Electronic Transactions on Numerical Analysis.
Volume 60, pp. 40–58, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol60s40

ETNA
Kent State University and
Johann Radon Institute (RICAM)

THE STABILITY OF SPLIT-PRECONDITIONED FGMRES IN FOUR PRECISIONS*

ERIN CARSON† AND IEVA DAUŽICKAITĖ†

Abstract. We consider the split-preconditioned FGMRES method in a mixed-precision framework, in which four potentially different precisions can be used for computations with the coefficient matrix, application of the left preconditioner, application of the right preconditioner, and the working precision. Our analysis is applicable to general preconditioners. We obtain bounds for the backward and forward errors in the split-preconditioned FGMRES method. Our analysis further provides insight into how the various precisions should be chosen; under certain assumptions, a suitable selection guarantees a backward error of the order of the working precision.

Key words. mixed precision, FGMRES, iterative methods, roundoff error, split-preconditioned

AMS subject classifications. 65F08, 65F10, 65F50, 65G50, 65Y99

1. Introduction.

We consider the problem of solving a linear system of equations

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsymmetric and $x, b \in \mathbb{R}^n$. When A is large and sparse, the iterative generalised minimal residual method (GMRES) or its flexible variant (FGMRES) are often used for solving (1.1); see, for example, [16]. In these and other Krylov subspace methods, preconditioning is an essential ingredient. Given a preconditioner $P = M_L M_R$, the problem (1.1) is transformed to

$$M_L^{-1} A M_R^{-1} \tilde{x} = M_L^{-1} b, \qquad \text{where} \qquad M_R^{-1}\tilde{x} = x. \tag{1.2}$$

Note that a particular strength of FGMRES is that it allows the right preconditioner to change throughout the iterations. Although for simplicity, we consider the case here where the preconditioners are static, our results could be extended to allow dynamic preconditioning.

The emergence of mixed-precision hardware has motivated work in developing mixed-precision algorithms for matrix computations; see, e.g., the recent surveys [1, 10]. Modern GPUs offer double, single, half, and even quarter precision, along with specialized tensor core instructions; see, e.g., [14]. The use of lower precision can offer significant performance improvements, although this comes at a numerical cost. With fewer bits, we have a greater unit roundoff and a smaller range of representable numbers. The goal is thus to selectively use low precision in algorithms such that performance is improved without adversely affecting the desired numerical properties.

Mixed-precision variants of GMRES and FGMRES with different preconditioners have been proposed and analyzed in multiple papers. Arioli and Duff [4] analyzed a two-precision variant of FGMRES in which the right-preconditioner is constructed using an LU decomposition computed in single precision and applied in either single or double precision and other computations are performed in double precision. They proved that in this setting, a backward error of the order of double precision is attainable.

*Received March 23, 2023. Accepted November 26, 2023. Published online on February 12, 2024. Recommended by D. Szyld. Co-funded by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, and by the European Union (ERC, inEXASCALE, 101075632). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

†Faculty of Mathematics and Physics, Charles University, Sokolovská 366/84, 180 00, Prague, Czech Republic ({carson, dauzickaite}@karlin.mff.cuni.cz).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The authors of [13] develop a mixed-precision variant of left-preconditioned GMRES in a mix of single and double precisions, requiring only a few operations to be performed in double precision. Their numerical experiments show that they can obtain a backward error of the level of double precision. Variants of left-preconditioned GMRES using various numbers of precisions have been analyzed as inner solvers within GMRES-based iterative refinement for solving linear systems of equations; Vieublé [17] analyzed left-preconditioned GMRES in four precisions with a general preconditioner, following the earlier works [6] and [3], which analyzed left-preconditioned GMRES with an LU preconditioner in two and three precisions, respectively. In general, different precisions can be used for computing the preconditioner, matrix-vector products with A, matrix-vector products or solves with the general preconditioner(s) and the remaining computations. We refer the readers to the recent surveys [1, 10] for other examples.

The structure of some problems and/or application requirements makes it desirable to construct and apply a split-preconditioner rather than left or right ones alone. For example, the condition number of a split-preconditioned matrix can be significantly smaller than when the same preconditioner is applied on the left [7, Section 3.2]. Such preconditioning is usually used for symmetric problems solved via short-recurrence symmetric solvers such as MINRES. However, MINRES may be inferior compared to GMRES when a high-accuracy solution for an ill-conditioned problem is required [7]. We also emphasize that analyzing split-preconditioning provides a uniform framework for analyzing cases with full left- or full right-preconditioning. The stability of split-preconditioned GMRES and FGMRES has not been analyzed in either uniform or mixed precision. The work [5] showed that uniform precision FGMRES with a specific right-preconditioner is backward stable while this is not the case for GMRES and that FGMRES is more robust than GMRES. We thus focus on split-preconditioned FGMRES in this paper and develop a mixed-precision framework allowing for four potentially different precisions for the following operations: computing matrix-vector products with A, applying the left-preconditioner M_L, applying the right-preconditioner M_R, and all other computations.

FGMRES computes a series of approximate solutions x_k from Krylov subspaces to (1.1). The Arnoldi method is employed to generate the basis for the Krylov subspaces like in GMRES, but FGMRES stores the right-preconditioned basis as well. The particular algorithm is shown in Algorithm 1. Our analysis considers general preconditioners, only requiring an assumption on the error in applying its inverse to a vector, and is thus widely applicable.

The paper is outlined as follows. We bound the backward errors in Section 2 while also providing guidance for setting the four precisions such that a backward error to the desired level is attainable. To make the results of the analysis more concrete, in Section 3 we bound the quantities involved for the example of LU preconditioners and then present a set of numerical experiments on both dense problems and problems from SuiteSparse [8]. In Section 4 we make concluding remarks.

2. Finite-precision analysis of FGMRES in four precisions. From the Rigal-Gaches Theorem (see [9, Theorem 7.1]), the normwise relative backward error is given by

$$\min\{\varepsilon : (A+\Delta A)x_k = b+\Delta b, \|\Delta A\| \le \varepsilon\|A\|, \|\Delta b\| \le \varepsilon\|b\|\} = \frac{\|r_k\|}{\|A\|\|x_k\| + \|b\|},$$

where $r_k = b - Ax_k$. We aim to bound this quantity when x_k is the approximate solution produced by Algorithm 1. To account for various ways in which the preconditioner can be computed and some constraints on A resulting in the need for different precisions, we assume that

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 1 Split-preconditioned FGMRES for solving (1.1) in four precisions.

Input: matrix $A \in \mathbb{R}^{n\times n}$, right-hand side $b \in \mathbb{R}^n$, preconditioner $P = M_L M_R$, maximum number of iterations *maxit*, convergence tolerance τ, precisions u, u_A, u_L, and u_R
Output: approximate solution x_k

initialize x_0
$t = Ax_0$ ▷ u_A
$t^{(p)} = M_L^{-1}t$ ▷ u_L
$b^{(p)} = M_L^{-1}b$ ▷ u_L
$r_0 = b^{(p)} - t^{(p)}$ ▷ u
$\beta = \|r_0\|$; $v_1 = r_0/\beta$ ▷ u
$k = 0$; *convergence* = *false*
while *convergence* = *false* and $k < maxit$ **do**
  $k = k + 1$
  $z_k = M_R^{-1}v_k$ ▷ u_R
  $s = Az_k$ ▷ u_A
  $w = M_L^{-1}s$ ▷ u_L
  for $i = 1, \ldots, k$ **do**
    $h_{i,k} = v_i^T w$ ▷ u
    $w = w - h_{i,k}v_i$ ▷ u
  end for
  $h_{k+1,k} = \|w\|$ ▷ u
  $Z_k = [z_1, \ldots, z_k]$; $H_k = \{h_{i,j}\}_{1\le i\le j+1; 1\le j\le k}$
  $y_k = \arg\min_y \|\beta e_1 - H_k y\|$ ▷ u
  if $\|\beta e_1 - H_k y_k\| \le \tau\beta$ **then**
    $x_k = x_0 + Z_k y_k$ ▷ u
    $t = Ax_k$ ▷ u_A
    $r = b - t$ ▷ u
    convergence = *true*
  else
    $v_{k+1} = w/h_{k+1,k}$; $V_{k+1} = [v_1, \ldots, v_{k+1}]$ ▷ u
  end if
end while

- computations with A are performed in precision with unit roundoff u_A;
- computations with M_L are performed in precision with unit roundoff u_L;
- computations with M_R are performed in precision with unit roundoff u_R;
- the precision for other computations (the working precision) has unit roundoff u.

Note that when these precisions differ, some conversion between them is required. This may be done implicitly or explicitly depending on the particular precisions and the underlying hardware and software. We also highlight that in Algorithm 1, computing r_0 requires computing $M_L^{-1}Ax_0$ and $M_L^{-1}b$ separately instead of the usual $M_L^{-1}(b - Ax_0)$; the prior choice is computationally more expensive, and we only do this to avoid the term $u\||M_L^{-1}||M_L|\|$ in the analysis. If $x_0 = 0$, then M_L^{-1} needs to be applied only once.

