the system is called overdetermined, and if $m < n$, it is underdetermined. Weighted LS (WLS) is used when there are discrepant rows in $A$. In this case, weights can be assigned to these rows to minimize discrepancy. In classical least squares, there is an assumption that perturbations are confined to the vector $b$. This is not necessarily realistic in practice. If $A$ and $b$ may both be perturbed ($\hat{A}, \hat{b}$, respectively) so that $\hat{b}$ is in column space of $\hat{A}$, this problem is called total LS (TLS).

Krylov subspace methods work by selecting approximate solutions from a Krylov subspace. The search space is formed via nested Krylov subspaces, and the solution is obtained from a sequence of projections onto the search space. Although these solvers can be fast and/or stable, for large problems, they may not be memory efficient and slow down performance. To speed up and exploit parallelism, techniques such as mixed-precision can be used.

Error analysis is important for determining how rounding errors propagate in computations and identifying potential sources of amplification. For a function $f : \mathbb{R}^n \to \mathbb{R}^n$, the backward error in the approximation $y$ to $f(x)$ is the smallest $\Delta x$ such that $y = f(x + \Delta x)$, i.e., [10]

$$\eta(y) = \min\{\epsilon : y = f(x + \Delta x), \|\Delta x\| \leq \epsilon\|x\|\}.$$

Backward error analysis [9] aims to derive a bound on the backward error. If the backward error is small, then we say the algorithm is backward stable. The forward error measures the difference between the computed and the exact solution. As defined in [10], the relative forward error of $y \approx f(x)$ can bounded in terms of the relative backward error $\eta(y)$ by

$$\frac{\|y - f(x)\|}{\|f\|} \leq cond(f, x)\eta(y) + O(\eta(y))^2,$$

where

$$cond(f, x) = lim_{\epsilon \to 0} \sup_{\|\Delta x\| \leq \epsilon \|x\|} \frac{\|f(x + \Delta x) - f(x)\|}{\epsilon \|f(x)\|}$$

is the condition number, which measures the sensitivity of the solution to small perturbations in the input data.

### Mixed-precision Rayleigh quotient iteration for total least squares problems

We first focus on the use of Rayleigh quotient iteration (RQI) to solve the TLS problem, which is the approach advocated in [2] for large-scale problems, and introduce a mixed-precision variant of the RQI-PCGTLS algorithm (RQI-PCGTLS-MP) [8]. This approach solves the eigenvalue problem

$$\begin{bmatrix} A^T A & A^T b \\ b^T A & b^T b \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} = \lambda \begin{bmatrix} x \\ -1 \end{bmatrix}$$

to find $x = x_{TLS}$, where $\lambda = \sigma_{n+1}^2$, and $\sigma_{n+1}^2$ is the smallest singular value of $[A, b]$. Our approach potentially decreases the computational cost of RQI-PCGTLS by using up to three different precisions in the algorithm. Moreover, to enable the use of lower precision for more ill-conditioned systems, we use the R-factor from the Householder QR factorization of $A$ instead of the Cholesky factorization of $A^T A$ within RQI-PCGTLS-MP. We discuss the convergence and accuracy of our algorithm and derive two theoretical constraints on the precision that can be used for the construction of the preconditioner within the inner solver. To evaluate to what extent the computational cost can be reduced by using the mixed-precision variant with Householder QR factorization, we construct a performance model. Our numerical experiments and performance model show that one can get up to 4× speedup while keeping the working precision accuracy when fp16 is used in computing QR factors.

### GMRES-based iterative refinement and its variants

Another variant of Newton's method is the iterative refinement (IR) algorithm. As RQI, IR algorithms require a linear solver in each outer iteration. The standard IR (we refer to as SIR) algorithm in [9] first computes the initial approximation using Gaussian elimination with partial pivoting and uses approximate LU factors of $A$ to solve for the correction term which then refines the current solution. To increase the range of problems that can be solved with IR, a Krylov subspace method, such as preconditioned GMRES, can be used to solve the linear systems as in RQI-PCGTLS; this three-precision approach is called GMRES-IR [3]. GMRES-IR uses precisions with unit round-off $u_f$ for LU factorization, $u_r$ for residual computation, and $u$ for storing data and solution. For stability analysis of methods such as IR variants, we can derive forward and error bounds under a constraint on the conditioning of the coefficient matrix, $\kappa(A)$. For a non-singular square matrix, the condition number is defined as $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$ with the associated norm

$p$. As long as $\kappa_\infty(A) \leq u^{-1/2}u_f^{-1}$ and $u_r = u^2$, GMRES-IR provides accurate solutions with the forward and (normwise) backward errors

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \approx \mathcal{O}(u) \quad \text{and} \quad \frac{\|b - A\hat{x}\|_\infty}{\|A\|_\infty\|x\|_\infty + \|b\|_\infty} \approx \mathcal{O}(u),$$

respectively, while SIR is guaranteed to have this forward error only if $\kappa_\infty(A) \leq u_f^{-1}$ and $u_r = u^2$.

GMRES-IR can be much more expensive than SIR, depending on the number of iterations performed. One way to speed up the convergence of the GMRES solver is using recycling. In an effort to reduce the overall computational cost of the GMRES solves within GMRES-IR, we introduce a recycled GMRES-based iterative refinement algorithm called RGMRES-IR [6]. The algorithm starts with computing the LU factors of $A$ and computing the initial approximate solution in the same manner as GMRES-IR. Instead of preconditioned GMRES, however, the algorithm uses preconditioned GCRO-DR to solve the correction equation. In the RGMRES-IR algorithm, as in GMRES-IR, we use three precisions. Numerical experiments show that RGMRES-IR decreases the total GMRES iterations performed, especially when the matrix is badly conditioned. Even when GMRES-IR cannot converge, we observe that our variant can still converge.

