A New Analysis of Iterative Refinement and its Application to Accurate Solution of Ill-Conditioned Sparse Linear Systems

Carson, Erin and Higham, Nicholas J.

2017

MIMS EPrint: **2017.12**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

A NEW ANALYSIS OF ITERATIVE REFINEMENT AND ITS APPLICATION TO ACCURATE SOLUTION OF ILL-CONDITIONED SPARSE LINEAR SYSTEMS*

ERIN CARSON† AND NICHOLAS J. HIGHAM‡

Abstract. Iterative refinement is a long-standing technique for improving the accuracy of a computed solution to a nonsingular linear system $Ax = b$ obtained via LU factorization. It makes use of residuals computed in extra precision, typically at twice the working precision, and existing results guarantee convergence if the matrix A has condition number safely less than the reciprocal of the unit roundoff, u. We identify a mechanism that allows iterative refinement to produce solutions with normwise relative error of order u to systems with condition numbers of order u^{-1} or larger, provided that the update equation is solved with a relative error sufficiently less than 1. A new rounding error analysis is given and its implications are analyzed. Building on the analysis, we develop a GMRES-based iterative refinement method (GMRES-IR) that makes use of the computed LU factors as preconditioners. GMRES-IR exploits the fact that even if A is extremely ill conditioned the LU factors contain enough information that preconditioning can greatly reduce the condition number of A. Our rounding error analysis and numerical experiments show that GMRES-IR can succeed where standard refinement fails, and that it can provide accurate solutions to systems with condition numbers of order u^{-1} and greater. Indeed in our experiments with such matrices—both random and from the University of Florida Sparse Matrix Collection—GMRES-IR yields a normwise relative error of order u in at most 3 steps in every case.

Key words. ill-conditioned linear system, iterative refinement, multiple precision, mixed precision, rounding error analysis, backward error, forward error, GMRES, preconditioning

AMS subject classifications. 65G50, 65F10

1. Introduction.

Ill-conditioned linear systems $Ax = b$ arise in a wide variety of science and engineering applications, ranging from geomechanical problems [13] to computational number theory [5]. When A is ill conditioned the solution x to $Ax = b$ is extremely sensitive to changes in A and b. Indeed, when the condition number $\kappa(A) = \|A\|\|A^{-1}\|$ is of order u^{-1}, where u is the unit roundoff, we cannot expect any accurate digits in a solution computed by standard techniques. This poses an obstacle in applications where an accurate solution is required, of which there is a growing number [4], [15], [23], [24], [34].

Iterative refinement (Algorithm 1.1) is frequently used to obtain an accurate solution to a linear system $Ax = b$. Typically, one computes an initial approximate solution $\widehat{x}$ using Gaussian elimination (GE), saving the factorization $A = LU$. Here and throughout, to simplify the notation we assume that A is a nonsingular matrix for which any required row or column interchanges have been carried out in advance (that is, "$A \equiv PAQ$", where P and Q are appropriate permutation matrices). After computing the residual $r = b - A\widehat{x}$ in higher precision[1] $\overline{u}$, where typically $\overline{u} = u^2$, one reuses L and U to solve the system $Ad = r$, rewritten as $LUd = r$, by substi-

*Version of July 26, 2017. **Funding**: The work of the second author was supported by MathWorks, European Research Council Advanced Grant MATFUN (267526), and Engineering and Physical Sciences Research Council grants EP/I01912X/1 and EP/P020720/1. The opinions and views expressed in this publication are those of the authors, and not necessarily those of the funding bodies.

†Courant Institute of Mathematical Sciences, New York University, New York, NY. (erinc@cims.nyu.edu, http://math.nyu.edu/~erinc).

‡School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk, http://www.maths.manchester.ac.uk/~higham).

[1]We are not concerned here with iterative refinement in fixed precision, which can also benefit accuracy, though to a lesser extent [17, chap. 12], [35].

tution. The original approximate solution is then refined by adding the corrective term, $\widehat{x} \leftarrow \widehat{x} + d$. This process is repeated until a desired backward or forward error criterion is satisfied (see [9] for a detailed discussion of stopping criteria).

Algorithm 1.1 Iterative Refinement

Input: $n \times n$ matrix A; right-hand side b; maximum number of refinement steps $i_{\mathtt{max}}$.
Output: Approximate solution $\widehat{x}$ to $Ax = b$.

1: Compute LU factorization $A = LU$.
2: Solve $Ax_0 = b$ by substitution.
3: **for** $i = 0 : i_{\max} - 1$ **do**
4: Compute $r_i = b - Ax_i$ in precision $\overline{u}$; store in precision u.
5: Solve $Ad_i = r_i$.
6: Compute $x_{i+1} = x_i + d_i$.
7: **if** converged **then** return x_{i+1}, **quit**, **end if**
8: **end for**
9: % Iteration has not converged.

If A is very ill conditioned however, the iterative refinement process may not converge, and indeed all existing results on the convergence of iterative refinement require A to be safely less than u^{-1}. Nevertheless, despite the ill conditioning of A, there is still useful information contained in the LU factors and their inverses (perhaps implicitly applied). It has been observed that if $\widehat{L}$ and $\widehat{U}$ are the computed factors of A from LU factorization with partial pivoting then $\kappa(\widehat{L}^{-1}A\widehat{U}^{-1}) \approx 1 + \kappa(A)u$ even for $\kappa(A) \gg u^{-1}$, where $\widehat{L}^{-1}A\widehat{U}^{-1}$ is computed by substitution; for discussion and further experiments see [26], [30], [31]. This approximation can also be made in the case where left-preconditioned is used, that is, $\kappa(\widehat{U}^{-1}\widehat{L}^{-1}A) \approx 1 + \kappa(A)u$. The experimental results shown in Figure 1.1 illustrate the quality of this approximation.

The results in Figure 1.1 lead us to question the conventional wisdom that iterative refinement cannot work in the regime where $\kappa(A) > u^{-1}$. If the LU factors contain useful information in that regime, can iterative refinement be made to work there too? We will give a new rounding error analysis that identifies a mechanism by which iterative refinement can indeed work when $\kappa(A) > u^{-1}$, provided that we can solve the equations for the updates on line 5 of Algorithm 1.1 with some relative accuracy.

We will then use the analysis to develop an implementation of Algorithm 1.1 that enables accurate solution of sparse, very ill conditioned systems. We use the generalized minimal residual (GMRES) method [33] preconditioned by the computed LU factors to solve for the corrections. We refer to this method as GMRES-based iterative refinement (GMRES-IR).

We show that in the case where the condition number $\kappa_\infty(A)$ is around u^{-1}, our approach can obtain a solution $\widehat{x}$ to $Ax = b$ for which the forward error $\|\widehat{x}-x\|_\infty/\|x\|_\infty$ is of order u. Extra precision (assumed to be with a unit roundoff of u^2) need only be used in computing the residual, in the triangular solves involved in applying the preconditioner, and in matrix-vector multiplication with A. All other computations are performed in the working precision u.

An advantage of our approach is that it can succeed when standard iterative refinement (using the LU factors to solve $Ad = r$) fails or, in the words of Ma et al. [24], "is on the brink of failure". Ma et al. [24] need to solve linear programming problems arising in a biological application and their attempts to use iterative refinement in the underlying linear system solutions were only partially successful. GMRES-IR offers

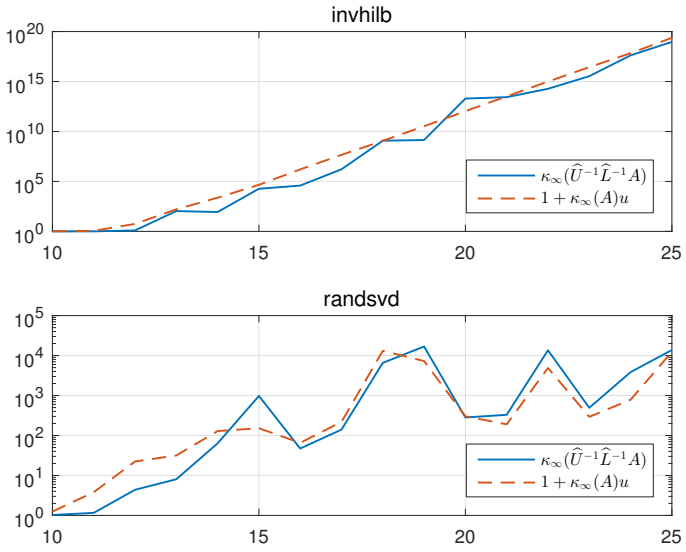


FIG. 1.1. *Comparison of* $\kappa_\infty(\widehat{U}^{-1}\widehat{L}^{-1}A)$ *and* $1+\kappa_\infty(A)u$ *for a computed LU factorization with partial pivoting, for matrices of dimension shown on the* x*-axis. These matrices are very ill conditioned with* $\kappa_\infty(A)$ *ranging from* 10^{13} *to* 10^{35}*. Computations were in double precision arithmetic. Top: inverse Hilbert matrix. Bottom: matrices generated as* `gallery('randsvd',n,10^(n+5))` in MATLAB.

the potential for better results in this application.

We note that there is growing interest in using lower precisions such as single or even half precision in climate and weather modeling [29] and machine learning [14], but this brings an increased likelihood that the problems encountered will be very ill conditioned relative to the working precision. Our work, which is applicable for any u, could be especially relevant in these contexts.

In section 2 we present the new rounding error analysis and investigate its implications. In section 3 we explain how we use preconditioned GMRES within iterative refinement and give, using the results of section 2, theoretical justification that this GMRES-based iterative refinement can yield an error of the order of u even in cases where $\kappa_\infty(A) \geq u^{-1}$. Since we are primarily concerned with sparse linear systems, we discuss in section 3.1 various choices of pivoting strategy and how these affect the numerical behavior. In section 4 we discuss other, related work on iterative refinement. Numerical experiments presented in section 5 confirm our theoretical analysis. Our numerical experiments motivate a two-stage iterative refinement approach, which we briefly discuss in section 5.3, that first attempts the less expensive standard iterative refinement and switches to a GMRES-IR stage in the case of slow convergence or divergence.

2. Error analysis of iterative refinement. The most general rounding error analysis of iterative refinement is that of Higham [17, sec. 12.1], which appeared first in [16]. That analysis cannot provide the result we want; convergence is guaranteed only when $\kappa_\infty(A) \leq u^{-1}$. We therefore carry out a new analysis with different assumptions. A key observation is that an inequality used without comment in previous analyses can be very weak. We introduce a quantity $\mu_i^{(p)}$, in (2.2) below, that captures the sharpness of the inequality and allows us to draw stronger conclusions.

We first define some notation that will be used in the remaining text. Given an integer k, we define

$$\gamma_k = ku/(1-ku), \quad \bar{\gamma}_k = k\overline{u}/(1-k\overline{u}), \quad \tilde{\gamma}_k = cku/(1-cku),$$

where c is some small constant independent of the problem size. For a matrix A and vector x, we define the condition numbers

$$\kappa_p(A) = \|A^{-1}\|_p\|A\|_p, \quad \mathrm{cond}_p(A) = \||A^{-1}||A|\|_p, \quad \mathrm{cond}_p(A,x) = \frac{\||A^{-1}||A||x|\|_p}{\|x\|_p},$$

where $|A| = (|a_{ij}|)$. If p is not specified the ∞-norm is implied.

