ISSN 1749-9097

Error Analysis For Standard and GMRES-Based Iterative Refinement in Two and Three-Precisions

Nicholas J. Higham*

November 5, 2019

Abstract

We give a concise summary of conditions for the convergence of iterative refinement and GMRES-based iterative refinement in three precisions, as well as the limiting forward errors and backward errors. All combinations of precisions of practical interest are included. As well as known results, we include new results for GMRES-based iterative refinement with the preconditioner applied at the working precision and the residual computed at the working precision.

1 Introduction

The purpose of this note is twofold: to summarize in an easy-to-assimilate form the conditions from Carson and Higham [2] for convergence of iterative refinement in three precisions and GMRES-based iterative refinement (GMRES-IR) in three precisions and also to derive conditions for the convergence of GMRES-IR when the residual is computed, and the preconditioned operator applied, at the working precision.

We assume the availability of floating-point arithmetics in three precisions, with unit roundoffs u_ℓ, u, and u_r satisfying $u_r \le u \le u_\ell$. Unless otherwise stated, two of the precisions, or all three of them, may be equal.

We concentrate on normwise backward errors and forward errors, though the analysis that we summarize also applies to componentwise backward errors and forward errors.

As well as the normwise condition number $\kappa(A) = \|A\|\|A^{-1}\|$, we will need the componentwise condition number

$$\mathrm{cond}(A,x) = \frac{\|\,|A^{-1}||A||x|\,\|_\infty}{\|x\|_\infty} \le \kappa_\infty(A), \tag{1.1}$$

where $|A| = (|a_{ij}|)$.

The arithmetics of current interest are IEEE half, single, double, and quadruple precision arithmetics [5] along with bfloat16 [6]. Key parameters for these arithmetics are given in Table 1.1.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).

Table 1.1: Parameters for floating-point arithmetics: number of bits in significand (mantissa), including the implicit most significant bit; number of bits in exponent; the unit roundoff u to three significant figures; and an approximation to u^{-1} used in later tables.

	Significand	Exponent	u	u^{-1}
bfloat16	8	8	3.91×10^{-3}	2×10^{2}
fp16	11	5	4.88×10^{-4}	2×10^{3}
fp32	24	8	5.96×10^{-8}	10^{7}
fp64	53	11	1.11×10^{-16}	10^{16}
fp128	113	15	9.63×10^{-35}	10^{34}

2 Iterative Refinement

We first consider iterative refinement in traditional form, albeit with three precisions, which is described in Algorithm 2.1. The algorithm uses LU factorization for the solver, which we assume is combined with whatever form of pivoting is necessary for stability; for simplicity, permutations are omitted from our equations. The results we state below hold with LU factorization replaced by any backward stable factorization, such as Cholesky factorization for a symmetric positive definite system.

Algorithm 2.1. *Let the nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^{n}$ be given in precision u. This algorithm uses iterative refinement to generate a sequence of approximations x_i, all stored in precision u, to the solution of $Ax = b$.*

```
Compute an LU factorization $A = LU$ in precision $u_\ell$.
Solve $Ax_0 = b$ in precision $u_\ell$ using the LU factors and store $x_0$ at precision $u$.
for $i = 0\colon \infty$
    Compute $r_i = b - Ax_i$ at precision $u_r$ and round $r_i$ to precision $u_\ell$.
    Solve $Ad_i = r_i$ at precision $u_\ell$ using the LU factors and store $d_i$
    at precision $u$.
    $x_{i+1} = x_i + d_i$ at precision $u$.
end
```

Table 2.1 summarizes sufficient conditions from [2, Table 5.1, secs 3, 4, 7] for Algorithm 2.1 to converge and the limiting forward and backward errors. To be precise, what Table 2.1 and later tables say is that if the convergence condition is satisfied then the forward error or backward error will decrease until it reaches the level of the limiting error. The constant p is the maximum number of nonzeros in any row of $[A\ b]$; thus $p = n + 1$ for a dense problem.

Note that if $u_r = u^2$ then $u_r \operatorname{cond}(A, x) \le u^2\kappa_\infty(A) \le uu_\ell\kappa_\infty(A) < u$ if the convergence condition holds, so in this case we obtain a forward error of order u.

