Notes for 2016-04-08

Life beyond Newton

Newton's method has many attractive properties, particularly when we combine it with a globalization strategy. Unfortunately, Newton steps are not cheap. At each step, we need to:

- Form the function f *and* the Jacobian. This involves not only computational work, but also analytical work – someone needs to figure out those derivatives!
- Solve a linear system with the Jacobian. This is no easier than any other linear solve problem! Indeed, it may be rather expensive for large systems, and factorization costs cannot (in general) be amortized across Newton steps.

The Jacobian (or the Hessian if we are looking at optimization problems) is the main source of difficulty. Today we consider several iterations that deal with this difficulty in one way or the other.

Almost Newton analysis

In these notes, we will be somewhat careful about the analysis, but in general you are *not* responsible for remembering this level of detail. We will try to highlight the points that are important in practice for understanding when solvers might run into trouble, and why.

A common theme in our analysis of "almost Newton" iterations is that we will build on Newton convergence. In order to simplify life, we assume throughout that f is C^1 and the Jacobian is Lipschitz with constant M and has a bounded inverse (i.e. $\|f'(x)^{-1}\| \leq K$ for x in some neighborhood of x^*).

Let's start by looking again at the error from a Newton step. A Newton step starting from x near x^* is

$$p(x) = -f'(x)^{-1}f(x) = -f'(x)^{-1}[f(x) - f(x^*)].$$

By the mean value theorem, there is a point $\tilde{x} = x^* + \xi(x - x^*)$ on the line segment between x and x^* such that

$$f(x) - f(x^*) = f'(\tilde{x})(x - x^*).$$

Thus, we have

$$\begin{aligned} p(x) &= -f'(x)^{-1} f'(\tilde{x})(x - x^*) \\ &= -(x - x^*) + f'(x)^{-1}[f'(x) - f'(\tilde{x})](x - x^*) \quad = -(x - x^*) + d(x). \end{aligned}$$

Applying the Lipschitz condition and bounded inverse assumption,

$$\|d(x)\| \leq BM\|x - \tilde{x}\|\|x - x^*\| \leq BM\|x - x^*\|^2.$$

Therefore,

$$x + p(x) = x^* + d(x)$$

and $d(x) = O(\|x - x^*\|^2)$; that is, the iteration $x \mapsto x + p(x)$ converges quadratically from starting points near enough x^*.

We can also see that a sufficient condition for convergence is that the initial error is less than $1/(BM)$. This differs from our earlier bound of $2/(3BM)$ only because we assumed a uniform bound on the inverse of the Jacobian in the relevant region rather than assuming a bound at the solution and using the Lipschitz behavior to get everything else.

Now suppose we have an iteration

$$x^{k+1} = x^k + \hat{p}^k$$

where $\hat{p}^k$ is an approximation to the Newton step $p(x^k)$. Subtracting x^* from both sides and adding $p(x^k) - p(x^k)$ to the right side gives

$$e^{k+1} = e^k + p(x^k) - (p(x^k) - \hat{p}^k),$$

and taking norms gives

$$\|e^{k+1}\| \leq BM\|e^k\|^2 + \|p(x^k) - \hat{p}^k\|.$$

Therefore, we can think of our convergence analysis in two steps: we first analyze the error in the Newton iteration, then analyze how close our approximate Newton step is to a true Newton step.

In general, to get linear convergence, we need that $\|p(x^k) - \hat{p}^k\| < \alpha\|e^k\|$ for some $\alpha < 1$. If we have this condition, then the approximate Newton iteration will converge whenever $\|e^0\| \leq (1 - \alpha)/BM$.