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and let $\widehat{x}$ be a computed solution to $Ax = b$. Define $x_0 = \widehat{x}$ and consider the following iterative refinement process: $r_i = b - Ax_i$ (compute in precision $\overline{u}$ and round result to precision u), solve $Ad_i = r_i$ (precision u), $x_{i+1} = x_i + d_i$ (precision u), for $i = 1, 2, \ldots$. For traditional iterative refinement, $\overline{u} = u^2$.

The only assumption we will make on the solver for $Ad_i = r_i$ is that the computed solution $\widehat{d}_i$ satisfies

$$\frac{\|d_i - \widehat{d}_i\|_\infty}{\|d_i\|_\infty} = \theta_i u, \quad \theta_i u \le 1, \tag{2.1}$$

where θ_i is a constant depending on A, r_i, n, and u. Thus the solver need not be LU factorization, or even a factorization method.

From this point until the statement of Theorem 2.1 we define r_i, d_i, and x_i to be the *computed* quantities, in order to avoid a profusion of hats.

For any p-norm we define $\mu_i^{(p)}$ by

$$\|A(x - x_i)\|_p = \mu_i^{(p)}\|A\|_p\|x - x_i\|_p \tag{2.2}$$

and note that

$$\kappa_p(A)^{-1} \le \mu_i^{(p)} \le 1.$$

As we argue in the next subsection, $\mu_i^{(p)}$ may be far below its upper bound, and this is the key reason why iterative refinement can work when $\kappa(A) \gtrsim u^{-1}$.

Consider first the computation of r_i. There are two stages. First, $s_i = fl(b - Ax_i) = b - Ax_i + \Delta s_i$ is formed in precision $\overline{u}$, so that $|\Delta s_i| \le \overline{\gamma}_{n+1}(|b| + |A||x_i|)$ [17, sec. 3.5]. Second, the residual is rounded to the working precision: $r_i = fl(s_i) = s_i + f_i$, where $|f_i| \le u|s_i|$. Hence

$$r_i = b - Ax_i + \Delta r_i, \qquad |\Delta r_i| \le u|b - Ax_i| + (1+u)\overline{\gamma}_{n+1}(|b| + |A||x_i|). \tag{2.3}$$

For the second step we have, by (2.1) and (2.2),

$$\begin{aligned}
\|d_i - A^{-1}r_i\|_\infty &\le \theta_i u\|A^{-1}r_i\|_\infty \\
&= \theta_i u\|A^{-1}(b - Ax_i + \Delta r_i)\|_\infty \\
&\le \theta_i u\big[\|x - x_i\|_\infty + u\mu_i^{(\infty)}\kappa_\infty(A)\|x - x_i\|_\infty \\
&\qquad + (1+u)\overline{\gamma}_{n+1}\| \, |A^{-1}|(|b| + |A||x_i|) \, \|_\infty\big] \\
&\le \theta_i u\big(1 + u\mu_i^{(\infty)}\kappa_\infty(A)\big)\|x - x_i\|_\infty \\
&\quad + \theta_i u(1+u)\overline{\gamma}_{n+1}\| \, |A^{-1}|(|b| + |A||x_i|) \, \|_\infty.
\end{aligned} \tag{2.4}$$

For the last step, using the variant [17, Eq. (2.5)] of the rounding error model we have

$$x_{i+1} = x_i + d_i + \Delta x_i, \qquad |\Delta x_i| \le u|x_{i+1}|.$$

Rewriting gives

$$\begin{aligned} x_{i+1} &= x_i + A^{-1}r_i + d_i - A^{-1}r_i + \Delta x_i \\ &= x + A^{-1}\Delta r_i + d_i - A^{-1}r_i + \Delta x_i. \end{aligned}$$

Hence, using (2.3) and (2.4),

$$\begin{aligned} \|x - x_{i+1}\|_\infty \le{}& \| \, |A^{-1}\big[u|A(x - x_i)| + (1+u)\overline{\gamma}_{n+1}(|b| + |A||x_i|)\big] \, \|_\infty \\ &+ \theta_i u\big(1 + u\mu_i^{(\infty)}\kappa_\infty(A)\big)\|x - x_i\|_\infty \qquad (2.5) \\ &+ \theta_i u(1+u)\overline{\gamma}_{n+1}\| \, |A^{-1}|(|b| + |A||x_i| \, \|_\infty + u\|x_{i+1}\|_\infty \qquad (2.6) \\ \le{}& u\big(\mu_i^{(\infty)}\kappa_\infty(A) + \theta_i(1 + u\mu_i^{(\infty)}\kappa_\infty(A))\big)\|x - x_i\|_\infty \\ &+ \overline{\gamma}_{n+1}(1+u)(1+\theta_i u)\| \, |A^{-1}|(|b| + |A||x_i|) \, \|_\infty + u\|x_{i+1}\|_\infty. \qquad (2.7) \end{aligned}$$

We summarize the analysis in the following theorem. As a reminder, we will now use hats to denote computed quantities.

THEOREM 2.1. *Let iterative refinement in precisions u and $\overline{u} \le u$ be applied to a linear system $Ax = b$ with nonsingular $A \in \mathbb{R}^{n\times n}$ and a given approximation x_0 to x, and assume that the solver used on step 5 of Algorithm 1.1 satisfies (2.1). Then for $i \ge 0$ the computed iterate $\widehat{x}_{i+1}$ satisfies*

$$\begin{aligned} \|x - \widehat{x}_{i+1}\|_\infty \lesssim{}& \big(2\mu_i^{(\infty)}\kappa_\infty(A)u + \theta_i u\big)\|x - \widehat{x}_i\|_\infty \\ &+ n\overline{u}(1+\theta_i u)\| \, |A^{-1}|(|b| + |A||\widehat{x}_i|) \, \|_\infty + u\|\widehat{x}_{i+1}\|_\infty. \qquad (2.8) \end{aligned}$$

Proof. The result follows from (2.7) on dropping second order terms, since $\theta_i u \le 1$ by assumption. ∎

We conclude that as long as

$$2\mu_i^{(\infty)}\kappa_\infty(A)u + \theta_i u < 1 \qquad (2.9)$$

for all i, the error will contract until a limiting normwise relative error of order

$$n\overline{u}(1+\theta u)\| \, |A^{-1}|(|b| + |A||x|) \, \|_\infty / \|x\|_\infty + u \le 2n\overline{u}(1+\theta u)\,\mathrm{cond}_\infty(A,x) + u$$

is achieved, where θ is an upper bound on the θ_i terms (here we have used $\widehat{x}_i = \widehat{x}_{i+1} + O(u) = x + O(u)$ in the limit). Thus the normwise relative error will be of order u as long as $\mathrm{cond}_\infty(A,x)\overline{u} \lesssim u$, or equivalently $\mathrm{cond}_\infty(A,x)u \lesssim 1$ when $\overline{u} = u^2$.

To achieve (2.9) we need $\theta_i u$ to be sufficiently less than 1, which is a condition on the solver and the data, and $\mu_i^{(\infty)}\kappa_\infty(A)u$ to be sufficiently less than 1, which is essentially a condition on the iteration. In the next subsection we consider the latter condition.

Note that the limiting accuracy is essentially independent of θ, as long as $\theta u < 1$. Therefore it is not necessary to solve the correction equation $Ad_i = r_i$ to high accuracy in order to achieve a final relative error of order u.

2.1. Bounding μ_i. Now we consider the size of $\mu_i^{(p)}$ in (2.2). We will focus on the 2-norm, but by equivalence of norms our conclusions also apply to the ∞-norm (although note that translating between norms involves factors depending on the dimension of the problem). Let A have the singular value decomposition $A = U\Sigma V^T$ and denote the jth columns of the matrices of left singular vectors U and right singular vectors V by u_j and v_j, respectively. (Note that in this subsection only, U denotes the matrix of left singular vectors of A rather than the upper triangular factor from an LU factorization of A.) Since we are interested in the case where A is ill conditioned (but nonsingular), the singular values satisfy $0 < \sigma_n \ll \sigma_1$.

Denote by $r_i = b - A\widehat{x}_i = A(x - \widehat{x}_i)$ the exact residual for the computed $\widehat{x}_i$. Then we can rewrite (2.2) for $p = 2$ as

$$\|r_i\|_2 = \mu_i^{(2)} \|A\|_2 \|x - \widehat{x}_i\|_2. \tag{2.10}$$

We have

$$x - \widehat{x}_i = V\Sigma^{-1}U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j},$$

and so

$$\|x - \widehat{x}_i\|_2^2 \geq \sum_{j=n+1-k}^{n} \frac{(u_j^T r_i)^2}{\sigma_j^2} \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} (u_j^T r_i)^2 = \frac{\|P_k r_i\|_2^2}{\sigma_{n+1-k}^2},$$

where $P_k = U_k U_k^T$ with $U_k = [u_{n+1-k}, \ldots, u_n]$. Hence from (2.10) we have

$$\mu_i^{(2)} \leq \frac{\|r_i\|_2}{\|P_k r_i\|_2} \frac{\sigma_{n+1-k}}{\sigma_1}.$$

The bound tells us that $\mu_i^{(2)}$ will be much less than 1 if r_i contains a significant component in the subspace $\operatorname{span}(U_k)$ for any k such that $\sigma_{n+1-k} \approx \sigma_n$.

This argument says that we can expect $\mu_i^{(2)} \ll 1$ when r_i is a "typical" vector—one having sizeable components in the direction of every left singular vector of A—in which case $x - \widehat{x}_i$ is not typical, in that it has large components in the direction of the right singular vectors of A corresponding to small singular values. We cannot prove that r_i is typical, but we can verify it numerically, which we do in section 5.

We can gain further insight from backward error considerations. For any backward stable solver (such as LU factorization with appropriate pivoting for stability, or GMRES with Householder orthogonalization [10] or modified Gram-Schmidt orthogonalization [28]) we know that the backward error $\|r_i\|_2/(\|A\|_2\|\widehat{x}_i\|_2)$ of the computed solution $\widehat{x}_i$ to $Ax = b$ will be small, yet the forward error may be large. For the refinement, the initial backward error will be small and the same will be true for each iterate $\widehat{x}_i$, as refinement does not degrade the backward error. So for an ill-conditioned system we would expect to see that

$$\frac{\|r_i\|_2}{\|A\|_2\|\widehat{x}_i\|_2} \approx u \ll \frac{\|x - \widehat{x}_i\|_2}{\|x\|_2},$$

or equivalently $\mu_i^{(2)} \ll 1$ assuming $\|\widehat{x}_i\|_2 \approx \|x\|_2$, at least in the early stages of the refinement when $\widehat{x}_i$ is not very accurate. However, close to convergence both the residual and the error will be small, so that

$$\|r_i\|_2 \approx \|A\|_2\|x - \widehat{x}_i\|_2,$$

or $\mu_i^{(2)} \approx 1$. Therefore we can expect $\mu_i^{(2)}$ to increase as the refinement steps progress and this could result in a slowing of the convergence.

