Note on Iterative Refinement*

Zhengbo Zhou†

1 Iterative Refinement

Let us introduce iterative refinement before getting to linear system. Iterative refinement is usually used to bring the error or residual down to the level of the unit roundoff. The reason why the initial error exceeds the unit roundoff could be ill conditioning of the problem, numerical instability in the method used, or the use of lower precision arithmetic.

We begin with the Newton's method since iterative refinement can be seen as the application of the Newton's method.

1.1 Newton's method

Definition 1.1 (Jacobian matrix [4, Def. 4.3])**.** Let $\boldsymbol{F} = (F_1, \dots, F_n)^T : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a function defined and continuous in an (open) neighborhood of $\boldsymbol{\xi} \in \mathbb{R}^n$. Suppose further that the first partial derivatives $\partial F_i/\partial x_j$, $j = 1, \dots, n$, of F_i exist at $\boldsymbol{\xi}$ for $i = 1, \dots, n$. The *Jacobian matrix* $J_{\boldsymbol{F}}(\boldsymbol{\xi})$ of $\boldsymbol{F}$ at $\boldsymbol{\xi}$ is the $n \times n$ matrix defined by

$$\big(J_{\boldsymbol{F}}(\boldsymbol{\xi})\big)_{ij} = \frac{\partial F_i}{\partial x_j}(\boldsymbol{\xi}), \quad i, j = 1, \dots, n.$$

We drop the subscript of the Jacobian matrix if the function is obvious.

Definition 1.2 (Newton's method [4, Def. 4.5])**.** For a nonlinear system $\boldsymbol{F}(\boldsymbol{x}) = 0$, where $\boldsymbol{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$. The Newton's method is defined by

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \big(J(\boldsymbol{x}_i)\big)^{-1} \boldsymbol{F}(\boldsymbol{x}_i), \quad i = 0, 1, \dots, \tag{1.1}$$

or equivalently,

$$J(\boldsymbol{x}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) = -\boldsymbol{F}(\boldsymbol{x}_i),$$

The next theorem provides the convergence result for the Newton's method.

Theorem 1.3 ([3, Thm. 5.1.1])**.** *For a nonlinear system $\boldsymbol{F}(\boldsymbol{x}) = 0$. Provided that:*

- *there exists a solution $\boldsymbol{x}_*$ to the system,*

*Date: April 21, 2025, Manchester, UK

†Department of Mathematics, University of Manchester, Manchester M13 9PL, United Kingdom. zhengbo.zhou@student.manchester.ac.uk

- $\boldsymbol{F}'(\boldsymbol{x})$ *is Lipschitz continuous with Lipschitz constant* γ*, and*
- $\boldsymbol{F}'(\boldsymbol{x}_*)$ *is nonsingular.*

Then there exists $\delta > 0$ *such that if* $\|\boldsymbol{x}_i - \boldsymbol{x}_*\| < \delta$*, then the Newton iterate from* $\boldsymbol{x}_0$ *given by (1.1) satisfies*

$$\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_*\| \leq \gamma \|J(\boldsymbol{x}_*)^{-1}\| \|\boldsymbol{x}_i - \boldsymbol{x}_*\|^2. \tag{1.2}$$

Here, δ needs to be small enough such that [3, Lem. 4.3.1] holds.

Algorithm 1. *Mixed-precision Newton's method*

Input: A vector function $\boldsymbol{F}$; a starting vector $\boldsymbol{x}_0$; three precisions u_ℓ, u and u_r ($u_r \leq u \leq u_\ell$).
Output: A solution $\boldsymbol{x}_\infty$ to the system $\boldsymbol{F}(\boldsymbol{x}) = 0$.

1: **for** $i = 1 : \infty$ **do**
2: Compute $\boldsymbol{f}_i = \boldsymbol{F}(\boldsymbol{x}_i)$ in precision u_r.
3: Solve $J(\boldsymbol{x}_i)\boldsymbol{d}_i = -\boldsymbol{f}_i$ in precision u_ℓ.
4: Update $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{d}_i$ in precision u.
5: **end for**

Algorithm 1 is a mixed precision implementation of the Newton's method. By using more than one precision, the algorithm gives flexibility to become faster or more accurate. The floating point treatment of the Newton's method is given by Tisseur [5] or in the review by Higham and Mary [2].

