NEWTON'S METHOD IN MIXED-PRECISION

C. T. KELLEY*

Abstract. We investigate the use of reduced precision arithmetic to solve the linear equation for the Newton step. If one neglects the backward error in the linear solve, then well-known convergence theory implies that using single precision in the linear solve has very little negative effect on the nonlinear convergence rate.

However, if one considers the effects of backward error, then the usual textbook estimates are very pessimistic and even the state-of-the-art estimates using probabilistic rounding analysis do not fully conform to experiments. We report on experiments with a specific example. We store and factor Jacobians in double, single, and half precision. In the single precision case we observe that the convergence rates for the nonlinear iteration do not degrade as the dimension increases and that the nonlinear iteration statistics are essentially identical to the double precision computation. In half precision we see that the nonlinear convergence rates, while poor, do not degrade as the dimension increases.

Audience. This paper is intended for students who have completed or are taking an entry-level graduate course in numerical analysis and for faculty who teach numerical analysis. The important ideas in the paper are O notation, floating point precision, backward error in linear solvers, and Newton's method.

Key words. Newton's Method, Mixed-Precision Arithmetic, Backward Error, Probabilistic Rounding Analysis

AMS subject classifications. 65H10, 65F05, 45G10,

1. Introduction.

The entry level numerical analysis curriculum at the graduate level typically includes

- a description of IEEE floating point arithmetic,
- direct methods for linear equations, especially Gaussian elimination and the LU factorization,
  - estimates of backward error in terms of the size of the problem, and
- Newton's method for nonlinear equations.

However these courses do not usually connect these topics. The purpose of this paper is to do that and to apply recent results on probabilistic rounding analysis [12–14, 17] to the convergence analysis of the nonlinear Newton iteration. In particular, we will show how the precision used for the linear solve for the Newton step can be less than that for computing the nonlinear residual with no loss in the speed of convergence or the quality of the solution of the nonlinear iteration.

In § 2 we review how the classic [19] convergence estimate for Newton's method is affected by the error in the Jacobian. In § 2.2 we connect that estimate with the backward error in the linear solver. We then review the standard estimates [7, 10] for this error and explain how the new results in [13, 14, 17] affect the nonlinear convergence analysis.

Finally in § 3 we illustrate the results with a numerical example using double, single, and half precision [16, 30] for the linear solve. These results and the theory in § 2 indicate that one can safely do the linear solve in single precision if the Jacobian itself is computed to single precision accuracy. This example is large enough to see the effects of increasing the dimension of the problem, at least in half precision, but small enough that the reader can do the computation on a desktop machine.

The theory breaks down if the Jacobian is singular at the solution and we also present an example of that case to illustrate the effects of singularities.

1.1. Notation.

In this paper we denote vectors by boldfaced lower case letters and matrices by boldfaced upper case letters, for example $\mathbf{x}$ and $\mathbf{A}$. We denote the ith component of $\mathbf{x}$ by x_i to distinguish it from the ith member of a sequence of vectors $\mathbf{x}_i$. We denote the ijth entry of $\mathbf{A}$ by $\mathbf{A}_{ij}$.

2. Local Error Estimates for Newton's Method.

Most of the material is this section is standard and one can find more details in [8, 19–21, 29]. The novelty will come in § 2.2, where we explore the connection between the nonlinear convergence estimate, the backward error in the nonlinear solver, and new probabilistic rounding results from [13, 14].

[*] North Carolina State University, Department of Mathematics, Box 8205, Raleigh, NC 27695-8205, USA (Tim_Kelley@ncsu.edu). This work was partially supported by Army Research Office grant W911NF-16-1-0504, Department of Energy grant DE-NA003967, and National Science Foundation Grants OAC-1740309, DMS-1745654, and DMS-1906446.

We consider nonlinear systems of equations

$$\mathbf{F}(\mathbf{x}) = 0. \tag{2.1}$$

In (2.1) $\mathbf{F} : D \to R^N$ where D is an open convex subset of R^N. We will let $\mathbf{F}'$ denote the Jacobian matrix

$$\mathbf{F}'(\mathbf{x})_{ij} = \partial f_i(\mathbf{x})/\partial x_j$$

where

$$\mathbf{F} = (f_1, f_2, \ldots, f_N)^T.$$

We will impose a norm $\| \cdot \|$ on R^N and let $\| \cdot \|$ also denote the induced matrix norm.

The Newton iteration for solving (2.1) takes a current approximation $\mathbf{x}_c$ of a solution to a new approximation $\mathbf{x}_+$ via

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c). \tag{2.2}$$

The Newton iteration is defined if $\mathbf{F}$ is differentiable at $\mathbf{x}_c$ and $\mathbf{F}'(\mathbf{x}_c)$ is nonsingular. In this paper we assume that we compute the Newton step

$$\mathbf{s} = -\mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c)$$

by solving the linear equation

$$\mathbf{F}'(\mathbf{x}_c)\mathbf{s} = -\mathbf{F}(\mathbf{x}_c) \tag{2.3}$$

with Gaussian elimination with column pivoting [7,11].

We make the standard assumptions [8,19,29] for local convergence:

Assumption 2.1. *There is* $\mathbf{x}^* \in D$ *such that*

- $\mathbf{F}(\mathbf{x}^*) = 0$*,*
- $\mathbf{F}'(\mathbf{x}^*)$ *is nonsingular, and*
- $\mathbf{F}'(\mathbf{x})$ *is Lipschitz continuous with Lipschitz constant* γ*,* i. e.

$$\|\mathbf{F}'(\mathbf{x}) - \mathbf{F}'(\mathbf{y})\| \le \gamma \|\mathbf{x} - \mathbf{y}\|, \tag{2.4}$$

for all $\mathbf{x}, \mathbf{y} \in D$*.*

Assumption 2.1 implies that the Newton iteration (2.2) is defined for all $\mathbf{x}_c$ sufficiently near $\mathbf{x}^*$.

The convergence estimates in this section neglect any error in the linear solver and assume that the solution of (2.3) is exact. We will use the standard notation for errors

$$\mathbf{e} = \mathbf{x} - \mathbf{x}^* \text{ for } \mathbf{x} \in D.$$

For example, if $\mathbf{x}_c$ is the current point in the iteration, then $\mathbf{e}_c = \mathbf{x}_c - \mathbf{x}^*$ is the current error.

We will begin by quoting the classic local convergence theorem. We will also give the proof because it is illuminating and uses a familiar result from an entry level numerical linear algebra course.

Lemma 2.1. *Suppose* $\mathbf{A}$ *is nonsingular and*

$$\|\mathbf{A} - \mathbf{B}\| \le \frac{1}{2\|\mathbf{A}^{-1}\|} \tag{2.5}$$

then $\mathbf{B}$ *is nonsingular,* $\|\mathbf{B}^{-1}\| < 2\|\mathbf{A}^{-1}\|$*, and*

$$\|\mathbf{A}^{-1} - \mathbf{B}^{-1}\| \le 2\|\mathbf{A}^{-1}\|^2 \|\mathbf{A} - \mathbf{B}\|. \tag{2.6}$$

Theorem 2.2. *Assume that Assumption 2.1 holds, then*

$$\|\mathbf{e}_c\| \le \frac{1}{2\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\gamma}, \tag{2.7}$$

and that the ball

$$\{\mathbf{x} \mid \|\mathbf{e}\| \leq \|\mathbf{e}_c\|\} \subset D. \tag{2.8}$$

Then

$$\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|/2 \leq \|\mathbf{F}'(\mathbf{x}_c)^{-1}\| \leq 2\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|. \tag{2.9}$$

Moreover, if $\mathbf{e}_+$ *is the Newton iterate from* $\mathbf{x}_c$ *(2.2), then*

$$\|\mathbf{e}_+\| \leq \gamma\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\|\mathbf{e}_c\|^2 \leq \|\mathbf{e}_c\|/2. \tag{2.10}$$

Proof. We can use Lipschitz continuity (2.4) and (2.7) to invoke Lemma 2.1 because

$$\|\mathbf{F}'(\mathbf{x}_c) - \mathbf{F}'(\mathbf{x}^*)\| \leq \gamma\|\mathbf{e}_c\| \leq \frac{1}{2\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|}.$$

Hence $\mathbf{F}'(\mathbf{x}_c)$ is nonsingular and

$$\|\mathbf{F}'(\mathbf{x}_c)^{-1}\| \leq 2\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|. \tag{2.11}$$

Since $\mathbf{x}^* + t\mathbf{e}_c \in D$ for all $0 \leq t \leq 1$ by assumption (2.8), the fundamental theorem of calculus implies that

$$\mathbf{F}(\mathbf{x}_c) = \int_0^1 \mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c)\mathbf{e}_c \, dt. \tag{2.12}$$

Subtract $\mathbf{x}^*$ from both sides of (2.2) and use (2.12) to obtain

$$\begin{aligned} \mathbf{e}_+ &= \mathbf{e}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c) = \mathbf{e}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\int_0^1 \mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c)\mathbf{e}_c \, dt \\ &= \mathbf{F}'(\mathbf{x}_c)^{-1}\left(\int_0^1 (\mathbf{F}'(\mathbf{x}_c) - \mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c)) \, dt\mathbf{e}_c\right). \end{aligned}$$

Hence, using (2.11) and Lipschitz continuity again

$$\begin{aligned} \|\mathbf{e}_+\| &\leq \|\mathbf{F}'(\mathbf{x}_c)^{-1}\|\gamma \textstyle\int_0^1 (1-t)\, dt\|\mathbf{e}_c\|^2 \\ &= \|\mathbf{F}'(\mathbf{x}_c)^{-1}\|\gamma/2\|\mathbf{e}_c\|^2 \leq \gamma\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\|\mathbf{e}_c\|^2, \end{aligned}$$

which completes the proof using (2.7). □

In many courses (2.10) is expressed with O-notation

$$\|\mathbf{e}_+\| = O(\|\mathbf{e}_c\|^2).$$

This is appropriate when the asymptotic convergence rate is more important in the discussion than the *prefactor* $\gamma\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|$. That will be the case in this paper and we will use O-notation throughout. Moreover, the precise condition (2.7) can be replaced by "$\mathbf{x}_0$ is sufficiently close to $\mathbf{x}^*$" for the discussion in this paper. Having said that, the presence of $\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|$ is a clear and correct indicator that Theorem 2.2 does not hold if $\mathbf{F}'(\mathbf{x}^*)$ is singular. We will warn the reader a few more times about the presence of $\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|$ in the O-terms.

