Numerical Methods in Economics

MIT Press, 1998

Notes for nonlinear equations

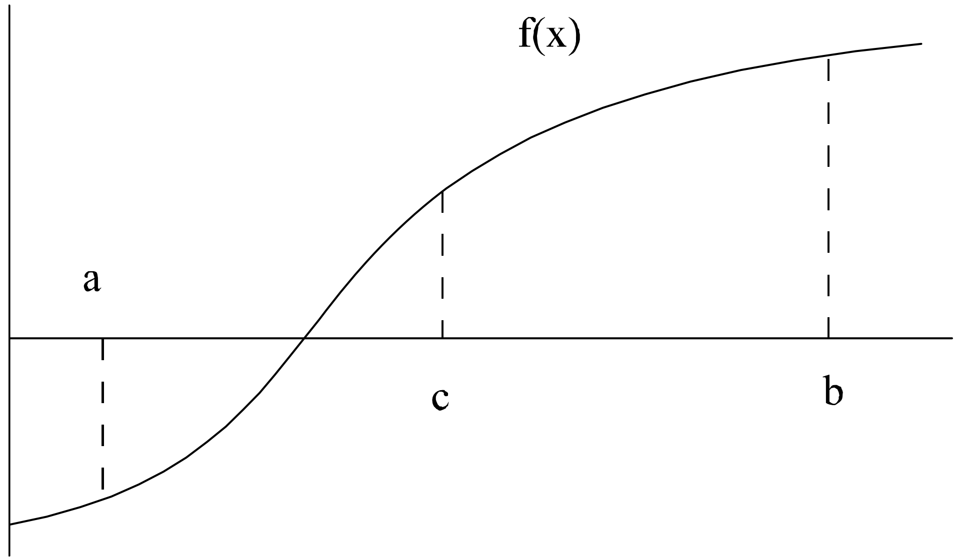
March 2, 2020

Nonlinear Equations

- Two forms of equations: zeros and fixed points of $f : R^n \to R^n$
  - A *zero of f* is any x such that $f(x) = 0$
  - A *fixed point* of f is any x such that $f(x) = x$.
  - Note: x is a fixed point of $f(x)$ iff it is a zero of $f(x) - x$.
- Existence of solutions is examined in Brouwer's theorem and its extensions.
- Examples
  - Arrow-Debreu general equilibrium: find a price at which excess demand is zero
  - Nash equilibrium of games with continuous strategies
  - Transition paths of deterministic dynamic systems
  - Approximate policy functions in nonlinear dynamic problems

One-Dimensional Problems: Bisection

- Suppose that $f(a) < 0 < f(b)$
- Step 1: Pick a point $c \in (a, b)$
  - If $f(c) = 0$, stop
  - If $f(c) < 0$, reduce interval to (c, b)
  - If $f(c) > 0$, reduce interval to (a, c)
- Repeat



One-Dimensional Problems: Newton's Method

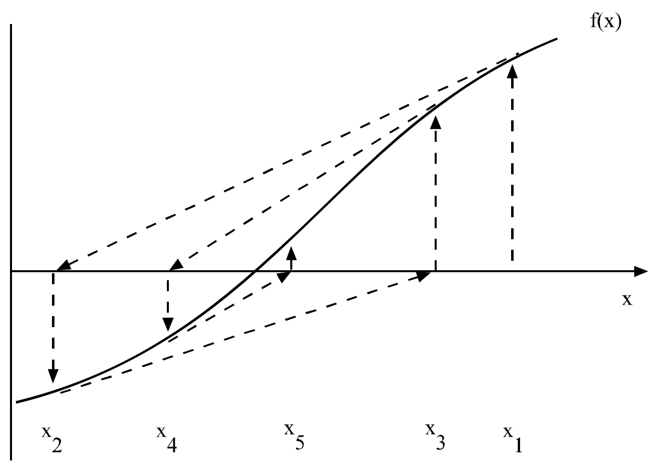
- Given guess x_k, compute linear approximation

$$f(x) \doteq f(x_k) + f'(x_k)(x - x_k)$$

and let x_{k+1} be zero of linear approximation:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{5.2.1}$$

- Graph of Newton's method:



- Convergence: Suppose f is C^2 and $f(x^*) = 0$. If x_0 is close to x^*, $f'(x^*) \neq 0$, and $|f''(x^*)/f'(x^*))| < \infty$, then (5.2.1) converges to x^* quadratically; that is,

$$\limsup_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \frac{1}{2}\frac{|f''(x^*)|}{|f'(x^*)|} < \infty\ . \tag{5.2.2}$$