Using the approach in [17], we assume that the application of M_L^{-1} and M_R^{-1} can be computed in a way such that

$$fl(M_L^{-1}w_j) = M_L^{-1}w_j + \Delta M_{L,j}w_j, \qquad |\Delta M_{L,j}| \le c(n)u_L E_{L,j}, \tag{2.1}$$

$$fl(M_R^{-1}w_j) = M_R^{-1}w_j + \Delta M_{R,j}w_j, \qquad |\Delta M_{R,j}| \le c(n)u_R E_{R,j}, \tag{2.2}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $fl(\cdot)$ denotes the quantity computed in floating point arithmetic, $E_{L,j}$ and $E_{R,j}$ have positive entries, $w_j \in \mathbb{R}^n$, and $c(n)$ is a constant that depends on n only. We define

$$\widetilde{A} := M_L^{-1}A \qquad \text{and} \qquad \tilde{b} := M_L^{-1}b$$

and assume that matrix-vector products with $\widetilde{A}$ can be computed so that

$$fl(\widetilde{A}z_j) = (M_L^{-1} + \Delta M_{L,j})(A + \Delta A_j)z_j.$$

Denoting

$$u_A\psi_{A,j} = \frac{\|M_L^{-1}\Delta A_j z_j\|}{\|\widetilde{A}\|\|z_j\|} \qquad \text{and} \qquad u_L\psi_{L,j} = \frac{\|\Delta M_{L,j}Az_j\|}{\|\widetilde{A}\|\|z_j\|},$$

where here and in the rest of the paper $\|\cdot\|$ denotes the 2-norm, and ignoring the second-order terms, we can write

$$fl(\widetilde{A}z_j) \approx \widetilde{A}z_j + f_j, \qquad \text{where} \quad \|f_j\| \le (u_A\psi_{A,j} + u_L\psi_{L,j})\|\widetilde{A}\|\|z_j\|.$$

In the following, a standard error analysis approach is used, e.g., [9], and we closely follow the analysis in [5] and [4]. The analysis is performed in the following stages:

1. Bounding the computed quantities in the modified Gram-Schmidt (MGS) algorithm that returns

$$C^{(k)} = \begin{bmatrix} \tilde{b} - \widetilde{A}x_0 & \widetilde{A}Z_k \end{bmatrix} = V_{k+1}R_k,$$

where

$$V_k^TV_k = I_k, \qquad R_k = \begin{bmatrix} \beta e_1 & H_k \end{bmatrix}, \qquad \text{and} \qquad e_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T.$$

2. Solving the least-squares problem

$$y_k = \arg\min_y \|\beta e_1 - H_k y\| \tag{2.3}$$

via a QR decomposition employing Givens rotations and analyzing its residual.
3. Computing $x_k = x_0 + Z_k y_k$.
4. Bounding $\|y_k\|$.

Throughout the paper, computed quantities are denoted with bars, that is, $\bar{C}^{(k)}$ is the computed $C^{(k)}$, and $\kappa(A) = \|A\|\|A^\dagger\|$ is the 2-norm condition number of A. The second-order terms in u_A, u_L, u_R, and u are ignored. We drop the subscripts j for $E_{L,j}$, $E_{R,j}$, $\Delta M_{L,j}$, $\Delta M_{R,j}$, ΔA_j, $\psi_{A,j}$, and $\psi_{L,j}$ and replace these quantities by their maxima over all j. It is assumed that no overflow or underflow occurs. We present the main result here and refer the reader to Appendix A for the proof.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

THEOREM 2.1. *Let $\bar{x}_k$ be the approximate solution to (1.2) computed by Algorithm 1. Under the assumptions (2.1), (2.2),*

$$2.12(n+1)u < 0.01 \quad \text{and} \quad c_0(n)u\kappa(C^{(k)}) < 0.1, \qquad \forall k, \tag{2.4}$$

$$|\bar{s}_k| < 1-u, \qquad \forall k, \tag{2.5}$$

where $c_0(n) = 18.53n^{3/2}$ and $\bar{s}_k$ are the sines computed for the Givens rotations and

$$\rho := 1.3c_{13}(n,k)\|M_R\|\left(u\|\bar{Z}_k\| + u_R\|E_R\|\right) < 1, \tag{2.6}$$

where $c_{13}(n,k)$ is defined in (A.4), the residual for the left-preconditioned system is bounded by

$$\|\tilde{b} - \widetilde{A}\bar{x}_k\| \lesssim \frac{1.3c(n,k)}{1-\rho}\left(\zeta_1 + \zeta_2\right), \tag{2.7}$$

where

$$\zeta_1 := \left(u + u_L\|E_L M_L\|\right)\|\tilde{b}\| \quad \text{and}$$

$$\zeta_2 := \left(u + u_A\psi_A + u_L\psi_L\right)\|\widetilde{A}\|\left(\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\| + \|\bar{x}_0\|\right), \tag{2.8}$$

and the normwise relative backward error for the left-preconditioned system is bounded by

$$\frac{\|\tilde{b} - \widetilde{A}\bar{x}_k\|}{\|\tilde{b}\| + \|\widetilde{A}\|\|\bar{x}_k\|} \lesssim \frac{1.3c(n,k)}{1-\rho}\zeta, \tag{2.9}$$

where

$$\zeta := \frac{\zeta_1 + \zeta_2}{\|\tilde{b}\| + \|\widetilde{A}\|\|\bar{x}_k\|}. \tag{2.10}$$

We expect (2.9) to be dominated by ζ_2, mainly due to the term $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$. As observed in [4, 5] and in our experiments (Sections 3.1–3.3), $\|\bar{Z}_k\|$ remains small in early iterations but can be large if many iterations are needed for convergence. We expect the quantity $\|M_R(\bar{x}_k - \bar{x}_0)\|$ to aid in partially mitigating the size of $\|\bar{Z}_k\|$, so that ζ_2 still gives good guarantees for the backward error. Note that if we were to obtain $\|\widetilde{A}\|\|\bar{x}_k\|$ in ζ_2 by using $\|M_R(\bar{x}_k - \bar{x}_0)\| \le \|M_R\|(\|\bar{x}_k\| + \|\bar{x}_0\|)$, then we would introduce the term $\|M_R\|\|\bar{Z}_k\|$. Depending on the preconditioner, $\|M_R\|$ can be close to $\|A\|$, and for some problems $\|M_R\|\|\bar{Z}_k\|$ can grow rapidly, thus making (2.9) a large overestimate. We comment on how (2.7) compares with other bounds for FGMRES available in the literature in the following section. The condition (2.6), the quantities ψ_A, ψ_L, and the role of different precisions are discussed in Section 2.2.

The bound (2.9) can be formulated with respect to the original system, that is, without the left preconditioner, using the inequalities

$$\|b - A\bar{x}_k\| \le \|M_L\|\|\tilde{b} - \widetilde{A}\bar{x}_k\| \quad \text{and} \quad \|b\| + \|A\|\|\bar{x}_k\| \ge (\|\tilde{b}\| + \|\widetilde{A}\|\|\bar{x}_k\|)/\|M_L^{-1}\|.$$

Alternatively, we can use the fact that the relative backward error is bounded by the relative forward error (see Section 2.3). We state the bound in the following corollary.

COROLLARY 2.2. *If the conditions in Theorem 2.1 are satisfied, then the normwise relative backward error for the system (1.1) is bounded by*

$$\frac{\|b - A\bar{x}_k\|}{\|b\| + \|A\|\|\bar{x}_k\|} \lesssim \frac{1.3c(n,k)}{1-\rho}\zeta\min\{\kappa(M_L), \kappa(\widetilde{A})\}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The condition number of the left preconditioner weakens the result, yet for some preconditioners, $\kappa(M_L)$ can be expected to be small, for example when an LU decomposition is used and $M_L = L$. Note that a small backward error with respect to the preconditioned system and small $\kappa(\widetilde{A})$ implies a small backward error with respect to the original system.

2.1. Comparison with existing bounds. We wish to compare our result with the bound in inequality (5.6) in [5] for FGMRES with a general right-preconditioner and the bound in inequality (3.32) in [4] for FGMRES right-preconditioned with an LU factorization computed in single precision. We set $M_L = I$, $u = u_A$; then $u_L = 0$ and $u_A\psi_A = u$. The bound (2.7) becomes

$$\|b - A\bar{x}_k\| \lesssim \frac{1.3c(n,k)u}{1-\rho}\left(\|b\| + \|A\|\left(\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\| + \|\bar{x}_0\|\right)\right).$$

We thus recover the bound in [5, (5.6)] but ignoring the term $u^2\|\bar{x}_0\|$ and with a slightly different ρ. If we further set $\Gamma = \frac{\|M_R\|}{\|A\|}$ and use

$$\|M_R(\bar{x}_k - \bar{x}_0)\| \le \|M_R\|(\|\bar{x}_k\| + \|\bar{x}_0\|),$$

then our bound becomes

$$\|b - A\bar{x}_k\| \lesssim \frac{1.3c(n,k)u}{1-\rho}\left(\|b\| + \|A\|\left(\|\bar{x}_k\| + \|\bar{x}_0\|\right)\left(1 + \Gamma\|A\|\|\bar{Z}_k\|\right)\right).$$

The main aspect in which this bound differs from [4, (3.32)] is that in [4], the term $\Gamma\|A\|\|\bar{Z}_k\|$ is controlled by a factor depending on u_R and the precision in which the LU decomposition used as M_R is computed. This comes from substitutions that rely on the specific M_R when bounding $\|\bar{y}_k\|$. Thus, when more information on M_R is available, reworking the bound for $\|\bar{y}_k\|$ may result in an improved bound.

2.2. Choosing the precisions. We provide guidance on how the precisions should be set when the target backward error is of order u. In our experiments we observe that the achievable backward error is determined by $u + u_A\psi_A + u_L\psi_L$, and we hence ignore the term $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$ in this section. We also note that because of the structure of the former term, we do not expect the backward error to be reduced by setting u_A or u_L so that $u_A\psi_A \ll u$ or $u_L\psi_L \ll u$. The aim is thus to have $u \approx u_A\psi_A \approx u_L\psi_L$ in (2.8).

- u_A. The precision for computations with A should be chosen so that $u_A \approx u/\psi_A$. Numerical experiments with left-preconditioned GMRES in [17] show that for large $\kappa(A)$ and $\kappa(M_L)$, the quantity ψ_A can be large and is driven by $\kappa(M_L)$. In such situations $u_A \ll u$ may be required. If, on the other hand, $\kappa(M_L)$ is small, then setting $u_A \approx u$ may be sufficient.
- u_L. Guidance for setting u_L comes from balancing $u \approx u_A\psi_A \approx u_L\psi_L$ and $u \approx u_L\|E_LM_L\|$. Based on the first expression, $u_L \approx u_A\psi_A/\psi_L$. Vieublé argues that $\psi_L \le \psi_A$ is likely, and if $\kappa(A)$ and $\kappa(M_L)$ are large, then we may observe $\psi_L \ll \psi_A$ [17]. In these cases we can set $u_L \ge u_A$ and $u_L \gg u_A$, respectively. The quantity $\|E_LM_L\|$ depends on M_L and the error in computing matrix-vector products with M_L^{-1}, which may be large for an ill-conditioned M_L. In this case, we thus may require $u_L \approx u$, which is consistent with the guidance for setting $u_A \ll u$.
- u_R. Our insight on u_R comes from the condition (2.6) (see Sections 3.1–3.3 for examples). It requires that $\|M_R\|\|E_R\| \ll u_R^{-1}$. Numerical experiments show that condition (2.6) is sufficient but not necessary; note that a similar condition appears in [4], and it is needed to express $\|\bar{y}_k\|$ via $\|\bar{x}_k\|$. We can obtain a less restrictive

ETNA
Kent State University and
Johann Radon Institute (RICAM)

condition for u_R by keeping $\|\bar{y}_k\|$ but replacing $\|\bar{Z}_k\|$ in (A.2). Using the triangle inequality in (A.3) to bound $\|\bar{Z}_k\|$, we obtain the term

$$(u + u_A\psi_A + u_L\psi_L)\|\widetilde{A}\| \left(\|M_R^{-1}\| + u_R\|E_R\|\right) \|\bar{y}_k\|.$$

Thus, as long as

$$\frac{\|E_R\|}{\|M_R^{-1}\|} \leq u_R^{-1},$$

the choice of u_R should not limit the backward error. E_R depends on the forward error of matrix-vector products with M_R. If $\kappa(M_R)$ is large, then we may need a small u_R for the condition to be satisfied. When $\|M_R\|$ and $\kappa(M_R)$ are small, a large value for u_R may suffice. Note that these comments take into account the backward error only and not the FGMRES iteration count.