We note that Wilkinson [38] comments that "The successive r derived during the course of iterative refinement become progressively more deficient in components corresponding to the smaller singular values of A". This claim is equivalent to saying that $\mu_i^{(2)}$ will increase with i, but Wilkinson does not justify the claim or make any further use of it.

2.2. The role of θ_i. In standard iterative refinement the LU factorization of A is reused to solve $Ad_i = \widehat{r}_i$ by substitution in each refinement step, where here and in the remaining text $\widehat{r}_i$ denotes the computed residual vector. We will now show that no matter how much precision is used in the substitutions, a relative error less than 1 for the computed solution $\widehat{d}_i$ cannot be guaranteed. Indeed, it suffices to assume that the substitutions with the computed LU factors $\widehat{L}$ and $\widehat{U}$ are carried out exactly. Then, since by [17, Thm. 9.3]

$$A + \Delta A = \widehat{L}\widehat{U}, \quad |\Delta A| \le \gamma_n |\widehat{L}||\widehat{U}|, \tag{2.11}$$

we have

$$\widehat{d}_i = \widehat{U}^{-1}\widehat{L}^{-1}\widehat{r}_i = (A + \Delta A)^{-1}\widehat{r}_i.$$

Hence

$$\frac{\|\widehat{d}_i - A^{-1}\widehat{r}_i\|_\infty}{\|A^{-1}\widehat{r}_i\|_\infty} \approx \frac{\|A^{-1}\Delta A A^{-1}\widehat{r}_i\|_\infty}{\|A^{-1}\widehat{r}_i\|_\infty} \le \gamma_n \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty. \tag{2.12}$$

The term $\| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty$, which is at least as large as $\mathrm{cond}_\infty(A)$, will usually be of similar size to $\kappa_\infty(A)$, unless A has poor row scaling. Therefore if $\kappa_\infty(A) \ge u^{-1}$ then (2.12) does not guarantee any relative accuracy in $\widehat{d}_i$, so we have $\theta_i u > 1$ in (2.1) and our analysis suggests that iterative refinement may fail. The culprit is the ΔA term, which comes from the LU factorization in precision u.

We conclude that if the correction equation is solved using the original LU factors then standard iterative refinement may fail to converge for very ill conditioned A, no matter how much precision is used in the triangular solves and regardless of the size of the μ_i values.

One way to satisfy (2.1) is to use higher precision in computing the LU factorization, but this is very expensive. In the following section we present an alternative approach. We show that the correction equations can be solved with some relative accuracy even for numerically singular A by using a different solver: GMRES preconditioned by the existing (precision u) LU factors. This approach can be motivated by the observation, mentioned in section 1, that even if A is very ill conditioned the computed LU factors still contain useful information.

3. GMRES iterative refinement. In this section we will show that we can use GMRES [33] to solve $Ad_i = \widehat{r}_i$ in iterative refinement in such a way that Theorem 2.1 guarantees accurate solution of ill-conditioned systems. We will use the computed LU factors as left preconditioners, so that GMRES solves the preconditioned system

$$\widetilde{A}d_i = s_i, \tag{3.1}$$

where $\widetilde{A} = \widehat{U}^{-1}\widehat{L}^{-1}A$ and $s_i = \widehat{U}^{-1}\widehat{L}^{-1}\widehat{r}_i$. The GMRES method presented in Algorithm 3.1 is a simplified variant in which no restarting is used and we assume that the

Algorithm 3.1 Left-preconditioned GMRES

Input: $n \times n$ matrix A; right-hand-side b; maximum number of iterations m; tolerance τ; approximate LU factors L and U.
Output: Approximate solution $\widehat{x}$ to $Ax = b$.

Compute $r_0 = U^{-1}(L^{-1}b)$ in precision $\overline{u}$; store in precision u.
$\beta = \|r_0\|_2$, $v_1 = r_0/\beta$
for $k = 1, \dots, m$ **do**
  Compute $z = U^{-1}(L^{-1}(Av_k))$ in precision $\overline{u}$; store in precision u.
  for $\ell = 1, \dots, k$ **do**
    $h_{\ell,k} = z^* v_\ell$
    $z = z - h_{\ell,k} v_\ell$
  end for
  $h_{k+1,k} = \|z\|_2$, $v_{k+1} = z/h_{k+1,k}$
  Let $V = [v_1, \dots, v_k]$ and $H = \{h_{i,\ell}\}_{1 \le i \le k+1, 1 \le \ell \le k}$.
  Update decomposition $Q = HR$ (via Givens rotations).
  if $|e_{k+1}^T Q e_1| \le \tau$ **then break**, **end if**
end for
Solve $y = \operatorname{argmin}_{\bar{y}} \|g - R\bar{y}\|_2$.
$\widehat{x} = Vy$
Return $\widehat{x}$.

iteration is started with the zero vector as the initial guess. Additionally, the method uses precision $\overline{u} = u^2$ in the triangular solves with $\widehat{L}$ and $\widehat{U}$ and in matrix-vector multiplication with A. The remaining computations are performed in precision u and all quantities are stored in precision u. For clarity, in this section we use hats to decorate all quantities computed in finite precision. To avoid confusion, in the remaining text we will use the word *iterations* and indices j and k in association with GMRES and the word *steps* and index i in association with the iterative refinement process.

Our analysis proceeds in three main steps. First, we show that $\kappa_\infty(\widetilde{A})$ is small. Then we show that the error $\|\widehat{s}_i - s_i\|_\infty$ in the computed right-hand side $\widehat{s}_i = fl(\widehat{U}^{-1} fl(\widehat{L}^{-1}\widehat{r}_i))$ is small. Then we use the analysis of [28] to show that our GMRES variant provides a backward stable solution to $\widetilde{A}d_i = \widehat{s}_i$. These three results allow us to conclude that $\widetilde{A}d_i = s_i$ can be solved with some degree of relative accuracy, that is, (2.1) is satisfied. To simplify the analysis we assume in this section that $\kappa_\infty(A)$ is not too much larger than u^{-1}, although our experiments in section 5 suggest that the GMRES-based approach can work even when $\kappa_\infty(A)$ is a few orders of magnitude larger than u^{-1}.

We begin by showing that the matrix $\widetilde{A}$ is well conditioned. Using (2.11) we can write

$$\widetilde{A} = \widehat{U}^{-1}\widehat{L}^{-1}A = (A + \Delta A)^{-1}A \approx I - A^{-1}\Delta A,$$
$$\widetilde{A}^{-1} = A^{-1}\widehat{L}\widehat{U} = A^{-1}(A + \Delta A) = I + A^{-1}\Delta A,$$

which give the bounds

$$\|\widetilde{A}\|_\infty \lesssim 1 + \gamma_n \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty,$$
$$\|\widetilde{A}^{-1}\|_\infty \lesssim 1 + \gamma_n \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty,$$

and then

$$\kappa_\infty(\widetilde{A}) \lesssim (1+\gamma_n \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty)^2. \tag{3.2}$$

Therefore even if A is so ill conditioned that $\gamma_n \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty$ is of order 100 (say), we still expect $\kappa_\infty(\widetilde{A})$ to be of modest size. (Note that by comparison with the observation in section 1 that $\kappa(\widehat{U}^{-1}\widehat{L}^{-1}A) \approx 1+\kappa(A)u$ for the computed matrix, here we have a strict bound for the corresponding exact matrix.)

Of course, the matrix $\widetilde{A}$ is not explicitly formed in preconditioned GMRES. Since GMRES only requires matrix-vector products with the preconditioned coefficient matrix, we compute $\widetilde{A}v_i$ by forming $w_i = Av_i$ and performing the triangular solves $\widehat{L}y_i = w_i$ and $\widehat{U}z_i = y_i$, all at precision $\overline{u} = u^2$.

Unlike $\widetilde{A}$, the right-hand side s_i is explicitly formed at the beginning of the preconditioned GMRES algorithm. Using precision $\overline{u}$, this computation yields

$$\widehat{s}_i = (\widehat{U}+\Delta U)^{-1}(\widehat{L}+\Delta L)^{-1}\widehat{r}_i,$$

where $|\Delta U| \le \bar{\gamma}_n|\widehat{U}|$ and $|\Delta L| \le \bar{\gamma}_n|\widehat{L}|$. Some manipulation gives

$$\begin{aligned}
\widehat{s}_i &\approx (\widehat{U}^{-1} - \widehat{U}^{-1}\Delta U\widehat{U}^{-1})(\widehat{L}^{-1} - \widehat{L}^{-1}\Delta L\widehat{L}^{-1})\widehat{r}_i \\
&\approx \widehat{U}^{-1}\widehat{L}^{-1}r_i - \widehat{U}^{-1}\widehat{L}^{-1}\Delta L\widehat{L}^{-1}r_i - \widehat{U}^{-1}\Delta U\widehat{U}^{-1}\widehat{L}^{-1}\widehat{r}_i \\
&= s_i - \widehat{U}^{-1}\widehat{L}^{-1}(\Delta L\widehat{U} + \widehat{L}\Delta U)s_i \\
&= s_i - (A+\Delta A)^{-1}(\Delta L\widehat{U} + \widehat{L}\Delta U)s_i,
\end{aligned}$$

so

$$s_i - \widehat{s}_i \approx A^{-1}(\Delta L\widehat{U} + \widehat{L}\Delta U)s_i.$$

Hence

$$\|s_i - \widehat{s}_i\|_\infty \lesssim \bar{\gamma}_{2n} \| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty \|s_i\|_\infty. \tag{3.3}$$

Again, the quantity $\| \, |A^{-1}||\widehat{L}||\widehat{U}| \, \|_\infty$ can be as large as $\kappa_\infty(A)$. Nevertheless, since precision $\overline{u}$ is used, we still expect $\|s_i - \widehat{s}_i\|_\infty \lesssim \tilde{\gamma}_n\|s_i\|_\infty$ as long as $\kappa_\infty(A)$ is not too much larger than u^{-1}.