In this table, and later tables, the bounds for $\kappa_\infty(A)$ are approximate, for two main reasons. First, in the analysis they are multiplied by $f(n)$ terms that depend on the solver and are low degree polynomials in n, and we ignore these terms. Second, the exact bounds include other terms and we report only the dominant term.

Table 2.1: Summary of the results from [2] (see Table 5.1 therein): conditions for convergence and the limiting size of the forward error and normwise backward error.

	Convergence condition	Bound on limiting value
Forward error	$u_\ell \kappa_\infty(A) < 1$	$4pu_r \operatorname{cond}(A,x) + u$
Backward error	$u_\ell \kappa_\infty(A) < 1$	pu

Table 2.2: Bounds on $\kappa_\infty(A)$ such that Algorithm 2.1 using IEEE arithmetic precisions given in the first three columns is guaranteed to converge with the limiting backward and forward errors shown in the final two columns, where "half", "single", and "double" denote quantities of the order of the unit roundoffs for half precision, single precision, and double precision, respectively. In the u_ℓ column, half can be replaced by bfloat16, in which case the bound on $\kappa_\infty(A)$ must be replaced by 2×10^2.

u_ℓ	u	u_r	$\kappa_\infty(A)$	Backward error	Forward error
half	half	half	2×10^3	half	$\operatorname{cond}(A,x) \times$ half
half	half	single	2×10^3	half	half
half	single	single	2×10^3	single	$\operatorname{cond}(A,x) \times$ single
half	single	double	2×10^3	single	single
half	double	double	2×10^3	double	$\operatorname{cond}(A,x) \times$ double
half	double	quad	2×10^3	double	double
single	single	single	10^7	single	$\operatorname{cond}(A,x) \times$ single
single	single	double	10^7	single	single
single	double	double	10^7	double	$\operatorname{cond}(A,x) \times$ double
single	double	quad	10^7	double	double
double	double	double	10^{16}	double	$\operatorname{cond}(A,x) \times$ double
double	double	quad	10^{16}	double	double

Nevertheless, numerical experiments reported in [2], [3], [4] show that these bounds are indicative of the behavior in practice.

Here, and in later tables, there is a further approximation: we approximate the bounds for $\kappa_\infty(A)$ for specific precisions using the values for u^{-1} given in the last column of Table 1.1. Note that in [2] the cruder approximations $u^{-1} \approx 10^4$ for half precision and $u^{-1} \approx 10^8$ for single precision were used.

Table 2.2 specializes Table 2.1 to specific precisions.

3 GMRES-Based Iterative Refinement

Now we consider GMRES-IR, which is described in Algorithm 3.1.

Algorithm 3.1 (GMRES-IR). *Let the nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ be given in precision u. This algorithm uses GMRES-based iterative refinement using LU factors as preconditioners to generate a sequence of approximations x_i, all stored in precision*

Table 3.1: Bounds on $\kappa_\infty(A)$ such that GMRES-IR (Algorithm 3.1) using IEEE arithmetic precisions given in the first three columns is guaranteed to converge with the indicated limiting backward or forward errors, where "half", "single", and "double" denote quantities of the order of the unit roundoffs for half precision, single precision, and double precision, respectively. In the u_ℓ column, half can be replaced by bfloat16, in which case the bound on $\kappa_\infty(A)$ must be replaced by 2×10^2.

			Backward error		Forward error	
u_ℓ	u	u_r	$\kappa_\infty(A)$	Limit	$\kappa_\infty(A)$	Limit
half	half	single	2×10^3	half	10^5	half
half	single	single	10^3	single	10^4	$\mathrm{cond}(A, x) \times$ single
half	single	double	10^7	single	10^7	single
half	double	double	10^6	double	10^7	$\mathrm{cond}(A, x) \times$ double
half	double	quad	10^{16}	double	10^{11}	double
single	single	double	10^7	single	10^{11}	single
single	double	double	10^7	double	10^{10}	$\mathrm{cond}(A, x) \times$ double
single	double	quad	10^{16}	double	10^{15}	double
double	double	quad	10^{16}	double	10^{16}	double