Approximate Jacobians

We start with a family of iterations $x^{k+1} = x^k + \hat{p}^k$ where

$$(f'(x^k) + E_k)\hat{p}^k = -f(x^k).$$

That is, the error in the step comes from a small error in the Jacobian. As in our analysis of Newton, we replace the right hand side using the mean value theorem

$$(f'(x^k) + E_k)\hat{p}^k = -f'(\tilde{x})(x^k - x^*).$$

where $\tilde{x}$ lies on the segment between x^k and x^*. Rearranging as before, we have

$$\hat{p}^k = -e^k + \left(f'(x^k) + E_k)^{-1}\right)^{-1} \left(f'(x^k) + E - f'(\tilde{x})\right) e^k.$$

Therefore, the error iteration $e^{k+1} = e^k + \hat{p}^k$ becomes

$$e^{k+1} = \left(f'(x^k) + E_k)^{-1}\right)^{-1} \left(f'(x^k) + E_k - f'(\tilde{x})\right) e^k.$$

Taking several bounds gives us (for $B\|E_k\| < 1$),

$$\|e^{k+1}\| \leq \left(\frac{BM\|e^k\| + \|E^k\|}{1 - B\|E_k\|}\right) \|e^k\| = O(\|e^k\|^2) + O(\|e^k\|\|E^k\|).$$

The constant in parentheses is less than one if

$$\|e^k\| \leq \frac{1 - (B+1)\|E^k\|}{BM},$$

which for the zero error case reverts to the bound we saw before.

How should we read these bounds? If the bound on B is large (i.e. f' can be very nearly singular), then the method can blow up either if the initial error is too big or if $\|E^k\|$ can get big; after all, for $\|E^k\| > B^{-1}$ the system for an approximate step might be singular! We can also see issues with the rate of convergence if f' moves around quickly (i.e. M is large); but this is an issue with Newton iteration as well, and switching to an approximate Newton iteration makes things neither better nor worse.

Chord iteration

The *chord iteration* is

$$x^{k+1} = x^k - f'(x^0)^{-1} f(x^k).$$

Written in this way, the method differs from Newton in only one character — but what a difference it makes! By re-using the Jacobian at x^0 for all steps, we degrade the progress per step, but each step becomes cheaper. In particular, we can benefit from re-using a factorization across several steps:

```
% Given a function [f,J] = f(x) that returns the Jacobian only if
% it is used as a second argument, run a chord iteration starting
% from some initial x0
function [x] = chord_solve(f, x0, nsteps)

  [f0,J0] = f(x0);
  [L,U,P] = lu(J0);
  x = x0;
  for k = 1:nsteps
    fx = f(x);
    p = - U\(L\(P*fx);
    x = x + p;
  end
```

In terms of the approximate Newton framework, the chord iteration involves errors $\|E_k\| = \|f'(x^k) - f'(x^0)\| \leq M\|e^0\|$. Therefore, the iteration is guaranteed to converge for starting points such that $\|e^0\| < 1/(2BM + M)$, and the error in successive iterates is bounded by.

$$\|e^{k+1}\| \leq \left(\frac{BM\|e^k\| + M\|e^0\|}{1 - BM\|e^0\|} \right) \|e^k\| = O(\|e^0\|\|e^k\|).$$

Shamanskii iteration

The chord method involves using one approximate Jacobian forever. The Shamanskii method involves freezing the Jacobian for m steps before getting a new Jacobian; that is, one step of Shaminskii looks like

$$\begin{aligned} x^{k+1,0} &= x^k \\ x^{k+1,j+1} &= x^{k+1,j} - f'(x^k)^{-1} f(x^{k+1,j}) \\ x^{k+1} &= x^{k+1,m}. \end{aligned}$$

Like the chord iteration, Shaminskii is guaranteed to converge for starting points such that $\|e^0\| < 1/(2BM+M)$. The error for each iteration (from x^k to x^{k+1}, not from $x^{k+1,j}$ to $x^{k+1,j+1}$) satisfies

$$\|e^{k+1}\| \leq \left(\frac{BM+M}{1-BM\|e^k\|} \right) \|e^k\|^{m+1} = O(\|e^k\|^{m+1}).$$

Beyond the chord and Shaminskii iterations, the idea of re-using Jacobians occurs in several other methods.