All one needs to describe the entire Newton iteration is an estimate like (2.10) that describes the evolution of the error from one iteration the next. Repeated applications of Theorem 2.2 imply Corollary 2.3.

COROLLARY 2.3. *Assume that Assumption 2.1 holds. Then if* $\mathbf{x}_0$ *is sufficiently near* $\mathbf{x}^*$*, the Newton iteration exists* (i. e. $\mathbf{F}'(\mathbf{x}_n)$ *is nonsingular for all* n) *and converges to* $\mathbf{x}^*$*. Moreover the convergence is* q-quadratic

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2)$$

In § 3 we plot on a semi-log scale the histories of relative residuals $\|\mathbf{F}(\mathbf{x}_n)\|/\|\mathbf{F}(\mathbf{x}_0)\|$ as a function of n. In the q-quadratic case the curve is concave, as we see in the figures. The relative residual is a good surrogate for the relative error if $\mathbf{F}'(\mathbf{x}^*)$ is well-conditioned. In fact, if Assumption 2.1 and (2.7) hold, then (see [19] page 72)

$$\frac{\|\mathbf{e}_n\|}{4\kappa(\mathbf{F}'(\mathbf{x}^*))\|\mathbf{e}_0\|} \le \frac{\|\mathbf{F}(\mathbf{x}_n)\|}{\|\mathbf{F}(\mathbf{x}_0)\|} \le \frac{4\kappa(\mathbf{F}'(\mathbf{x}^*))\|\mathbf{e}_n\|}{\|\mathbf{e}_0\|}. \tag{2.13}$$

In (2.13)

$$\kappa(\mathbf{F}'(\mathbf{x}^*)) = \|\mathbf{F}'(\mathbf{x}^*)\|\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|$$

is the condition number of $\mathbf{F}'(\mathbf{x}^*)$.

2.1. Errors in the Function and Jacobian.

Theorem 2.2 gives an idealized description of what one can expect in computations. Even so, the predictions are very accurate for all but the final step or two of a nonlinear iteration. Theorem 2.4 makes this precise. The objective of this paper is to explore the effects of errors in the Jacobian and in the linear solver on the idealized analysis in Theorem 2.2. To that end, we consider an iteration

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{J}(\mathbf{x}_c)^{-1}(\mathbf{F}(\mathbf{x}_c) + \mathbf{E}(\mathbf{x}_c)). \tag{2.14}$$

In (2.14) $\mathbf{J}(\mathbf{x}_c)$ is an approximation of $\mathbf{F}'(\mathbf{x}_c)$. One example is a finite-difference approximation of the Jacobian. The term $\mathbf{E}(\mathbf{x}_c)$ is the error in $\mathbf{F}(\mathbf{x}_c)$.

We will assume that the errors in the function and Jacobian are uniformly bounded

$$\|\mathbf{E}(\mathbf{x})\| \le \epsilon_F \text{ and } \|\mathbf{J}(\mathbf{x}) - \mathbf{F}'(\mathbf{x})\| \le \epsilon_J \le \frac{1}{4\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|} \tag{2.15}$$

for all $\mathbf{x}$ sufficiently near $\mathbf{x}^*$. The reader may think of ϵ_F as double precision floating point error, even though that is generally optimistic. The Jacobian error bound ϵ_J depends, as we will see, on the method for approximating $\mathbf{F}'(\mathbf{x}_c)$.

We we will give a result from [19, 21, 31] about the progress of the iteration in the presence of these errors. The case of interest in this paper is simple and we will give the proof.

Theorem 2.4. *Let Assumption 2.1 hold. Let (2.7) and (2.15) hold. Then $\mathbf{J}_c = \mathbf{J}(\mathbf{x}_c)$ is nonsingular and $\mathbf{x}_+$, as defined by (2.14), satisfies*

$$\|\mathbf{e}_+\| = O\left(\|\mathbf{e}_c\|^2 + \epsilon_J\|\mathbf{e}_c\| + \epsilon_F\right). \tag{2.16}$$

Proof. We express $\mathbf{e}_+$ as the sum of the error from Newton's method

$$\mathbf{e}_+^N = \mathbf{e}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c) = O(\|\mathbf{e}_c\|^2)$$

and the correction

$$(\mathbf{J}_c^{-1} - \mathbf{F}'(\mathbf{x}_c)^{-1})\mathbf{F}(\mathbf{x}_c) - \mathbf{J}_c^{-1}\mathbf{E}(\mathbf{x}_c).$$

Equation (2.15) and (2.9) imply that

$$\|\mathbf{J}_c - \mathbf{F}'(\mathbf{x}_c)\| \le \frac{1}{2\|\mathbf{F}'(\mathbf{x}_c)^{-1}\|},$$

and hence we may apply Lemma 2.1 with $\mathbf{A} = \mathbf{F}'(\mathbf{x}_c)$ and $\mathbf{B} = \mathbf{J}_c$. We may then conclude that

$$\|\mathbf{J}_c^{-1}\| \le 2\|\mathbf{F}'(\mathbf{x}_c)^{-1}\| \le 4\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|$$

and

$$\|\mathbf{J}_c^{-1} - \mathbf{F}(\mathbf{x}_c)^{-1}\| \le 2\|\mathbf{F}'(\mathbf{x}_c)^{-1}\|^2\epsilon_J \le 8\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|^2\epsilon_J = O(\epsilon_J).$$

Note that the prefactor in this O-term contains $\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|^2$, which, while not so important in this paper, tells us something about the effects of ill-conditioning on one's freedom to approximate the Jacobian.

We now apply (2.12), (2.7), and Lipschitz continuity to obtain

$$\begin{aligned} \|\mathbf{F}(\mathbf{x}_c)\| \le \|\mathbf{F}(\mathbf{x}^*)\mathbf{e}_c\| \quad & + \textstyle\int_0^1 \|\mathbf{F}(\mathbf{x}^* + t\mathbf{e}_c) - \mathbf{F}(\mathbf{x}^*)\| \, dt \|\mathbf{e}_c\| \\ & \le (\|\mathbf{F}(\mathbf{x}^*)\| + \gamma\|\mathbf{e}_c\|)\|\mathbf{e}_c\| = O(\|\mathbf{e}_c\| + \|\mathbf{e}_c\|^2). \end{aligned}$$

Combining the terms and using (watch the $\|\mathbf{F}(\mathbf{x}^*)^{-1}\|$ in the prefactor)

$$\|\mathbf{J}_c^{-1}\mathbf{E}(\mathbf{x}_c)\| \le \|\mathbf{J}_c^{-1}\|\epsilon_F \le 4\|\mathbf{F}(\mathbf{x}^*)^{-1}\|\epsilon_F = O(\epsilon_F)$$

completes the proof. □

The corollary describing the entire iteration is not a convergence result because the error does not converge to zero, rather the iteration *stagnates* when $\|\mathbf{e}_n\| = O(\epsilon_F)$. Results of this type are called *local improvement* results in [9].

Corollary 2.5. *Let the assumptions of Corollary 2.3 hold. Assume that* (2.15) *holds and that ϵ_J is sufficiently small. Then, for all n,*

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2 + \epsilon_J\|\mathbf{e}_n\| + \epsilon_F), \tag{2.17}$$

where the prefactor in the O-term is independent of n.

If, for example, $\epsilon_F = 0$ (exact arithmetic) and ϵ_J is sufficiently small, then the convergence of the iteration will be *q-linear* with *q-factor* $\le \epsilon_J$. This means that either $\|\mathbf{e}_n\| = 0$ for some $n < \infty$ or

$$\limsup_{n\to\infty} \frac{\|\mathbf{e}_{n+1}\|}{\|\mathbf{e}_n\|} \le \epsilon_J.$$

In a semilog plot of the relative residual history, a linear curve is a sign of q-linear convergence.

One case of interest in this paper is when $\epsilon_J = O(\sqrt{\epsilon_F})$. In that case

$$\epsilon_J\|\mathbf{e}_n\| = O(\sqrt{\epsilon_F}\|\mathbf{e}_n\|)$$

and hence

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2 + \epsilon_F).$$

Thefore the error in the Jacobian approximation can be neglected in the sense that the estimate for $\|\mathbf{e}_{n+1}\|$ in (2.17) is $O(\|\mathbf{e}_n\|^2 + \epsilon_F)$ with the Jacobian error playing no important role at all. We will clearly see this in the computations in § 3.