We now want to show that GMRES provides a backward stable solution to $\widetilde{A}d_i = \widehat{s}_i$. We will use the analysis of [28], where it is proved that the variant of GMRES that uses modified Gram-Schmidt orthogonalization (MGS-GMRES) is backward stable. This proof relies on the observation that, given a matrix A and right-hand side b, carrying out $k-1$ iterations of the Arnoldi process is equivalent to applying k steps of modified Gram-Schmidt to the matrix

$$[b, fl(A\widehat{V}_{k-1})] = [b, AV_{k-1}] + [0, \Delta V_{k-1}],$$

where $\widehat{V}_{k-1} = [\widehat{v}_1, \ldots, \widehat{v}_{k-1}]$ is the matrix of computed basis vectors and $V_{k-1} = [v_1, \ldots, v_{k-1}]$ is $\widehat{V}_{k-1}$ with its columns correctly normalized, that is, for $j \le k-1$,

$$\widehat{v}_j = v_j + \Delta v_j', \quad \|\Delta v_j'\|_2 \le \tilde{\gamma}_n. \tag{3.4}$$

The term $\Delta V_{k-1} = [\Delta v_1, \ldots, \Delta v_{k-1}]$ contains both errors in applying the matrix A to vectors $\widehat{v}_j$ and errors in normalizing $\widehat{v}_j$, for $j \le k-1$. Assuming that matrix-vector products and inner products are computed in floating point arithmetic in the

usual way, $\|\Delta V_{k-1}\| \le k^{1/2}\gamma_n\|A\|_F$. In [28], this bound is then combined with results on the finite precision behavior of MGS, including the loss of orthogonality in the MGS process and the backward stability of MGS for solving linear least squares problems, to show the backward stability of MGS-GMRES for solving $Ax = b$.

We now consider the case where MGS-GMRES is used to solve $\widetilde{A}d_i = \widehat{s}_i$. The only thing that will change computationally is the error in applying $\widetilde{A}$ to a vector, which is done in this case without explicitly forming $\widetilde{A}$. Other aspects of the MGS-GMRES algorithm, such as the MGS orthogonalization process and least squares solve, remain unchanged. Therefore if we can show that

$$\|\Delta V_{k-1}\|_F \le k^{1/2}\gamma_n\|\widetilde{A}\|_F, \tag{3.5}$$

then carrying through the remaining analysis in [28] shows that the MGS-GMRES backward error results of [28] hold for the left-preconditioned GMRES method run with $\widetilde{A}$ and $\widehat{s}_i$, that is, for some $k \le n$, we have

$$(\widetilde{A} + \Delta\widetilde{A})\widehat{d}_i = \widehat{s}_i + \Delta\widehat{s}_i, \quad \|\Delta\widetilde{A}\|_F \le \tilde{\gamma}_{kn}\|\widetilde{A}\|_F, \quad \|\Delta\widehat{s}_i\|_2 \le \tilde{\gamma}_{kn}\|\widehat{s}_i\|_2. \tag{3.6}$$

We now show that if precision $\overline{u}$ is used in implicitly applying $\widetilde{A}$ to $\widehat{v}_j$, then ΔV_{k-1} indeed satisfies the required bound (3.5). Using precision $\overline{u}$, we compute

$$\begin{aligned}
(A + \Delta A)\widehat{v}_j &= \widehat{w}_j, \quad |\Delta A| \le \bar{\gamma}_n|A|,\\
(\widehat{L} + \Delta L)\widehat{y}_j &= \widehat{w}_j, \quad |\Delta L| \le \bar{\gamma}_n|\widehat{L}|,\\
(\widehat{U} + \Delta U)\widehat{z}_j &= \widehat{y}_j, \quad |\Delta U| \le \bar{\gamma}_n|\widehat{U}|.
\end{aligned}$$

The computed vector $\widehat{z}_j$ can therefore be written

$$\begin{aligned}
\widehat{z}_j &= (\widehat{U} + \Delta U)^{-1}(\widehat{L} + \Delta L)^{-1}(A + \Delta A)\widehat{v}_j\\
&\approx (\widehat{U}^{-1} - \widehat{U}^{-1}\Delta U\widehat{U}^{-1})(\widehat{L}^{-1} - \widehat{L}^{-1}\Delta L\widehat{L}^{-1})(A + \Delta A)\widehat{v}_j\\
&= (\widetilde{A} + \Delta\widetilde{A}')\widehat{v}_j,
\end{aligned}$$

where

$$\begin{aligned}
\Delta\widetilde{A}' &\approx \widehat{U}^{-1}\widehat{L}^{-1}\Delta A - \widehat{U}^{-1}\widehat{L}^{-1}\Delta L\widehat{L}^{-1}A - \widehat{U}^{-1}\Delta U\widehat{U}^{-1}\widehat{L}^{-1}A\\
&= \widetilde{A}A^{-1}\Delta A - \widehat{U}^{-1}\widehat{L}^{-1}\Delta L\widehat{U}\widetilde{A} - \widehat{U}^{-1}\Delta U\widetilde{A},
\end{aligned}$$

from which we obtain

$$\|\Delta\widetilde{A}'\|_F \le \bar{\gamma}_n\big(\kappa_F(A) + \kappa_F(\widehat{U})\kappa_F(\widehat{L}) + \kappa_F(\widehat{U})\big)\|\widetilde{A}\|_F. \tag{3.7}$$

If $\kappa_F(A) \approx u^{-1}$ and $\kappa_F(\widehat{L})$ is of modest size (which will usually be the case, as $\widehat{L}$ is unit triangular with off-diagonal elements bounded by 1), then since $\kappa_F(\widehat{U}) \lesssim \kappa_F(A)\kappa_F(\widehat{L})$, we have

$$\|\Delta\widetilde{A}'\|_F \lesssim \tilde{\gamma}_n\|\widetilde{A}\|_F.$$

Accounting for the errors in normalization, with (3.4) we have

$$\widehat{z}_j \approx (\widetilde{A} + \Delta\widetilde{A}')(v_j + \Delta v_j') \approx \widetilde{A}v_j + \Delta\widetilde{A}'v_j + \widetilde{A}\Delta v_j' = \widetilde{A}v_j + \Delta v_j,$$

with $\Delta v_j = \Delta\widetilde{A}' v_j + \widetilde{A}\Delta v_j'$. Using (3.7), this gives

$$\|\Delta v_j\|_2 \le \|\Delta\widetilde{A}'\|_2\|v_j\|_2 + \|\widetilde{A}\|_2\|\Delta v_j'\|_2 \approx \tilde{\gamma}_n\|\widetilde{A}\|_F.$$

Then after $k-1$ iterations,

$$\widehat{Z}_{k-1} = [\widehat{z}_1, \ldots, \widehat{z}_{k-1}] = \widetilde{A}V_{k-1} + \Delta V_{k-1}, \quad \|\Delta V_{k-1}\|_F \le k^{1/2}\tilde{\gamma}_n\|\widetilde{A}\|_F.$$

Therefore (3.5) is satisfied, and so the backward error result (3.6) holds.

We now want to show that the computed $\widehat{d}_i$ is a backward stable solution to $\widetilde{A}d_i = s_i$ (with the exact preconditioned residual s_i rather than the computed preconditioned residual $\widehat{s}_i$ as the right-hand side). Writing $\widehat{s}_i = s_i + (\widehat{s}_i - s_i)$, from (3.6) we have

$$s_i - \widetilde{A}\widehat{d}_i = \Delta\widetilde{A}\widehat{d}_i - (\widehat{s}_i - s_i) - \Delta\widehat{s}_i,$$

which, using (3.3) and (3.6) gives the bound

$$\begin{aligned}
\|s_i - \widetilde{A}\widehat{d}_i\|_\infty &\le \|\Delta\widetilde{A}\|_\infty\|\widehat{d}_i\|_\infty + \|\widehat{s}_i - s_i\|_\infty + \|\Delta\widehat{s}_i\|_\infty \\
&\lesssim n\tilde{\gamma}_{kn}\|\widetilde{A}\|_\infty\|\widehat{d}_i\|_\infty + \tilde{\gamma}_n\|s_i\|_\infty + n^{1/2}\tilde{\gamma}_{kn}\|\widehat{s}_i\|_\infty \\
&\lesssim n\tilde{\gamma}_{kn}\|\widetilde{A}\|_\infty\|\widehat{d}_i\|_\infty + \tilde{\gamma}_n\|s_i\|_\infty + n^{1/2}\tilde{\gamma}_{kn}(1+\tilde{\gamma}_n)\|s_i\|_\infty \\
&\lesssim n\tilde{\gamma}_{kn}(\|\widetilde{A}\|_\infty\|\widehat{d}_i\|_\infty + \|s_i\|_\infty).
\end{aligned}$$

Thus the normwise relative backward error for the system (3.1) is

$$\frac{\|s_i - \widetilde{A}\widehat{d}_i\|_\infty}{\|\widetilde{A}\|_\infty\|\widehat{d}_i\|_\infty + \|s_i\|_\infty} \lesssim n\tilde{\gamma}_{kn},$$

and therefore the relative error of the computed $\widehat{d}_i$ can be bounded by

$$\frac{\|d_i - \widehat{d}_i\|_\infty}{\|d_i\|_\infty} \lesssim n\tilde{\gamma}_{kn}\kappa_\infty(\widetilde{A}).$$

Thus in (2.1) we can take $\theta_i u = n\tilde{\gamma}_{kn}\kappa_\infty(\widetilde{A})$. Since, as we have shown, $\kappa_\infty(\widetilde{A})$ is small (see (3.2)), we expect that $\theta_i u \ll 1$. We note in passing that it can be shown that if the normwise relative backward error is small for the system $\widetilde{A}\widehat{d}_i = s_i$ then it is also small for the system $A\widehat{d}_i = \widehat{r}_i$.

We conclude that this variant of preconditioned GMRES can solve for the correction vector with sufficient accuracy to allow convergence of the iterative refinement process. Thus we define a new iterative refinement scheme, where in Algorithm 1.1, the solve in line 5 is performed by invoking GMRES (Algorithm 3.1) with input matrix A, right-hand side r_i, preconditioners $\widehat{L}$, $\widehat{U}$, and a specified tolerance τ and maximum number of iterations m. We call this method *GMRES-based iterative refinement* (GMRES-IR). In section 5, we show experimentally that GMRES-IR can indeed converge to an accurate solution to $Ax = b$ even when $\kappa_\infty(A)$ is a few orders of magnitude larger than u^{-1}.

In discussing the backward stability of GMRES, we have used results from [28] that are specific to the MGS variant of GMRES. We conjecture however that one could prove similar results (that is, show $\theta_i u \ll 1$) when certain other GMRES variants are used to solve for the corrective term with LU preconditioning, such as Householder

GMRES [37], and flexible GMRES (FGMRES) [32], which were proved to be backward stable in [10] and [1], respectively. Although out of the scope of this paper, one could potentially use existing results to prove that our approach will work with other Krylov subspace methods besides GMRES. One such potential method is the full orthogonalization method (FOM), for which bounds on the forward error have been given in [3].

As an alternative to $\widetilde{A}$ we could use right preconditioning or split preconditioning. Consider the split preconditioning case, where $\bar{A} = \widehat{L}^{-1}A\widehat{U}^{-1}$. It is straightforward to show that

$$|\bar{A} - I| \le \gamma_n |\widehat{L}^{-1}||\widehat{L}||\widehat{U}||\widehat{U}^{-1}|,$$
$$|\widetilde{A} - I| \le \gamma_n |\widehat{U}^{-1}||\widehat{L}^{-1}||\widehat{L}||\widehat{U}|.$$

The first of these two bounds is the more favorable as it allows the diagonal of U, which has elements of potentially widely varying magnitude, to cancel, whereas in the second bound the $\widehat{L}$-based term intervenes. A related observation is that for the exact LU factors we have $AD = LUD$ for diagonal D, and D does not affect the pivot sequence. Therefore, since $|U||U^{-1}| = |UD||(UD)^{-1}|$, the bound for $\bar{A} - I$ has the desirable property of being insensitive to the column scaling of A, so split preconditioning might be the best choice when the matrix is badly-scaled.

