Newton's Method in Mixed Precision

C. T. Kelley
NC State University
tim_kelley@ncsu.edu

Supported by ARO, DOE, NSF

Copper Mountain, April 5, 2022

Outline

1. Nonlinear Equations and Backward Error
   - Newton's Method
   - Inexact function and Jacobian
2. Linear Solver Woes
   - This Talk's Problem
   - The Backward Error Bites You
3. Examples: you figure it out.
4. Summary

Nonlinear Equations

Objective: solve

$$\mathrm{F}(\mathrm{x}) = 0$$

where

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

Newton's method is

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

Jacobian:

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

Local Convergence to distinguished root x^*

Standard assumptions for local convergence:
There is $x^* \in D$ such that

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

$$\|F'(x) - F'(y)\| \leq \gamma \|x - y\|,$$

for all $x, y \in D$.

Rules for talking about Newton's method

- x^* is the solution in SA
  which may not be the one you want
- $e = x - x^*$ is the error
- Convergence theorems in terms of change from
  - current iteration x_c to
  - next iteration x_+

Famous local convergence theorem

SA + small $\|e_0\|$ imply

$$\|e_+\| = O(\|e_c\|^2).$$

Local quadratic convergence.

What does this mean?

In an ideal world where

- precision is infinite,
- derivatives are analytic,
- linear solvers are exact,

Newton's method works great with good initial data.
But . . .

. . . you'll be doing it wrong.

In practice, you get

$$x_+ = x_c - J_c^{-1}(F(x_c) + E_c)$$

where

- $J_c \approx F'(x_c)$ (maybe badly)
- E_c is the (usually small) error in F

A less famous theorem

Same assumptions as for Newton plus $\|\mathrm{J}_c - \mathrm{F}'(\mathrm{x}_c)\|$ not too large imply

$$\|e_+\| = O(\|e_c\|^2 + \|\mathrm{J}_c - \mathrm{F}'(\mathrm{x}_c)\|\|e_c\| + \epsilon_F)$$

where $\|\mathrm{E}_c\| \leq \epsilon_F\|$.
Local improvement.
The theorem does not predict convergence, rather stagnation.

Examples

- $\epsilon_F = 0$, $\mathrm{J}_n = \mathrm{F}(\mathrm{x}_n)$: Newton
- $\epsilon_F > 0$, floating point error: Newton in practice
- $\epsilon_F > 0$, J_n finite difference Jacobian, step h
  - Use optimal $h = \sqrt{\epsilon_F}$ and
  - $\|\mathrm{e}_{n+1}\| = O(\|\mathrm{e}_n\|^2 + h\|\mathrm{e}_n\| + \epsilon_F)$
  - Same behavior as Newton until stagnation.
- $\epsilon_F > 0$, $\mathrm{J}_n = \mathrm{F}'(\mathrm{x}_0)$, chord method

Implementation: ignore ϵ_F

```
Intialize $x_0$, $n = 0$, termination criteria
while Not happy do
    Evaluate $F(x_n)$; terminate?
    Evaluate $J_n \approx F'(x_n)$
    Solve $J_n s = -F(x_n)$
    $x_{n+1} = x_n + s$
end while
```

Genius Idea!

- Store J in reduced precision.
- Solve in reduced precision.
    - Cut $O(N^2)$ storage by factor of 2 (single)
    - Cut $O(N^3)$ work by factor of 2 (single)
- How can you lose? Why isn't this in all the books?

The case in this talk

- ϵ_F floating point double precision roundoff
- $\mathrm{J}_c = \mathrm{J}_N + \Delta_{be}$ where
- Solver is double, single, or half precision LU
  - J_N is the nominal approximation you give the linear solver $\mathrm{F}'(\mathrm{x}_c)$ in double or finite-difference approximation
  - The solver returns the solution of $(\mathrm{J}_N + \Delta_{be})\mathrm{s} = -\mathrm{F}(\mathrm{x}_c) - \mathrm{E}_c$

So the less famous theorem says . . .

$$\|e_{n+1}\| = O\left(\|e_n\|^2 + (\|J_{Nn} - F'(x_n)\| + \|\Delta_{be}\|)\|e_n\| + \epsilon_F \right).$$

The Jacobian you think you have is harmless

- Analytic Jacobian: $\|J_{Nn} - F'(x_n)\| = O(\epsilon_F)$
- Difference Jacobian: $\|J_{Nn} - F'(x_n)\| = O(\epsilon_F^{1/2})$
- But what about the backward error?
- Large backward error → slow nonlinear convergence.
  Can we see this numerically?