In this paper we consider a forward-difference approximation to $\mathbf{F}'$ as the alternative to an analytic expression. It's useful to look at the scalar case. Let the computed $f(x)$ be

$$\hat{f}(x) = f(x) + e(x) \text{ where } |e(x)| \le \epsilon_F.$$

Then

$$\begin{aligned} \tfrac{\hat{f}(x+h) - \hat{f}(x)}{h} \quad & = \tfrac{f(x+h) - f(x)}{h} + O(\epsilon_F/h) \\ & = f'(x) + O(h + \epsilon_F/h). \end{aligned}$$

If, as is usually the case, the prefactors in the O-terms are benign, then the error is minimized when

$$h = O(\sqrt{\epsilon_F}),$$

in which case

$$\frac{\hat{f}(x+h) - \hat{f}(x)}{h} = f'(x) + O(\sqrt{\epsilon_F}). \tag{2.18}$$

We warn the reader that the prefactor in the $O(h)$ term for the finite difference approximation is not guaranteed to be harmless [26]. We also warn the reader of a few assumptions hidden in the derivation of (2.18). We assume in the derivation that $|x|$ is $O(1)$, *i. e.* not too large nor too small, and that $|f'(x)|$ is not too small. If $|x|$ is not $O(1)$ then h will need to be scaled to conform to x [19], a detail we can ignore in this paper because the scaling of the solution in our example problem is $O(1)$. If $|f'(x)|$ is small, then the error term in (2.18) could be as large as the main term. That is trouble, as we will see in § 3.

The forward difference approximation to the Jacobian approximates $\mathbf{F}'(x)$ by $\mathbf{J}(x)$ where the kth column of $\mathbf{J}$ is

$$\mathbf{J}_k = \frac{\hat{\mathbf{F}}(\mathbf{x} + h\tilde{\mathbf{u}}_k) - \hat{\mathbf{F}}(\mathbf{x})}{h} \tag{2.19}$$

where $\hat{\mathbf{F}}(x) = \mathbf{F}(x) + \mathbf{E}(x)$ and $\tilde{\mathbf{u}}_k$ is the unit vector in the kth coordinate direction. If $h = O(\sqrt{\epsilon_F})$ then the error in the Jacobian is

$$\epsilon_J = O(\sqrt{\epsilon_F}).$$

Hence, Theorem 2.4 predicts that there will be no significant difference in the convergence of the nonlinear iteration between a double precision analytic Jacobian with the linear solve done in double and a forward difference approximate Jacobian with the linear solve done in single precision. We will see this in the examples in § 3.2.

As an example, consider solving the linear equation for the Newton step with Gaussian elimination. Think of computing the Jacobian (either analytically or with finite differences) in double precision and then storing and factoring it in either single or double precision. The discussion above indicates that there will be no loss in the nonlinear convergence rate if one uses single precision instead of double. There are two benefits. There is a clear reduction by half in storage if you use single precision. As for cost, there are two extreme cases of interest.

- If the cost of evaluating $\mathbf{F}$ and $\mathbf{F}'$ (either analytically or via finite differences) is $o(N^3)$, then the matrix factorization will be the dominant cost of the computation. Solving the equation for the Newton step in single precision instead of double will then cut the cost of the nonlinear solve almost in half. The example in § 3 is like this.
- Suppose the evaluation of $\mathbf{F}$ is $O(N^2)$ work and a finite difference Jacobian computation is the only option. Then the cost of evaluating $\mathbf{F}'$ is $O(N^3)$ because each column of the finite difference Jacobian uses a call to $\mathbf{F}$. In this case the benefit of a single precision linear solve is less significant. If the evaluation of $\mathbf{F}'$ is more than $O(N^3)$, then there is little value in a reduced precision linear solve in terms of cost.

2.2. Backward Error Estimates for *LU* Factorization. The local improvement estimate (2.17) does not take the backward error in the linear solver into account. In fact, most of the literature in nonlinear equations (for example [6, 8, 19, 20, 27, 29]) makes the implicit assumption that the backward error in the solver can be neglected and focuses instead on either the forward error in the Jacobian itself $\|\mathbf{J}_c - \mathbf{F}'(\mathbf{x}_c)\|$ or a formulation in terms of the inexact Newton condition,

$$\|\mathbf{F}'(\mathbf{x}_c)\mathbf{s} + \mathbf{F}(\mathbf{x}_c)\| \le \eta \|\mathbf{F}(\mathbf{x}_c)\|,$$

which is a small residual condition on the linear equation for the Newton step (2.3).

While either of these expressions of error could include the backward error as part of the estimate, that is not done explicitly and is not part of the discussion or the examples in those papers. One purpose of this paper is to question the assumption that the backward error can be neglected.

The missing component in (2.17) is the backward error in the solver. We let $\hat{\mathbf{L}}$ and $\hat{\mathbf{U}}$ be the computed *LU* factors of $\mathbf{J}$ and $\hat{\mathbf{J}} = \hat{\mathbf{L}}\hat{\mathbf{U}}$. The backward error in the solver is

$$\delta\mathbf{J} = \hat{\mathbf{J}} - \mathbf{J}.$$

The reader should think of $\mathbf{J}$ as an analytic Jacobian or a forward-difference approximation. We will assume, as is the case in the example in § 3, that $\|\mathbf{J}\|$ is uniformly bounded in the dimension N of the problem.

We can incorporate the backward error into (2.17) and obtain,

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2 + (\epsilon_J + \|\delta\mathbf{J}\|)\|\mathbf{e}_n\| + \epsilon_F).$$

Since $\epsilon_J = O(\sqrt{\epsilon_F})$ in this paper, we can neglect ϵ_J and have

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2 + \|\delta\mathbf{J}\|\|\mathbf{e}_n\| + \epsilon_F), \tag{2.20}$$

clearly exposing the role, if any, of the backward error. The estimate (2.20) suggests that one could attempt to detect a large backward error via examination of the convergence of the nonlinear iteration, which we do in § 3.

Now let ϵ_p be the precision of the linear solver. This means that we store the Jacobian and do the factorization and triangular solves with precision ϵ_p. The classic estimate [7, 11] uses the L^1 norm and contains the dimension N in a nontrivial manner. The first step is an estimate for the component-wise backward error

$$|\delta\mathbf{J}|_{ik} \le \gamma_N(|\hat{\mathbf{L}}||\hat{\mathbf{U}}|)_{ik}, \tag{2.21}$$

where, for a matrix $\mathbf{A}$, $|\mathbf{A}|$ is the matrix with entries $|\mathbf{A}_{ij}|$, and

$$\gamma_N = \frac{N\epsilon_p}{1 - N\epsilon_p}. \tag{2.22}$$

The starting point for this estimate of the component-wise backward error is estimation of products of the form

$$\prod_{i=1}^{N}(1+\delta_i)^{\rho_i}$$

where $|\delta_i| \le \epsilon_p$ and $\rho_i = \pm 1$. The standard estimate is ([11], page 63)

$$\left|\prod_{i=1}^{N}(1+\delta_i)^{\rho_i}\right| \le 1 + \gamma_N. \tag{2.23}$$

The final step in the proof of (2.21) is to count the floating point operations in the factorization and use (2.23).

The classic worst case bound for $\|\delta\mathbf{J}\|$ uses the L^1 matrix norm, *i. e.* the maximum column sum. We will use the L^1 norm in this part of the paper for that reason. However, the rest of the paper is norm-independent and uses the L^1 estimates as guidance, a reasonable idea since only the magnitude of the bound is important in most applications [11]. With (2.22), the norm estimate is

$$\|\delta\mathbf{J}\|_1 \le \gamma_N\|\hat{\mathbf{L}}\|_1\|\hat{\mathbf{U}}\|_1.$$

The magnitudes of the entries of $\hat{L}$ are bounded by 1. The worst case would be if $|\hat{\mathbf{L}}_{i1}| = 1$ for all i. Then

$$\|\hat{\mathbf{L}}\|_1 = N.$$

Following [7], we define the growth factor

$$g = \max_{1\le i,j\le N}\frac{\max|\hat{\mathbf{U}}_{ij}|}{|\mathbf{J}_{ij}|}.$$

Hence, again using the worst case estimate for the L^1 norm of $\mathbf{U}$

$$\|\hat{\mathbf{U}}\|_1 \le gN\|\mathbf{J}\|_1.$$

So, at this point we have

$$\|\delta\mathbf{J}\|_1 \le \gamma_N N^2 g\epsilon_p = O(\epsilon_p g N^3).$$

While the growth factor g can be as large as 2^{N-1}, that is a worst-case bound first seen in a famous example and only rarely seen in practice ([11], page 177–178). However, one can justify neglecting g in most applications, so we will do that.

Since $\|\mathbf{J}\|_1 = O(1)$, we obtain, neglecting g,

$$\|\delta\mathbf{J}\|_1 \le \gamma_N N^2 \epsilon_p = O(N^3 \epsilon_p). \tag{2.24}$$

This is, as the textbooks clearly say, ridiculous. For example, if $\epsilon_J \approx 10^{-16}$, then the backward error is $O(1)$ for any $N > 250,000$ and $> .001$ for $N > 21,000$. This would tell us that we should expect Gaussian elimination with column pivoting to return only three figures of accuracy for some fairly small problems and says that $\|\delta\mathbf{J}\|$ could cause some real trouble with slow convergence of Newton's method. This pessimism is not confirmed by practice.

