Scientific Computing with Case Studies

SIAM Press, 2009

http://www.cs.umd.edu/users/oleary/SCCSwebpage

Lecture Notes for Unit VI

Nonlinear Equations and Continuation Methods

Dianne P. O'Leary

The problem

Given a function $\mathbf{F} : \mathcal{R}^n \rightarrow \mathcal{R}^n$, find a point $\mathbf{x} \in \mathcal{R}^n$ such that

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

Note: The one-dimensional case ($n = 1$) is covered in CMSC/AMSC 460. The best software for this problem is some variant on Richard Brent's `zeroin`, available in Netlib. In Matlab, it is called `fzero`.

We'll assume from now on that $n > 1$.

Goals

To develop algorithms for solving nonlinear systems of equations.

Note: Solving nonlinear equations is a close kin to solving optimization problems.

- Easier than optimization, since a "local solution" is just fine.
- Harder than optimization, since there is no natural merit function $f(\mathbf{x})$ to measure our progress.

Important note: If **F** is a polynomial in the variables **x**, then use special purpose software that enables you to find all of the solutions reliably.

Example of a polynomial system:

$$\begin{aligned} x^2y^3 + xy &= 2 \\ 2xy^2 + x^2y + xy &= 0 \end{aligned}$$

Pointers: Watson's homotopy method; Traub's software.

What we know

We already know a lot about solving nonlinear equations. The main tools are Newton's method and Newton-like methods.

Differences from the methods we have studied:

- Instead of the Hessian matrix, we have the Jacobian matrix of first derivatives:

$$\mathbf{J}_{ik} = \frac{\partial F_i}{\partial x_k} .$$

  This matrix is generally not symmetric.

- Line searches are more difficult to guide, since we can't measure progress using the function $f(\mathbf{x})$. Some attempts have been made to use

$$\|\mathbf{F}(\mathbf{x})\|$$

  as a merit function, but there are difficulties with this approach.

The Plan

- Newton-like methods (for easy problems): Section 24.3
- Globally-convergent continuation methods (for hard problems): Section 24.4
- Case study: Chapter 25 compares these methods on the problem of determining orientations of a truss.
- Case study: Chapter 26 uses these methods to determine parameters in an ecological model of a colony of flour beetles.

Note: The nonlinear least squares problem, discussed in Section 24.2 and Chapter 13, is one approach to solving nonlinear equations for which

- there are more equations than unknowns.
- measurement error contaminates the equations.

Newton-like methods

- applied to nonlinear least squares
- applied to nonlinear equations

Nonlinear least squares

Note that we can solve $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ by solving

$$\min_{\mathbf{x}} \|\mathbf{F}(\mathbf{x})\|_2^2$$

using any of our methods from Unit 3, looking for a point that gives a function value of zero.

Advantages:

- Uses all of our old machinery.
- Generalizes to overdetermined systems in which the number of equations is greater than the number of variables.

Disadvantage: Derivatives are rather expensive: if $f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$, then

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= 2\mathbf{J}(\mathbf{x})^T\mathbf{F}(\mathbf{x}) \\ \mathbf{H}(\mathbf{x}) &= 2\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) + \mathbf{Z}(\mathbf{x}) \end{aligned}$$

where $\mathbf{Z}(\mathbf{x})$ involves 2nd derivatives of $\mathbf{F}$.

Newton-like methods for nonlinear equations

Recall our general scheme for function minimization:

Until $\mathbf{x}^{(k)}$ is a good enough solution,

> Find a downhill search direction $\mathbf{p}^{(k)}$.
> Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$, where α_k is a scalar chosen to guarantee that progress is made.

The Newton search direction was defined by $\mathbf{H}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{g}(\mathbf{x}^{(k)})$.

Complications:

- We derived Newton by fitting a quadratic function to a function that we were trying to minimize. Equivalently, we fit a linear function to the gradient $\mathbf{g}$:

  $$\mathbf{g}(\mathbf{x}^{(k)} + \mathbf{p}) \approx \mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{H}(\mathbf{x}^{(k)})\mathbf{p} = \mathbf{0} \;\rightarrow\; \mathbf{H}(\mathbf{x}^{(k)})\mathbf{p} = -\mathbf{g}(\mathbf{x}^{(k)})\,.$$

  Now we can fit a linear function to $\mathbf{F}$:

  $$\mathbf{F}(\mathbf{x}^{(k)} + \mathbf{p}) \approx \mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{J}(\mathbf{x}^{(k)})\mathbf{p} = \mathbf{0} \;\rightarrow\; \mathbf{J}(\mathbf{x}^{(k)})\mathbf{p} = -\mathbf{F}(\mathbf{x}^{(k)})\,.$$

- There is no notion of "downhill" because there is no function f.
- There is no notion of "progress" because there is no function f.