2.3. Forward error. A rule of thumb says that the forward error can be bounded by multiplying the backward error by the condition number of the coefficient matrix; see, for example, [9]. Using (2.9) thus gives the bound

$$\frac{\|x - \bar{x}_k\|}{\|x\|} \leq \frac{1.3c(n,k)}{1-\rho}\zeta\kappa(\widetilde{A}), \tag{2.11}$$

where x is the solution to (1.2) and $\bar{x}_k$ is the output of FGMRES. Note that the bound depends on the condition number of the left-preconditioned matrix $\widetilde{A}$. If $\kappa(\widetilde{A})$ and ζ are small, then the forward error is small too, and thus $\bar{x}_k \approx x$. Then $b - A\bar{x}_k = A(x - \bar{x}_k)$ is small and implies a small backward error with respect to the original system as previously noted.

The forward error bound can also be formulated with respect to the split-preconditioned matrix $\widehat{A} \coloneqq M_L^{-1}AM_R^{-1}$ as follows:

$$\frac{\|x - \bar{x}_k\|}{\|x\|} \leq \frac{1.3c(n,k)}{1-\rho}\zeta\kappa(\widehat{A})\kappa(M_R). \tag{2.12}$$

Note that (2.12) is weaker than (2.11) as $\kappa(\widetilde{A}) \leq \kappa(\widehat{A})\kappa(M_R)$. However, (2.12) may be useful if $\kappa(\widehat{A})$ and $\kappa(M_R)$ or their estimates are known and such information is not available for $\kappa(\widetilde{A})$. The bounds (2.11) and (2.12) suggest that guaranteeing a small forward error requires controlling the backward error and constructing the preconditioners so that either $\kappa(\widetilde{A})$ or both $\kappa(\widehat{A})$ and $\kappa(M_R)$ are small (depending on which condition numbers can be evaluated). If A is ill-conditioned, then achieving a small $\kappa(\widetilde{A})$ in (2.11) requires an M_L with a high condition number. Note that in this case, as discussed in the previous section, we may have to set $u_A \ll u$ and can get away with $u_L \gg u_A$. The bound (2.12) indicates that if we achieve a small $\kappa(\widehat{A})$ at the price of $\kappa(M_R) \approx \kappa(A)$, then we cannot guarantee a smaller forward error than when no preconditioning is used because unpreconditioned FGMRES is equivalent to unpreconditioned GMRES in uniform precision with backward error bounded by $\frac{cnu}{1-cnu}$, where c is a constant [15], n is the dimension of A, and u is the unit roundoff.

2.4. Left-, right-, or split-preconditioning. As previously stated, the split-preconditioning approach allows us to analyze the left- and right-preconditioned cases as well. In this section, we explore which preconditioning strategy may be preferred under certain objectives. The discussion is based on the bounds for the backward and forward error. We first simplify these for left- and right-preconditioning.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

If only left-preconditioning is used, then $M_R = I$, $E_R = 0$, $\bar{Z}_k = \bar{V}_k$, and $\rho = 1.3c(n,k)u$, and Algorithm 1 is equivalent to left-preconditioned GMRES. The dominant term in the relative backward error for the preconditioned system is thus

$$\begin{aligned}\zeta_2 &= (u + u_A\psi_A + u_L\psi_L)\,\|\widetilde{A}\|\,(\|\bar{x}_k - \bar{x}_0\| + \|\bar{x}_0\|)\\ &\approx (u + u_A\psi_A + u_L\psi_L)\,\|\widetilde{A}\|\|\bar{x}_k\|,\end{aligned} \tag{2.13}$$

where the approximation holds if $\|\bar{x}_k\| \approx \|\bar{x}_0\|$ or $\bar{x}_0 = 0$. This is equivalent to the result for left-preconditioned GMRES in [17, Theorem 7.1].

In the right-preconditioning case, we have $M_L = I$, $E_L = 0$ and $\psi_L = 0$, and $u_A\psi_A = \max_j\{\|\Delta A_j z_j\|/\|A\|\|z_j\|\}$. Then (2.9) gives the bound for the relative backward error for the original system (1.1), and

$$\zeta_1 = u\|b\| \qquad \text{and} \qquad \zeta_2 = (u + u_A\psi_A)\,\|A\|\left(\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\| + \|\bar{x}_0\|\right).$$

We make the following observations.

- Consider the case where a small backward error is the main concern and A is ill-conditioned. If we have a 'good' preconditioner, so that $\kappa(\widetilde{A})$ is small and we can afford setting u_A and u_L to precisions that are high enough to neutralize the ψ_A and ψ_L terms, then Corollary 2.2 and (2.13) can guarantee a small relative backward error when full left-preconditioning is used. If, however, we cannot afford setting u_A and u_L to high precisions but can construct a split-preconditioner such that $\kappa(M_L)$ is small, then split-preconditioning (note that in this case ψ_A and ψ_L may be smaller too) or full right-preconditioning may be preferential. Note, however, that a small backward error for these options can only be guaranteed if $\|\bar{Z}_k\|$ is expected to be small or if the bounds for $\|\bar{y}_k\|$ are reworked taking into account a specific preconditioner in order to control $\|\bar{Z}_k\|$ (as for the right-preconditioning with an LU decomposition in [4]).
- If we are aiming for a small forward error and can obtain a small relative backward error with full left-preconditioning as detailed above, then this approach also gives a small forward error. If we, however, do not have the flexibility of setting u_A and u_L, then it is not clear from the bounds which preconditioning approach gives the best results.
- Assume that our main concern is applying the preconditioner in lower than the working precision. This may be relevant, for example, when A is very sparse and the preconditioner uses some dense factors. In this case, the bounds suggest that full left-preconditioning should not be used as $u_A\psi_A$ and $u_L\psi_L$ may be large. Full right-preconditioning may be suitable in this case although the bound is affected by $\|\bar{Z}_k\|$.

3. Example: LU preconditioner. We supplement the theoretical analysis in the previous section with an example. Assume that an approximate LU decomposition of A is computed, for example in low precision, and the computed factors $\bar{L}$ and $\bar{U}$ are used for preconditioning. We choose this preconditioning due to its effectiveness and ease of application; note that there is no structural advantage to applying it as a left-, split- or right-preconditioner.

If split-preconditioning is used, then $M_L = \bar{L}$ and $M_R = \bar{U}$. In Algorithm 1, products with A are computed in precision u_A, and hence

$$\psi_{A,j} = \frac{\|M_L^{-1}\Delta A_j z_j\|}{u_A\|\widetilde{A}\|\|z_j\|} \le \tilde{c}_1(n)\frac{\||\bar{L}^{-1}||A|\|\|z_j\|}{\|\bar{L}^{-1}A\|\|z_j\|} = \tilde{c}_1(n)\frac{\||\bar{L}^{-1}||A|\|}{\|\bar{L}^{-1}A\|}, \tag{3.1}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\tilde{c}_i(n)$ is a constant that depends on n. We expect $\||\bar{L}^{-1}||A|\|/\|\bar{L}^{-1}A\|$ to be moderate for many systems, and in this case setting $u_A = u$ may be sufficient.

We apply M_L by solving a triangular system $\bar{L}w_j = (A + \Delta A_j)z_j$ via substitution in precision u_L. From standard results we know that the computed $\bar{w}_j$ satisfies

$$(\bar{L} + \Delta L_j)\bar{w}_j = (A + \Delta A_j)z_j, \qquad \text{where} \qquad |\Delta L_j| \le \tilde{c}_2(n)u_L|\bar{L}|.$$

Thus,

$$\Delta M_{L,j} = \bar{L}^{-1} - (\bar{L} + \Delta L_j)^{-1} \approx \bar{L}^{-1}\Delta L_j \bar{L}^{-1}.$$

We use this to bound $\psi_{L,j}$ as

$$\begin{aligned} \psi_{L,j} &= \frac{\|\Delta M_{L,j}Az_j\|}{u_L\|\widetilde{A}\|\|z_j\|} \approx \frac{\|\bar{L}^{-1}\Delta L_j\bar{L}^{-1}Az_j\|}{u_L\|\bar{L}^{-1}A\|\|z_j\|} \\ &\le \frac{\|\bar{L}^{-1}\Delta L_j\|\|\bar{L}^{-1}A\|\|z_j\|}{u_L\|\bar{L}^{-1}A\|\|z_j\|} \le \tilde{c}_3(n)\kappa_2(\bar{L}). \end{aligned} \tag{3.2}$$

The bound (3.2) is obtained using the bound for the forward error of solving a triangular system. In general, such systems are solved to high accuracy, and thus we expect (3.2) to be a large overestimate. Note that the bounds (3.1) and (3.2) hold for every j.