3.1. Pivoting strategies for sparse LU. In the sparse case, it is common to use a pivoting strategy that allows for minimizing fill-in of the triangular factors and preallocating data structures. One such technique is static pivoting [11], [12], [22], in which a strict pivot ordering is decided during a structural analysis phase. If a pivot is encountered that is too small then a small perturbation can be added to the diagonal in order to limit the element growth. Another technique for sparse matrices is threshold pivoting, in which an entry a_{pq} is selected as a pivot only if $|a_{pq}| \ge \phi \max_p |a_{pj}|$, where $0 < \phi < 1$. This limits the growth factor to $(1 + 1/\phi)^{n-1}$.

Another point of interest is the use of an incomplete LU factorization, where the nonzero structure of L and U is restricted based on the nonzero structure of A^k for some fill level $k \ge 0$. One possibility is to use the complete LU factors for the initial solve and to drop entries from L and U for their use as preconditioners in GMRES-IR. This could allow the preconditioned system to remain reasonably well-conditioned while reducing the cost of applying the preconditioner in some cases. The investigation of incomplete LU factorizations for our purposes remains future work.

4. Related work. Kobayashi and Ogita [20], [21] have designed an iterative refinement method for linear systems $Ax = b$ with ill-conditioned A. They compute an LU factorization with partial pivoting of A^T, perform an initial solve, then carry out standard iterative refinement. If iterative refinement fails to converge in a set number of steps then a second phase is entered: $W = U^{-T}$ is computed, the products $Z = WA$ and $d = Wb$ are formed using special techniques that yield greater accuracy, and the system $Zx = d$ is solved by LU factorization with partial pivoting followed by iterative refinement. An alternative approach requiring fewer flops is given in [21], in which the preconditioned matrix is constructed by computing an accurate residual of the LU factorization. However, both methods require explicit construction of the preconditioned system, making them unsuitable for sparse problems, and they need a second LU factorization of the preconditioned coefficient matrix.

No analysis is given in [20], [21] to support the method. However, our analysis is applicable, as we briefly indicate. We need to determine a bound on θu in (2.1) for

solution of the update equation $Ad_i = r_i$, which is actually solved via $(WA)d_i = Wr_i$. Here, both WA and Wr_i are effectively computed at precision $\overline{u}$ and then rounded to precision u, and an LU factorization of WA is used. Relative errors of order roughly $\kappa(WA)u + \overline{u}\|Z^{-1}\|\|W\|\|A\|$ are incurred. It is an assumption of this method that $\kappa(WA) \ll \kappa(A)$, and if this inequality is true we can expect $\theta u \ll 1$.

Ogita [26] and Oishi, Ogita, and Rump [27] develop algorithms for accurate solution of ill-conditioned linear systems that build approximate inverses of A or its LU factors. These algorithms are very different from that developed here and are not applicable to sparse matrices because of the need to form explicit approximations to inverses.

Arioli and Duff [1] show that FGMRES implemented in double precision and preconditioned with an LU factorization computed in single precision can deliver backward stability at double precision, even for ill conditioned systems. This work builds on the earlier work of Arioli et al. [2], which focuses on the symmetric indefinite case.

Based on this work, Hogg and Scott [18] have implemented an algorithm for symmetric indefinite systems that computes a solution using a direct solver in single precision, performs iterative refinement using the factorization of A, and then uses mixed precision FGMRES preconditioned by the direct solver to solve the original system. The stopping criteria are backward error-based.

Turner and Walker [36] frame restarted GMRES as a type of "abstract improvement algorithm". They show that restarted GMRES can be viewed as an iterative refinement process where, in each step, the corrective term is found using a fixed number of GMRES iterations. They use this connection with standard iterative refinement to justify the use of high-accuracy computations in selected parts of restarted GMRES. They do not, however, give any supporting analysis, nor do they consider preconditioned versions of GMRES. It is also worth noting that restarted GMRES may not converge.

Our approach is related to those in [1], [2], and [18] in the sense that restarted GMRES can be viewed as an iterative refinement process (see [36]). However our approach differs from those in [1], [2], and [18] in a number of ways. First, we analyze the convergence of the iterative refinement process where a preconditioned GMRES solver is used for refinement, rather than analyze the convergence of GMRES (right) preconditioned by the triangular factors. Second, our emphasis is on solving sparse nonsymmetric linear systems, whereas the algorithms in [2] and [18] are aimed at the sparse symmetric case. Finally—and most importantly—our focus is on the forward error as opposed to the backward error. Our goal is to obtain a forward error of order the unit roundoff, u, whereas a backward error of order u only guarantees a forward error of order $\kappa_\infty(A)u$.

5. Numerical experiments. In this section we compare the convergence of the forward error in standard iterative refinement and GMRES-IR for problems where the matrix is very ill conditioned. Our test problems include both random dense matrices generated in MATLAB and real-life problems from the University of Florida Sparse Matrix Collection [7], [8]. We test two combinations of u and $\overline{u}$: single/double precision ($u = 2^{-24}$, $\overline{u} = 2^{-53}$) and double/quadruple precision ($u = 2^{-53}$, $\overline{u} = 2^{-113}$). Single and double quantities and computations use built-in MATLAB datatypes and routines. For quadruple precision, we use the Advanpix Multiprecision Computing Toolbox [25] with the setting `mp.Digits(34)`, which is compliant with the IEEE 754-2008 standard [19].

For all the test problems in this section, the right-hand-side is generated in MATLAB by `b = randn(n,1)` and then normalized so that $\|b\|_\infty \approx 1$. This results in a true solution x for which $\|x\|_\infty$ is large. We also carried out the same experiments using a small-normed x, by choosing x as a random vector and constructing the right-hand side $b = Ax$ using extra precision. The results were similar to the results for large-normed x presented in this section. At the start of each experiment we use the MATLAB command `rng(1)` to seed the random number generator for reproducibility. We use the MATLAB LU function to compute the LU factorization with partial pivoting.

All quantities are stored in the working precision u. The computation of the residual at the start of each refinement step is done in precision $\overline{u}$. Within the GMRES method, the matrix-vector multiplication with A and the triangular solves with $\widehat{L}$ and $\widehat{U}$ are also performed in precision $\overline{u}$ (as explained in section 3). All other computations are performed in precision u.

In the figures, plots on the left show the relative error $e_i = \|x - \widehat{x}_i\|_\infty / \|x\|_\infty$ for standard iterative refinement (red line and circles) and GMRES-IR (blue line and squares), both started from the initial solution obtained via LU factorization with partial pivoting. Here, x is a reference solution computed in precision $\overline{u}^2$ (and stored in precision u). We let the process run until the forward error converges to the level $\epsilon = n^{1/2}u$ (indicated by a dashed black line) or the maximum number of refinement steps is reached. Plots in the middle show the computed values of $\mu_i^{(\infty)}$ (in (2.2)), and plots on the right show the computed values of $\theta_i u$ (in (2.1)) for the solves for the correction terms. In these plots, the dashed black line marks 1, which is an upper bound on $\mu_i^{(\infty)}$ and a constraint on $\theta_i u$ for convergence of the iterative refinement process.

In all tests in this section, we set the maximum number of iterative refinement steps (parameter $i_{\max}$ in Algorithm 1.1) to 15. For GMRES-IR, the maximum number of GMRES iterations m in each iterative refinement step is set to n, although convergence always occurs well before n iterations. The GMRES convergence tolerance (the parameter τ in Algorithm 3.1) is set to 10^{-4}. As discussed just after Theorem 2.1, it is not necessary to solve the correction equation to high accuracy. Since we expect the preconditioned matrix $\widetilde{A}$ to be very well conditioned the forward error of the correction will be not too much larger than the backward error, so GMRES can be terminated long before the backward error is at the level $O(u)$. In these tests, we found that $\tau = 10^{-4}$ provided a good balance between ensuring convergence of the GMRES-IR process and minimizing the number of GMRES iterations required. In practice, this parameter may be adjusted depending on the application, the conditioning of A, and the relative costs of standard iterative refinement and GMRES-IR steps.

5.1. Random dense matrices. We begin by testing random dense matrices of dimension $n = 100$ using $u = 2^{-24}$ (single precision) and $\overline{u} = 2^{-53}$ (double precision). The test matrices were generated using the MATLAB command `gallery('randsvd', n, kappa(i), 3)`, where `kappa` is a list of the desired 2-norm condition numbers 10^7, 10^8, 10^9, and 10^{10}. Our test results are shown in Figure 5.1.

Table 5.1 shows the number of standard iterative refinement (SIR) steps, the number of GMRES-IR steps, and the number of GMRES iterations summed over all GMRES-IR steps. In the parenthetical list next to the number of GMRES iterations, element i gives the number of GMRES iterations in GMRES-IR step i. Dashes in the table indicate that the method did not converge to the level $\epsilon = n^{1/2}u$ within the

TABLE 5.1
Comparison of refinement steps for each method shown in Figure 5.1.

Test	$\kappa_\infty(A)$	$\kappa_\infty(\widetilde{A})$	SIR steps	GMRES-IR steps	GMRES its
1	$6.7 \cdot 10^7$	2.3	4	2	6 (3,3)
2	$1.0 \cdot 10^9$	$6.2 \cdot 10^1$	-	2	12 (5,7)
3	$1.8 \cdot 10^{10}$	$6.4 \cdot 10^3$	-	2	37 (16,21)
4	$1.3 \cdot 10^{10}$	$1.4 \cdot 10^4$	-	3	104 (33,36,35)

maximum number of refinement steps (15 in all experiments).

From Figure 5.1, we can see that when the condition number of A is close to but still less than u^{-1} (Test 1), standard iterative refinement converges within 4 steps. GMRES-IR converges in 2 steps, each of which consists of 3 iterations of preconditioned GMRES. When the condition number of A grows to u^{-1} and larger, however, standard iterative refinement no longer converges within 15 steps (in fact it diverges in Tests 2, 3, and 4). From the plots on the right we can see that $\theta_i u > 1$ for standard iterative refinement in these tests, and so by Theorem 2.1, we should not expect standard iterative refinement to converge. For GMRES-IR, however, $\theta_i u < 1$ for all tests, and GMRES-IR converges in at most 3 refinement steps, though Table 5.1 shows that as $\kappa_\infty(A)$ grows larger more GMRES iterations are required for convergence in each refinement step.

The middle plots display $\mu_i^{(\infty)}$ (see (2.2) and section 2.1) for each iterative refinement step. We see that $\mu_i^{(\infty)}$ starts out close to $\kappa_\infty(A)^{-1}$ and grows at a rate proportional to the rate of the decrease of the error e_i. So if the iterative refinement process is converging, the error becomes small and $\mu_i^{(\infty)}$ increases towards 1. In Tests 2, 3, and 4, where standard iterative refinement does not converge, $\mu_i^{(\infty)}$ stays small.