We can obtain a more realistic bound than (2.24) if we replace the worst case bound for $\|\hat{\mathbf{L}}\|_1$ and $\|\hat{\mathbf{U}}\|_1$ with the best-case, $\|\hat{\mathbf{L}}\|_1 = O(1)$ and $\|\hat{\mathbf{U}}\|_1 = O(\|\mathbf{J}\|_1) = O(1)$. Then we have

$$\|\delta\mathbf{J}\|_1 \le \gamma_N \epsilon_p = O(\epsilon_p N). \tag{2.25}$$

This is much better. Remember we want $\|\delta\mathbf{J}\|_1 = O(\sqrt{\epsilon_F})$. If ϵ_F is double precision unit roundoff (1.1×10^{-16}), $\epsilon_J = O(\sqrt{\epsilon_F})$ (think of a forward difference approximation), and $\epsilon_p = \epsilon_F$ (i.e. we do the solve in double precision), then (2.25) tells us that $\|\delta\mathbf{J}\| = O(\sqrt{\epsilon_F})$ as long as $N < 10^8$. Problems with dimension $N > 10^8$ are far too large for dense matrix Gaussian elimination on a typical desktop computer, so we can expect the backward error to have little effect on the nonlinear iteration.

We will consider doing the linear solver in a lower precision after making our estimate of $\|\delta\mathbf{J}\|$ even more optimistic. New results in probabilistic roundoff analysis [13,14] attempt to make theory better reflect practice.

The new formulation of (2.23) in [13] is a probabilistic statement. The advantage is that one can replace γ_N with

$$\tilde{\gamma}_N(\lambda) = \exp\left(\lambda\sqrt{N}\epsilon_p + \frac{N\epsilon_p^2}{1-\epsilon_p}\right) - 1 = \lambda\sqrt{N}\epsilon_p + O(\epsilon_p^2),$$

where λ can be tuned as we will see below. The analog to (2.23) (Theorem 2.4, page A2819 in [13]) is

Theorem 2.6. *Let $\{\delta_j\}_{j=1}^N$ be independent random variables with mean zero and bounded in absolute value by ϵ_p. Then, for any $\lambda > 0$ the bound*

$$\left|\prod_{i=1}^{N}(1+\delta_i)^{\rho_i}\right| \le 1 + \tilde{\gamma}_N(\lambda) \tag{2.26}$$

holds with probability at least

$$P(\lambda) = 1 - 2exp\left(\frac{-\lambda^2(1-\epsilon_p)^2}{2}\right).$$

Since λ is a free parameter and $P(\lambda) \to 1$ very rapidly as $\lambda \to \infty$, one can increase λ to make $P(\lambda)$ near one and still obtain a bound of $O(\sqrt{N}\epsilon_p)$ with high probability for the left side of (2.26). We will give a concrete example when we state the result from [13] for the backward error in the LU factorization.

The application to the backward error for LU is not as straightforward as in the deterministic case. While counting operations is still the way to obtain the bound, the probability term is more complicated. Define

$$Q(\lambda, N) = 1 - N(1 - P(\lambda).$$

Theorem 2.7 (Theorem 3.6, page A2824 in [13]) is the component-wise backward error estimate.

Theorem 2.7. *Assume that all errors in every binary operation in Gaussian elimination are independent random variables of mean zero. Let $\lambda > 0$ be given. Then the computed LU factors from Gaussian elimination on $\mathbf{J} \in R^{N\times N}$ satisfy*

$$\hat{\mathbf{L}}\hat{\mathbf{U}} = \hat{\mathbf{J}} = \mathbf{J} + \delta\mathbf{J},$$

where

$$(2.27) \qquad |\delta \mathbf{J}| \le \tilde{\gamma}_N(\lambda) |\hat{\mathbf{L}}||\hat{\mathbf{U}}| = (\lambda \sqrt{N} \epsilon_p + O(\epsilon_p^2)) |\hat{\mathbf{L}}||\hat{\mathbf{U}}|$$

holds with probability at least $Q(\lambda, N^3/3 + N^2/2 + N/6)$.

Now one is free to adjust λ. Using $\lambda = \sqrt{\log(N)}$ for the largest N of interest is one approach. An example from [13,14] illustrates the result. If we set $\lambda = 13$, then the probability that (2.27) fails to hold is

$$(N^3/3 + N^2/2 + 7N/6) P(13) \approx 1.3 * 10^{-7} \text{ for } N \le 10^{10} \ .$$

In this case (2.25) can be improved to

$$(2.28) \qquad |\delta \mathbf{J}| \le (13 \sqrt{N} \epsilon_p + O(\epsilon_p^2)) |\hat{\mathbf{L}}||\hat{\mathbf{U}}|,$$

for $N \le 10^{10}$, and hence for all desktop-sized problems.

Returning to general norms and the case $\|\mathbf{F}'\| = O(1)$, the idea for this paper is that (2.28) implies that, with high probability, we can use (neglecting the ϵ_p^2 terms)

$$(2.29) \qquad \|\delta \mathbf{J}\| \le 13 \sqrt{N} \epsilon_p$$

for the values of N of interest under our best-case assumptions that $\|\hat{\mathbf{U}}\|$ and $\|\hat{\mathbf{L}}\|$ are $O(1)$. We will explore some consequences of that below and report on numerical observations in § 3.

Now consider the case where $\epsilon_p = \epsilon_s = 6.0 \times 10^{-8} = O(\sqrt{\epsilon_F})$ is single precision unit roundoff. In that case (2.28) tells us that we cannot completely neglect the backward error unless N is very small, say < 10 However, (2.20) implies that the $\|\delta \mathbf{J}\|$ term on the right side of (2.20) will only become important when $\|\mathbf{e}_n\| \approx \|\delta \mathbf{J}\|$ and this will only happen at the end of the iteration. For example, if $N = 10000$ and we neglect the norms of the LU factors, then, with high probability, $\|\delta \mathbf{J}\| \le 7.8 \times 10^{-5}$. In that case (2.25) and (2.20) indicate that the convergence will be q-linear, but still fast enough to be useful. The estimate also shows that that $\|\delta \mathbf{J}\| \|\mathbf{e}_n\|$ will be the dominant term in (2.20) only if $\|\mathbf{e}_n\| \le 8 \times 10^{-5}$, *i. e.* for the last one or two iterations before stagnation. Figure 3.4 illustrates this, but the effect is visible only very near stagnation.

Many computing environments support half precision computations. Unlike double and single precision, which conform to the IEEE standard [16,30], there are many half precision formats. This paper will focus on IEEE half precision (see Table 3.5, page 23 in [16]). If we do the linear solves in half precision, then $\epsilon_p = \epsilon_h = 4.9 \times 10^{-4}$. We can invoke (2.28) and (2.20) to predict that the nonlinear iteration will see the effects of large N much earlier than a single precision computation, so we can expect to see the reduction in convergence rate more readily. If, for example, $N = 10000$, we should expect to converge slowly, if at all, because the estimate is that $|\delta \mathbf{J}| \le .64$ The largest half precision computation we could do for this paper had size $N = 16,384$, for which the estimate for $|\delta \mathbf{J}| \approx .8$. So as the dimension increases, the deterioration in the convergence rate should be clearly visible. This is something we can test on a desktop computer. The results in § 3 show that this estimate is still pessimistic.

We will explore these estimates in § 3 by solving a nonlinear problem and increasing the dimension to see if one can observe changes in the nonlinear convergence rates. (2.29) suggests that we will see very little differences between single precision and double precision and a significant difference between half precision and either single or double precision.

3. Example: Chandresekhar H-Equation.

As an example we consider the mid-point rule discretization of the Chandrasekhar H-equation [3],

$$(3.1) \qquad \mathcal{F}(H)(\mu) = H(\mu) - \left(1 - \frac{c}{2} \int_0^1 \frac{\mu H(\mu)}{\mu + \nu} \, d\nu \right)^{-1} = 0.$$

The nonlinear operator $\mathcal{F}$ is defined on $C[0,1]$, the space of continuous functions on $[0,1]$.

This equation has a well-understood dependence on the parameter c [4,28]. The equation has unique solutions at $c = 0$ and $c = 1$ and two solutions for $0 < c < 1$. There is a simple fold singularity [18] at $c = 1$. Only one [2,3] of the two solutions for $0 < c < 1$ is of physical interest and that is the one easiest to find numerically. One must perform a continuation computation to find the other one. The structure of the singularity is preserved if one discretizes the integral with any quadrature rule with positive weights that integrates constants exactly.

For the purposes of this paper the composite midpoint rule will suffice. The N-point composite midpoint rule is

$$\int_0^1 f(\nu)\, d\nu \approx \frac{1}{N} \sum_{j=1}^{N} f(\nu_j)$$

where $\nu_j = (j - 1/2)/N$ for $1 \leq j \leq N$. This rule is second-order accurate for sufficiently smooth functions f. The solution of (3.1) is, however, not smooth enough. $H'(\mu)$ has a logarithmic singularity at $\mu = 0$. We will use the L^2 norm to compute $\|\mathbf{F}(\mathbf{x})\|$ in the tables and figures.