But if we still find the Newton direction from solving $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$, then we are still fitting a linear function to $\mathbf{F}$.

As for the line search, we will just eliminate it and set $\alpha_k = 1$, or we will use $\|\mathbf{F}(\mathbf{x})\|$ as a merit function.

The resulting algorithm is Newton's method for nonlinear equations:

Until $\mathbf{x}^{(k)}$ is a good enough solution,

> Find a direction $\mathbf{p}^{(k)}$ by solving $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$.
> Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ where $\alpha_k = 1$ or α_k is determined by a line search.

Convergence result: Under some "standard assumptions" (2nd derivatives exist, simple zero, etc.), the algorithm is guaranteed to converge to a solution if started close enough to it, and the convergence rate is quadratic.

Newton-like methods

- Finite Difference Newton method: If $\mathbf{J}$ is not available, we can approximate it using finite differences. Again, this is not recommended; use the next method instead.

- Inexact Newton method: Instead of solving the linear system $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$ exactly, we can use an iterative method to obtain an approximate solution. The usual choice of iterative method is GMRES, a relative of conjugate gradients, and matrix-vector products are evaluated by differencing $\mathbf{F}$.

- Quasi-Newton methods: If storage is not a problem, then instead of the inexact Newton method, we can store and update an approximation to $\mathbf{J}$. The usual formula is Broyden's method:

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \frac{(\mathbf{y} - \mathbf{B}^{(k)}\mathbf{s})\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}}$$

where $\mathbf{y} = \mathbf{F}(\mathbf{x}^{(k+1)}) - \mathbf{F}(\mathbf{x}^{(k)})$ and $\mathbf{s} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$. Then

$$\mathbf{B}^{(k+1)}\mathbf{s} = \mathbf{y} \quad \text{Secant condition}$$

and if $\mathbf{s}^T\mathbf{v} = 0$, then

$$\mathbf{B}^{(k+1)}\mathbf{v} = \mathbf{B}^{(k)}\mathbf{v}\,.$$

Convergence result: As in the minimization case, we get superlinear convergence if the direction is (asymptotically) close enough to the Newton direction.

Globally-convergent continuation methods

Globally-convergent continuation methods

The basic idea:

- Want to solve $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.
- Know the solution to some easy problem $\mathbf{F_a}(\mathbf{x}) = \mathbf{0}$.
  Example: $\mathbf{F_a}(\mathbf{x}) = \mathbf{x} - \mathbf{a}$ for some constant vector $\mathbf{a}$.
- Formulate the problem

$$\boldsymbol{\rho}_\mathbf{a}(\lambda, \mathbf{x}) = \lambda \mathbf{F}(\mathbf{x}) + (1 - \lambda)\mathbf{F_a}(\mathbf{x})$$

  where λ is a real number in the interval $[0, 1]$.
- Note that the solution to $\boldsymbol{\rho}_\mathbf{a}(0, \mathbf{x}) = \mathbf{0}$ is known (for our example, it is just $\mathbf{a}$), and the solution to $\boldsymbol{\rho}_\mathbf{a}(1, \mathbf{x}) = \mathbf{0}$ is the solution to our desired problem.

We have just seen one example of how to construct a homotopy function $\boldsymbol{\rho}_\mathbf{a}(\lambda, \mathbf{x})$ so that solving $\boldsymbol{\rho}_\mathbf{a}(\lambda, \mathbf{x}) = \mathbf{0}$ is easy when $\lambda = 0$ and solves the desired problem when $\lambda = 1$.

There are many other useful ways to define a homotopy, and we will use some alternatives in the case studies.

The basic algorithm

Initialize $\lambda = 0$ and $\mathbf{x} =$ solution to $\mathbf{F_a}(\mathbf{x}) = \mathbf{0}$.

While $\lambda < 1$,

- Increase λ by a small amount.
- Solve $\boldsymbol{\rho}_\mathbf{a}(\lambda, \mathbf{x}) = \mathbf{0}$ using the previous solution vector as a starting point to solve the new problem.

The hope

- We also hope that for each of the intermediate problems, a finite solution exists.
- Because we change λ by just a little each time, we hope that the solution $\mathbf{x}(\lambda)$ also changes by just a little, so that the new nonlinear equation can be solved easily at each iteration.