Failure of GMRES-IR is possible if $\kappa_\infty(A)$ becomes large enough relative to u^{-1}, but the algorithm often does better than we might hope. For this class of matrices, GMRES-IR exhibits slower convergence and/or stagnation of the error once $\kappa_\infty(A) \gtrsim 5 \cdot 10^{10}$.

We now perform an analogous experiment using $u = 2^{-53}$ (double precision) and $\overline{u} = 2^{-113}$ (quadruple precision). The problems are the same size ($n = 100$) and are generated in the same way as before using the MATLAB `gallery('randsvd')` function, but now with `kappa` values 10^{15}, 10^{16}, 10^{17}, and 10^{18}.

The results are shown in Figure 5.2. We give the total number of standard iterative refinement steps, GMRES-IR steps, and GMRES iterations required for convergence in each test in Table 5.2.

The observations from the single/double experiments hold for the double/quad case as well. In these tests, standard iterative refinement converged in Test 1 and 2 but not in Tests 3 and 4. In Test 3, we can see that it appears that standard iterative refinement may eventually converge to level $\epsilon = n^{1/2}u$ if allowed enough refinement steps. Interestingly, the corresponding plot for $\theta_i u$ shows that $\theta_i u$ is very close to 1 (it is around 0.4 in each step), confirming that the iterative refinement process can still make progress despite not having $\theta_i u \ll 1$.

GMRES-IR converges in 3 refinement steps in all the double/quad tests. Table 5.2 shows that the number of GMRES iterations required per GMRES-IR step increases with $\kappa_\infty(A)$, although when both standard iterative refinement and GMRES-IR converge, the total number of GMRES iterations is about the same as the number of

TABLE 5.2
Comparison of refinement steps for each method shown in Figure 5.2.

Test	$\kappa_\infty(A)$	$\kappa_\infty(\widetilde{A})$	SIR steps	GMRES-IR steps	GMRES its
1	$5.3 \cdot 10^{15}$	1.1	6	3	6 (2,2,2)
2	$4.8 \cdot 10^{16}$	2.5	10	3	9 (3,3,3)
3	$2.9 \cdot 10^{17}$	$2.7 \cdot 10^{1}$	-	3	15 (5,5,5)
4	$1.6 \cdot 10^{18}$	$8.5 \cdot 10^{2}$	-	3	34 (10, 12, 12)

TABLE 5.3
Properties of the test matrices from the University of Florida Sparse Matrix Collection. The quantities $\mathrm{cond}(A)$, $\kappa_\infty(A)$, *and* $\kappa_\infty(\widetilde{A})$ *given in the table were computed in single precision for the first four rows and double precision for the last four rows (corresponding to the working precision in the corresponding tests).*

Matrix	Application	n	$\mathrm{cond}(A)$	$\kappa_\infty(A)$	$\kappa_\infty(\widetilde{A})$
radfr1	chem. eng.	1048	$2.1 \cdot 10^{8}$	$1.0 \cdot 10^{11}$	$1.6 \cdot 10^{3}$
adder_dcop_06	circuit sim.	1813	$1.3 \cdot 10^{10}$	$7.2 \cdot 10^{12}$	2.8
adder_dcop_19	circuit sim.	1813	$2.8 \cdot 10^{8}$	$9.1 \cdot 10^{11}$	1.0
adder_dcop_26	circuit sim.	1813	$4.3 \cdot 10^{8}$	$7.9 \cdot 10^{11}$	$4.5 \cdot 10^{1}$
mhda416	MHD	416	$1.1 \cdot 10^{19}$	$1.9 \cdot 10^{25}$	$7.3 \cdot 10^{9}$
oscil_dcop_06	circuit sim.	430	$1.7 \cdot 10^{18}$	$1.1 \cdot 10^{21}$	$4.5 \cdot 10^{1}$
oscil_dcop_42	circuit sim.	430	$6.7 \cdot 10^{17}$	$5.0 \cdot 10^{20}$	2.3
oscil_dcop_43	circuit sim.	430	$1.0 \cdot 10^{18}$	$7.7 \cdot 10^{20}$	2.1

standard iterative refinement steps. In Test 4, we see that GMRES-IR can converge to a relative error of order $n^{1/2}u$ even when $\kappa_\infty(A)$ is orders of magnitude larger than u^{-1} (in this test, $\kappa_\infty(A) = 1.6 \cdot 10^{18}$).

5.2. University of Florida Sparse Matrix Collection tests. We now test the two iterative refinement schemes on some ill-conditioned problems from the University of Florida Sparse Matrix Collection [7], [8]. Properties of the test matrices are listed in Table 5.3.

We first test matrices that are close to numerically singular for $u = 2^{-24}$ (single precision) and $\overline{u} = 2^{-53}$ (double precision); see the the first four rows of Table 5.3.

Figure 5.3 and Table 5.4 show the results in the same format as previous experiments. For the matrices adder_dcop_06 and adder_dcop_26, standard iterative refinement does not converge, as we would expect since $\theta_i u \geq 1$. For matrices radfr1 and adder_dcop_19, $\theta_i u$ is close to but still less than 1, and standard iterative refinement converges slowly. In all tests, GMRES-IR converges in only a *single* refinement step consisting of at most 3 iterations of GMRES.

The last four matrices in Table 5.3 were tested using $u = 2^{-53}$ (double precision) and $\overline{u} = 2^{-113}$ (quadruple precision). The results can be found in Figure 5.4 and Table 5.5. Again, we see that the behavior of standard iterative refinement and GMRES-IR is as expected. In short, GMRES-IR enables the accurate solution of very ill conditioned problems even when standard iterative refinement fails.

5.3. Two-stage iterative refinement. From our numerical experiments, we can see that in some cases, even though A is close to numerically singular, standard iterative refinement still converges (see, e.g., Test 1 in Figures 5.1–5.4). Since a

TABLE 5.4
Comparison of refinement steps for each method shown in Figure 5.3.

Matrix name	SIR steps	GMRES-IR steps	GMRES its
radfr1	13	1	1 (1)
adder_dcop_06	-	1	2 (2)
adder_dcop_19	-	1	1 (1)
adder_dcop_26	-	1	3 (3)

TABLE 5.5
Comparison of refinement steps for each method shown in Figure 5.4.

Matrix name	SIR steps	GMRES-IR steps	GMRES its
mhda416	5	2	3 (1,2)
oscil_dcop_06	-	2	7 (3,4)
oscil_dcop_42	-	3	9 (2,3,4)
oscil_dcop_43	-	3	10 (2,4,4)

step of standard iterative refinement is likely to be less expensive than a step of GMRES-IR (how much less expensive depends on the number of GMRES iterations in each GMRES-IR step), it may be preferable in such cases to use standard iterative refinement.

We therefore propose a *two-stage* iterative refinement process, which starts by trying standard iterative refinement and switches to GMRES-IR (and makes use of the existing LU factorization) in case of slow convergence or divergence. The decision of whether to switch from standard iterative refinement to GMRES-IR can be based on the stopping criteria suggested by Demmel et al. [9], which detect when standard refinement is converging too slowly or not at all. The optimal parameters to use in these stopping criteria will be dependent on the particular application.

6. Conclusions and future work. There is an argument in numerical analysis that a nearly singular problem does not deserve to be solved accurately, because if the data is inexact there may be an exactly singular problem within the region of uncertainty. The problem should therefore be reformulated or regularized. While this argument is often valid, there is an increasing number of applications where very ill conditioned problems do arise and an accurate solution is warranted, as explained in section 1. Moreover, the trend towards trading precision for performance (single precision for double precision, or half precision for single precision) means that problems that are only moderately ill conditioned at one precision become extremely ill conditioned at the reduced precision.

We have shown that, contrary to the conventional wisdom, iterative refinement can provide a highly accurate solution to a linear system $Ax = b$ with condition number of order u^{-1}. Our new rounding error analysis shows that it is sufficient to obtain some correct significant digits in solving the correction equation, thanks to a special property of the residuals of the iterates that enables much smaller error bounds to be obtained. Our use of GMRES to solve the correction equation preconditioned by the LU factors (GMRES-IR) yields the necessary accuracy for refinement to work, so it expands the range of accurately solvable linear systems.

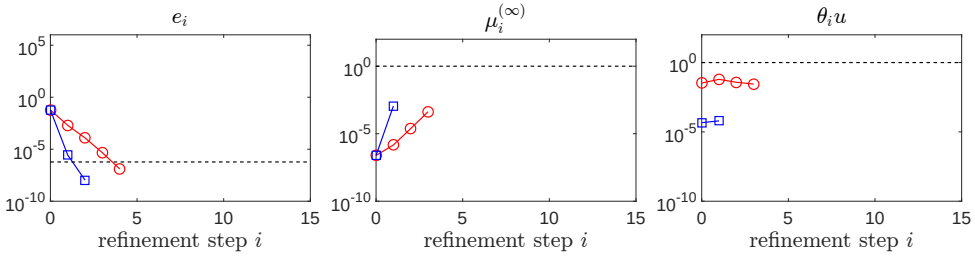
More work is required on practical implementation of GMRES-IR. As noted in section 3.1, various pivoting strategies as well as incomplete LU factorization might

be used, and the two-stage process proposed in section 5.3 requires various choices of parameters.

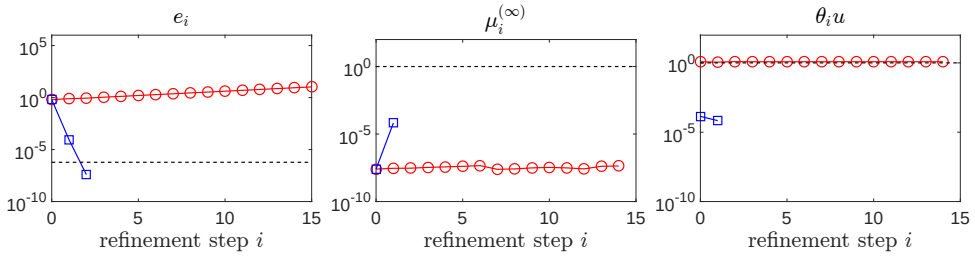
Finally, we note that in further work we have shown that GMRES-IR can tolerate the use of an LU factorization computed at less than the working precision. See [6] for details.

Our MATLAB codes are available at https://github.com/eccarson/ir3.

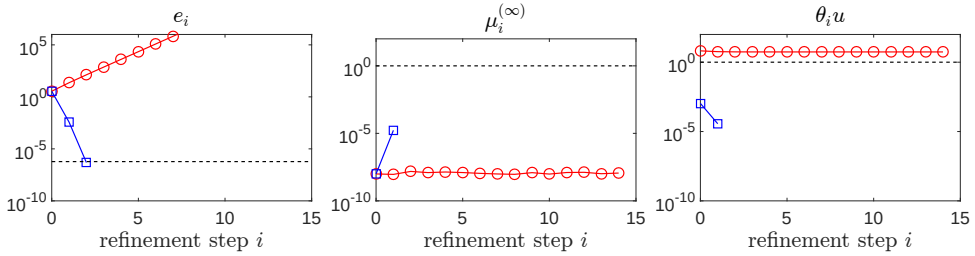
Acknowledgments. We thank the referees for their helpful suggestions.