Pathological Examples

- Newton's method works well when it works, but it can fail.
- Example: $f(x) = x^{1/3}e^{-x^2}$.
  - Unique zero of f is at $x = 0$.
  - Newton's method is

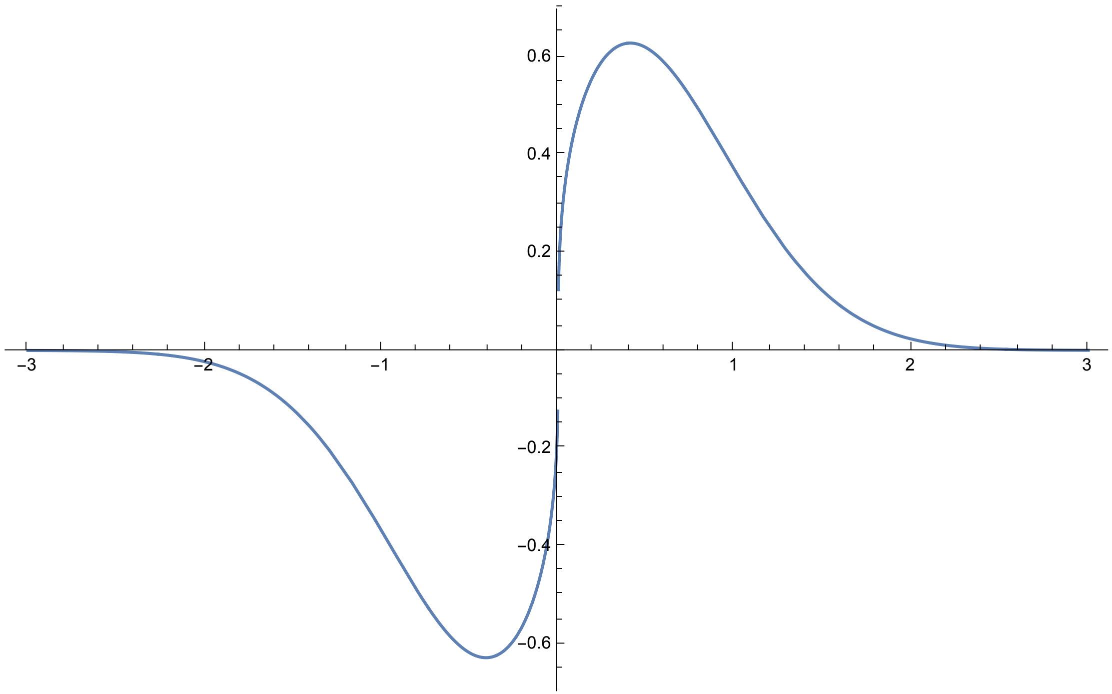
$$x_{n+1} = x_n \left(1 - \frac{3}{1 - 6x_n^2}\right) \tag{5.2.4}$$

    which has two pathologies.
    - For x_n small, (5.2.4) reduces to $x_{n+1} = -2x_n$; hence, (5.2.4) converges to 0 only if $x_0 = 0$ is the initial guess.
    - For x_n large, (5.2.4) becomes $x_{n+1} = x_n(1 + \frac{2}{x_n^2})$, which diverges, but will eventually satisfy stopping rule at some large x_n.
  - Divergence due to $f''(0)/f'(0) = \infty$
  - "Convergence" arises because e^{-x^2} factor squashes f at large x; in some sense, since $f(\pm\infty) = 0$.

Pathological example of Newton's method

In[79]:= `Plot[Sign[x] Abs[x]^(1/3) Exp[-x x], {x, -3, 3}, ImageSize → Large]`

Out[79]=



Define the Newton iteration function.