u, to the solution of $Ax = b$.

```
Compute an LU factorization $A = LU$ in precision $u_\ell$.
Solve $Ax_0 = b$ in precision $u_\ell$ using the LU factors and store $x_0$ at precision $u$.
for $i = 0\colon \infty$
    Compute $r_i = b - Ax_i$ at precision $u_r$ and round $r_i$ to precision $u$.
    Solve $\widetilde{A}d_i \equiv \widehat{U}^{-1}\widehat{L}^{-1}Ad_i = \widehat{U}^{-1}\widehat{L}^{-1}r_i$ by GMRES at precision $u$, with
    matrix–vector products with $\widetilde{A}$ computed at precision $u_r$, and
    store $d_i$ at precision $u$.
    $x_{i+1} = x_i + d_i$ at precision $u$.
end
```

Table 3.1 summarizes sufficient conditions for Algorithm 3.1 to converge and the limiting forward and backward errors. Most of the conditions in the table are from [2, sec. 8], which makes use of [1].

The bound on $\kappa_\infty(A)$ for the forward error column in the table is

$$\kappa_\infty(A) < u^{-1/2} u_\ell^{-1} \tag{3.1}$$

[2, line 3 of p. A832]. The lines of the table where $u = u_r$ are not covered by [1] or [2]. We derive the bounds for these cases in the next subsection.

3.1 GMRES-IR in Two Precisions

The analysis in [1, sec. 3] assumes that the matrix–vector products with the preconditioned matrix $\widetilde{A} = \widehat{U}^{-1}\widehat{L}^{-1}A$ in Algorithm 3.1 are carried out at twice the working precision, that is, $u_r = u^2$. Here we are interested in running Algorithm 3.1 entirely in hardware-supported floating-point arithmetic, so in order to support a working precision of double precision we do not want to use extra precision in applying $\widetilde{A}$.

What can be said about the behavior of Algorithm 2.1 with $u_r = u$?

The analysis in [1, sec. 3] is parametrized: it denotes by $\overline{u}$ the unit roundoff for the precision at which $\widetilde{A}$ is applied. However, at some stages of the analysis $\kappa(A)\overline{u} \lesssim u$ is used (since $\overline{u} = u^2$ is assumed). so we cannot simply set $\overline{u} = u$ in the final result.

We note, first, that the analysis in [1] invokes the backward error analysis in [7] for the modified Gram–Schmidt variant of GMRES and, in order to do so, shows that

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

after $k-1$ steps of GMRES, where V_{k-1} is an $n \times (k-1)$ matrix of unit 2-norm Arnoldi basis vectors, with

$$\|\Delta V_{k-1}\|_F \lesssim k^{1/2} nu \|\widetilde{A}\|_F, \tag{3.2}$$

the latter equation being [1, eq. (3.5)]. The backward error for the computed solution of $\widetilde{A}d_i = s_i$, where $s_i = \widehat{U}^{-1}\widehat{L}^{-1}\widehat{r}_i$ and $\widehat{r}_i$ is the computed residual $b - A\widehat{x}_i$, is shown to be of order kn^2u.

Inspecting the analysis, we see that for $\overline{u} = u$ [1, eq. (3.7)] leads to

$$\|\Delta V_{k-1}\|_F \lesssim k^{1/2} nu\kappa_F(A) \|\widetilde{A}\|_F$$

in place of (3.2).

The question now is whether the GMRES backward error analysis implies a backward error bounded by $kn^2u\kappa(A)$ for the GMRES solve, instead of kn^2u when (3.2) holds. Strictly speaking, we cannot draw this conclusion from what is in [1, sec. 3] without carefully checking the details of [7], which is not an easy task. However, if we assume that all the computations in the GMRES solve (products with $\widetilde{A}$ and the GMRES computations) are done at a precision with unit roundoff $\kappa(A)u$, which is a more pessimistic scenario than we actually have, then we can directly draw the desired conclusion about the backward error of the GMRES solve.