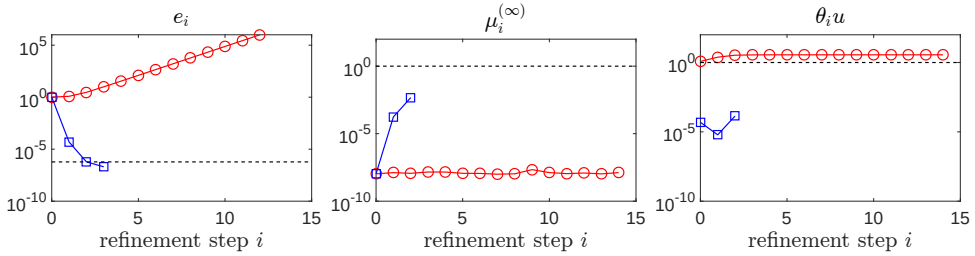
Test 1: `gallery('randsvd',100,1e7,3)`



Test 2: `gallery('randsvd',100,1e8,3)`

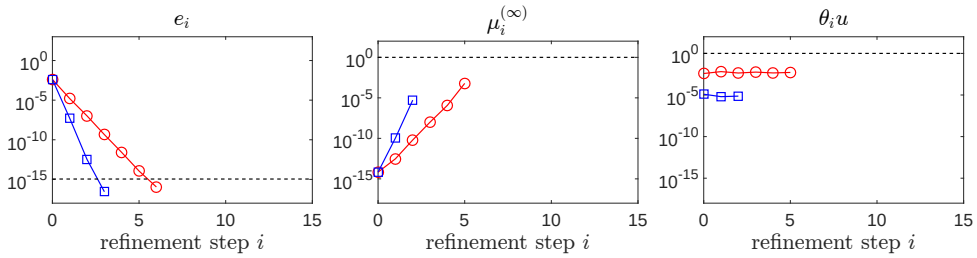


Test 3: `gallery('randsvd',100,1e9,3)`

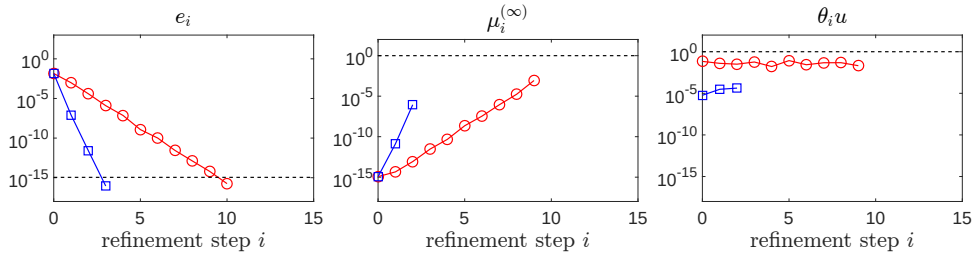


Test 4: `gallery('randsvd',100,1e10,3)`

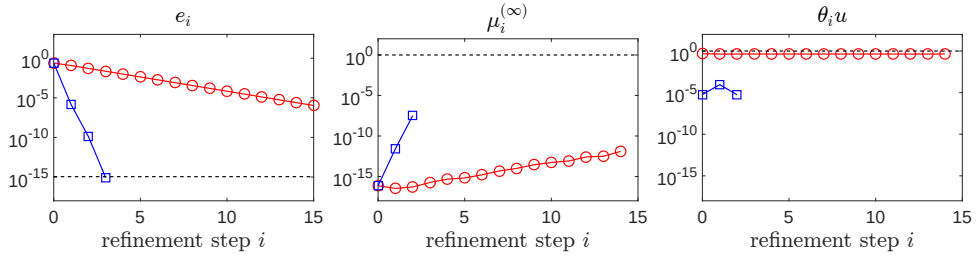
FIG. 5.1. *Relative error* $e_i = \|x - \widehat{x}_i\|_\infty / \|x\|_\infty$ *(left),* $\mu_i^{(\infty)}$ *(middle), and* $\theta_i u$ *(right) versus refinement step* i *for tests generated using the MATLAB function* `randsvd`*, with condition numbers (from top to bottom)* 10^7*,* 10^8*,* 10^9*, and* 10^{10}*. Here* $u = 2^{-24}$ *(single precision) and* $\overline{u} = 2^{-53}$ *(double precision). Red circles correspond to standard iterative refinement and blue squares correspond to GMRES-IR.*



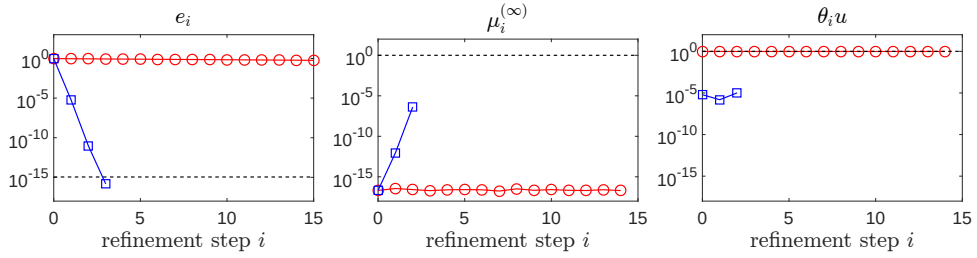
Test 1: `gallery('randsvd',100,1e15,3)`



Test 2: `gallery('randsvd',100,1e16,3)`

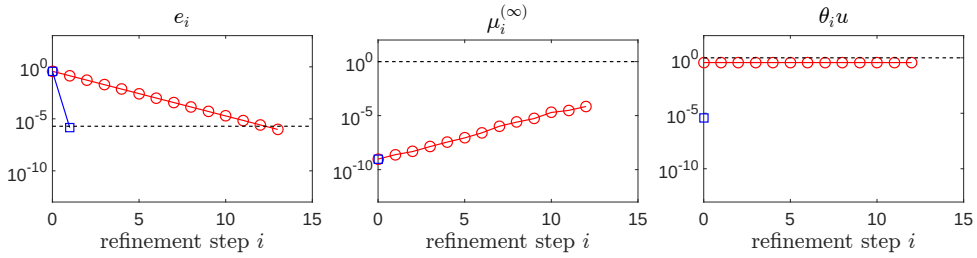


Test 3: `gallery('randsvd',100,1e17,3)`

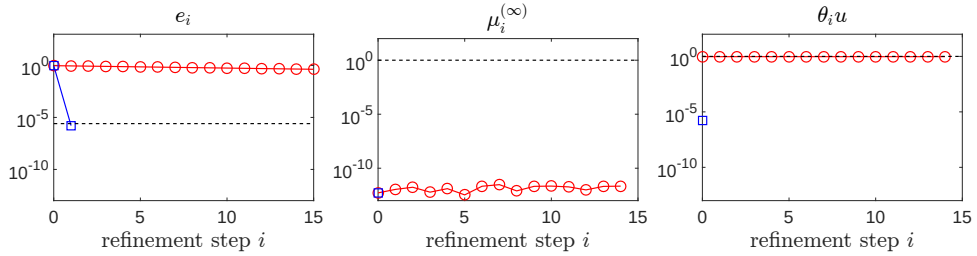


Test 4: `gallery('randsvd',100,1e18,3)`

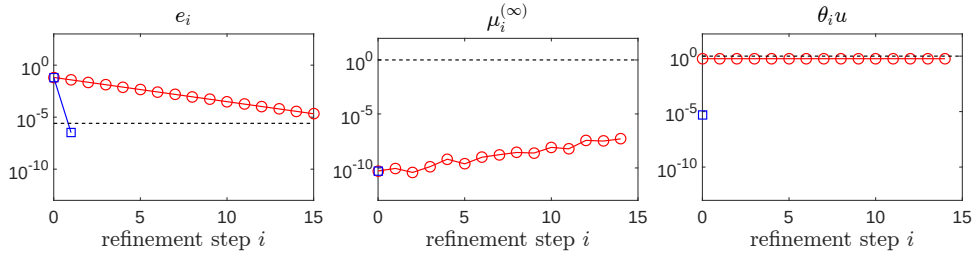
FIG. 5.2. *Relative error* $e_i = \|x - \widehat{x}_i\|_\infty / \|x\|_\infty$ *(left),* $\mu_i^{(\infty)}$ *(middle), and* $\theta_i u$ *(right) versus refinement step* i *for tests generated using the MATLAB function* `randsvd`*, with condition numbers (from top to bottom)* 10^{15}*,* 10^{16}*,* 10^{17}*, and* 10^{18}*. Here* $u = 2^{-53}$ *(double precision) and* $\overline{u} = 2^{-113}$ *(quadruple precision). Red circles correspond to standard iterative refinement and blue squares correspond to GMRES-IR.*