Increasing N has no effect on the conditioning of the Jacobian nor, if the backward error in the linear solve can truly be neglected, on the iteration statistics [1, 25]. Hence we can clearly, but indirectly, observe the effects of N on the Jacobian backward error through the performance of the nonlinear solver.

The discrete problem is

$$\text{(3.2)} \qquad \mathbf{F}(\mathbf{x})_i \equiv x_i - \left(1 - \frac{c}{2N} \sum_{j=1}^{N} \frac{x_j \mu_i}{\mu_j + \mu_i} \right)^{-1} = 0.$$

One can simplify the approximate integral operator in (3.2) and expose some useful structure. Since

$$\frac{c}{2N} \sum_{j=1}^{N} \frac{x_j \mu_i}{\mu_j + \mu_i} = \frac{c(i-1/2)}{2N} \sum_{j=1}^{N} \frac{x_j}{i+j-1},$$

the approximate integral operator is the product of a diagonal matrix and a Hankel matrix and one can use a fast Fourier transform to evaluate the operator-vector product with $O(N \log(N))$ work [10, 21].

We can express the approximation of the integral operator in matrix form

$$\mathbf{M}(\mathbf{x})_{ij} = \frac{c(i-1/2)}{2N} \sum_{j=1}^{N} \frac{x_j}{i+j-1}$$

and compute the Jacobian analytically as

$$\mathbf{F}'(\mathbf{x}) = \mathbf{I} - \text{diag}(\mathbf{G}(\mathbf{x}))^2 \mathbf{M}(\mathbf{x}),$$

where

$$\mathbf{G}(\mathbf{x})_i = \left(1 - \frac{c}{2N} \sum_{j=1}^{N} \frac{x_j \mu_i}{\mu_j + \mu_i} \right)^{-1}.$$

Hence the data for the Jacobian is already available after one computes $\mathbf{F}(\mathbf{x}) = \mathbf{x} - \mathbf{G}(\mathbf{x})$ and the Jacobian can be computed with $O(N^2)$ work. We do that in this example and therefore the only part of the solve that requires $O(N^3)$ work is the matrix factorization.

One could also approximate the Jacobian with forward differences using (2.19) at a cost of N function evaluations. As we saw in § 2, if one computes $\mathbf{F}$ in double precision with unit roundoff ϵ_F, then $h = O(\sqrt{\epsilon_F})$ is a reasonable choice [19]. In that case the error in the Jacobian is $O(\sqrt{\epsilon_F}) = O(\epsilon_s)$ where ϵ_s is unit roundoff in single precision. The cost of a finite difference Jacobian in this example is $O(N^2 \log(N))$ work.

The analysis in § 2 suggests that there is no significant difference in the nonlinear iteration from either the choice of analytic or finite difference Jacobians or the choice of single or double precision for the linear solver. The results in § 3.2 support that suggestion.

One should be more cautious with half precision because the error in the solver is larger than single precision roundoff, so we would expect linear convergence prior to stagnation at best. In § 3.3 we see linear convergence and show that the convergence rate of the nonlinear solver does degrade with dimension for small problems sizes, but eventually stabilizes.

In all cases the initial iterate $\mathbf{x}_0$ had all components equal to one. We consider three cases. If $c = .5$ or $c = .99$ the Jacobian is nonsingular and the theory in § 2 is applicable. The case $c = 1.0$ is different because the Jacobian is singular at the solution.

3.1. Computations. The computations reported in this section were done in Julia v 1.5.3 on a 2019 Apple iMac with eight cores and 64GB of memory. Julia supports half precision in software and so computations in half precision are very slow. We report on computations for dimensions $N = 2^{10} \dots 2^{14}$. The results for half precision required two weeks of computer time and increasing the dimension beyond 2^{14} was not practical. In all the figures we plot the relative residual $\|\mathbf{F}(\mathbf{x}_n)\|/\|\mathbf{F}(\mathbf{x}_0)\|$ as a function of the iteration counter n. This is a reasonable surrogate for the errors in the nonsingular cases $c = .5$ and $c = .99$ in view of (2.13). In the singular case $c = 1$, $\|\mathbf{F}(\mathbf{x}_n)\|/\|\mathbf{F}(\mathbf{x}_0)\| = O(\|\mathbf{e}_n\|^2)$ [4]. However, even in that case we can observe the effects, if any, of backward error in the Jacobian using the relative residual.

In the computations we computed the analytic Jacobian in double precision and then stored and factored the Jacobian in double, single, or half precision (the solver precision). We computed the columns of the forward difference Jacobian in double precision using (2.19) and then stored them in the solver precision to build the forward difference Jacobian. The factorization and triangular solves were carried out in the solver precision. We converted the residual to the solver precision before computing the step. This conversion keeps the solver from promoting the intermediate steps in the solve in Julia and is important for performance. By the way, Matlab does this conversion automatically. In half precision one must also scale the residual before the conversion to avoid underflow errors [15]. After the solve the step was automatically promoted to double precision upon addition to the current nonlinear iteration.

The computations used the author's SIAMFANLEquation.jl Julia package [22–24]. The files (codes, data, and an IJulia notebook) for these results are available at
https://github.com/ctkelley/MPResults.

The solvers with the SIAMFANLEquation.jl package are available at
https://github.com/ctkelley/SIAMFANLEquations.jl.

3.2. Can You Tell the Difference Between Single and Double? We consider two cases $c = .5$ and $c = .99$ with nonsingular Jacobian. Theorem 2.4 is applicable. We see a difference in the convergence because the Jacobian for $c = .99$ is nearer to singularity that than for $c = .5$. In these cases Figures 3.2 and 3.1 show very little dependence of the iteration histories on either the precision of the factorization or the use of a finite difference or analytic Jacobian. The only meaningful difference is the third iteration for $c = .5$, The iteration is very near stagnation in that case and the analytic Jacobian combined with a factorization in double precision reaches stagnation before the other three methods, all of which have a Jacobian with error $O(\sqrt{\epsilon_F})$. In all cases, if one were to terminate the iteration when the relative residual fell below 10^{-8}, then all the iterations in Figures 3.1 and 3.2 would stop at the same iteration (3 for $c = .5$ and 5 for $c = .99$).

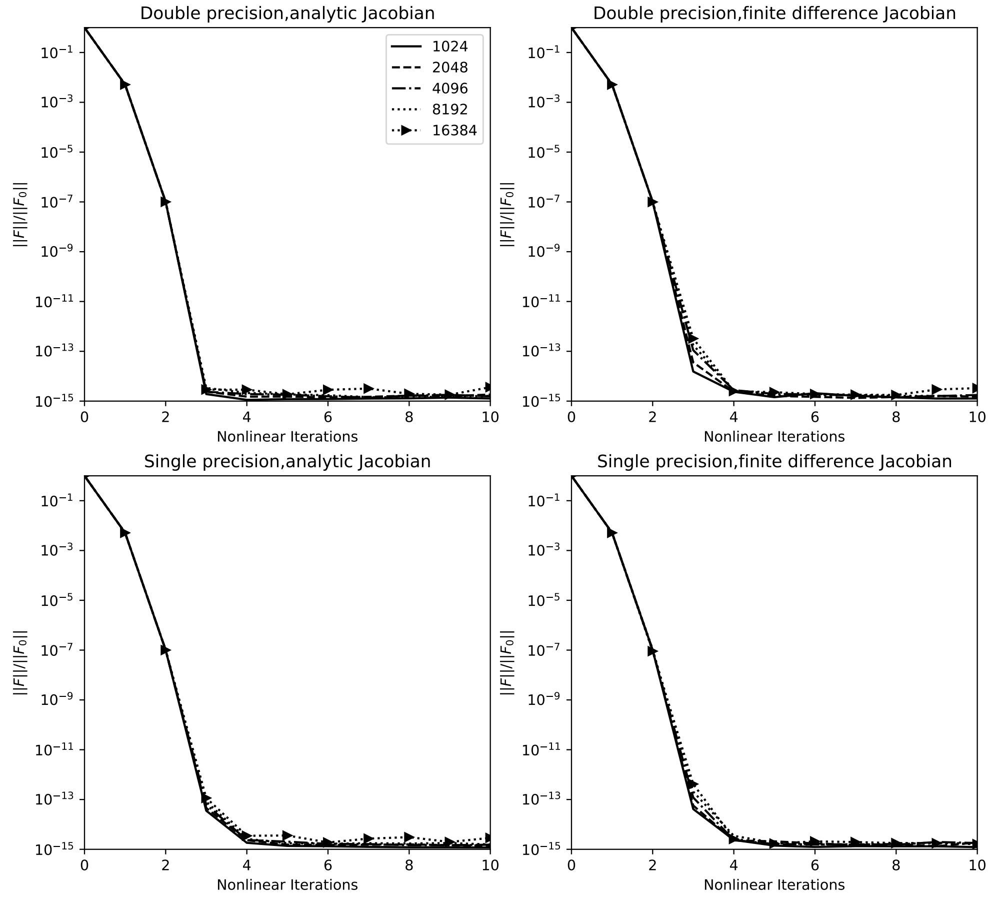
The singular case $c = 1$ is very different [4]. The assumptions of Theorem 2.4 do not hold and we do not see the behavior that the theorem predicts. To begin with, the convergence is not quadratic, but q-linear with q-factor 1/2. Moreover, the initial iterate must not only be near the solution, but the initial error must be mostly in the direction of the null space of the Jacobian at the solution. We see q-linear convergence in Figure 3.3. One way to understand the convergence rate is to solve $x^2 = 0$ with Newton's method. The iteration is

$$x_{n+1} = x_n - \frac{x_n^2}{2x_n} = x_n/2$$

giving a q-factor of 1/2. The structure of the singularity of the H-equation is very similar to this in the component of the error in the direction of the null space of the Jacobian at the solution.