When using full left-preconditioning, $M_L = \bar{L}\bar{U}$ and $M_R = I$. This case is considered in [17]. We bound $\psi_{A,j}$ in a similar way as in the split-preconditioned case, i.e.,

$$\psi_{A,j} \le \tilde{c}_4(n)\frac{\||\bar{U}^{-1}\bar{L}^{-1}||A|\|}{\|\bar{U}^{-1}\bar{L}^{-1}A\|},$$

which can be expected to be moderate in many cases as well. Applying M_L now requires consecutively solving two triangular systems of equations, that is,

$$\begin{aligned} &(\bar{L} + \Delta L_j)(\bar{U} + \Delta U_j)\bar{w}_j = (A + \Delta A_j)z_j, \qquad \text{where} \\ &|\Delta U_j| \le \tilde{c}_5(n)u_L|\bar{U}| \qquad \text{and} \qquad |\Delta L_j| \le \tilde{c}_6(n)u_L|\bar{L}|. \end{aligned}$$

This gives

$$\begin{aligned} \Delta M_{L,j} &= \bar{U}^{-1}\bar{L}^{-1} - (\bar{U} + \Delta U_j)^{-1}(\bar{L} + \Delta L_j)^{-1} \\ &\approx \bar{U}^{-1}\bar{L}^{-1}\Delta L_j\bar{L}^{-1} + \bar{U}^{-1}\Delta U_j\bar{U}^{-1}\bar{L}^{-1}, \end{aligned} \tag{3.3}$$

where we omit the term involving $\Delta L_j \Delta U_j$, and hence

$$\begin{aligned} \psi_{L,j} \le& \frac{\|\bar{U}^{-1}\bar{L}^{-1}\Delta L_j\bar{L}^{-1}Az_j\| + \|\bar{U}^{-1}\Delta U_j\bar{U}^{-1}\bar{L}^{-1}Az_j\|}{u_L\|\bar{U}^{-1}\bar{L}^{-1}A\|\|z_j\|} \\ \le& \frac{\|\bar{U}^{-1}\|\||\bar{L}^{-1}||L|\|\|\bar{L}^{-1}A\|}{\|\bar{U}^{-1}\bar{L}^{-1}A\|} + \||\bar{U}^{-1}||U|\| \\ =& \frac{\|\bar{U}^{-1}\|\|\bar{L}^{-1}A\|}{\|\bar{U}^{-1}\bar{L}^{-1}A\|}\text{cond}(\bar{L}) + \text{cond}(\bar{U}), \end{aligned}$$

where $\text{cond}(B) = \||B^{-1}||B|\|$. As discussed in [17], $\text{cond}(\bar{L})$ and $\text{cond}(\bar{U})$ are modest when using partial pivoting. The term $\|\bar{U}^{-1}\|$, however, can be large for an ill-conditioned A.

For full right-preconditioning, $M_L = I$ and $M_R = \bar{L}\bar{U}$. Then $\psi_{L,j} = 0$, and

$$\psi_{A,j} = \frac{\|\Delta A_j z_j\|}{u_A\|A\|\|z_j\|} \le \frac{\||A|\|}{\|A\|} \le \sqrt{n}, \tag{3.4}$$

where the final inequality is due to $\||A|\|_2 \le \sqrt{\text{rank}(A)}\|A\|_2$; cf. [9, Lemma 6.6]. We can thus use $u_A = u$ in most cases.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.1
Unit roundoff u for IEEE floating point arithmetics.

Arithmetic	u
fp16 (half)	$2^{-11} \approx 4.88 \times 10^{-4}$
fp32 (single)	$2^{-24} \approx 5.96 \times 10^{-8}$
fp64 (double)	$2^{-53} \approx 1.11 \times 10^{-16}$
fp128 (quadruple)	$2^{-113} \approx 9.63 \times 10^{-35}$

3.1. Numerical example: synthetic dense systems with split-preconditioning. We perform numerical experiments in MATLAB R2021a[1] using a setup similar to an example from [4]. An $n \times n$ coefficient matrix $A = UDV$ is constructed by generating random orthogonal $n \times n$ matrices U and V and setting D to be diagonal with elements $10^{-c(j-1)/(n-1)}$, for $j = 1, 2, \ldots, n$. The condition number of A is 10^c, and we vary its value. The right-hand side b is a random vector with uniformly distributed entries. The preconditioner is computed as a low-precision LU factorization. Namely, for $c \in \{1, 2, \ldots, 5\}$, we use $[\bar{L}, \bar{U}] = lu(mp(A, 4))$, where $mp(\cdot, 4)$ calls the Advanpix Multiprecision Computing Toolbox [2] and simulates precision accurate to four decimal digits; note that this has a smaller unit roundoff than IEEE half precision (see Table 3.1 for the unit roundoff values). For $c \in \{6, 7, \ldots, 10\}$, we compute an LU factorization in single precision using the built-in MATLAB single-precision data type. We set $M_L = \bar{L}$ and $M_R = \bar{U}$, and $E_L = |\bar{L}^{-1}||\bar{L}||\bar{L}^{-1}|$ and $E_R = |\bar{U}^{-1}||\bar{U}||\bar{U}^{-1}|$. The left-preconditioner can slightly reduce the condition number of the coefficient matrix whereas the split-preconditioner achieves a high reduction (Table 3.2).

We set the working precision u to double. Bounds for ψ_A in Table 3.2 indicate that there is no need for $u_A < u$, thus we choose $u_A = u$. The preconditioners are applied using all combinations of half, single, double, and quadruple precisions. Half precision is simulated via the *chop* function [11], and Advanpix is used for quadruple precision. We expect $\kappa(M_L)$ to be a large overestimate for ψ_L. $\kappa(M_R)$ suggests that the condition $\rho < 1$ in (2.6) should be satisfied with u_R set to any of the four precisions, except half for large c-values. We approximate $\|E_R\|/\|M_R^{-1}\|$ by $\min\{\||\bar{U}^{-1}||\bar{U}|\|, \||\bar{U}||\bar{U}^{-1}|\|\}$ in Table 3.2, where we round to the nearest whole number. If u_R is set to half precision, then $u_R^{-1} = 2048$, which is slightly smaller than the values of $\min\{\||\bar{U}^{-1}||\bar{U}|\|, \||\bar{U}||\bar{U}^{-1}|\|\}$ for $c \geq 6$. This indicates that applying M_R in half precision may affect the backward error, however, note that our choice for E_R is expected to be an overestimate. The solver tolerance τ (see Algorithm 1) is set to $4u$, and we use $x_0 = 0$. For the unpreconditioned system, FGMRES converges in 200 iterations when $c = 1$ and does not converge for other c-values.

We show results for $c = 5$ for all precision combinations (Figure 3.1) and for all c-values with u_L set to single and u_R set to double (Table 3.3) and u_L set to double and u_R set to single (Table 3.4). We report the relative backward error (BE) of the original problem, that is,

$$\frac{\|b - A\bar{x}_k\|}{\|b\| + \|A\|\|\bar{x}_k\|}$$

and compute the dominant part of the backward error bound ζ (as defined in (2.10)). Note that ζ bounds the relative backward error for the left-preconditioned system.

From Figure 3.1, we can see that the achievable backward error and subsequently the forward error depends on u_L. As expected from theory, u_R does not affect the achievable backward error, however, u_R influences the iteration count. Setting u_L to half results in extra iterations when $c = 1$, $c = 2$, and $c = 6$ (not shown). Note that setting u_L to quadruple

[1] The code is available at https://github.com/dauzickaite/mpfgmres/.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.2
Synthetic problems. Condition numbers of the unpreconditioned and preconditioned coefficient matrices and preconditioners, and bounds for ψ_A and ψ_L ($\kappa(M_L)$ is also the bound for ψ_L). The preconditioners are computed in precision accurate to four decimal digits for $c < 6$ and in single precision for $c \geq 6$.

c	$\kappa(A)$	$\kappa(\widetilde{A})$	$\kappa(\widehat{A})$	$\kappa(M_R)$	$\kappa(M_L)$	ψ_A bound	$\frac{\|E_R\|}{\|M_R^{-1}\|}$ approx.
1	10	2.69×10^2	1.06	2.69×10^2	3.05×10^2	4.85×10^1	615
2	10^2	4.80×10^2	1.15	4.80×10^2	3.23×10^2	6.30×10^1	1551
3	10^3	1.69×10^3	1.66	1.69×10^3	3.09×10^2	7.27×10^1	1853
4	10^4	1.40×10^4	1.06×10^1	1.43×10^4	2.79×10^2	1.64×10^2	1949
5	10^5	7.73×10^4	3.25×10^2	1.10×10^5	2.85×10^2	1.18×10^2	2048
6	10^6	4.68×10^5	1.14	4.69×10^5	2.64×10^2	2.54×10^2	2144
7	10^7	2.76×10^6	2.34	2.74×10^6	3.50×10^2	1.80×10^2	2336
8	10^8	3.98×10^7	7.00×10^1	4.04×10^7	3.41×10^2	1.03×10^2	2551
9	10^9	3.19×10^8	5.44×10^3	4.78×10^8	3.95×10^2	1.37×10^2	2716
10	10^{10}	4.49×10^9	5.88×10^4	5.72×10^8	3.06×10^2	1.46×10^2	2768

TABLE 3.3
Synthetic problems with u_L single and u_R double. IC denotes the iteration count, BE is the relative backward error, and FE is the relative forward error. For $c = 10$, the solver is terminated at 200 iterations without satisfying the convergence criteria.

c	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ψ_L	ρ
1	6	2.45×10^{-7}	1.61×10^{-6}	2.91×10^{-7}	5.04×10^1	1.21	2.22	1.44×10^{-15}
2	7	3.82×10^{-8}	1.31×10^{-6}	9.34×10^{-8}	2.92×10^2	2.08	7.46×10^{-1}	3.59×10^{-15}
3	9	5.76×10^{-9}	1.46×10^{-6}	2.56×10^{-8}	2.53×10^3	3.79	1.88×10^{-1}	1.51×10^{-14}
4	15	5.45×10^{-10}	1.27×10^{-6}	3.14×10^{-9}	2.30×10^4	3.55	2.19×10^{-2}	1.46×10^{-13}
5	35	1.03×10^{-10}	2.59×10^{-6}	2.51×10^{-9}	8.39×10^5	4.58	4.28×10^{-3}	2.66×10^{-12}
6	7	6.37×10^{-12}	1.58×10^{-6}	5.30×10^{-11}	1.35×10^6	4.81	5.08×10^{-4}	4.88×10^{-12}
7	11	5.88×10^{-13}	9.77×10^{-7}	8.25×10^{-12}	1.42×10^7	8.09	6.88×10^{-5}	3.28×10^{-11}
8	21	7.11×10^{-14}	8.24×10^{-7}	1.72×10^{-12}	2.71×10^8	6.08	6.95×10^{-6}	8.31×10^{-10}
9	92	7.97×10^{-14}	8.82×10^{-6}	1.32×10^{-11}	4.02×10^{10}	6.82	3.33×10^{-6}	4.48×10^{-8}
10	200	1.21×10^{-13}	2.09×10^{-4}	2.30×10^{-11}	5.95×10^{11}	8.28	3.68×10^{-6}	6.97×10^{-8}

and u_R to double or quadruple does not give any benefit. As mentioned, the backward error bound (2.9) is dominated by ζ_2. From Tables 3.3 and 3.4, we can see that the quantity $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$ can become large, but it stays of the order of $\|\bar{x}_k\|$ or close to it (not shown), and thus ζ gives a good estimate of the backward error. If, however, $\|x\|$ is small, then the term $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$ can impair the bound. The increase in the forward error compared to the backward error is well estimated by $\kappa(\widetilde{A})$, whereas $\kappa(\widehat{A})\kappa(M_R)$ is an overestimate. From Figure 3.1, we see that the condition $\rho < 1$ is sufficient but not necessary as it is not satisfied when $c = 5$ and u_R is set to half.