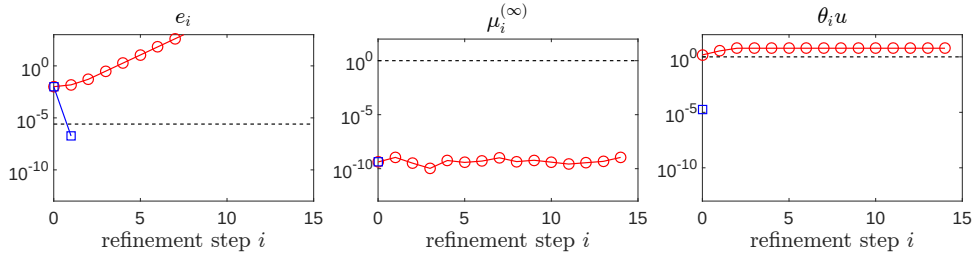
Test 1:radfr1



Test 2: adder_dcop_06

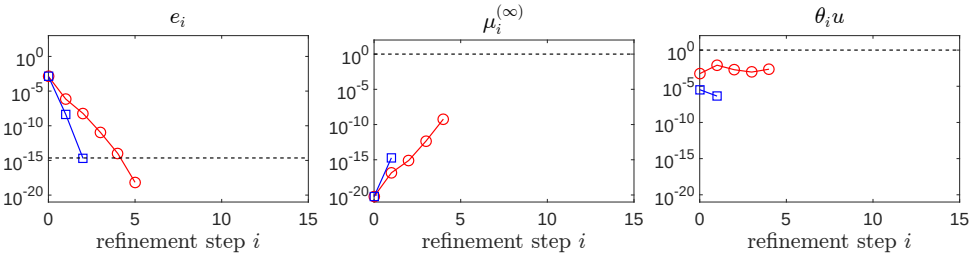


Test 3: adder_dcop_19

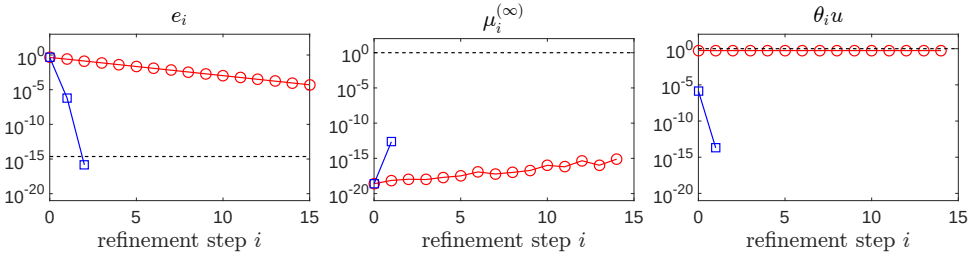


Test 4: adder_dcop_26

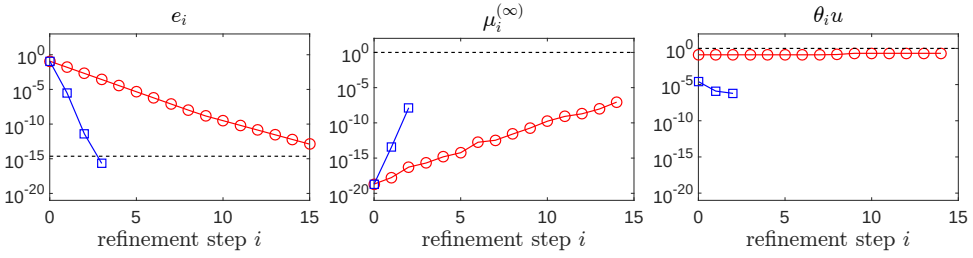
FIG. 5.3. *Relative error* $e_i = \|x - \widehat{x}_i\|_\infty / \|x\|_\infty$ *(left),* $\mu_i^{(\infty)}$ *(middle), and* $\theta_i u$ *(right) versus refinement step* i *for tests from the University of Florida Sparse Matrix Collection. Here* $u = 2^{-24}$ *(single precision) and* $\overline{u} = 2^{-53}$ *(double precision). Red circles correspond to standard iterative refinement and blue squares correspond to GMRES-IR.*



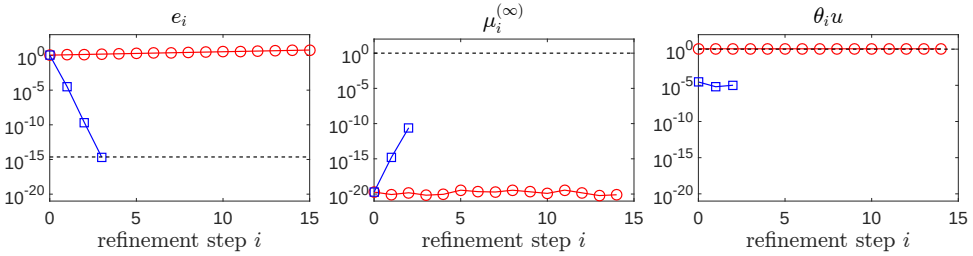
Test 1: mhda416



Test 2: oscil_dcop_06



Test 3: oscil_dcop_42



Test 4: oscil_dcop_43

FIG. 5.4. *Relative error* $e_i = \|x - \widehat{x}_i\|_\infty / \|x\|_\infty$ *(left),* $\mu_i^{(\infty)}$ *(middle), and* $\theta_i u$ *(right) versus refinement step* i *for tests from the University of Florida Sparse Matrix Collection. Here* $u = 2^{-53}$ *(double precision) and* $\overline{u} = 2^{-113}$ *(quadruple precision). Red circles correspond to standard iterative refinement and blue squares correspond to GMRES-IR.*

REFERENCES

[1] M. Arioli and I. S. Duff, *Using FGMRES to obtain backward stability in mixed precision*, Electron. Trans. Numer. Anal., 33 (2009), pp. 31–44, https://eudml.org/doc/130614.

[2] 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, https://doi.org/10.1137/060661545.

[3] M. Arioli and C. Fassino, *Roundoff error analysis of algorithms based on Krylov subspace methods*, BIT, 36 (1996), pp. 189–206, https://doi.org/10.1007/BF01731978.

[4] D. H. Bailey and J. M. Borwein, *High-precision arithmetic in mathematical physics*, Mathematics, 3 (2015), pp. 337–367, https://doi.org/10.3390/math3020337.

[5] G. Beliakov and Y. Matiyasevich, *A parallel algorithm for calculation of determinants and minors using arbitrary precision arithmetic*, BIT, 56 (2015), pp. 33–50, https://doi.org/10.1007/s10543-015-0547-z.

[6] E. Carson and N. J. Higham, *Accelerating the solution of linear systems by iterative refinement in three precisions*, MIMS EPrint 2017.24, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, July 2017, http://eprints.ma.man.ac.uk/2562.

[7] T. A. Davis, *University of Florida Sparse Matrix Collection.* http://www.cise.ufl.edu/research/sparse/matrices.

[8] T. A. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), pp. 1:1–1:25, https://doi.org/10.1145/2049662.2049663.

[9] J. Demmel, Y. Hida, W. Kahan, X. S. Li, S. Mukherjee, and E. J. Riedy, *Error bounds from extra-precise iterative refinement*, ACM Trans. Math. Software, 32 (2006), pp. 325–351, https://doi.org/10.1145/1141885.1141894.

[10] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of GMRES*, BIT, 35 (1995), pp. 309–330, https://doi.org/10.1007/BF01732607.

[11] I. S. Duff, *MA57—A code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Software, 30 (2004), pp. 118–144, https://doi.org/10.1145/992200.992202.

[12] I. S. Duff and S. Pralet, *Towards stable mixed pivoting strategies for the sequential and parallel solution of sparse symmetric indefinite systems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1007–1024, https://doi.org/10.1137/050629598.

[13] M. Ferronato, C. Janna, and G. Pini, *Parallel solution to ill-conditioned FE geomechanical problems*, International Journal for Numerical and Analytical Methods in Geomechanics, 36 (2012), pp. 422–437, https://doi.org/10.1002/nag.1012.

[14] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, *Deep learning with limited numerical precision*, in Proceedings of the 32nd International Conference on Machine Learning, vol. 37 of JMLR: Workshop and Conference Proceedings, 2015, pp. 1737–1746, http://www.jmlr.org/proceedings/papers/v37/gupta15.html.

[15] Y. He and C. H. Q. Ding, *Using accurate arithmetics to improve numerical reproducibility and stability in parallel applications*, J. Supercomputing, 18 (2001), pp. 259–277, https://doi.org/10.1023/A:1008153532043.

[16] N. J. Higham, *Iterative refinement for linear systems and LAPACK*, IMA J. Numer. Anal., 17 (1997), pp. 495–509, https://doi.org/10.1093/imanum/17.4.495.

[17] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002, https://doi.org/10.1137/1.9780898718027.

[18] J. D. Hogg and J. A. Scott, *A fast and robust mixed-precision solver for the solution of sparse symmetric linear systems*, ACM Trans. Math. Software, 37 (2010), pp. 17:1–17:24, https://doi.org/10.1145/1731022.1731027.

[19] *IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008 (revision of IEEE Std 754-1985)*, IEEE Computer Society, New York, 2008, https://doi.org/10.1109/IEEESTD.2008.4610935.

[20] Y. Kobayashi and T. Ogita, *A fast and efficient algorithm for solving ill-conditioned linear systems*, JSIAM Letters, 7 (2015), pp. 1–4, https://doi.org/10.14495/jsiaml.7.1.

[21] Y. Kobayashi and T. Ogita, *Accurate and efficient algorithm for solving ill-conditioned linear systems by preconditioning methods*, Nonlinear Theory and Its Applications, IEICE, 7 (2016), pp. 374–385, https://doi.org/10.1587/nolta.7.374.

[22] X. S. Li and J. W. Demmel, *Making sparse Gaussian elimination scalable by static pivoting*, in Proceedings of the 1998 ACM/IEEE Conference on Supercomputing, IEEE Computer Society, Washington, DC, USA, 1998, pp. 1–17, http://dl.acm.org/citation.cfm?id=509058.509092. CD ROM.

[23] D. Ma and M. Saunders, *Solving multiscale linear programs using the simplex method in quadruple precision*, in Numerical Analysis and Optimization, M. Al-Baali, L. Grandinetti, and A. Purnama, eds., no. 134 in Springer Proceedings in Mathematics and, Springer-Verlag, Berlin, 2015, pp. 223–235, https://doi.org/10.1007/978-3-319-17689-5_9.

[24] D. Ma, L. Yang, R. M. T. Fleming, I. Thiele, B. O. Palsson, and M. A. Saunders, *Reliable and efficient solution of genome-scale models of metabolism and macromolecular expression*, Scientific Reports, 7:40863 (2017), https://doi.org/10.1038/srep40863.

[25] *Multiprecision Computing Toolbox*. Advanpix, Tokyo. http://www.advanpix.com.

[26] T. Ogita, *Accurate matrix factorization: Inverse LU and inverse QR factorizations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2477–2497, https://doi.org/10.1137/090754376.

[27] S. Oishi, T. Ogita, and S. M. Rump, *Iterative refinement for ill-conditioned linear systems*, Japan J. Indust. Appl. Math., 26 (2009), pp. 465–476, https://doi.org/10.1007/BF03186544.

[28] 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, https://doi.org/10.1137/050630416.

[29] T. N. Palmer, *More reliable forecasts with less precise computations: A fast-track route to cloud-resolved weather and climate simulators?*, Phil. Trans. R. Soc. A, 372 (2014), https://doi.org/10.1098/rsta.2013.0391.

[30] S. M. Rump, *Approximate inverses of almost singular matrices still contain useful information*, Tech. Report 90.1, Hamburg University of Technology, 1990, https://doi.org/10.15480/882.319.

[31] S. M. Rump, *Inversion of extremely ill-conditioned matrices in floating-point*, Japan Journal of Industrial and Applied Mathematics, 26 (2009), pp. 249–277, https://doi.org/10.1007/BF03186534.

[32] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469, https://doi.org/10.1137/0914028.

[33] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869, https://doi.org/10.1137/0907058.

[34] S. A. Sarra, *Radial basis function approximation methods with extended precision floating point arithmetic*, Engineering Analysis with Boundary Elements, 35 (2011), pp. 68–76, https://doi.org/10.1016/j.enganabound.2010.05.011.

[35] R. D. Skeel, *Iterative refinement implies numerical stability for Gaussian elimination*, Math. Comp., 35 (1980), pp. 817–832, https://doi.org/10.1090/S0025-5718-1980-0572859-4.

[36] K. Turner and H. F. Walker, *Efficient high accuracy solutions with GMRES(m)*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 815–825, https://doi.org/10.1137/0913048.

[37] H. F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 152–163, https://doi.org/10.1137/0909010.

[38] J. H. Wilkinson, *The use of the single-precision residual in the solution of linear systems*. Unpublished manuscript, 1977.