One can also see that convergence history is very different for the forward difference approximation to the Jacobian. In the scalar case, for example, if $f'(x) = 0$, then the relative error in the finite difference approximation can be large and the estimate (2.18) is true, but not very useful. This is especially the case if ϵ_F is an absolute error, which is often the case. As an example, let $f(x) = cos(x)$, $x = 10^{-6}$, and $h = 10^{-7}$. The $f'(x) = -\sin(x) \approx -x$. The finite difference approximation is $\approx -1.05 \times 10^{-6}$ and has only two figures of accuracy.

Fig. 3.1. *Residual Histories: Single and Double Precision, $c = .5$*



As in the scalar case, if $\mathbf{F}'(\mathbf{x})$ is singular or nearly so then the finite-difference approximation may be poor in directions in the null space of $\mathbf{F}'(\mathbf{x})$. Moreover, the estimate (2.17) for the nonlinear iteration depends on nonsingularity of the Jacobian. We should not be surprised when things go wrong and see an example of this in Figure 3.3.

FIG. 3.2. *Residual Histories: Single and Double Precision,* $c = .99$

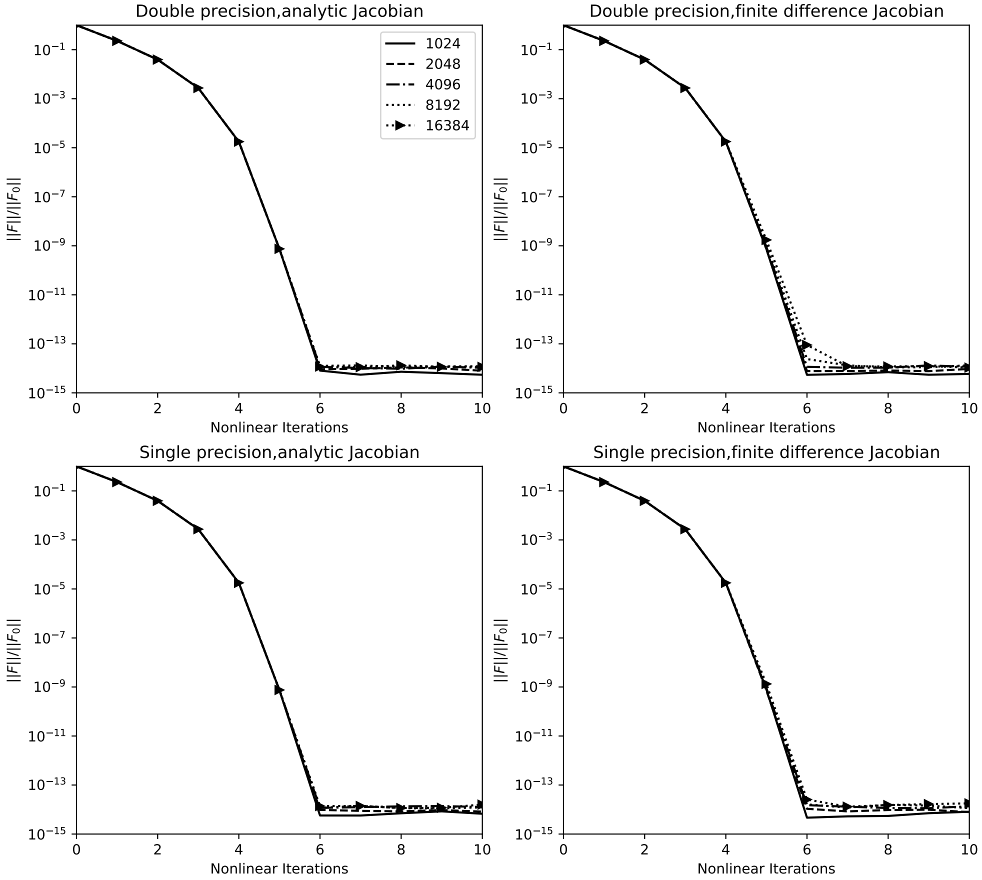
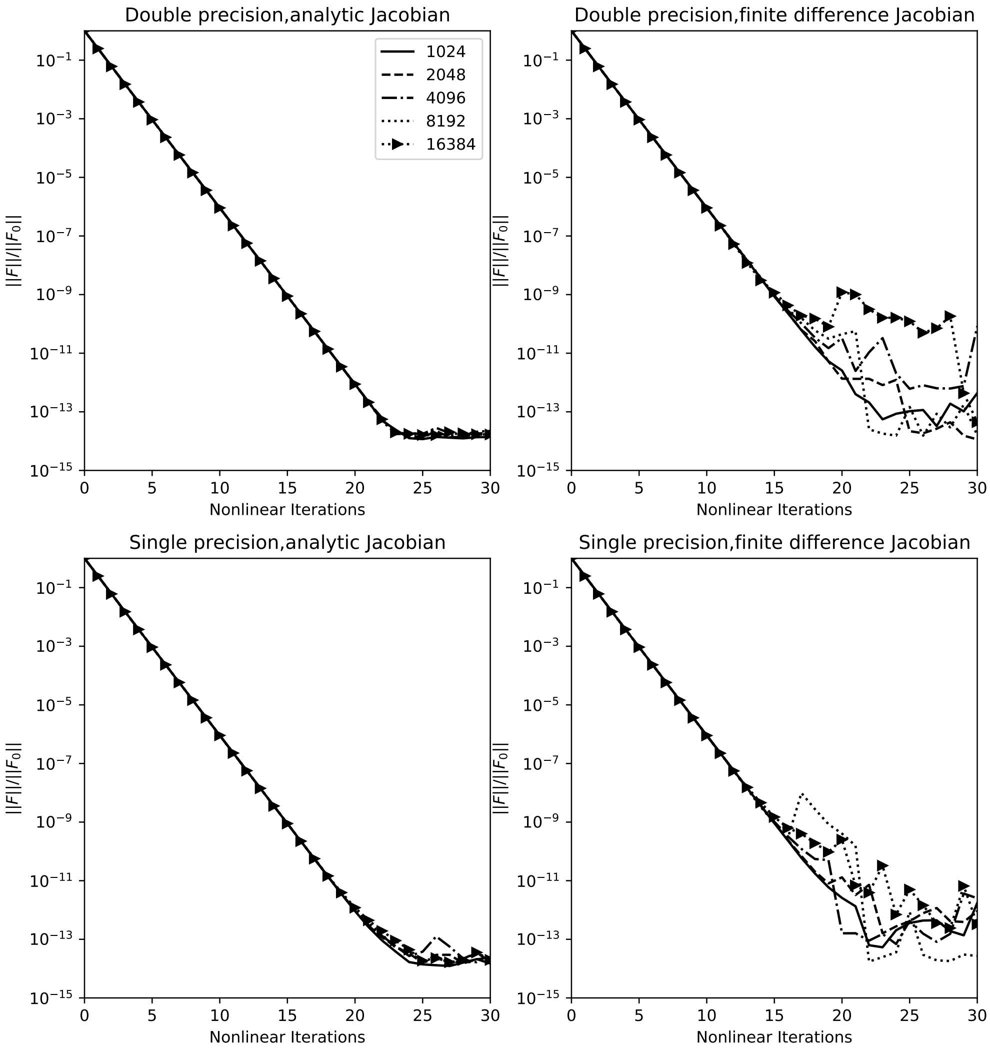