3.2. Numerical example: left- and right-preconditioning. We also perform experiments with full left- (i.e., $M_L = \bar{L}\bar{U}$ and $M_R = I$) and full right-preconditioning (i.e., $M_L = I$ and $M_R = \bar{L}\bar{U}$) using the same set-up as in the previous section. For the left-preconditioning case using (3.3), we set

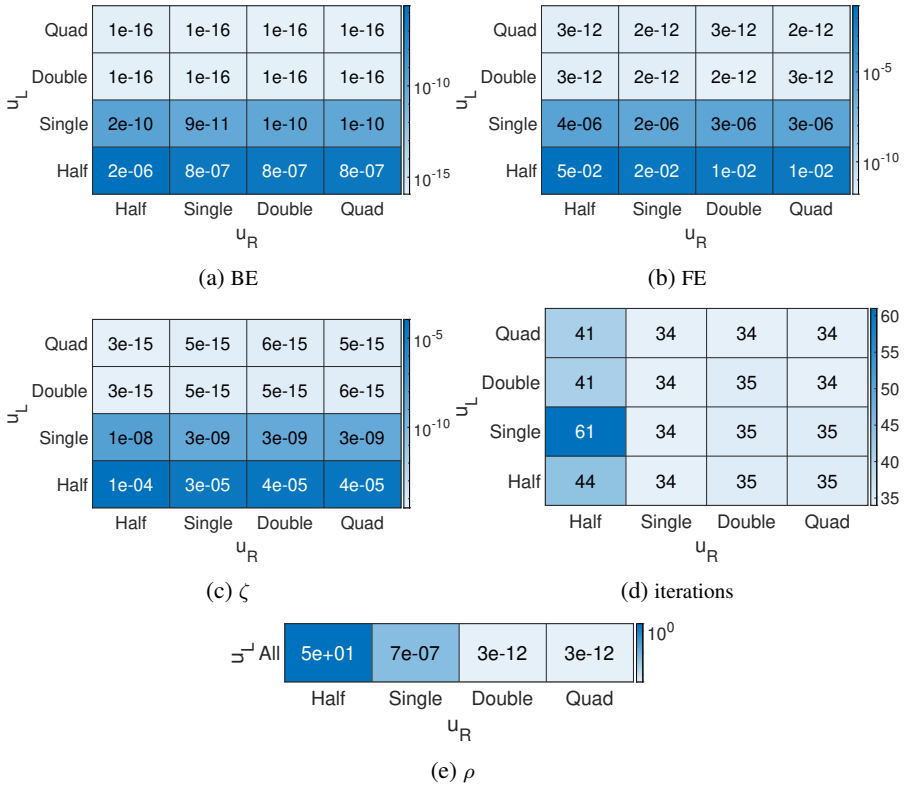
$$E_L = |\bar{U}^{-1}\bar{L}^{-1}||\bar{L}||\bar{L}^{-1}| + |\bar{U}^{-1}||\bar{U}||\bar{U}^{-1}\bar{L}^{-1}| \qquad \text{and} \qquad E_R = 0.$$

Equivalently, for the right-preconditioning case,

$$E_R = |\bar{U}^{-1}\bar{L}^{-1}||\bar{L}||\bar{L}^{-1}| + |\bar{U}^{-1}||\bar{U}||\bar{U}^{-1}\bar{L}^{-1}| \qquad \text{and} \qquad E_L = 0.$$

Both left- and right-preconditioners are effective in reducing the condition number of the coefficient matrix (see Table 3.5). Note that the bounds for ψ_A and ψ_L in the left-preconditioning

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) BE (b) FE

(c) ζ (d) iterations

(e) ρ

FIG. 3.1. *Synthetic problem, $c = 5$. BE is the relative backward error, and FE is the relative forward error, ζ is as defined in (2.10), and ρ is as defined in (2.6).*

case indicate that ψ_L may have to be applied in a precision such that $u_L \leq u$ for all problems, and we may need $u_A \leq u$ for highly ill-conditioned cases. For right-preconditioning, from (3.4), we know that ψ_A can be bounded by $\sqrt{n} \approx 14$, and thus we can set $u_A = u$. We use $\||\bar{L}||\bar{L}^{-1}|\| + \||\bar{U}^{-1}||\bar{U}|\|$ to approximate $\|E_R\|/\|M_R^{-1}\|$ in Table 3.5. As in the split-preconditioning case, the values are larger than u_R^{-1} for half precision.

Experiments with $u = u_A = u_L$ set to double (Table 3.6) show that the bounds for ψ_L in Table 3.5 largely overestimate the error in applying the preconditioner, and even though the bounds for ψ_A are quite close to the obtained values, we still obtain an $\mathcal{O}(u)$ relative backward error for the unpreconditioned system. Note that the ζ-values are increased by ψ_A. If we keep $u = u_A$ set to double and set u_L to single, then the relative backward error for the unpreconditioned system reaches $\mathcal{O}(u_L)$, and $u_L\psi_L$ determines ζ (Table 3.3). This agrees with the split-reconditioning results.

We keep $u = u_A$ set to double for right-preconditioning experiments (Tables 3.8 and 3.9). Note that the relative backward error reaches $\mathcal{O}(u)$ with u_R set to both double and single, except for $\kappa(A) = 10^{10}$ with u_R set to single. The number of iterations when u_R is set to single grows for highly ill-conditioned problems. Though the term $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$ grows as in the split-preconditioning case, it is balanced by $\|\bar{x}_k\|$ here. Note that in these experiments the bound in [4, Equation (3.22)] is applicable for $c \geq 6$. Here $\Gamma = \||L||U|\|/\|A\| = \mathcal{O}(10)$, and the term $\sqrt{u}\Gamma\|A\|\|\bar{Z}_k\|$ is thus small ensuring a tight bound for the backward error.

The results for $c = 5$ for all choices of u_L and u_R are presented in Figure 3.2. Comparing these with the skew diagonals (/) of the respective heatmaps in Figure 3.1, that is, when the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.4
Synthetic problems with u_L double and u_R single. IC denotes the iteration count, BE is the relative backward error and FE is the relative forward error. For $c = 10$, the solver is terminated at 200 iterations without satisfying the convergence criteria.

c	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ψ_L	ρ
1	6	5.14×10^{-16}	3.04×10^{-15}	1.17×10^{-15}	5.05×10^{1}	1.26	2.48	1.29×10^{-9}
2	7	1.20×10^{-16}	3.61×10^{-15}	7.69×10^{-16}	2.92×10^{2}	1.66	6.33×10^{-1}	3.06×10^{-9}
3	9	8.62×10^{-17}	2.07×10^{-14}	1.19×10^{-15}	2.53×10^{3}	3.46	2.45×10^{-1}	1.31×10^{-8}
4	15	6.91×10^{-17}	1.36×10^{-13}	1.12×10^{-15}	2.30×10^{4}	3.19	2.44×10^{-2}	1.05×10^{-7}
5	34	1.42×10^{-16}	2.45×10^{-12}	5.20×10^{-15}	8.20×10^{5}	3.86	5.87×10^{-3}	6.88×10^{-7}
6	7	5.17×10^{-17}	8.85×10^{-12}	9.97×10^{-16}	1.31×10^{6}	4.28	7.65×10^{-4}	3.81×10^{-6}
7	11	4.97×10^{-17}	6.91×10^{-11}	1.59×10^{-15}	1.40×10^{7}	6.23	1.11×10^{-4}	2.40×10^{-5}
8	21	6.58×10^{-17}	5.29×10^{-10}	2.41×10^{-15}	2.29×10^{8}	5.18	1.71×10^{-5}	3.38×10^{-4}
9	158	2.91×10^{-16}	4.98×10^{-8}	7.65×10^{-14}	4.28×10^{10}	8.79	7.92×10^{-6}	3.16×10^{-3}
10	200	4.69×10^{-16}	5.71×10^{-7}	8.81×10^{-14}	5.24×10^{11}	7.62	7.24×10^{-6}	4.38×10^{-3}

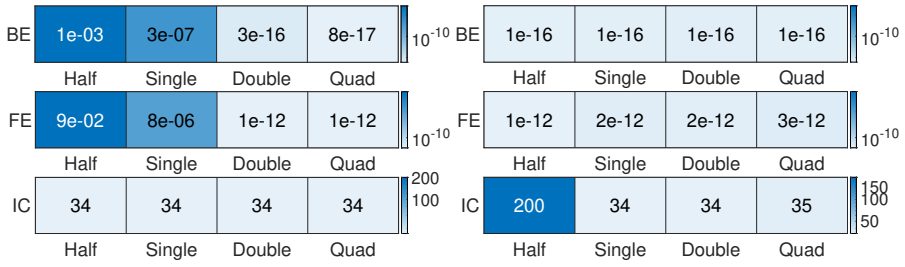
TABLE 3.5
Synthetic problems. Condition numbers of the preconditioned coefficient matrices with full left- and right-preconditioning, and bounds for ψ_A and ψ_L (left-preconditioning only). The preconditioners are computed in precision accurate to four decimal digits for $c < 6$ and in single precision for $c \geq 6$.

