Nonlinear Equations

C. T. Kelley
NC State University
tim_kelley@ncsu.edu
Research Supported by NSF, DOE, ARO, USACE

DTU ITMAN, 2011

Outline

References I

- P. N. Brown and A. C. Hindmarsh, Reduced storage matrix methods in stiff ODE systems, J. Appl. Math. Comp., 31 (1989), pp. 40–91.
- 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.
- S. C. Eisenstat and H. F. Walker, Globally convergent inexact Newton methods, SIAM J. Optim., 4 (1994), pp. 393–422.

References II

- C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.
- J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

What's in Friday's directory?

- This lecture.
- Corrections to the project lecture (typos)
- Matlab files, including knl.m + revised kl.m

Notation

Objective: find a solution of

$$F(x) = 0$$

where $F : R^N \rightarrow R^N$.
We write $F = (f_1, \dots, f_N)^T$. The Jacobian matrix F' is

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

Newton's Method

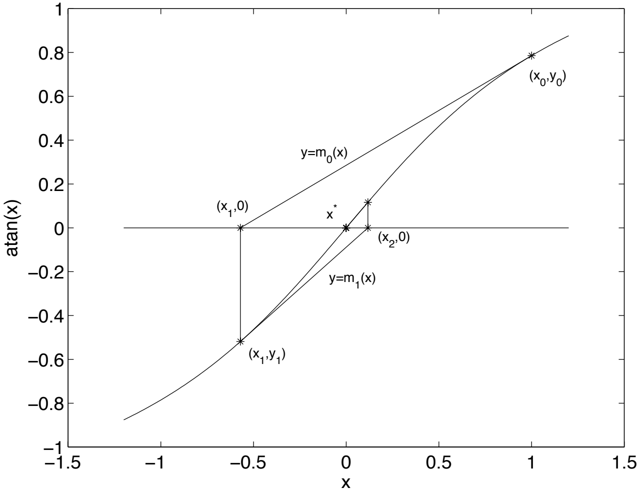
Transition from current point x_c to new one x_+.

$$x_+ = x_c - F'(x_c)^{-1} F(x_c).$$

Interpretation: x_+ is the root of the local linear model at x_c

$$M_j(x) = F(x_j) + F'(x_j)(x - x_j)$$

$f(x) = \arctan(x)$



Implementation of a Newton Iteration

Evaluate $F(x_c)$; terminate?
Solve $F'(x_c)s = -F(x_c)$
$x_+ = x_c + s$

Formulations of Newton differ in the way they solve for s.

Convergence Theory for Exact Linear Solves

Standard Assumptions (SA):

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

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

Convergence Theory: I

Theorem: SA implies that if

$$\|e_c\| \equiv \|x_c - x^*\| \le \frac{\|F'(x^*)^{-1}\|}{2\gamma}$$

then $F'(x_c)$ is nonsingular and $\|F'(x_c)^{-1}\| \le 2\|F'(x^*)^{-1}\|$
Proof: Lipschitz continuity implies that

$$\|F'(x_c) - F'(x^*)\| \le \gamma\|e_c\| \le \frac{\|F'(x^*)^{-1}\|}{2}$$

and so ...

Convergence Theory: II

$$\|I - F'(x^*)^{-1} F'(x_c)\| \le \|F'(x^*)^{-1}\| \|F'(x_c) - F'(x^*)\| \le 1/2$$

so $F'(x^*)^{-1}$ is an approximate inverse of $F'(x_c)$, and

$$\|F'(x_c)^{-1}\| \le \frac{\|F'(x^*)^{-1}\|}{2}.$$

Local Convergence Theory: III

Recall the fundamental theorem of calculus:

$$F(x) - F(x^*) = \int_0^1 F'(x^* + t(x - x^*))(x - x^*)\, dt.$$

$F'(x^*) = 0$, so let $x = x_c$ and ...

$$\begin{aligned} F(x_c) &= \int_0^1 F'(x_c + te_c)e_c\, dt \\ &= F'(x_c)e_c + \int_0^1 (F'(x_c + te_c) - F'(x_c))e_c\, dt \end{aligned}$$

Local Convergence Theory: III

We are done since

$$
\begin{aligned}
e_+ &= e_c - F'(x_c)^{-1}(F'(x_c)e_c + \int_0^1 (F'(x_c + te_c) - F'(x_c))e_c \, dt) \\
&= -F'(x_c)^{-1}(\int_0^1 (F'(x_c + te_c) - F'(x_c))e_c \, dt)
\end{aligned}
$$

So,

$$\|e_+\| \le \frac{\|F'(x^*)^{-1}\|\gamma}{2}\|e_c\|^2 \le \|e_c\|/2.$$

Bottom Line

SA and good data ($\|e_0\|$ small) imply that

- $x_n \rightarrow x^*$
- Convergence is q-quadratic

Things change if initial iterate is not close to x^* or you use an iterative method to compute s.