What is that backward error for Ax = b?

Begin with componentwise backward error

- NICHOLAS J. HIGHAM, Accuracy and Stability of Numerical Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

Page 175-177: Componentwise backward error (ignore permutation matrix)

$$|\delta \mathrm{A}| \le 2\gamma_N |\hat{L}||\hat{U}|$$

where $\hat{\mathrm{L}}\hat{\mathrm{U}} = \mathrm{A} + \delta \mathrm{A}$ and

$$\gamma_N = \frac{N\epsilon_S}{1 - N\epsilon_S}$$

Short story, each element in LU is $O(N)$ work.

Worse case bounds

The growth factor (Demmel) is

$$\hat{g}_{PP} = \frac{\max|\hat{\mathrm{U}}_{ij}|}{\max|\mathrm{A}_{ij}|}$$

The estimate you see in books is

$$\|\delta \mathrm{A}\| \le 2\hat{g}_{PP} N^2 \gamma_N = \frac{2\hat{g}_{PP} N^3 \epsilon_S}{1 - N\epsilon_S} \|\mathrm{A}\|$$

Then the text books say this is silly and rarely observed.

Optimistic Estimates: $\|A\| = O(1)$

The optimistic estimate is

$$\|\delta A\| = O(N\epsilon_S)?$$

Higham-Mary 2019 say even more . . .
If rounding errors are independent, then with with high probability for most desktop-sized problems and Newton's method

$$\|\Delta_{be}\|_\infty = O(\epsilon_S\sqrt{N})$$

What should we observe if $\sqrt{N}$ is the right thing?

- Trouble (slow nonlinear convergence) when $\sqrt{N}\epsilon_S \ge .1$
  - Double: $N \approx 10^{30}$. Not on my computer.
  - Single: $N \approx 10^{14}$. Not on my computer.
  - Half: $N \approx 10^{6}$. Maybe if we push it.
- Expectation: Single just as good as double.
- Expect to see deterioration with N for half.

Examples and Solvers from . . .

- C. T. Kelley, Newton's method in mixed precision, SIAM Review, 64 (2022), pp. 191–211.
- C. T. Kelley, Notebook for Solving Nonlinear Equations with Iterative Methods: Solvers and Examples in Julia, SIAM, 2022(3).

Solving Nonlinear Equations with Iterative Methods: Solvers and Examples in Julia

SIAM: Publication sometime in 2022/3

Three parts

- Print book: sequel to FA1:
  C. T. KELLEY, Solving Nonlinear Equations with Newton's Method, number 1 in Fundamentals of Algorithms, SIAM, Philadelphia, 2003.
- IJulia (aka Jupyter) notebook at
  `https://github.com/ctkelley/NotebookSIAMFANL`
- Julia package with solvers+test problems+examples
  `https://github.com/ctkelley/SIAMFANLEquations.jl`

Chandrasekhar H-equation

Midpoint rule discretization

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

- Defined on $C[0,1]$
- $\mathcal{F}'$ nonsingular for $0 \le c < 1$.
  Simple fold singularity at $c = 1$.
- Any sensible discretization inherits the singularity structure.

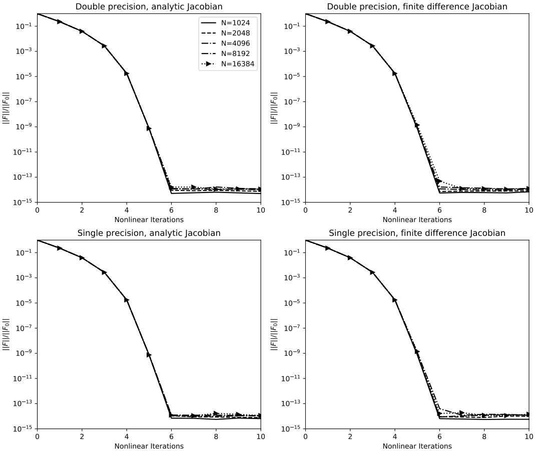
Discrete Problem

- composite midpoint rule
- function evaluation is $O(N \log N)$ for Hankel matrix-vector product
- Jacobian is $O(N^2)$ analytic
- Factorization is $O(N^3)$ in the solver precision