Following through the rest of the analysis in [1, bottom two equations of page A2844], we find that the backward error of the computed solution $\widehat{d}_i$ of $\widetilde{A}d_i$ satisfies

$$\omega(\widetilde{A}, \widehat{d}_i, s_i) := \frac{\|s_i - \widetilde{A}\widehat{d}_i\|_\infty}{\|\widetilde{A}\|_\infty \|\widehat{d}_i\|_\infty + \|s_i\|_\infty} \lesssim kn^2\kappa_\infty(A)u \tag{3.3}$$

and therefore the relative error of $\widehat{d}_i$ is bounded by

$$\frac{\|d_i - \widehat{d}_i\|_\infty}{\|d_i\|_\infty} \lesssim kn^2u\kappa_\infty(A)\kappa_\infty(\widetilde{A}), \tag{3.4}$$

which has an extra factor $\kappa_\infty(A)$ in the bound compared with [1, last displayed equation on p. A2844]. This is the price we pay for not using extra precision in applying $\widetilde{A}$.

An important point to note is that u_ℓ has not appeared in these equations. Though the LU factorization is computed at precision u_ℓ, we obtain a solution to the update equation whose accuracy is independent of u_ℓ. In the analysis of [2], which carries a fourth precision u_s (which is essentially the precision of the solve for the update equation), Carson and Higham are therefore able to set $u_s = u$.

We will need a bound on the backward error for the original update equation $A\widehat{d}_i = \widehat{r}_i$. One can be obtained by noting that $\widehat{r}_i - A\widehat{d}_i = \widehat{L}\widehat{U}(s_i - \widetilde{A}\widehat{d}_i)$, so that, using a bound for $\|\widetilde{A}\|_\infty$ from [1, eq. (3.2)],

$$\begin{aligned}\|\widehat{r}_i - A\widehat{d}_i\|_\infty &\le \|\widehat{L}\widehat{U}\|_\infty \omega(\widetilde{A}, \widehat{d}_i, s_i)(\|\widetilde{A}\|_\infty \|\widehat{d}_i\|_\infty + \|s_i\|_\infty) \\ &\lesssim \|A\|_\infty \omega(\widetilde{A}, \widehat{d}_i, s_i)\big((1 + nu_\ell \kappa_\infty(A))\|\widehat{d}_i\|_\infty + \|A^{-1}\|_\infty \|\widehat{r}_i\|_\infty\big) \\ &\approx \|A\|_\infty \omega(\widetilde{A}, \widehat{d}_i, s_i)\big(nu_\ell \kappa_\infty(A)\|\widehat{d}_i\|_\infty + \|A^{-1}\|_\infty \|\widehat{r}_i\|_\infty\big) \\ &\lesssim \omega(\widetilde{A}, \widehat{d}_i, s_i)\big(nu_\ell \kappa_\infty(A)\|A\|_\infty \|\widehat{d}_i\|_\infty + \kappa_\infty(A)\|\widehat{r}_i\|_\infty\big). \end{aligned} \tag{3.5}$$

We now consider separately the convergence analyses based on backward error and forward error.

3.1.1 Backward Error

The backward error analysis in [2, sec. 4] assumes that the computed solution of the update equation $Ad_i = r_i$ satisfies [2, eq. (2.4)]

$$\|\widehat{r}_i - A\widehat{d}_i\|_\infty \le u_s(c_1\|A\|_\infty \|\widehat{d}_i\|_\infty + c_2\|\widehat{r}_i\|_\infty).$$

By (3.3) and (3.5) we have $u_s = u$, $c_1 = kn^3\kappa_\infty(A)^2 u_\ell$, and $c_2 = kn^2\kappa_\infty(A)^2$. The condition for convergence of the backward error is [2, Cor. 4.2] $(c_1\kappa_\infty(A) + c_2)u_s < 1$, that is,

$$kn^2(n\kappa_\infty(A)^3 u_\ell + \kappa_\infty(A)^2)u < 1,$$

so that we certainly need

$$\kappa_\infty(A)^2(\kappa_\infty(A)u_\ell + 1)u < 1, \tag{3.6}$$

and hence in particular we need $\kappa_\infty(A)^2 u < 1$. The $\kappa_\infty(A)$ bounds for $u_r = u$ in the backward error column of Table 3.1 are based on (3.6). Condition (3.6) is much more stringent than the condition $\kappa_\infty(A)u < 1$ for convergence of the backward error when $u_r = u^2$. In fact, condition (3.6) is clearly pessimistic. This is not too surprising given that, as we have already mentioned, the argument based on "unit roundoff $\kappa(A)u$" is pessimistic.