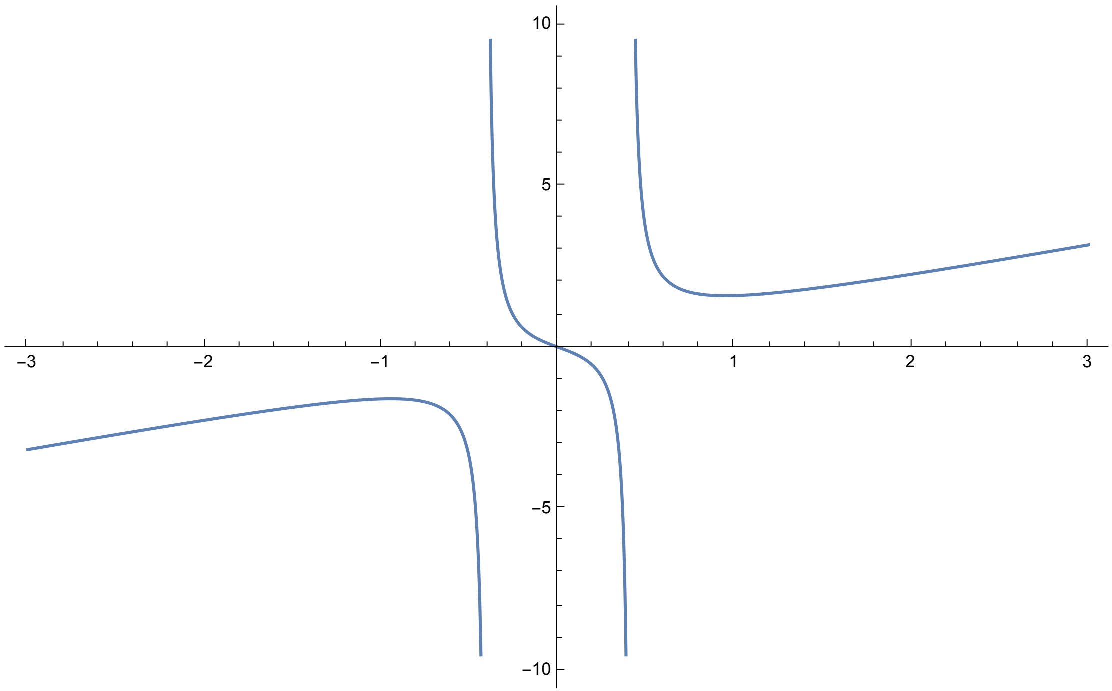
In[80]:=
```
fiter[x_] = x (1 - (3 / (1 - 6 x^2)))
```