Newton-Iterative Methods

Replace exact (or direct) solution of

$$F'(x_c)s = -F(x_c)$$

with an iterative method.
Terminate the linear (inner) iteration when the inexact Newton condition

$$\|F'(x_c)s + F(x_c)\| \le \eta_c \|F(x_c)\|$$

holds.
η is called the forcing term.

Options

Examples: Newton-GMRES, Newton-MG, Newton-Krylov-Schwarz
Jacobian-vector product:

$$F'(x)v \approx \frac{F(x+hv) - F(x)}{h}$$

where h is scaled to capture the low-order bits.

$$h = \|x\| \sqrt{\epsilon_{mach}} / \|v\|.$$

Convergence Theory

Theorem: Assume that $\eta_n \le \bar{\eta} < 1$, SA, and good data. Then the Newton iteration converges to x^* and

$$\|e_{n+1}\| = O(\|e_n\|^2 + \eta_n \|e_n\|)$$

Proof: The idea is to compare the inexact Newton iteration to a Newton iteration. The inexact iteration is

$$u_+ = u_c + s = u_c - F'(x_c)^{-1}F(x_c) + (F'(x_c)^{-1}F(x_c) + s)$$

Convergence Theory: II

The difference between the Newton iteration u_+^N and the inexact iteration can be bounded

$$
\begin{aligned}
\|(F'(x_c)^{-1}F(x_c) + s)\| &= \|F'(x_c)^{-1}\|\|F'(x_c)s + F(x_c)\| \\
&\le \|F'(x_c)^{-1}\|\eta_c\|F(x_c)\| \\
&\le 2\bar{\eta}\|F'(x^*)^{-1}\|\|F(x_c)\|
\end{aligned}
$$

if x_c is sufficiently near x^*.

Convergence Theory: III

The final step is to estimate $\|F(x_c)\|$. Calculus says

$$\|F(x_c)\| = \| \int_0^1 F'(x_c + te_c)e_c \, dt\| \leq \int_0^1 \|F'(x^* + te_c)\| \, dt \|e_c\|$$

Since, by SA,

$$\|F'(x^* + te_c) - F'(x^*)\| \leq \gamma t \|e_c\|$$

we see that

$$\|F'(x^* + te_c)\| \leq \|F'(x^*)\| + \gamma t \|e_c\|$$

Convergence Theory: IV

Bottom line:

$$\|F(x_c)\| \le (\|F'(x^*)\| + \gamma t \|e_c\|)\|e_c\|$$

and so

$$\begin{aligned} \|(F'(x_c)^{-1}F(x_c) + s)\| &\le 2\bar{\eta}\|F'(x^*)^{-1}\|(\|F'(x^*)\| + \gamma t\|e_c\|)\|e_c\| \\ &= O(\eta_c\|e_c\| + \|e_c\|^2). \end{aligned}$$

Convergence Theory: V

So

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

which proves the result.

Remarks

- If $\eta_n = O(\|F(x_n)\|)$ the convergence is quadratic.
- If $\eta_n \to 0$ the convergence is q-superlinear,

$$\|e_{n+1}\|/\|e_n\| \to 0.$$

- It is usually a poor idea to over solve in the inner iteration.

Poor Data

Suppose you try to solve 'arctan$(x) = 0$ with Newton and $x_0 = 10$. The iterations are

$$10, -138, 2.9 \times 10^4, -1.5 \times 10^9, 9.9 \times 10^{17}.$$

What happened?