	left-preconditioning			right-preconditioning	
c	$\kappa(\bar{U}^{-1}\bar{L}^{-1}A)$	ψ_A bound	ψ_L bound	$\kappa(A\bar{U}^{-1}\bar{L}^{-1})$	$\|E_R\|/\|M_R^{-1}\|$ approx.
1	1.01	2.76×10^{2}	3.79×10^{5}	1.01	2.74×10^{3}
2	1.02	1.40×10^{3}	6.45×10^{5}	1.02	2.84×10^{3}
3	1.09	9.17×10^{3}	2.35×10^{6}	1.09	3.24×10^{3}
4	1.75	5.53×10^{4}	1.51×10^{7}	1.74	3.27×10^{3}
5	3.09×10^{1}	9.54×10^{4}	1.93×10^{7}	3.09×10^{1}	3.30×10^{3}
6	1.03	4.90×10^{6}	6.04×10^{8}	1.03	3.39×10^{3}
7	1.31	3.79×10^{7}	3.12×10^{9}	1.30	3.66×10^{3}
8	7.79	1.39×10^{8}	1.93×10^{10}	7.56	3.96×10^{3}
9	4.75×10^{3}	8.72×10^{7}	7.14×10^{9}	1.96×10^{3}	4.08×10^{3}
10	2.49×10^{5}	1.01×10^{8}	3.75×10^{9}	4.03×10^{4}	4.08×10^{3}

preconditioners are applied in the same precision, shows that applying the full preconditioner on the left gives larger relative backward and consequentially forward errors when the preconditioner is applied in low precision. This is not the case when the full preconditioner is applied on the right. It is curious that the iteration count with left-, right-, and split-preconditioning is essentially different only when the preconditioner is applied in half precision. The results for different c-values are similar.

3.3. Numerical example: application problems with split-preconditioning. We perform experiments with some ill-conditioned problems from the SuiteSparse collection [8]; see Table 3.10. We generate the right-hand side vector b in the same way as for the synthetic problems. The preconditioner is computed as in the previous section, but in single precision, and the matrix A is converted to a full matrix due to the lacking single-precision sparse matrix-vector product functionality in MATLAB. We report results for split-preconditioning only; these illustrate the trends in the left- and right-preconditioning cases well. Note that the split-preconditioner reduces the condition number to the theoretical minimum or close to it, except for the problem with the highest $\kappa(A)$. However $\kappa(\widetilde{A})$ and $\kappa(M_R)$ are of the order of $\kappa(A)$, and thus we expect to see its effect on the forward error. We set u_A and u to double based on the ψ_A bound; u_L and u_R are set as in the previous sections. Unpreconditioned FGMRES does not converge in n iterations for any of the problems.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Left-preconditioning (b) Right-preconditioning

FIG. 3.2. *As in Figure 3.1, but for full left- and right-preconditioning. The left panel shows results for different choices of u_L, and the right panel shows results for different choices of u_R. IC is the iteration count.*

TABLE 3.6
Synthetic problems with full left-preconditioning and u_L set to double. IC denotes the iteration count, BE is the relative backward error, and FE is the relative forward error.

c	IC	BE	FE	ζ	ψ_A	ψ_L
1	6	1.60×10^{-15}	6.77×10^{-15}	5.21×10^{-15}	6.41	4.68×10^{1}
2	7	9.17×10^{-16}	8.39×10^{-15}	8.42×10^{-15}	2.99×10^{1}	4.64×10^{1}
3	9	4.56×10^{-16}	2.31×10^{-14}	2.66×10^{-14}	1.95×10^{2}	4.41×10^{1}
4	15	3.53×10^{-16}	1.61×10^{-13}	1.55×10^{-13}	1.36×10^{3}	4.23×10^{1}
5	34	3.33×10^{-16}	1.06×10^{-12}	2.63×10^{-13}	2.35×10^{3}	1.35×10^{1}
6	7	1.31×10^{-16}	1.14×10^{-11}	1.13×10^{-11}	1.01×10^{5}	4.26×10^{1}
7	11	1.17×10^{-16}	7.71×10^{-11}	1.23×10^{-10}	1.10×10^{6}	5.43×10^{1}
8	21	1.12×10^{-16}	5.25×10^{-10}	4.03×10^{-10}	3.63×10^{6}	4.50×10^{1}
9	47	3.90×10^{-16}	8.10×10^{-9}	3.27×10^{-10}	2.94×10^{6}	8.02×10^{1}
10	124	6.69×10^{-16}	1.42×10^{-7}	5.78×10^{-10}	5.20×10^{6}	9.46×10^{3}

Approximations of $\|E_R\|/\|M_R^{-1}\|$ indicate that the backward error may be affected if we apply M_R in half precision for *arc130* and *west0132*. However, we cannot test this as for all problems except *rajat14*, M_R is singular with respect to u_R set to half. This may be amended by using scaling strategies when computing the preconditioner; see, for example, [12]. We observe similar tendencies (Figures 3.3 and 3.4) as for the dense problems, however, here we can achieve smaller backward error, and for *fs_183_3*, the backward error is $\mathcal{O}(u)$ even with u_L set to half. Note that for sparse problems, setting u_L to a low precision results in iteration-wise slower convergence. The term $\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$ grows as for the dense problems but is balanced by $\|\bar{x}_k\|$; see Tables 3.11 and 3.12.

In all of our sparse and dense examples, ψ_A/ψ_L does not become large enough to allow setting $u_L > u_A$ without it affecting the backward error. However for *fs_183_3*, both ψ_A and ψ_L are small enough that we can set u_A and u_L to single and expect an $\mathcal{O}(u)$ backward error. Numerical experiments confirm this even though the backward and forward errors become slightly larger compared to u_A set to double (not shown).

4. Concluding remarks. In light of great community focus on mixed-precision computations, we analyzed a variant of split-preconditioned FGMRES that allows using different precisions for computing matrix-vector products with the coefficient matrix A (unit roundoff u_A), the left-preconditioner M_L (unit roundoff u_L), the right-preconditioner M_R (unit roundoff u_R), and other computations (unit roundoff u). A backward error of a required level can be achieved by controlling these precisions.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.7
As in Table 3.6, but for u_L set to single. IC denotes the iteration count, BE is the relative backward error, and FE is the relative forward error.

c	IC	BE	FE	ζ	ψ_A	ψ_L
1	6	9.29×10^{-7}	3.85×10^{-6}	2.32×10^{-6}	6.48	4.52×10^{1}
2	7	6.07×10^{-7}	4.25×10^{-6}	2.56×10^{-6}	2.86×10^{1}	4.38×10^{1}
3	9	2.60×10^{-7}	2.76×10^{-6}	2.55×10^{-6}	2.07×10^{2}	4.29×10^{1}
4	15	1.72×10^{-7}	4.97×10^{-6}	2.15×10^{-6}	1.32×10^{3}	3.61×10^{1}
5	34	2.78×10^{-7}	7.99×10^{-6}	8.52×10^{-7}	3.08×10^{3}	1.43×10^{1}
6	7	1.00×10^{-7}	3.99×10^{-6}	2.84×10^{-6}	1.06×10^{5}	4.76×10^{1}
7	11	4.41×10^{-8}	4.41×10^{-8}	3.52×10^{-6}	9.85×10^{5}	5.90×10^{1}
8	21	5.01×10^{-8}	4.46×10^{-6}	2.67×10^{-6}	3.37×10^{6}	4.48×10^{1}
9	47	1.55×10^{-7}	3.23×10^{-5}	3.98×10^{-6}	3.99×10^{6}	6.67×10^{1}
10	114	2.56×10^{-7}	2.74×10^{-2}	4.55×10^{-3}	5.78×10^{6}	7.63×10^{4}

TABLE 3.8
Synthetic problems with full right-preconditioning and u_R set to double. IC denotes the iteration count, BE is the relative backward error, and FE is the relative forward error.

c	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ρ
1	6	1.18×10^{-16}	6.83×10^{-16}	4.31×10^{-16}	4.24×10^{1}	1.41	5.96×10^{-16}
2	7	9.15×10^{-17}	2.73×10^{-15}	5.10×10^{-16}	3.34×10^{2}	9.58×10^{-1}	4.69×10^{-15}
3	9	8.00×10^{-17}	1.82×10^{-14}	6.39×10^{-16}	3.47×10^{3}	8.48×10^{-1}	4.88×10^{-14}
4	15	8.13×10^{-17}	1.50×10^{-13}	1.50×10^{-13}	3.26×10^{4}	6.53×10^{-1}	4.60×10^{-13}
5	34	1.39×10^{-16}	1.65×10^{-12}	1.62×10^{-15}	7.79×10^{5}	5.95×10^{-1}	1.01×10^{-11}
6	7	5.08×10^{-17}	1.04×10^{-11}	5.77×10^{-16}	2.49×10^{6}	6.09×10^{-1}	3.50×10^{-11}
7	11	5.93×10^{-17}	7.32×10^{-11}	4.66×10^{-16}	1.92×10^{7}	5.48×10^{-1}	2.69×10^{-10}
8	21	4.87×10^{-17}	7.15×10^{-10}	9.40×10^{-16}	3.76×10^{8}	4.71×10^{-1}	5.25×10^{-9}
9	61	3.13×10^{-16}	4.04×10^{-8}	9.87×10^{-15}	3.77×10^{10}	4.32×10^{-1}	3.06×10^{-7}
10	200	8.15×10^{-16}	1.44×10^{-6}	4.71×10^{-14}	1.53×10^{12}	5.77×10^{-1}	1.62×10^{-6}

TABLE 3.9
As in Table 3.8, but for u_R set to single. IC denotes the iteration count, BE is the relative backward error, and FE is the relative forward error.