Worries

- turning points (fix by allowing λ to decrease sometimes).
- bifurcations (fix by careful choice of $\mathbf{F_a}$).
- solution might fail to exist for some $\lambda < 1$.
- solution curve might wander off to infinity.

Some history

- Theoretical foundation: Chow, Mallet-Paret, Yorke (UMD!)
- Practical implementation: Layne Watson (Virginia Polytechnic Inst.) and colleagues

The difficult part of the theory: How to construct $\boldsymbol{\rho_a}$ so that we can walk all the way from $\lambda = 0$ to $\lambda = 1$ without failing to find a solution.

The basis of the theory

A function $\mathbf{w}$ is transversal to zero on an open domain U if, for any point $\hat{\mathbf{u}} \in U$ such that $\mathbf{w}(\hat{\mathbf{u}}) = \mathbf{0}$, the Jacobian matrix at $\hat{\mathbf{u}}$ has full rank.

Note that the Jacobian matrix need not be square. Its dimensions are length$(\mathbf{w})\times$ length$(\hat{\mathbf{u}})$.

A function $\mathbf{z}$ is said to be in C^k on a domain U if the function and all of its first k derivatives exist at all points in U.

Our mapping is almost always "nice"

Parameterized Sard's Theorem: Let $U = \mathcal{R}^n \times [0,1) \times \mathcal{R}^n$ and define a function $\boldsymbol{\rho} : U \to \mathcal{R}^n$ in C^2 which is transversal to zero on U. (We'll call its variables $(\mathbf{a}, \lambda, \mathbf{z})$.) Choose a point $\mathbf{a} \in \mathcal{R}^n$ and define

$$\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{z}) = \boldsymbol{\rho}(\mathbf{a}, \lambda, \mathbf{z}).$$

Then the map $\boldsymbol{\rho}_{\mathbf{a}}$ is transversal to zero on $[0,1) \times \mathcal{R}^n$ for almost all $\mathbf{a}$.

What "for almost all" means:

- "except for a set of measure zero".
- If we choose a point $\mathbf{a}$ at random, it has probability zero of giving a function that is not transversal to zero.

Why we need this:

- If we encounter a zero that does not have a full-rank Jacobian, then it may be impossible to continue the algorithm:
  - The solution path may bifurcate.
  - The solution path may end.
- If we write a computer program and choose $\mathbf{a}$ at random, it will almost never fail to give a function that is transversal to zero, and if it does, choosing a different $\mathbf{a}$ will almost always work.

Our solution curve is almost always "nice"

Theorem: In addition to the hypotheses above, assume that the equation $\boldsymbol{\rho}_{\mathbf{a}}(0, \mathbf{z}) = \mathbf{0}$ has a unique solution $\mathbf{z}_0$. Then for almost all $\mathbf{a} \in \mathcal{R}^m$, there is a curve $\boldsymbol{\gamma}$ of solutions to $\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{z}) = \mathbf{0}$, emanating from $(0, \mathbf{z}_0)$, with the properties:

- The Jacobian matrix has (full) rank n at all points in $\boldsymbol{\gamma}$.
- The curve $\boldsymbol{\gamma}$ is smooth.
- The curve does not intersect itself or any other solution curve.
- The curve does not bifurcate.
- The curve either goes to infinity or reaches the hyperplane $\lambda = 1$.

In summary, the curve is very well-behaved! In fact, we can hope to walk along it.

Where does the solution curve go?

Theorem: Under the hypotheses above,

- If the solution curve γ is bounded, then it has an accumulation point $(1, \hat{\mathbf{z}})$.
- If the Jacobian is full rank at the accumulation point, then γ has finite length.

In other words, the curve is going just where we want to go, and it gets there finitely, as long as the Jacobians are bounded away from rank-deficient matrices.

Picture. The above theorem means that we have a solution curve that starts at $\lambda = 0$ and goes to our desired solution at $\lambda = 1$. But there may be a countable number of other solution curves, some closed and some with two endpoints at $\lambda = 1$.

Note that we need to be careful not to step onto one of these other curves.

An example: convex optimization

Suppose we want to find $\mathbf{x}^*$ to solve the problem

$$\min_{\mathbf{x}} f(\mathbf{x})$$

where $f : \mathcal{R}^n \rightarrow \mathcal{R}^1$ is a C^2 convex function that is bounded below. (This last assumption assures that a bounded solution exists.)

We define the mapping

$$\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{x}) = \lambda \nabla f(\mathbf{x}) + (1 - \lambda)(\mathbf{x} - \mathbf{a})$$

Will this homotopy generate a solution curve that terminates in $(1, \mathbf{x}^*)$?