Overdetermined standard least squares problems can be solved by using mixed-precision within the iterative refinement approach. It has been shown that mixed-precision GMRES-IR can also be used, in an approach termed GMRES-LSIR [4]. GMRES-LSIR solves the augmented system using GMRES preconditioned by a preconditioner $M$ computed using the QR factors of $A$:

$$M = \begin{bmatrix} \alpha I & Q_1 U \\ U^T Q_1^T & 0 \end{bmatrix},$$

where $A = Q_1 U$ is the thin QR factorization of $A$. As long as $\kappa_\infty(A) \leq u^{-1/2}u_f^{-1}$, and assuming $u_r = u^2$, GMRES-LSIR provides $\mathcal{O}(u)$ backward and forward error. Furthermore, using the left preconditioner $M$, the conditioning of the preconditioned augmented matrix can be bounded by

$$\kappa_\infty(M^{-1}\tilde{A}) \lesssim (1 + 2m\sqrt{n}\tilde{\gamma}_{mn}^f \kappa_\infty(A))^2, \quad \text{where} \quad \tilde{\gamma}_{mn}^f = \frac{cmn}{1 - mnu_f},$$

and $c$ is a small constant. In practice, we often encounter types of least squares problems beyond standard least squares, including the WLS problem $\min_x \|D^{1/2}(b - Ax)\|_2$, where $D^{1/2}$ is a diagonal matrix of weights, which is possibly ill-conditioned. WLS problems can be solved via the normal equations or the corresponding augmented system,

$$A^T D A x = A^T D b \quad \text{and} \quad \begin{bmatrix} \alpha D^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \alpha^{-1} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

respectively, where $y = D(b - Ax)$, $\alpha$ is the scaling factor for stability. We present the FGMRES-WLSIR algorithm, a variant of GMRES-LSIR for solving WLS problems using flexible GMRES (FGMRES), and discuss and analyze two different preconditioners [5]; a left preconditioner and a block diagonal split preconditioner,

$$M_l = \begin{bmatrix} \alpha D^{-1} & Q\hat{R} \\ \hat{R}^T Q^T & 0 \end{bmatrix}, \text{ and } M_b = \begin{bmatrix} \alpha D^{-1} & 0 \\ 0 & \hat{C} \end{bmatrix},$$

respectively, where $\hat{C} \approx \alpha^{-1} A^T D A$ is a symmetric positive definite approximation to the Schur complement.

### Multistage mixed-precision iterative refinement

In some cases, SIR can fail depending on the conditioning of the matrix and the precisions used. However, using GMRES-IR can be more expensive since one GMRES-IR iteration is more expensive than one SIR iteration. To benefit from both approaches and their variants, we propose a multistage IR approach (MSIR) to reduce the computation cost while improving applicability [7]. Our approach automatically switches between solvers and precisions if slow convergence (of the refinement scheme itself or of the inner GMRES solves) is detected using stopping criteria. With MSIR we attempt to use "stronger" solvers before resorting to increasing the precision of the factorization, and when executing a GMRES-based refinement algorithm, we modify the stopping criteria to also restrict the number of GMRES iterations per refinement step. Our experiments show that since the algorithmic variants often outperform what is dictated by the theoretical condition number constraints there can be an advantage to first trying other solvers before resorting to increasing the precision and refactorizing.

## References


[1] Åke Björck (1967) *Iterative refinement of linear least squares solutions i.* BIT Numerical Mathematics, 7(4):257–278.

[2] Åke Björck, Pinar Heggernes, and Pontus Matstoms (2000) *Methods for large scale total least squares problems.* SIAM Journal on Matrix Analysis and Applications, 22(2):413–429.

[3] Erin Carson and Nicholas J. Higham (2017) *A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems.* SIAM Journal on Scientific Computing, 39(6):A2834–A2856.

[4] Erin Carson, Nicholas J. Higham, and Srikara Pranesh (2020) *Three-precision GMRES-based iterative refinement for least squares problems.* SIAM Journal on Scientific Computing, 42(6):A4063–A4083.

[5] Erin Carson and Eda Oktay (2024) *Mixed precision FGMRES-based iterative refinement for weighted least squares.* arXiv preprint arXiv:2401.03755.

[6] Eda Oktay and Erin Carson (2022) *Mixed precision GMRES-based iterative refinement with recycling.* In Jan Chleboun and Pavel Kůs and Jan Papež and Miroslav Rozložník and Karel Segeth and Jakub Šístek, editors, Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar, pp.149–162. Institute of Mathematics CAS.

[7] Eda Oktay and Erin Carson (2022) *Multistage mixed precision iterative refinement.* Numerical Linear Algebra with Applications, 29(4):e2434.

[8] Eda Oktay and Erin Carson (2024) *Mixed precision Rayleigh quotient iteration for total least squares problems.* Numerical Algorithms, 96: 777–798.

[9] James Hardy Wilkinson (1963) *Rounding errors in algebraic processes.* Prentice-Hall.

[10] Nicholas J. Higham (2002) *Accuracy and stability of numerical algorithms.* SIAM.