c	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ρ
1	6	1.21×10^{-16}	6.77×10^{-16}	4.18×10^{-16}	4.25×10^{1}	1.34	1.09×10^{-7}
2	7	9.06×10^{-17}	3.21×10^{-15}	4.88×10^{-16}	3.34×10^{2}	8.72×10^{-1}	1.94×10^{-7}
3	9	7.87×10^{-17}	1.88×10^{-14}	6.15×10^{-16}	3.48×10^{3}	7.73×10^{-1}	7.15×10^{-7}
4	15	7.62×10^{-17}	1.25×10^{-13}	6.32×10^{-16}	3.26×10^{4}	6.72×10^{-1}	4.98×10^{-6}
5	34	1.15×10^{-16}	1.91×10^{-12}	1.63×10^{-15}	7.79×10^{5}	6.05×10^{-1}	2.39×10^{-5}
6	7	5.31×10^{-17}	9.20×10^{-12}	9.20×10^{-12}	1.25×10^{6}	4.42×10^{-1}	1.48×10^{-4}
7	11	4.63×10^{-17}	6.76×10^{-11}	3.39×10^{-16}	1.42×10^{7}	5.21×10^{-1}	7.40×10^{-4}
8	27	6.27×10^{-17}	5.69×10^{-10}	5.82×10^{-16}	2.35×10^{8}	4.58×10^{-1}	4.58×10^{-1}
9	200	1.97×10^{-16}	2.25×10^{-8}	1.86×10^{-14}	6.94×10^{10}	4.71×10^{-1}	1.77×10^{-1}
10	200	1.41×10^{-15}	2.29×10^{-6}	3.98×10^{-14}	1.32×10^{12}	5.43×10^{-1}	2.22×10^{-1}

TABLE 3.10
As in Table 3.2 but for SuiteSparse problems.

problem	n	$\kappa(A)$	$\kappa(\widetilde{A})$	$\kappa(\widehat{A})$	$\kappa(M_R)$	$\kappa(M_L)$	ψ_A bound	$\frac{\|E_R\|}{\|M_R^{-1}\|}$ approx.
rajat14	180	3.22×10^{8}	1.44×10^{8}	1.01	1.44×10^{8}	9.72×10^{1}	1.13	33
arc130	130	6.05×10^{10}	6.05×10^{10}	1.00	6.05×10^{10}	2.64	1.00	479471
west0132	132	4.21×10^{11}	2.20×10^{11}	1.12	2.20×10^{11}	7.49	1.00	3619199
fs_183_3	183	3.27×10^{13}	2.39×10^{13}	1.00	2.39×10^{13}	4.73×10^{1}	6.74×10^{-1}	267

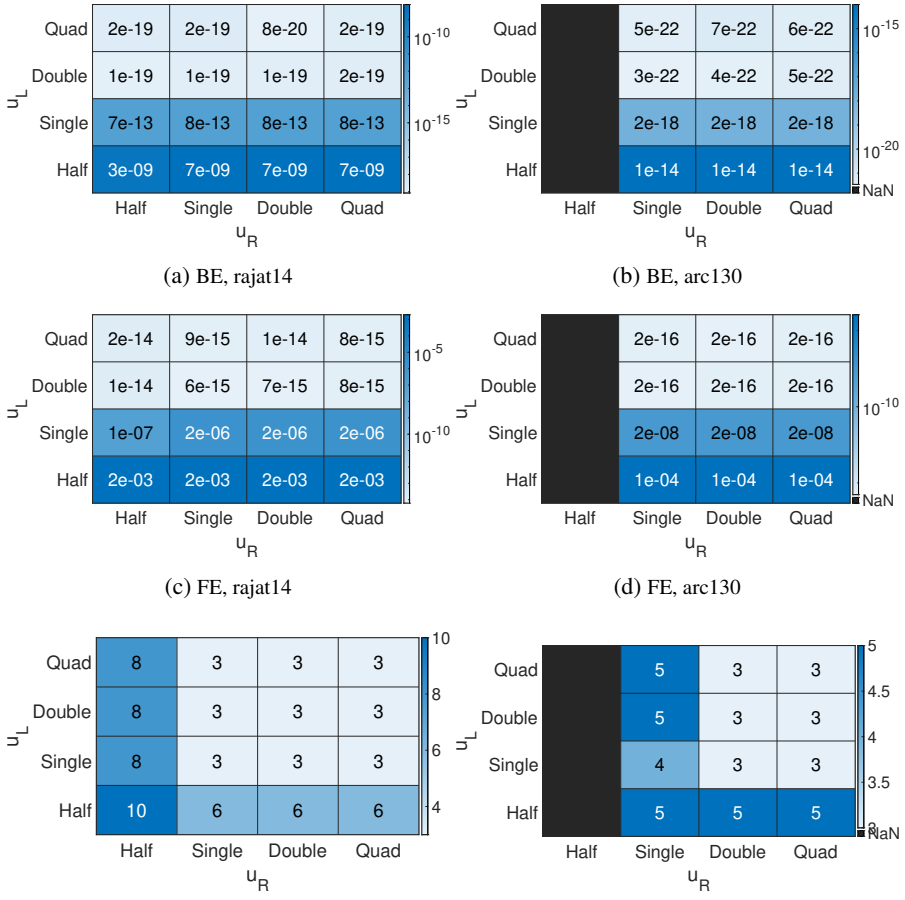
ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.11
As in Table 3.3 but for SuiteSparse problems: u_L is set to single, u_R is set to double.

problem	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ψ_L	ρ
rajat14	3	7.89×10^{-13}	1.84×10^{-6}	1.18×10^{-11}	5.14×10^{2}	1.42×10^{-2}	1.06×10^{-4}	3.11×10^{-11}
arc130	3	1.73×10^{-18}	2.14×10^{-8}	1.41×10^{-16}	1.02×10^{6}	5.99×10^{-6}	4.80×10^{-10}	4.02×10^{-6}
west0132	4	2.30×10^{-17}	2.93×10^{-6}	4.84×10^{-16}	6.46×10^{4}	1.87×10^{-5}	6.05×10^{-9}	3.00×10^{-7}
fs_183_3	3	2.41×10^{-20}	1.31×10^{-8}	1.31×10^{-16}	1.53×10^{5}	9.38×10^{-12}	5.32×10^{-13}	9.17×10^{-4}

TABLE 3.12
As in Table 3.4 but for SuiteSparse problems: u_L is set to double, u_R is set to single.

problem	IC	BE	FE	ζ	$\|\bar{Z}_k\|\|M_R(\bar{x}_k - \bar{x}_0)\|$	ψ_A	ψ_L	ρ
rajat14	3	1.25×10^{-19}	6.19×10^{-15}	1.19×10^{-16}	2.97×10^{2}	7.46×10^{-3}	1.18×10^{-4}	1.15×10^{-6}
arc130	5	3.43×10^{-22}	1.83×10^{-16}	1.11×10^{-16}	1.02×10^{6}	2.14×10^{-6}	5.51×10^{-11}	6.49×10^{-4}
west0132	5	1.48×10^{-21}	4.80×10^{-15}	1.13×10^{-16}	6.39×10^{4}	1.98×10^{-5}	6.16×10^{-9}	2.04×10^{-1}
fs_183_3	3	1.52×10^{-27}	1.05×10^{-15}	1.25×10^{-16}	1.47×10^{5}	1.87×10^{-11}	1.71×10^{-12}	1.38



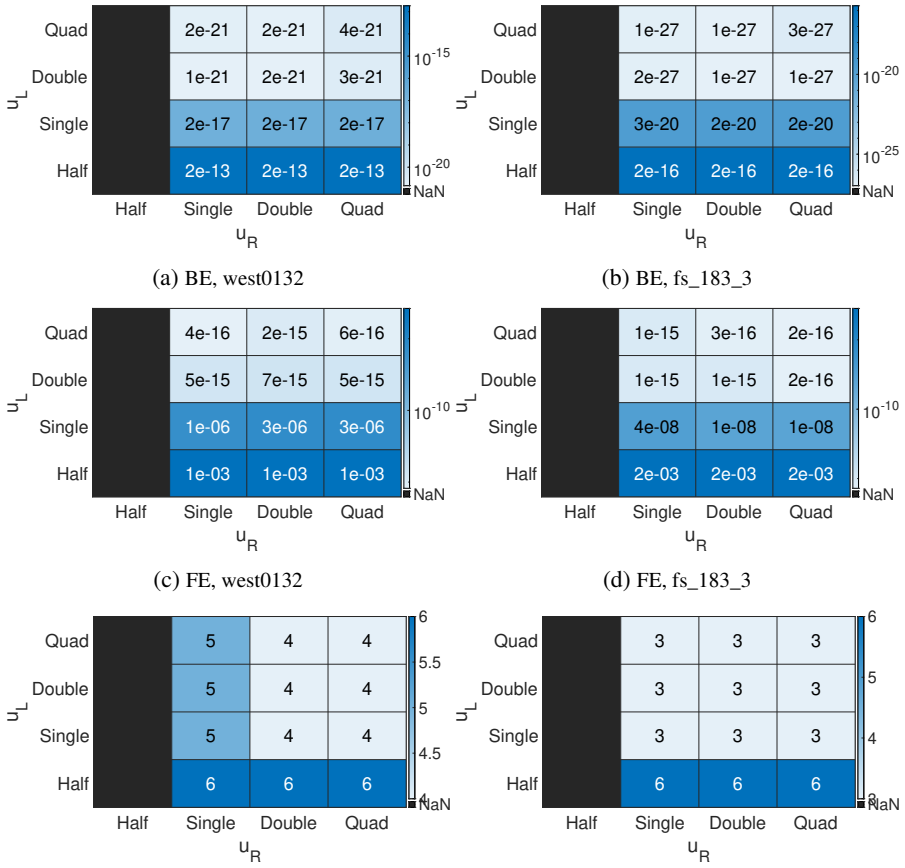
(a) BE, rajat14 (b) BE, arc130

(c) FE, rajat14 (d) FE, arc130

(e) iterations, rajat14 (f) iterations, arc130

FIG. 3.3. *SuiteSparse problems rajat14 and arc130. BE is the relative backward error, and FE is the relative forward error.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) BE, west0132 (b) BE, fs_183_3

(c) FE, west0132 (d) FE, fs_183_3

(e) iterations, west0132 (f) iterations, fs_183_3

FIG. 3.4. *SuiteSparse problems west0132 and fs_183_3. BE is the relative backward error, and FE is the relative forward error.*

Our analysis and numerical experiments show that the precision for applying M_L must be chosen in relation to u, u_A, and the required backward and forward errors, because u_L heavily influences the achievable backward error. We can be more flexible when choosing u_R as it does not influence the backward error directly. Our analysis holds under a sufficient but not necessary assumption on u_R in relation to M_R. As long as M_R is not singular in precision u_R (note that scaling strategies may be used to ensure this), setting u_R to a low precision is sufficient. Very low precisions u_L and u_R may delay the convergence iteration-wise, yet setting $u_L \leq u$ or $u_R \leq u$ does not improve the convergence in general. Note that these conclusions apply to the full left- and right-preconditioning cases as well.