It is sufficient to verify the hypotheses of our three theorems:

- Let $U = \mathcal{R}^m \times [0, 1) \times \mathcal{R}^n$ and define a function $\boldsymbol{\rho} : U \to \mathcal{R}^n$ in C^2 which is transversal to zero on U. Let's look at the Jacobian matrix for the mapping. The Jacobian with respect to the $\mathbf{x}$ variables is

$$\mathbf{J_X} = \lambda \mathbf{H}(\mathbf{x}) + (1 - \lambda)\mathbf{I}\,,$$

  where $\mathbf{H}$ is the Hessian matrix of f. Now $\mathbf{H}$ is positive semi-definite since f is convex, so $\mathbf{J_X}$ has rank n for $\lambda \in [0, 1)$, so the complete Jacobian has full-rank, as required.

- The equation $\boldsymbol{\rho}_{\mathbf{a}}(0, \mathbf{z}) = \mathbf{0}$ should have a unique solution $\mathbf{z}_0$. Since

$$\boldsymbol{\rho}_{\mathbf{a}}(0, \mathbf{z}) = \mathbf{z} - \mathbf{a},$$

this is clearly satisfied.

- The solution curve γ needs to be bounded. Choose a number M such that $\|\mathbf{x}^*\| < M$ and $\|\mathbf{a}\| < M$. Consider the ball defined by $\|\mathbf{x}\| = 3M$. For $\mathbf{x}$ on this ball, we have

$$(\mathbf{x} - \mathbf{x}^*)^T(\mathbf{x} - \mathbf{a}) > 0,$$

and, because $\nabla f(\mathbf{x}^*) = 0$ and f is convex, we have

$$(\mathbf{x} - \mathbf{x}^*)^T \nabla f(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T(\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^*)) \geq 0\,.$$

(Proof that $(\mathbf{x} - \mathbf{x}^*)^T \nabla f(\mathbf{x}) \geq 0$ in note below.) Put these two inequalities together to get

$$(\mathbf{x} - \mathbf{x}^*)^T \boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T(\lambda \nabla f(\mathbf{x}) + (1 - \lambda)(\mathbf{x} - \mathbf{a})) > 0\,.$$

Therefore, $\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{x}) \neq \mathbf{0}$ on the ball $\|\mathbf{x}\| = 3M$, so the solution curve $\boldsymbol{\gamma}$ cannot escape from the ball and remains bounded.

Therefore, this homotopy generates a solution curve that terminates in $(1, \mathbf{x}^*)$ for almost every $\mathbf{a}$.

Proof that $(\mathbf{x} - \mathbf{x}^*)^T \nabla f(\mathbf{x}) \geq 0$:

Definition: f is a convex function if and only if

$$f(\alpha \mathbf{y} + (1-\alpha)\mathbf{x}) \leq \alpha f(\mathbf{y}) + (1-\alpha) f(\mathbf{x})$$

for all $\mathbf{x}, \mathbf{y}$ in the domain of f and all $0 \leq \alpha \leq 1$.

Let $\mathbf{z} = \mathbf{y} - \mathbf{x}$. Then

$$f(\alpha \mathbf{z} + \mathbf{x}) = f(\alpha \mathbf{y} + (1-\alpha)\mathbf{x}) \leq \alpha f(\mathbf{y}) + (1-\alpha) f(\mathbf{x}) = \alpha(f(\mathbf{y}) - f(\mathbf{x})) + f(\mathbf{x})$$

so

$$f(\mathbf{x} + \alpha \mathbf{z}) - f(\mathbf{x}) \leq \alpha(f(\mathbf{y}) - f(\mathbf{x})).$$

Therefore,

$$\frac{f(\mathbf{x} + \alpha \mathbf{z}) - f(\mathbf{x}) - \alpha \mathbf{z}^T \nabla f(\mathbf{x})}{\alpha} \leq \frac{\alpha(f(\mathbf{y}) - f(\mathbf{x})) - \alpha \mathbf{z}^T \nabla f(\mathbf{x})}{\alpha}.$$

Taking the limit as $\alpha \to 0$, the left-hand side is zero, so

$$0 \leq (f(\mathbf{y}) - f(\mathbf{x})) - \mathbf{z}^T \nabla f(\mathbf{x}).$$

If $\mathbf{y} = \mathbf{x}^*$, then $f(\mathbf{y}) - f(\mathbf{x}) \leq 0$, so
$-\mathbf{z}^T \nabla f(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T \nabla f(\mathbf{x}) \geq 0$ as desired.