FIG. 3.3. *Residual Histories: Single and Double Precision,* $c = 1.0$

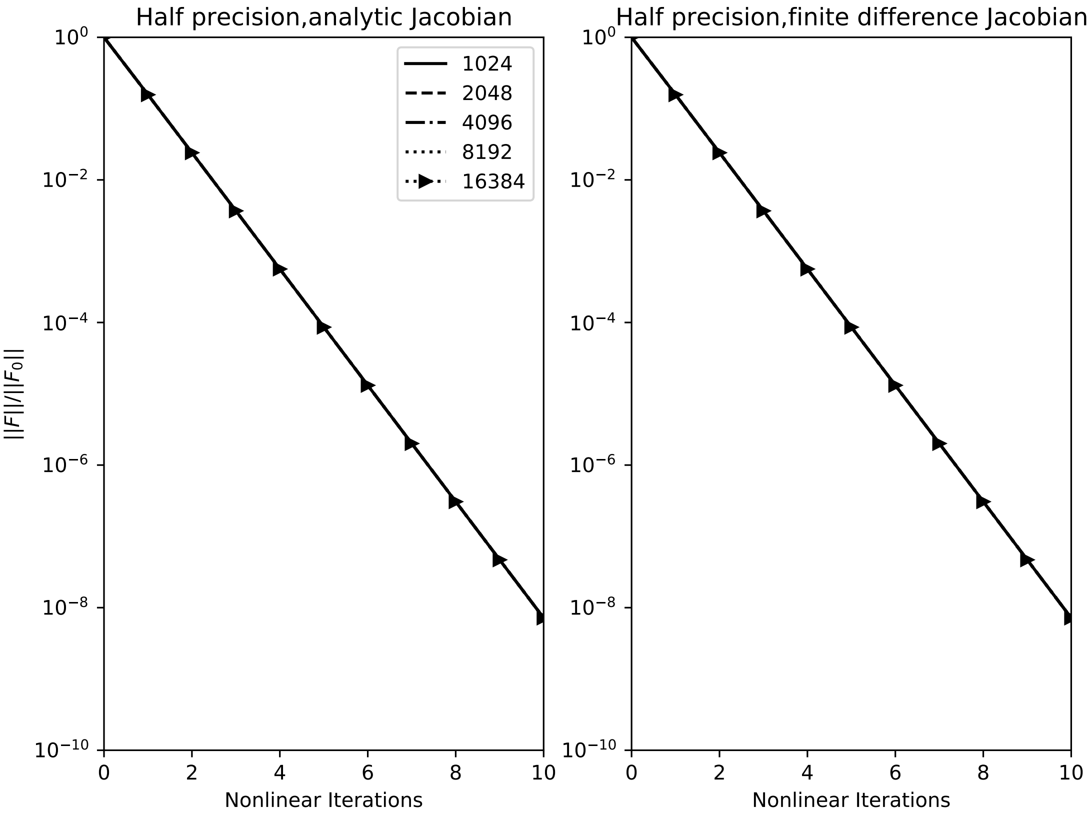


3.3. Half Precision: How low can you go? We begin with the case of nonsingular Jacobian: $c = .5$ and $c = .99$. As you can see from the figures, the convergence is not quadratic, but q-linear. This is because of the large Jacobian error. Also the results with a finite difference Jacobian were essentially the same, with no visible difference in the plots. The convergence rates agreed to three figures and we only present the rate estimates for the analytic Jacobian in the tables.

In the half precision computations we can see the difference in convergence speed between the $c = .5$ case and the case nearer to singularity $c = .99$. There was little difference between the analytic Jacobian and the forward difference approximation for these two cases. However, the nonlinear iteration statistics were very different and the change in convergence rate as a function of dimension for $c = .99$ is easy to see.

In Figures 3.4 and 3.5 we show the dependence of the nonlinear convergence rate on dimension when the matrix factorization is done in half precision. The remarkable thing about the plots is that the convergence rates do not seem to depend on dimension in the easy ($c = .5$) case and stop becoming slower as the dimension increases beyond $N = 4096$ for the nearly singular case ($c = .99$). Tables 3.1 and 3.2 show numerically that the convergence rates $\|\mathbf{F}(\mathbf{x}_{n+1})\|/\|\mathbf{F}(\mathbf{x}_n)\|$ are essentially independent of dimension for the $c = .5$ case and stabilize after $N = 4096$ for the $c = .99$ case. This explains the overlap in the plots after $N = 4096$. Both the plots and the tables indicate that the solver error is not increasing with dimension.

FIG. 3.4. *Residual Histories: Half Precision, $c = .5$*

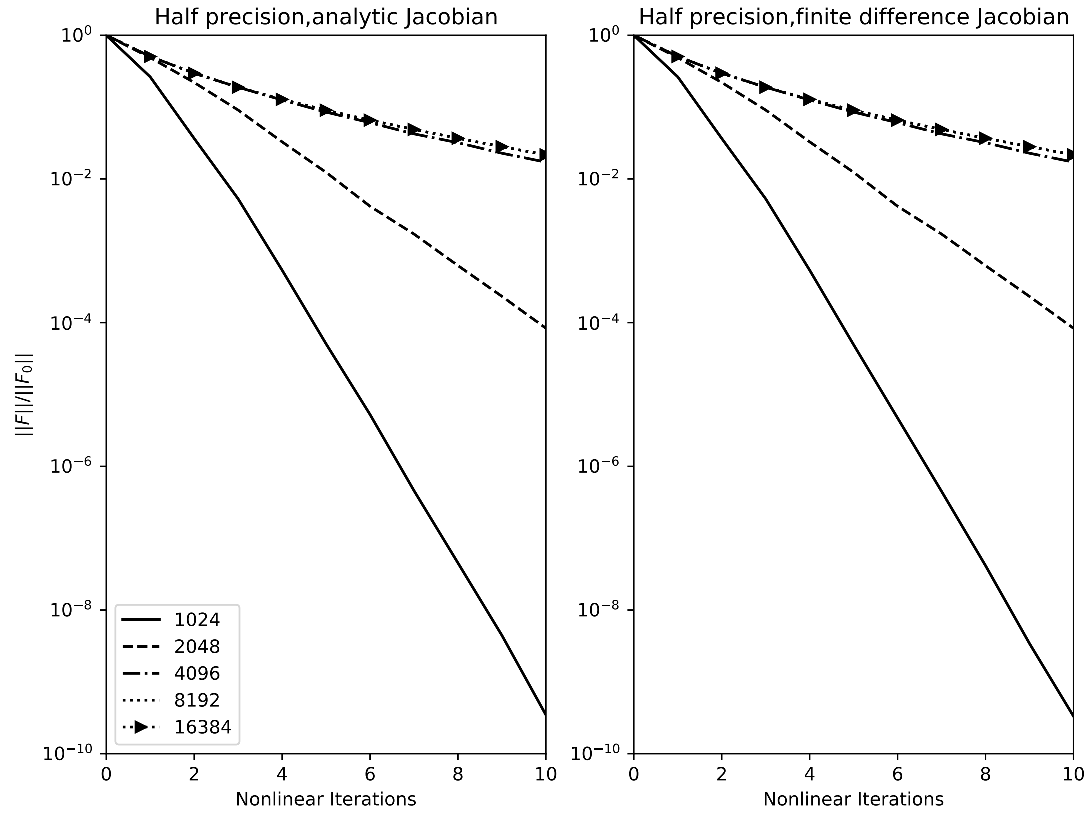


The singular case, $c = 1$, is particularly interesting in half precision for two reasons. The first is that the convergence rate seems, both from Figure 3.6 and Table 3.3 to be worse than q-linear. This is, in fact, what happens with singular problems of this type when the Jacobian approximation is poor [5]. The equation $f(x) = x^2 = 0$ is a good example. If $x_0 = 1$ and we approximate $f'(x)$ by $f'(x_0) = 2$, then it is easy to show

TABLE 3.1
Half Precision Computed Convergence Rates: $\|\mathbf{F}(\mathbf{x}_{n+1})\|/\|\mathbf{F}(\mathbf{x}_n)\|$, $c = .5$

n	1024	2048	4096	8192	16384
1	1.56706e-01	1.56708e-01	1.56708e-01	1.56705e-01	1.56706e-01
2	1.53569e-01	1.53573e-01	1.53579e-01	1.53578e-01	1.53576e-01
3	1.52949e-01	1.52944e-01	1.52946e-01	1.52948e-01	1.52949e-01
4	1.52853e-01	1.52848e-01	1.52844e-01	1.52847e-01	1.52843e-01
5	1.52831e-01	1.52829e-01	1.52832e-01	1.52830e-01	1.52830e-01
6	1.52828e-01	1.52825e-01	1.52826e-01	1.52830e-01	1.52827e-01
7	1.52830e-01	1.52824e-01	1.52827e-01	1.52826e-01	1.52825e-01
8	1.52832e-01	1.52832e-01	1.52824e-01	1.52825e-01	1.52828e-01
9	1.52838e-01	1.52830e-01	1.52831e-01	1.52830e-01	1.52826e-01
10	1.52828e-01	1.52828e-01	1.52827e-01	1.52829e-01	1.52827e-01

FIG. 3.5. *Residual Histories: Half Precision,* $c = .99$



that

$$\lim_{n\to\infty} \frac{x_n}{2/n} = 1.$$

This is very poor sublinear convergence.

Secondly, after 30 iterations the error is still too large for the effects of the forward difference approximate Jacobian to be seen. So both sides of Figure 3.6 are identical and the convergence statistics cease to depend on N after $N = 4096$.