Line searches and the Armijo rule

Now we make a distinction between the Newton direction

$$d = -F'(x_c)^{-1}F(x_c)$$

and the Newton step

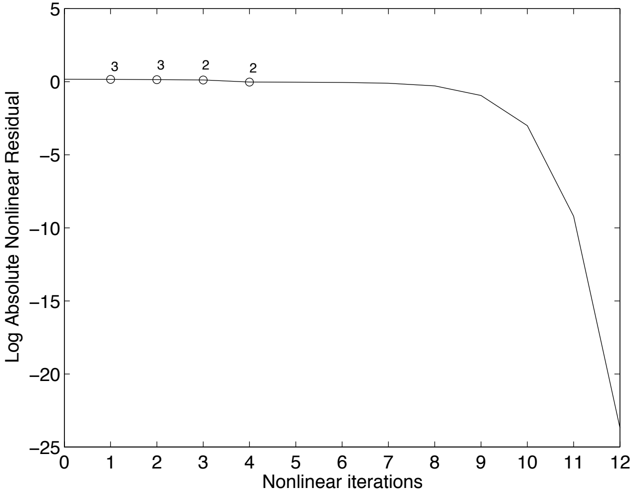
$$x = x_+ - x_c$$

Simply put, we find the least $\lambda = 2^{-m}$ for $m = 0, 1, \ldots$ so that

$$\|F(x_c + \lambda d)\| < \|F(x_c)\|$$

and use $s = \lambda d$. This process, a line search almost works.

Saving the ArcTan iteration



Sufficient Decrease and Termination

You need a little more to prove something. The sufficient decrease condition is

$$\|F(x_c + 2^{-m} d)\| < (1 - \alpha 2^{-m})\|F(x_c)\|,$$

Typical $\alpha = 10^{-4}$.
We will terminate when the nonlinear residual $\|F\|$ is small, i. e. when

$$\|F(x_n)\| \leq \tau_a + \tau_r \|F(x_0)\|.$$

Newton-Iterative Algorithm

nsolg(x, F, τ_a, τ_r)

evaluate $F(x)$; $\tau \leftarrow \tau_r |F(x)| + \tau_a$.
while $\|F(x)\| > \tau$ **do**
Find d such that $\|F'(x)d + F(x)\| \leq \eta \|F(x)\|$
If no such d can be found, terminate with failure.
$\lambda = 1$
while $\|F(x + \lambda d)\| > (1 - \alpha\lambda)\|F(x)\|$ **do**
$\lambda \leftarrow \sigma\lambda$ where $\sigma \in [1/10, 1/2]$ is computed by minimizing a polynomial model of $\|F(x_n + \lambda d)\|^2$.
end while
$x \leftarrow x + \lambda d$
end while

Polynomial Models for Line Search

Most codes do not use a constant reduction factor for d. Theyh approximate

$$\phi(\lambda) = \|F(x_c + \lambda d)\|^2$$

- After one failed iteration, you have enough data to model (x_c and $xc + d$) with a linear model.
- After two or more iterations you have x_c, $x_c + 2\lambda d$, and $x_c + \lambda d$ so can build a parabolic model.
- Minimize the model for $\lambda \in [.1, .5]$ to get the new λ.

Theory

Theorem: Suppose F is Lipschitz continuously differentiable, $\{x_n\}$ is the inexact Newton-Armijo sequence, $0 < \eta_n < \bar{\eta} < 1$. The there are only three possibilities:

- $\{x_n\}$ converges to a root x^* of F at which the standard assumptions hold, full steps ($\lambda = 1$) are taken for n sufficiently large, and the local convergence theory holds.
- The sequence $\{x_n\}$ is unbounded.
- The sequence $\{F'(x_n)^{-1}\}$ is unbounded.

A few examples

- $f(x) = e^x$; the Newton-Armijo sequence takes full steps and

$$x_+ = x_c - \frac{e^{x_c}}{e^{x_c}} = x_c - 1$$

  So $x_n \to -\inf$.
- $f(x) = x^2 + 1$; You dont get full steps and $x_n \to 0$. What happened?

Bottom line: you can't solve a problem with no solution.

Chosing the forcing term η

- Constants are often ok: $\eta = .1, .01$ (but not 10^{-8})
- Eisenstat-Walker version 1

$$\eta_n^{Res} = \gamma \|F(x_n)\|^2 / \|F(x_{n-1})\|^2$$

  - Tries to get faster nonlinear convergence at the end.
  - $\gamma \in (0,1]$ is a parameter you have to make up. $\gamma = .9$ is generally ok.
  - Trades work in the linear solver for fewer nonlinear iterations.
  - Can result is very volatile changes in η

Eisenstat-Walker v2

$$\eta_n^{Safe} = \min(\eta_{max}, \max(\eta_n^{Res}, .5\tau_t/\|F(x_n)\|)).$$

Keeps the forcing term from getting to large and limits decreases.
Typical choices: $\eta_{max} = .9$.