Note: This is a nice example, and the result means that the continuation method is one approach to solving convex optimization problems, but such problems are generally too easy to require such heavy machinery. Use one of the Newton-type algorithms to solve convex optimization problems.

What we have accomplished

To solve the nonlinear system of equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, under the above hypotheses, all we need to do is to follow the solution curve

$$\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{x}) = \lambda \mathbf{F}(\mathbf{x}) + (1 - \lambda)\mathbf{F}_{\mathbf{a}}(\mathbf{x}) = \mathbf{0}$$

until $\lambda = 1$.

Following the solution curve

We want to follow the solution curve from $\lambda = 0$ to $\lambda = 1$.

Note first that we don't need to be too careful when following the curve – we don't care about the values at any points in between $\lambda = 0$ and $\lambda = 1$.

Note second that we need to be careful enough that we don't start walking along any other solution curve, or we will lose our way!

Method 1

Given a function $\boldsymbol{\rho}_{\mathbf{a}}$, we set $\lambda = 0$ and $\hat{\mathbf{x}} = \mathbf{a}$. Then

$$\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \hat{\mathbf{x}}) = \mathbf{0}.$$

Until $\lambda = 1$,

Increase λ a little bit.
Solve $\boldsymbol{\rho}_{\mathbf{a}}(\lambda, \mathbf{x}) = \mathbf{0}$ using your favorite algorithm (Newton-like, etc.) with $\hat{\mathbf{x}}$ as a starting guess.
Call the solution $\hat{\mathbf{x}}$.

Upon termination, we have computed $\hat{\mathbf{x}}$ so that $\boldsymbol{\rho}_{\mathbf{a}}(1, \hat{\mathbf{x}}) = \mathbf{0}$, so $\hat{\mathbf{x}}$ solves the problem $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

Issues:

- choosing the stepsize in λ.
- choosing the tolerance in the nonlinear equation solver.

Method 2

Issues of choosing a stepsize and tolerance are troublesome, and we also face them in Unit 5, in solving ordinary differential equations (ode's). Can we use our ode machinery to solve the homotopy problem?

Let's differentiate our curve. We could differentiate with respect to λ, but then the derivative would fail to exist at any turning point and we might get in trouble.

Instead, we'll introduce a new independent variable and differentiate with respect to it. The most convenient one is $s =$ arc length.

Now

$$\boldsymbol{\rho}_{\mathbf{a}}(\lambda(s), \mathbf{x}(s)) = \mathbf{0}\,,$$

so

$$\frac{d}{ds}\boldsymbol{\rho}_{\mathbf{a}}(\lambda(s), \mathbf{x}(s)) = \mathbf{0}$$

and we have the intial conditions

$$\lambda(0) = 0 \;\;,\;\; \mathbf{x}(0) = \mathbf{a}\,.$$

When the solution curve reaches $\lambda(s) = 1$, we are finished; the resulting $\mathbf{x}(s)$ solves our nonlinear system.

For uniqueness, we normalize so that

$$\left\| (\frac{d\lambda}{ds}, \frac{d\mathbf{x}}{ds}) \right\| = 1\,.$$

What does this system of ode's look like when $\mathbf{F_a}(\mathbf{x}) = \mathbf{x} - \mathbf{a}$?

$$\frac{d\lambda}{ds}\mathbf{F}(\mathbf{x}) + \lambda\mathbf{J}(\mathbf{x})\frac{d\mathbf{x}}{ds} + \frac{d\lambda}{ds}(\mathbf{x} - \mathbf{a}) + (1 - \lambda)\frac{d\mathbf{x}}{ds} = \mathbf{0}$$

$$\left(\frac{d\lambda}{ds}\right)^2 + \sum_{i=1}^{n}\left(\frac{dx_i}{ds}\right)^2 = 1$$

The system is implicit.

In principle, we could just plug the system into our favorite ode solver, and this is good for prototyping algorithms, but in practice the ode solver should be tailored to this problem:

- If the arc length parameter s gets too long, restart with a new choice of a.
- Take care that we don't wander too far from the solution curve but that we don't work too hard in following it exactly.
- Use inverse interpolation at the end to compute the value $\hat{s}$ for which $\lambda(\hat{s}) = 1$.

Software

Hompack, by Layne Watson and co-workers, is a high-quality system for solving nonlinear equations by continuation algorithms.

Final Words

- Much of our work on optimization algorithms gave us insight into algorithms for solving nonlinear equations.
- We still need to understand the parameter stepping in continuation algorithms better; stay tuned for Unit 6.
- There is a lot of high-quality software: choose a good Newton-like method, or Hompack, or a method tailored to solving a polynomial system.