It is well-known that Newton's method must be provided with accurate function values in order to produce an accurate solution. Hence $u_r \leq u$.

Also, it is well-known that Newton's method can tolerate errors in the solving step (including forming the Jacobian and solving the linear system); variants of Newton's method use a finite difference approximation to the Jacobian or freeze the Jacobian, for example using $J(\boldsymbol{x}_0)$ over $J(\boldsymbol{x}_i)$, at the cost of possible reducing the rate of convergence to linear. Hence we take $u_\ell \geq u$.

1.1.1 Rounding Error Analysis

The computed iterate $\widehat{\boldsymbol{x}}_{i+1}$ from Algorithm 1 can be written as

$$\widehat{\boldsymbol{x}}_{i+1} = \widehat{\boldsymbol{x}}_i - \big(J(\widehat{\boldsymbol{x}}_i) + \Delta J_i\big)^{-1}\big(\boldsymbol{F}(\widehat{\boldsymbol{x}}_i) + \Delta \boldsymbol{f}_i\big) + \varepsilon_i, \tag{1.3}$$

where

- $\Delta \boldsymbol{f}_i$ is the error made in computing $\boldsymbol{F}(\widehat{\boldsymbol{x}}_i)$, which we assume satisfies a bound of the form, for some function ψ,

$$\|\Delta \boldsymbol{f}_i\| \leq \psi(\boldsymbol{F}, \widehat{\boldsymbol{x}}_i, u, u_r) + u_\ell \|\boldsymbol{F}(\widehat{\boldsymbol{x}}_i)\|;$$

- ΔJ_i combines the error incurred in forming $J(\widehat{\boldsymbol{x}}_i)$ with the backward error for solving the linear system for $\boldsymbol{d}_i$ and is bounded in terms of u and u_ℓ; and
- ε_i is the error in the addition that forms $\boldsymbol{x}_i$ and is bounded in terms of u.

We are interested in the *limiting accuracy*, the smallest relative error that is guaranteed to be achieved. Under conditions of Theorem 1.3, the limiting accuracy is

$$\frac{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}_*\|}{\|\boldsymbol{x}_*\|} \approx \frac{\|J(\boldsymbol{x}_*)^{-1}\|}{\|\boldsymbol{x}_*\|}\psi(\boldsymbol{F}, \boldsymbol{x}_*, u, u_r) + u. \tag{1.4}$$

We also interested in the *limiting residual*, the smallest residual that is guarantee to be achieved, which is

$$\|\boldsymbol{F}(\widehat{\boldsymbol{x}})\| \approx \psi(\boldsymbol{F}, \widehat{\boldsymbol{x}}, u, u_r) + u\|J(\widehat{\boldsymbol{x}})\|\|\widehat{\boldsymbol{x}}\|. \tag{1.5}$$

Notice that, neither the limiting accuracy nor the limiting residual depends on the errors in evaluating J or in solving the linear system.

1.2 Iterative refinement for linear system

We now consider the iterative refinement for linear system.

Lemma 1.4. *The Newton's method for the linear system $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ is defined as*

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{d}_i,$$

where $\boldsymbol{d}_i$ is obtained by solving $A\boldsymbol{d}_i = \boldsymbol{r}_i := \boldsymbol{b} - A\boldsymbol{x}_i$, for $i = 1, 2, \ldots$.

We call $\boldsymbol{r}_i$ the residual.

Proof. Consider the vector-valued function $\boldsymbol{F}(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b}$. Solving the linear system is equivalent to finding a zero for $\boldsymbol{F}(\boldsymbol{x})$. Let us now determine the Jacobian matrix for $\boldsymbol{F}$. We have $\boldsymbol{F}_i = \sum_{k=1}^n a_{ik}\boldsymbol{x}_k - \boldsymbol{b}_i$. Then by Theorem 1.1,

$$\frac{\partial (J_{\boldsymbol{F}})_i}{\partial \boldsymbol{x}_j} = \frac{\partial\Big(\sum_{k=1}^n a_{ik}\boldsymbol{x}_k - \boldsymbol{b}_i\Big)}{\partial \boldsymbol{x}_j} = a_{ij}.$$