Choosing a Solver

The most important issues in selecting a solver are

- the size of the problem,
- the cost of evaluating F and F', and
- the way linear systems of equations will be solved.

The items in the list above are not independent.

Rough guidelines

- Small N and cheap F; try direct methods and a forward difference Jacobian (but see the example at the end of this lecture for a different view).
- Large N or expensive F'; try matrix-free Newton-Krylov solvers.
- Large N, very sparse F, only bad preconditioners; try sparse differencing for the Jacobian and use a direct method.

knl

```
function [sol, it_hist, ierr, x_hist] = ...
     knl(x,f, nloptions,static_data)
```

Input:

- initial iterate = x
- function = f
  Calling sequence is either `fout = f(x)` or
  `fout=f(x, static_data)`
  Precomputed data is the optional argument.
- nloptions = options structure. See `knl_optset` for documentation.
- optional input: `static_data` = precomputed data for function evaluation, jacobian/preconditioner-vector product

Output

- `sol` = solution
- `it_hist(maxit,3)` = l2 norms of nonlinear residuals for the iteration, number of function evaluations, and number of steplength reductions
- `ierr` = 0 upon successful termination
  `ierr` = 1 failure after maxit iterations
  `ierr` = 2 failure in the line search.
- optional output: `x_hist` = matrix of the entire iteration history. The columns are the nonlinear iterates.

What it can do

- Choice of PCG, BiCGStab, TFQMR, GMRES Krylov solvers directly from KL.
- Use KL options for the linear solver.
- Based on 15 year old solvers from red book, but better organized and easier to learn and use.
- Can pass data to function, preconditioner, Jacobian-vector product with optional last arguemtn `static_data`.
- Ability to use an analytic Jacobian-vector product if you have one.
- Exciting opportunity to test new software and find bugs.

Setting Options

Use the `knl_optset.m` function to set

- Relative `rtol` and absolute `atol` termination tolerances.
- Limits on linear and nonlinear iterations.
- Forcing term limit `etamax` or allow for fancy forcing term control.
- Flags to inform KNL if you can give it information for matvecs or preconditioners.
- Flag for preconditioners that depend on the nonlinear iteration.
- Options for KL (except for forcing term)

Defaults

- `atol`: 10^{-12}; `rtol`: 10^{-6}
- Linear solver: GMRES; mgs + orth test
- Maximum iterations: linear (40), nonlinear (40)
- Forcing term: Eisenstadt-Walker v2 + limit of .9 (but see comments in code)

Chandrasekhar H-Equation

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

- $0 \leq \mu \leq 1$; $0 \leq c \leq 1$
- Unknown $H \in C[0,1]$
- Two solutions for $0 < c < 1$. Unique for $c = 0, 1$.

Discretization

We will approximate the integrals by the composite midpoint rule.

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

where $\mu_i = (i - 1/2)/N$ for $1 \leq i \leq N$.
Discrete equation in R^N:

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

Compact Form

Let

$$A_{ij} = \frac{c\mu_i}{2N(\mu_i + \mu_j)}.$$

Once A is stored, $F(x)$ can be rapidly evaluated as

$$F(x)_i = (x)_i - (1 - (Ax)_i)^{-1}.$$

The Jacobian-vector product is given by

$$(F'(x)v)_i = v_i - \frac{(Av)_i}{(1 - (Ax)_i)^2}.$$

So, after you compute $F(x)$, $F'(x)v$ takes very little work.

Matlab Demonstration

Exercises

Use KNL to solve

$$-u'' + \cos(u)u' = f(x), \; 0 < x < 1; u(0) = u(1) = 0$$

with $f(x)$ build so that the solution is $e^x \sin(\pi x)$

- Discretize with the standard centered difference formulae.
- Compare the performance of three preconditioners which only use the high-order term.
  - An exact solver for $-u'' = g$ (ie tridiagonal solve)
  - An single V-cycle for $-u'' = g$
  - Two-level additive Schwarz for $-u'' = g$