Fig. 3.6. *Residual Histories: Half Precision, $c = 1$*

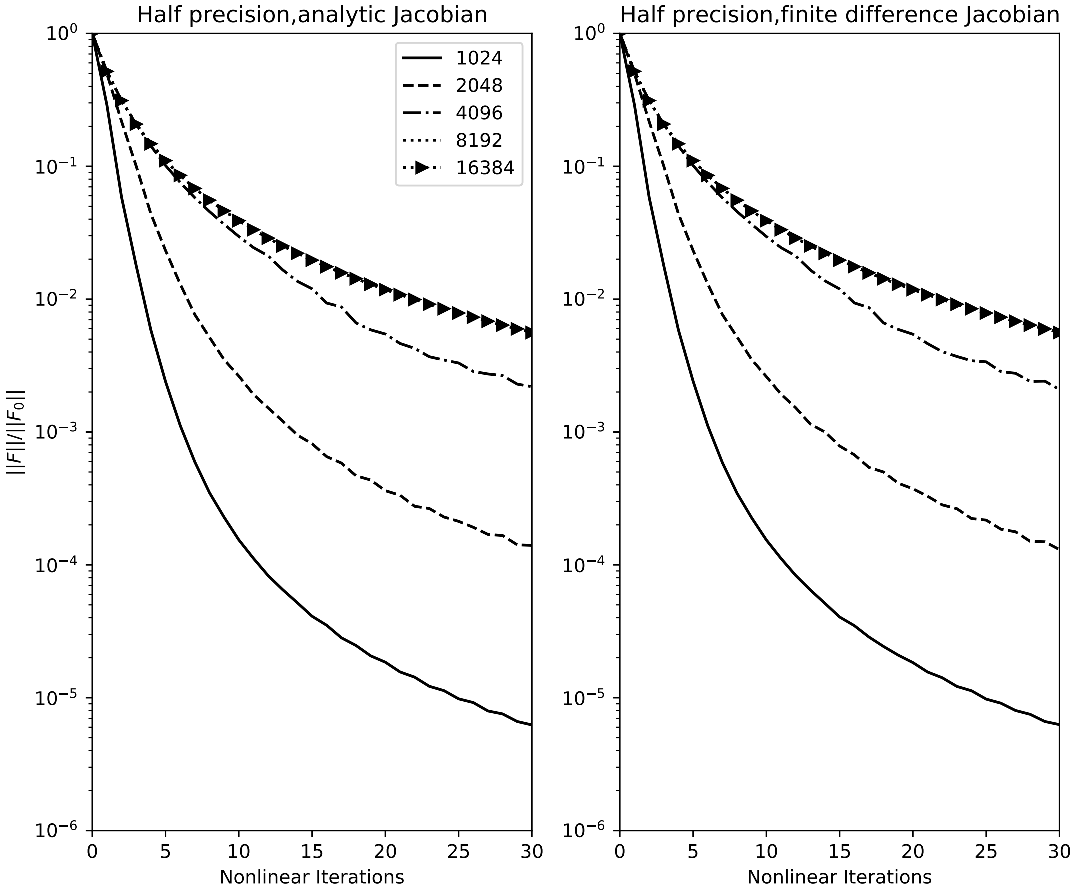


Table 3.2
Half Precision Computed Convergence Rates: $\|\mathbf{F}(\mathbf{x}_{n+1})\|/\|\mathbf{F}(\mathbf{x}_n)\|$, $c = .99$

n	1024	2048	4096	8192	16384
1	2.63294e-01	4.88603e-01	5.06480e-01	5.06480e-01	5.06480e-01
2	1.39099e-01	4.50348e-01	5.83959e-01	5.83962e-01	5.83961e-01
3	1.44278e-01	4.13186e-01	6.38900e-01	6.38900e-01	6.38898e-01
4	1.01754e-01	3.60960e-01	6.64343e-01	6.77977e-01	6.77976e-01
5	9.42216e-02	3.76718e-01	6.78547e-01	7.06190e-01	7.06192e-01
6	1.03855e-01	3.36916e-01	7.12325e-01	7.26959e-01	7.26957e-01
7	8.71357e-02	4.10487e-01	6.98753e-01	7.42506e-01	7.42507e-01
8	9.85171e-02	3.63814e-01	7.53512e-01	7.54294e-01	7.54296e-01
9	9.81771e-02	3.73218e-01	7.13812e-01	7.63337e-01	7.63337e-01
10	7.82248e-02	3.60341e-01	7.51307e-01	7.70318e-01	7.70319e-01

Table 3.3

Half Precision Computed Convergence Rates: $\|\mathbf{F}(\mathbf{x}_{n+1})\|/\|\mathbf{F}(\mathbf{x}_n)\|$, $c = 1$

n	1024	2048	4096	8192	16384
1	2.89271e-01	4.97487e-01	5.18347e-01	5.18347e-01	5.18347e-01
2	2.02306e-01	4.37133e-01	6.02756e-01	6.02754e-01	6.02755e-01
3	3.04726e-01	4.60824e-01	6.62228e-01	6.64944e-01	6.64946e-01
4	3.28286e-01	4.37880e-01	6.87719e-01	7.11345e-01	7.11344e-01
5	4.09534e-01	5.29573e-01	7.12647e-01	7.46800e-01	7.46800e-01
6	4.67802e-01	5.59574e-01	7.46514e-01	7.74644e-01	7.74642e-01
⋮	⋮	⋮	⋮	⋮	⋮
26	9.37645e-01	9.02116e-01	8.63996e-01	9.30262e-01	9.30264e-01
27	8.64444e-01	8.84138e-01	9.57200e-01	9.32602e-01	9.32599e-01
28	9.50195e-01	9.80767e-01	9.72447e-01	9.34786e-01	9.34786e-01
29	8.76453e-01	8.51197e-01	8.62460e-01	9.36834e-01	9.36837e-01
30	9.42673e-01	9.90469e-01	9.56378e-01	9.38760e-01	9.38762e-01

4. Conclusions. We showed how to indirectly observe the backward error in an LU factorization through the iteration statistics in Newton's method. For single precision, we confirm both the recent theory and folklore that storing and factoring the Jacobian in single precision has minimal effect on the performance of the nonlinear iteration. The backward error in the linear solver for the half precision case is large enough to degrade the nonlinear convergence to q-linear. Even so, we see that the results for the linear solver depend less on dimension than the theory predicts. Storing and factoring the Jacobian in half precision only seems useful for very well-conditioned problems.

5. Acknowledgments. The author is very grateful to Ilse Ipsen for her careful reading of a preliminary version of this paper.

REFERENCES

[1] E. L. Allgower, K. Böhmer, F. A. Potra, and W. C. Rheinboldt, *A mesh-independence principle for operator equations and their discretizations*, SIAM J. Numer. Anal., 23 (1986), pp. 160–169.

[2] I. W. Busbridge, *The Mathematics of Radiative Transfer*, no. 50 in Cambridge Tracts, Cambridge Univ. Press, Cambridge, 1960.

[3] S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.

[4] D. W. Decker and C. T. Kelley, *Newton's method at singular points I*, SIAM J. Numer. Anal., 17 (1980), pp. 66–70.

[5] ———, *Sublinear convergence of the chord method at singular points*, Numer. Math., 42 (1983), pp. 147–154.

[6] R.S. Dembo, S.C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[7] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[8] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[9] J. E. Dennis and H. F. Walker, *Inaccuracy in quasi-Newton methods: Local improvement theorems*, in Mathematical Programming Study 22: Mathematical programming at Oberwolfach II, North–Holland, Amsterdam, 1984, pp. 70–85.

[10] G. H. Golub and C. G. VanLoan, *Matrix Computations*, Johns Hopkins studies in the mathematical sciences, Johns Hopkins University Press, Baltimore, 3 ed., 1996.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[12] N. J. Higham, *A multiprecision world*, SIAM News, 50 (2017), p. 2.

[13] N. J. Higham and T. Mary, *A new approach to probabilistic rounding error analysis*, SIAM J. Sci. Comput., 1 (2019), pp. A2815–A2835.

[14] ———, *A new approach to probabilistic rounding error analysis*, February, 27 2019. Presentation at SIAM conference on Computational Science and Engineering.

[15] N. J. Higham, S. Pranesh, and M. Zounon, *Squeezing a matrix into half precision, with an application to solving linear systems*, SIAM J. Sci. Comput., 41 (2019), pp. A2536–A2551.

[16] IEEE Computer Society, *IEEE standard for floating-point arithmetic, IEEE Std 754–2019*, July 2019.

[17] I. C. F. Ipsen and H. Zhou, *Probabilistic error analysis for inner products*, Under revision for SIAM J. Matrix Anal. Appl., (2019). arXiv:1906.10465.

[18] H. B. Keller, *Lectures on Numerical Methods in Bifurcation Theory*, Tata Institute of Fundamental Research, Lectures

on Mathematics and Physics, Springer-Verlag, New York, 1987.

[19] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[20] ———, *Solving Nonlinear Equations with Newton's Method*, no. 1 in Fundamentals of Algorithms, SIAM, Philadelphia, 2003.

[21] ———, *Numerical methods for nonlinear equations*, Acta Numerica, 27 (2018), pp. 207–287.

[22] ———, *Notebook for Solving Nonlinear Equations with Iterative Methods: Solvers and Examples in Julia*. https://github.com/ctkelley/NotebookSIAMFANL, 2020. IJulia Notebook.

[23] ———, *SIAMFANLEquations.jl*. https://github.com/ctkelley/SIAMFANLEquations.jl, 2020. Julia Package.

[24] ———, *Solving Nonlinear Equations with Iterative Methods: Solvers and Examples in Julia*, 2020. Unpublished book ms, under contract with SIAM.

[25] C. T. Kelley and E. W. Sachs, *Mesh independence of Newton-like methods for infinite dimensional problems*, Journal of Integral Equations and Applications, 3 (1991), pp. 549–573.

[26] T. Kerkhoven and Y. Saad, *On acceleration methods for coupled nonlinear elliptic systems*, Numer. Math., 60 (1992), pp. 525–548.

[27] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton Krylov methods: A survey of approaches and appliciations*, J. Comp. Phys., 193 (2004), pp. 357–397.

[28] T. W. Mullikin, *Some probability distributions for neutron transport in a half space*, J. Appl. Prob., 5 (1968), pp. 357–374.

[29] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[30] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelphia, 2001.

[31] Jeffrey Willert, Xiaojun Chen, and C. T. Kelley, *Newton's method for Monte Carlo-based residuals*, SIAM J. Numer. Anal., 53 (2015), pp. 1738–1757.