Namely, $J_{\boldsymbol{F}}(\boldsymbol{x}) = A$. As a result,

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{d}_i, \quad \text{where } A\boldsymbol{d}_i = -\boldsymbol{F}(\boldsymbol{x}_i) := \boldsymbol{b} - A\boldsymbol{x}_i.$$

Denote the last part as $\boldsymbol{r}_i$ gives the iteration. □

Algorithm 2. *Newton's method for linear system.*

Input: A nonsingular matrix $A \in \mathbb{R}^{n\times n}$; $\boldsymbol{b} \in \mathbb{R}^n$; an initial approximation $\boldsymbol{x}_0 \in \mathbb{R}^n$; three precisions u_r, u and u_s ($u_r \le u \le u_s$). Initially, A, $\boldsymbol{b}$ and $\boldsymbol{x}_0$ are stored at u.

Output: A sequence of approximation $\boldsymbol{x}_i$, all stored in precision u, to the solution of $A\boldsymbol{x} = \boldsymbol{b}$.

1: **for** $i = 0 : \infty$ **do**
2: Compute $\boldsymbol{r}_i = \boldsymbol{b}_i - A\boldsymbol{x}_i$ in precision u_r.
3: Solve $A\boldsymbol{d}_i = \boldsymbol{r}_i$ in precision u_s.
4: Update $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{d}_i$ in precision u.
5: **end for**

1.2.1 Iterative Refinement for a General Linear System Solver

We begin by considering the use of an arbitrary method for solving the update equation $A\boldsymbol{d}_i = \boldsymbol{r}_i$.

In floating point arithmetic, Algorithm 2 does not, in general, converge, and if it does it may not converge to the exact solution, as the exact solution may not exactly representable in precision u. The focus is to know the limiting accuracy and the limiting residual.

Carson and Higham [1] give a rounding error analysis that takes account of all the errors in Algorithm 2. Let us summarize the result. We use hats to denote computed quantities. We assume the solver on line 3 of Algorithm 2 produce a computed $\widehat{\boldsymbol{d}}_i$ that satisfies the conditions

$$\frac{\|\boldsymbol{d}_i - \widehat{\boldsymbol{d}}_i\|_\infty}{\|\boldsymbol{d}_i\|_\infty} \le u_s\theta < 1, \tag{1.6}$$

$$\|\widehat{\boldsymbol{r}}_i - A\widehat{\boldsymbol{d}}_i\|_\infty \le u_s\big(f_1\|A\|_\infty\|\widehat{\boldsymbol{d}}_i\|_\infty + f_2\|\widehat{\boldsymbol{r}}_i\|_\infty\big), \tag{1.7}$$

where f_1 and f_2 are functions of $n, A, \widehat{\boldsymbol{r}}_i$ and u_s. (1.6) focused on the relative forward error, whereas (1.7) focused on the backward error. Let us define

$$\mu_i = \frac{\|A(\boldsymbol{x} - \widehat{\boldsymbol{x}}_i)\|_\infty}{\|A\|_\infty\|\boldsymbol{x} - \widehat{\boldsymbol{x}}_i\|_\infty} \le 1. \tag{1.8}$$

and the componentwise condition number

$$\mathrm{cond}(A, \boldsymbol{x}) = \frac{\| \, |A^{-1}| \, |A| \, |\boldsymbol{x}| \, \|_\infty}{\|\boldsymbol{x}\|_\infty}.$$

Theorem 1.5 (limiting accuracy)**.** *Let Algorithm 2 be applied to a linear system $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{R}^{n\times n}$. If*

$$\phi_i = (2\min\{\mathrm{cond}(A), \kappa_\infty(A)\mu_i\} + \theta)u_s$$

is sufficiently less than 1 for each i. Then the forward error is reduced on the ith iteration by a factor approximately ϕ_i until an iterate $\widehat{\boldsymbol{x}}$ is produced for which

$$\frac{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} \le u + 4n\mathrm{cond}(A, \boldsymbol{x})u_r.$$