3.1.2 Forward Error

In the forward error analysis of [2], in the underlying assumption [2, eq. (2.3)] we can take $u_s = u$ and $u_s\|E_i\|_\infty \equiv kn^2 u\kappa_\infty(A)\kappa_\infty(\widetilde{A})$, in view of (3.4). For convergence of the forward error in iterative refinement we therefore need [2, Cor. 3.3]

$$kn^2 u\kappa_\infty(A)\kappa_\infty(\widetilde{A}) < 1. \tag{3.7}$$

(Here, we are ignoring the first term in the convergence condition [2, eq. (3.9)], because it is known usually not to be dominant.) The condition (3.7) will be satisfied for problems not too ill conditioned with respect to the working precision, as long as $\widetilde{A}$ is not too ill conditioned, so in such cases GMRES-IR will still work well.

We note that by [1, eq. (3.2)],

$$\kappa_\infty(\widetilde{A}) \lesssim (1 + n u_\ell \kappa_\infty(A))^2 \approx \max(1, n^2 u_\ell^2 \kappa_\infty(A)^2).$$

For (3.7) to hold we therefore need, ignoring the constants, $\kappa_\infty(A)^3 u u_\ell^2 < 1$, that is,

$$\kappa_\infty(A) < u^{-1/3} u_\ell^{-2/3} \tag{3.8}$$

for convergence of the forward error. The limiting forward error is of order $\mathrm{cond}(A, x)u$ by [2, Cor. 3.3]. The $\kappa_\infty(A)$ bounds for $u_r = u$ in the forward error column of Table 3.1 are based on (3.8).

3.1.3 Discussion

The analysis for $u_r = u$ leads to more complicated convergence conditions than for $u_r = u$. The bounds for convergence of the backward error in Table 3.1 are mostly more stringent than those for convergence of the forward error. Furthermore, the bound for $\kappa_\infty(A)$ for $u_r = u$ is equal to or smaller than for u the next lower precision with the same u and u_ℓ, although the limiting backward errors and forward errors are in general smaller.

These differences appear to be caused by weakness of the analysis for $u = u_r$, but at present we do not have any other way to analyze the $u_r = u$ case.

For the forward error the limiting value is $\mathrm{cond}(A, x)u$ for $u_\ell = u$ instead of u when $u_r = u^2$. This is quite acceptable in most cases, since a standard solve with LU factorization at precision u only guarantees a forward error of $\mathrm{cond}(A, x)u$.

We note that in the experiments of [2], [3], and [4] GMRES is terminated when the backward error (or relative residual) for the update equation is of order a quantity much larger than u, whereas the analysis on which Table 3.1 is based assumes a backward error of order $u_s = u$. Yet GMRES-IR converges when the bounds predict it should, which implies that there is some slack in the analysis.

In conclusion, we expect GMRES-IR to be useful for $u_r = u$, though it might not converge for as wide a range of problems as when $u_r = u^2$.

References

[1] Erin Carson and Nicholas 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(6):A2834–A2856, 2017.

[2] Erin Carson and Nicholas J. Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM J. Sci. Comput.*, 40(2):A817–A847, 2018.

[3] Azzam Haidar, Stanimire Tomov, Jack Dongarra, and Nicholas J. Higham. Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, SC '18 (Dallas, TX), Piscataway, NJ, USA, 2018, pages 47:1–47:11. IEEE Press.

[4] Nicholas J. Higham, Srikara Pranesh, and Mawussi Zounon. Squeezing a matrix into half precision, with an application to solving linear systems. *SIAM J. Sci. Comput.*, 41(4):A2536–A2551, 2019.

[5] *IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008 (revision of IEEE Std 754-1985)*. IEEE Computer Society, New York, 2008. 58 pp. ISBN 978-0-7381-5752-8.

[6] Intel Corporation. BFLOAT16—hardware numerics definition, November 2018. White paper. Document number 338302-001US.

[7] Christopher C. Paige, Miro Rozložník, and Zdeněk Strakoš. Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES. *SIAM J. Matrix Anal. Appl.*, 28(1):264–284, 2006.