We observe that the forward error is determined by the backward error and the condition number of the left-preconditioned coefficient matrix. This motivates concentrating effort on constructing an appropriate left-preconditioner when aiming for a small forward error: the preconditioner should reduce the condition number sufficiently and needs to be applied in a suitably chosen precision.

Appendix A. Proof of Theorem 2.1. The analysis closely follows [5] and [4], and thus we provide the important results for each stage rather than a step-by-step analysis.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A.1. Left preconditioner. We start by accounting for the effect of M_L.

A.1.1. Stage 1: MGS. In this stage, we use precisions u, u_A, and u_L. MGS is applied to

$$\bar{C}^{(k)} = \left[fl(M_L^{-1}\bar{r}_0) \quad fl(\widetilde{A}\bar{Z}_k) \right].$$

MGS returns an upper triangular matrix $\bar{R}_k$, and there exists an orthonormal matrix $\widehat{V}_{k+1}$, that is $\widehat{V}_{k+1}^T\widehat{V}_{k+1} = I_{k+1}$, such that

$$\begin{aligned}
&\left[\tilde{b} - \widetilde{A}\bar{x}_0 \quad \widetilde{A}\bar{Z}_k\right] + \left[f_1 + f_2 + f_3 \quad F_k^{(1)} + F_k^{(2)}\right] = \widehat{V}_{k+1}\bar{R},\\
&\|f_1\| \le (u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{x}_0\|,\\
&\|f_2\| \le c_1(n)u_L\|E_LM_L\|\|\tilde{b}\| + u\left(\|\tilde{b}\| + (1 + u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{x}_0\|\right),\\
&\|f_3\| \le c_2(n)u\left(\|\tilde{b} - \widetilde{A}\bar{x}_0\| + (u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{x}_0\| + c_1(n)u_L\|E_LM_L\|\|\tilde{b}\|\right),\\
&\|F_k^{(1)}\| \le (u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{Z}_k\|,\\
&\|F_k^{(2)}\| \le c_3(n,k)u\left(\|\widetilde{A}\bar{Z}_k\| + (u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{Z}_k\|\right).
\end{aligned}$$

Here f_1 is the error in computing the matrix vector product $\widetilde{A}\bar{x}_0$, and f_2 accounts for computing $M_L^{-1}b$ and adding it to the computed $\widetilde{A}\bar{x}_0$. The error $F_k^{(1)}$ comes from computing $\widetilde{A}\bar{Z}_k$. f_3, and $F_k^{(2)}$ arise in the MGS process.

A.1.2. Stage 2: Least-squares. The least-squares problem is solved using precision u. From the analysis of [5], under assumptions (2.4) and (2.5), the norm of the residual of the least-squares problem (2.3)

$$\alpha_k = \|\bar{\beta}e_1 + g^{[k]} - (\bar{H}_k + \Delta\bar{H}_k)\bar{y}_k\|$$

monotonically converges to zero for a finite $k \le n$. We can express α_k in the following way:

$$\alpha_k = \|\tilde{b} - \widetilde{A}\bar{x}_0 + \delta\tilde{r}_0 - \widetilde{A}(\bar{Z}_k + \widehat{Z}_k)\bar{y}_k\|, \tag{A.1}$$

where

$$\begin{aligned}
\delta\tilde{r}_0 &= f_1 + f_2 + f_3 + \widehat{V}_{k+1}g^{[k]},\\
\widehat{Z}_k &= \widetilde{A}^{-1}\left(F_k^{(1)} + F_k^{(2)} + \widehat{V}_{k+1}\Delta\bar{H}_k\right),\\
\|g^{[k]}\| &\le c_5(k)u\|\tilde{b} - \widetilde{A}\bar{x}_0\| + c_5(k)u(u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{x}_0\|\\
&\qquad + c_6(n,k)uu_L\|E_LM_L\|\|\tilde{b}\|,\\
\|\Delta\bar{H}_k\| &\le c_4(k)u\|\widetilde{A}\bar{Z}_k\| + c_7(n,k)u(u_A\psi_A + u_L\psi_L)\|\widetilde{A}\|\|\bar{Z}_k\|.
\end{aligned}$$

A.1.3. Stage 3: Computing $\bar{x}_k$. When certain conditions on the residual norm are satisfied, precision u is used to compute x_k as

$$\bar{x}_k = \bar{x}_0 + \bar{Z}_k\bar{y}_k + \delta x_k, \qquad \|\delta x_k\| \le c_8(k)u\|\bar{Z}_k\|\|\bar{y}_k\| + u\|\bar{x}_0\|.$$

Using this to eliminate $\bar{Z}_k\bar{y}_k$ in (A.1), then applying the reverse triangle inequality to bound $\|\tilde{b} - \widetilde{A}\bar{x}_k\|$ and bounding $\|\delta\tilde{r}_0\|$, $\|\widetilde{A}\delta x_k\|$, and $\|\widetilde{A}\bar{Z}_k\bar{y}_k\|$ gives

$$\begin{aligned}
\|\tilde{b} - \widetilde{A}\bar{x}_k\| \le c_{11}(n,k)\Big(&\big(u + (1+u)(u_A\psi_A + u_L\psi_L)\big)\|\widetilde{A}\|\left(\|\bar{x}_0\| + \|\bar{Z}_k\|\|\bar{y}_k\|\right)\\
&+ \left(u + u_L(1+u)\|E_LM_L\|\right)\|\tilde{b}\|\Big).
\end{aligned} \tag{A.2}$$

We eliminate $\|\bar{y}_k\|$ from the bound in the following section.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A.2. Right preconditioner. We now extend the analysis to account for the effect of applying M_R. Under assumption (2.2), $\bar{Z}_k$ is computed such that

$$\bar{Z}_k = M_R^{-1}\bar{V}_k + \Delta M_R \bar{V}_k, \tag{A.3}$$

where $\|\Delta M_R\| \leq c_{12}(n)u_R\|E_R\|$. Then we can obtain

$$\begin{aligned}\|\bar{y}_k\| &\leq 1.3\left(\|M_R(\bar{x}_k - \bar{x}_0)\| + \|M_R\|\|\delta x_k\| + \|M_R\|\|\Delta M_R\bar{V}_k\|\|\bar{y}_k\|\right)\\ &\leq 1.3c_{13}(n,k)\big(\|M_R(\bar{x}_k - \bar{x}_0)\| + u\|M_R\|\|\bar{Z}_k\|\|\bar{y}_k\| + u\|M_R\|\|\bar{x}_0\|\\ &\qquad + u_R\|M_R\|\|E_R\|\|\bar{y}_k\|\big).\end{aligned} \tag{A.4}$$

Under assumption (2.6),

$$\|\bar{y}_k\| \leq \frac{1.3c_{13}(n,k)}{1-\rho}\left(\|M_R(\bar{x}_k - \bar{x}_0)\| + u\|M_R\|\|\bar{x}_0\|\right).$$

Using this in (A.2) and dropping the terms u^2, uu_L, and uu_A gives the required result.

REFERENCES

[1] A. ABDELFATTAH, H. ANZT, E. G. BOMAN, E. CARSON, T. COJEAN, J. DONGARRA, A. FOX, M. GATES, N. J. HIGHAM, X. S. LI, ET AL., *A survey of numerical linear algebra methods utilizing mixed-precision arithmetic*, Int. J. High Performance Comput. Appl., 35 (2021), pp. 344–369.
[2] ADVANPIX, *Advanpix multiprecision computing toolbox for MATLAB*. http://www.advanpix.com
[3] P. AMESTOY, A. BUTTARI, N. J. HIGHAM, J.-Y. L'EXCELLENT, T. MARY, AND B. VIEUBLÉ, *Five-precision gmres-based iterative refinement*, MIMS EPrint 2021.5, Manchester Institute for Mathematical Sciences, The University of Manchester, Manchester, April 2021.
[4] M. ARIOLI AND I. S. DUFF, *Using FGMRES to obtain backward stability in mixed precision*, Electron. Trans. Numer. Anal., 33 (2008/09), pp. 31–44. https://etna.ricam.oeaw.ac.at/vol.33.2008-2009/pp31-44.dir/pp31-44.pdf
[5] M. ARIOLI, I. S. DUFF, S. GRATTON, AND S. PRALET, *A note on GMRES preconditioned by a perturbed LDL^T decomposition with static pivoting*, SIAM J. Sci. Comput., 29 (2007), pp. 2024–2044.
[6] E. CARSON AND N. J. HIGHAM, *A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems*, SIAM J. Sci. Comput., 39 (2017), pp. A2834–A2856.
[7] E. CARSON, N. J. HIGHAM, AND S. PRANESH, *Three-precision GMRES-based iterative refinement for least squares problems*, SIAM J. Sci. Comput., 42 (2020), pp. A4063–A4083.
[8] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), Art. 1, 25 pages.
[9] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[10] N. J. HIGHAM AND T. MARY, *Mixed precision algorithms in numerical linear algebra*, Acta Numer., 31 (2022), pp. 347–414.
[11] N. J. HIGHAM AND S. PRANESH, *Simulating low precision floating-point arithmetic*, SIAM J. Sci. Comput., 41 (2019), pp. C585–C602.
[12] N. J. HIGHAM, S. PRANESH, AND M. ZOUNON, *Squeezing a matrix into half precision, with an application to solving linear systems*, SIAM J. Sci. Comput., 41 (2019), pp. A2536–A2551.
[13] N. LINDQUIST, P. LUSZCZEK, AND J. DONGARRA, *Improving the performance of the GMRES method using mixed-precision techniques*, in Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI, J. Nichols, B. Verastegui, A. Maccabe, O. Hernandez, S. Parete-Koon, T. Ahearn, eds., Revised Selected Papers 17, Comm. Comp. Inf. Sci., Springer, Cham, 2020, pp. 51–66.
[14] NVIDIA, *NVIDIA H100 Tensor Core GPU*, 2023. https://www.nvidia.com/en-us/data-center/h100/
[15] C. C. PAIGE, M. ROZLOŽNÍK, AND Z. STRAKOŠ, *Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[16] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[17] B. VIEUBLÉ, *Mixed Precision Iterative Refinement for the Solution of Large Sparse Linear Systems*, PhD. Thesis, INP Toulouse, University of Toulouse, 2022.