Remark 1.6. A backward stable solver such as LU decomposition with partial pivoting in precision u produces a computed solution $\widehat{\boldsymbol{x}}$ to $A\boldsymbol{x} = \boldsymbol{b}$ satisfies

$$\frac{\|\boldsymbol{b} - A\widehat{\boldsymbol{x}}\|_\infty}{\|A\|_\infty\|\widehat{\boldsymbol{x}}\|_\infty} \le c_n u,$$

for some modestly growing constant c_n. Iterative refinement would worsen the backward error. However, if A is ill conditioned then the early iterates are likely to have a large forward error, of order $\kappa_\infty(A)u$. We therefore expect that

$$\frac{\|\boldsymbol{b} - A\widehat{\boldsymbol{x}}_i\|_\infty}{\|A\|_\infty\|\widehat{\boldsymbol{x}}_i\|_\infty} \approx u \ll \frac{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} \approx \kappa_\infty(A)u,$$

in early iterations. By assuming $\|\widehat{x}_i\|_\infty \approx \|x\|_\infty$,

$$\frac{1}{\kappa_\infty(A)} \leq \frac{\|b - A\widehat{x}_i\|_\infty}{\|A\|_\infty \|x - \widehat{x}_i\|_\infty} \ll 1.$$

Namely, $\kappa_\infty(A)^{-1} \leq \mu_i \ll 1$.

The next theorem describe the limiting residual

Theorem 1.7 (limiting residual)**.** *Let Algorithm 2 be applied to a linear system $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{R}^{n\times n}$ is nonsingular. If $\psi = (f_1\kappa_\infty(A) + f_2)u_s$ is sufficiently less than 1, then the residual is reduced on each iteration by a factor approximately ψ until an iterate $\widehat{\boldsymbol{x}}$ is produced for which*

$$\frac{\|\boldsymbol{b} - A\widehat{\boldsymbol{x}}\|_\infty}{\|\boldsymbol{b}\|_\infty + \|A\|_\infty\|\widehat{\boldsymbol{x}}\|_\infty} \leq nu_r + u.$$

1.3 Generalized Minimal Residual method (GMRES)

The GMRES is a projection method based on taking $\mathcal{K} = \mathcal{K}_m$ and $L = A\mathcal{K}_m$, in which $\mathcal{K}_m$ is the mth Krylov subspace with $v_1 = r_0/\|r_0\|_2$. Such a technique minimizes the residual norm over all vectors in $x_0 + \mathcal{K}_m$.

1.3.1 GMRES-based Iterative Refinement

The conditions on ϕ_i and ψ in Theorems 1.5 and 1.7 mean that the use of low precision arithmetic with $u_s \gg u$ will succeed only when A is well conditioned, which is a significant limitation.

One cure is using approximate LU factors as preconditioner for the solving stage.

Algorithm 3. *Iterative refinement with GMRES (GMRES-IR5)*

Input: A nonsingular matrix $A \in \mathbb{R}^{n\times n}$ stored in precision u, $b \in \mathbb{R}^n$ stored in precision u, and five precisions u_p, u_r, u, u_g and u_ℓ $(\max(u_p, u_r) \leq u \leq u_g \leq u_\ell)$.

Output: A sequence of approximations x_i, all stored in precision u, to the solution $Ax = b$.

1: Compute the factorization $A = LU$ in precision u_ℓ.
2: Solve $LUx_0 = b$ by substitution in precision u_ℓ.
3: **for** $i = 1 : i_{\max}$ or until converge **do**
4: Compute $r_i = b - Ax_{i-1}$ in precision u_r.
5: Solve $\widetilde{A}d_i = \widehat{U}^{-1}\widehat{L}^{-1}Ad_i = \widehat{U}^{-1}\widehat{L}^{-1}r_i$ by GMRES in precision u_g, while performing the products with $\widetilde{A}$ in precision u_p.
6: Update $x_i = x_{i-1} + d_i$ in precision u.
7: **end for**

References

[1] Erin Carson and Nicholas J. Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018. (Cited on p. 4)

[2] Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022. (Cited on p. 2)

[3] Carl T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 1995. xiii+156 pp. ISBN 978-0-89871-352-7. (Cited on pp. 1, 2)

[4] Endre Süli and David Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, UK, August 2003. x+433 pp. ISBN 978-0-521-00794-8. (Cited on p. 1)

[5] Françoise Tisseur. Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 22 (4):1038–1057, 2001. (Cited on p. 2)