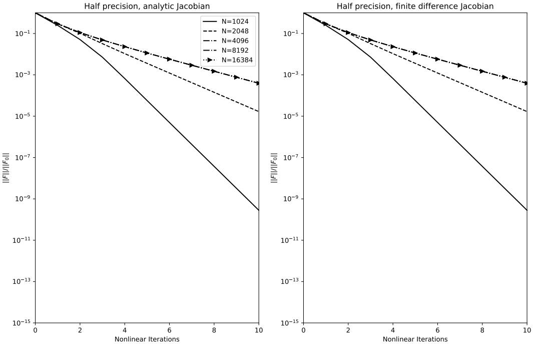
Experiments: from K. 22

- $c = .99$
- Analytic and forward difference Jacobians
  Theory predicts single as good as double
- Double, single, and half precision factor/solve
- Everything else in double
- $N = 2^p$, $p = 10, \dots, 14$, $2^{14} = 16384$
  Larger N took far too long in half.

$c = .99$, double and single



$c = .99$, half, Wait! What?



Timings: $c = .99$, $N = 4096$

Solve phase only: memory for Jacobian preallocated.

- Analytic, double, Newton
  909 ms
- Analytic, single, Newton
  436 ms

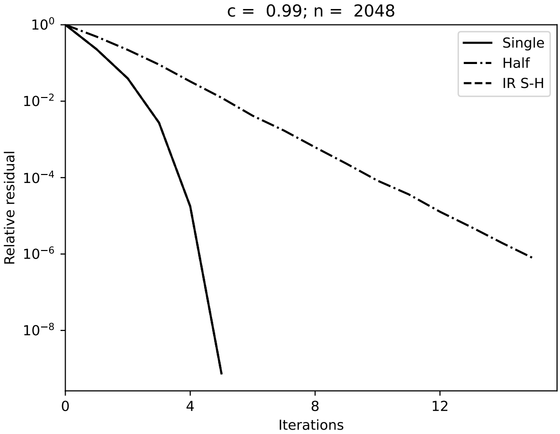
Speedup = 2x, storage is allocated in initialization phase.

Iterative Refinement: for $Ax = b$

Can half precision be saved?

- Store A in high and low precision
- $x_+ = x_c + I_{low}^{high} A_{low}^{-1} I_{high}^{low} (b - A_{high} x_c)$
- Details:
    - Factor in low precision. Free lunch?
    - Scaling and promotion/demotion
      Inter-precision transfers are not free.
    - Punishment for ill-conditioning
    - Poor infrastructure for testing
- Experiment for high=single, low=half . . .

Iterative Refinement: Nonlinear Iteration is happy



Details

- IR iteration count increases with
  - Nonlinear iteration, but not a lot
    No surprise, Jacobian at solution is poorly conditioned
  - Problem size (at least what I've measured)

Newton-Krylov methods

Terminate the linear iteration when

$$\|\mathrm{F}'(\mathrm{x}_n)\mathrm{d} + F(\mathrm{x}_n)\| \leq \eta_n \|\mathrm{F}(\mathrm{x}_n)\|.$$

and then the estimate is

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

Backward error issues: loss of orthogonality in GMRES

Timings: Newton-GMRES, $c = .5$, $N = 4096$

Solve phase only: memory for basis preallocated

- analytic, single, Newton, Gaussian Elimination: 258.299 ms
- Newton-GMRES, fixed $\eta = .1$, Double: 1.776 ms
- Newton-GMRES, fixed $\eta = .1$, Single: 1.805 ms

No difference: 1 Krylov/Newton, orthogonalization/storage negligible.

Summary

- Single precision linear solvers are just fine
    - Single precision → same nonlinear results
    - Half precision → not recommended
        - But there is hope . . .
        - Iterative Refinement needs more support
          LAPACK for half precision not here
- Gaussian elimination for dense problems gives expected results
  Costof (single) $\approx 1/2\times$ Costof (double)
- Newton-Krylov not so clear.