Finite-difference Newton

So far, we have assumed that we can compute the Jacobian if we want it. What if we just don't want to do the calculus to compute Jacobians? A natural idea is to approximate each column of the Jacobian by a finite difference estimate:

$$f'(x^k) e_j \approx \frac{f(x^k + he_j) - f(x^k)}{h}$$

Using Lipschitz bounds on f' gives the error bound

$$\left\| f'(x^k) - \frac{f(x^k + he_j) - f(x^k)}{h} \right\| \leq Mh,$$

and an approximation to $f'(x^k)$ based on finite difference approximation would have a two norm error of at most $\|E^k\| \leq \sqrt{n} Mh$. The method must converge for initial errors less than $(1 - \sqrt{n}(B+1)Mh)/(1 - \sqrt{n}BMh)$, and the convergence is bounded by

$$\|e^{k+1}\| \leq \left(\frac{BM\|e^k\| + \sqrt{n}Mh}{1 - \sqrt{n}BMh} \right) \|e^k\| = O(h\|e^k\|).$$

Inexact Newton

So far, we have considered approximations to the Newton step based on approximation of the Jacobian matrix. What if we instead used the exact Jacobian matrix, but allowed the update linear systems to be solved using an iterative solver? In this case, there is a small residual, i.e.

$$f'(x^k)\hat{p}^k = -f(x^k) + r^k$$

where $\|r^k\| \leq \eta_k \|f(x^k)\|$ (i.e. η_k is a relative residual tolerance on the solve). In this case,

$$\|\hat{p}^k - p(x^k)\| = \|f'(x^k)^{-1} r^k\| \leq B\|r^k\| \leq \eta_k B \|f(x^k)\|.$$

We also have that

$$\|f(x^k)\| = \|f(x^k) - f(x^*)\| = \|f'(\tilde{x})e^k\| \leq C\|e^k\|$$

where C is a bound on the norm of f'. Thus

$$\|\hat{p}^k - p(x^k)\| \leq \eta_k BC \|e^k\|,$$

which we combine with the bound from the start of the notes to give

$$\|e^{k+1}\| \leq B(M\|e^k\| + \eta_k C)\|e^k\| = O(\|e^k\|^2) + O(\eta_k \|e^k\|).$$

Hence, we have the following trade-off. If we solve the systems very accurately (η_k small), then inexact Newton will behave much like ordinary Newton. Thus, we expect to require few steps of the outer, nonlinear iteration; but the inner iteration (the linear solver) may require many steps to reach an acceptable residual tolerance. In contrast, if we choose η_k to be some modest constant independent of k, then we expect linear convergence of the outer nonlinear iteration, but each step may run relatively fast, since the linear systems are not solved to high accuracy.

One attractive feature of Krylov subspace solvers for the Newton system is that they only require matrix-vector multiplies with the Jacobian — also known as directional derivative computations. We can approximate these directional derivaties by finite differences to get a method that may be rather more attractive than computing a full Jacobian approximation by finite differencing. However, it is necessary to use a Krylov subspace method that tolerates inexact matrix vector multiplies (e.g. FGMRES).

Broyden

A *quasi-Newton* method updates approximations to both the solution and the Jacobian over the course of an iteration. The most famous such method, Broyden's method, is analogous to the secant iteration in 1D. The iteration is

$$x^{k+1} = x^k - B_k^{-1} f(x^k)$$

where B_k is computed by the update

$$B_{k+1} = B_k + \frac{f(x^{k+1})(p^k)^T}{\|p^k\|^2}$$

where $p^k = x^{k+1} - x^k$. This update satisfies the *secant equation*

$$B_{k+1} p^k = f(x^{k+1}) - f(x^k).$$

It is not the only solution to the secant equation, but it is a good one.

Broyden's method converges superlinearly, though not quadratically; the argument is beyond the scope of these notes, though none of the ideas are that complicated. For small problems, the implementations of Broyden are interesting in that they involve continuously updating factorizations to account for new low-rank additions to an approximate Jacobian. This can be done in $O(n^2)$ time (rather than $O(n^3)$) if one is clever about the numerical linear algebra. For large problems, the complexity of the matrix B_k gets out of hand as one proceeds, so we usually use tricks like *limited memory Broyden*, keeping only a few recent updates to the initial (often diagonal) Jacobian approximation.