Out[80]= $x\left(1-\frac{3}{1-6x^2}\right)$

Plot one iteration

In[81]:=
```
Plot[fiter[x], {x, -3, 3}, ImageSize → Large]
```

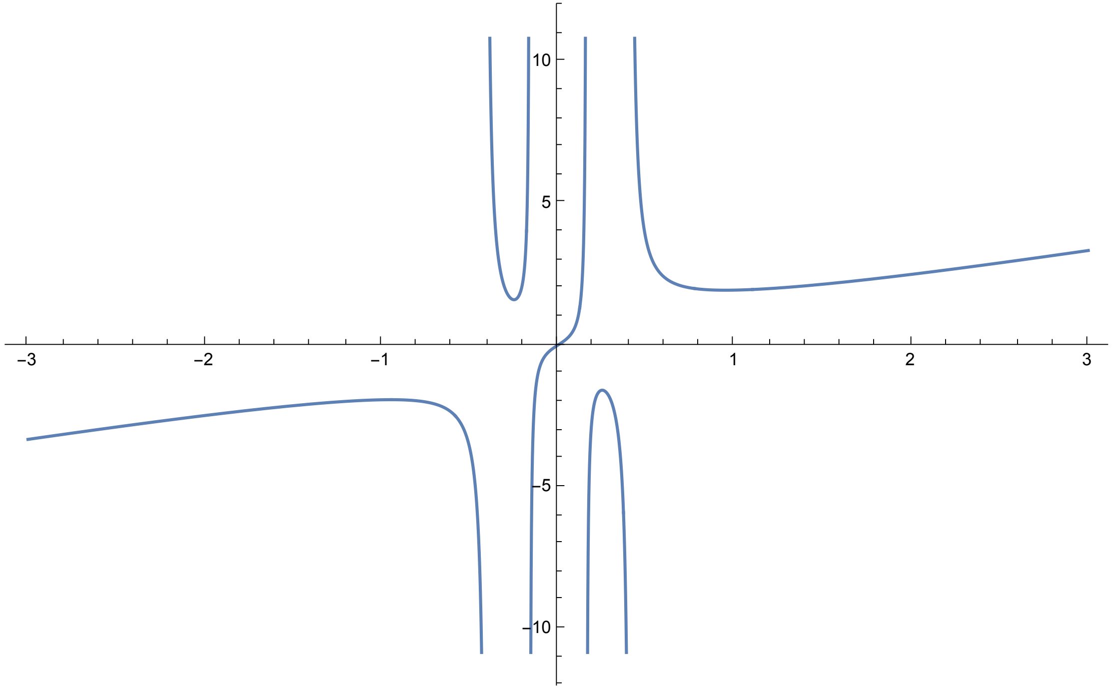
Out[81]=



Plot two iterations

In[82]:=
```
Plot[fiter[fiter[x]], {x, -3, 3}, ImageSize → Large]
```

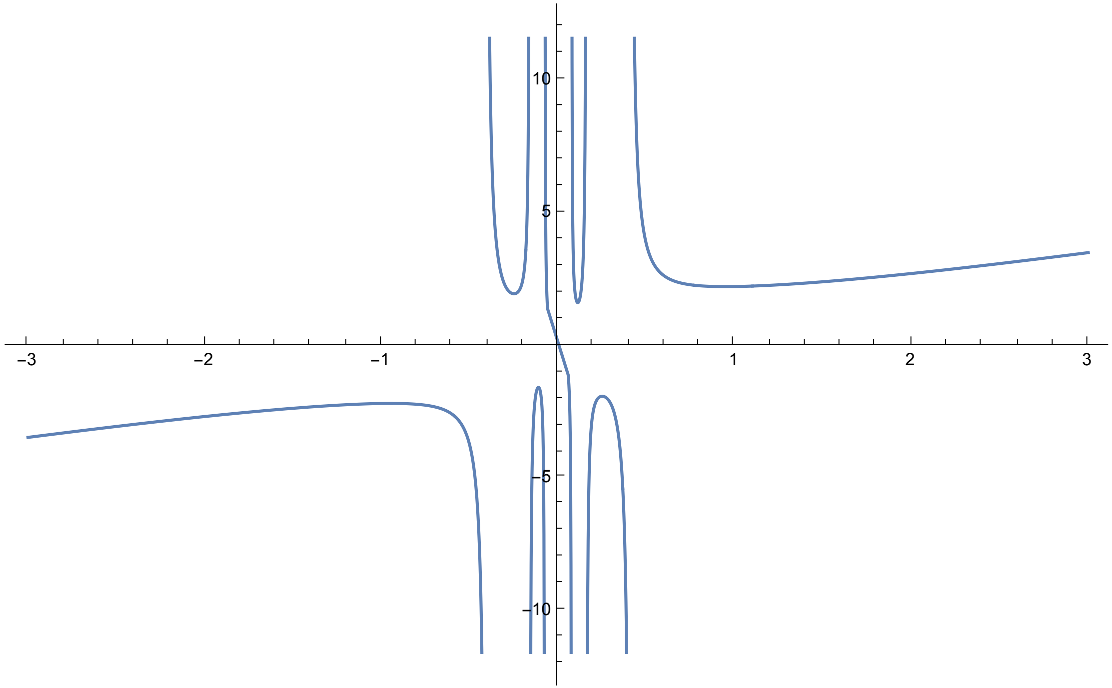
Out[82]=



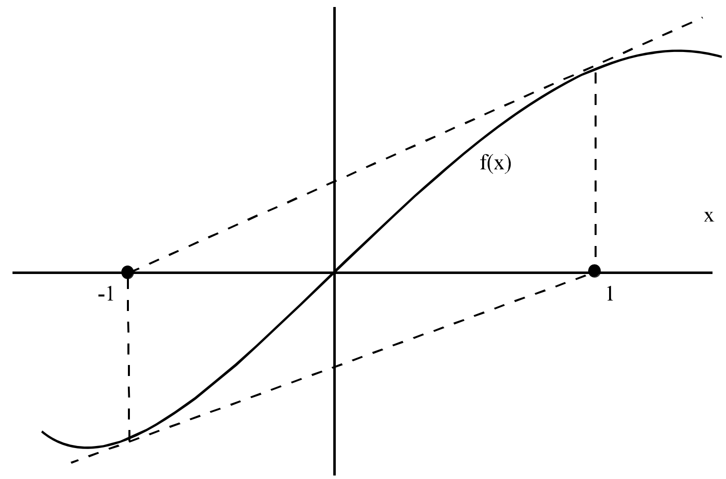
Plot three iterations

In[83]:=
```
Plot[fiter[fiter[fiter[x]]], {x, -3, 3}, ImageSize → Large]
```

Out[83]=



- Example: convergence to a cycle:



A General Equilibrium Example

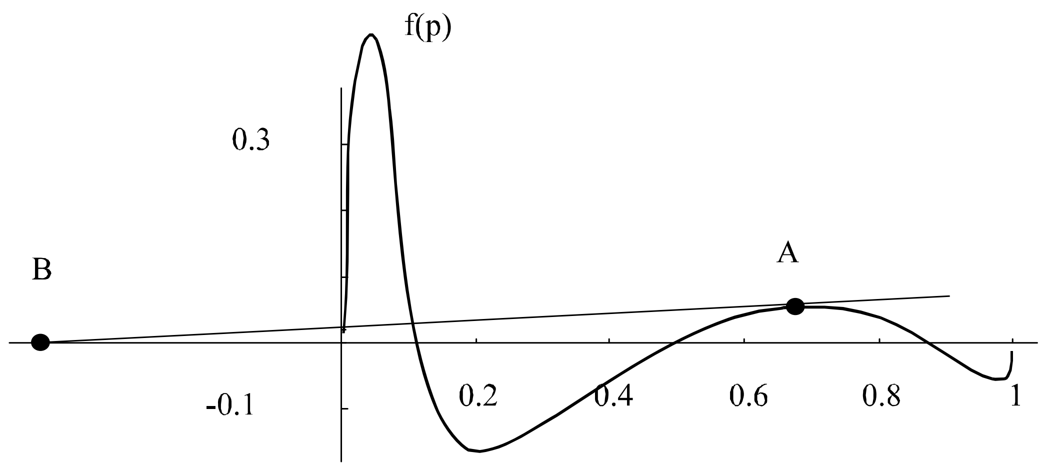
- Demand function is

$$d_j^i(p) = \theta_j^i I^i p_j^{-\eta_i}$$

$$\theta_j^i \equiv (a_j^i)^{\eta_i} \Big/ \sum_{\ell=1}^{2} (a_\ell^i)^{\eta_i} p_\ell^{(1-\eta_i)}$$

- Three equilibria:. (0.5, 0.5), (0.1129, 0.8871), (0.8871, 0.1129).
- Reduce to a one-variable problem by $p_2 = 1 - p_1$, producing

$$f(p_1) \equiv \sum_{i=1}^{2} d_1^i(p_1, 1 - p_1) - \sum_{i=1}^{2} e_1^i = 0 \tag{5.2.6}$$



- Notice: Newton's method may send p negative.

Secant Method

- Problem: $f'(x)$ may be costly.
- Solution: *secant method* approximates $f'(x_k)$ with secant of f between x_k and x_{k-1}:

$$x_{k+1} = x_k - \frac{f(x_k)\,(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \tag{5.3.1}$$

- Convergence: If $f(x^*) = 0$, $f'(x^*) \neq 0$, and $f''(x)$ is continuous near x^*, then (5.3.1) converges at rate $(1+\sqrt{5})/2$, that is

$$\limsup_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^{(1+\sqrt{5})/2}} < \infty \tag{5.3.3}$$

Multivariate Equations: Gauss-Jacobi Algorithm

- Suppose $f : R^n \rightarrow R^n$, and we want to solve $f(x) = 0$:

$$\begin{aligned} f^1(x_1, x_2, \cdots, x_n) &= 0, \\ &\vdots \\ f^n(x_1, x_2, \cdots, x_n) &= 0. \end{aligned} \tag{5.4.1}$$

- Gauss-Jacobi method.
  - Given kth iterate, x^k, use equation i to compute x_i^{k+1}:

$$\begin{aligned} &f^1(x_1^{k+1}, x_2^k, x_3^k, \cdots, x_n^k) &= 0, \\ &f^2(x_1^k, x_2^{k+1}, x_3^k, \cdots, x_n^k) &= 0, \\ &\vdots \\ &f^n(x_1^k, x_2^k, \cdots, x_{n-1}^k, x_n^{k+1}) &= 0. \end{aligned} \tag{5.4.2}$$

  - Gauss-Jacobi repeatedly solves n equations in one unknown.
  - Gauss-Jacobi is affected by the indexing scheme.
    * Otherwise, there are $n(n-1)/2$ different Gauss-Jacobi schemes.
    * Sometimes there is a natural scheme implying diagonal dominance (or, gross substitutes)
    * Strategy: choose indexing which makes Jacobian nearly diagonal

Multivariate Equations: Gauss-Seidel Algorithm

- Gauss-Jacobi: use new guess of x_i, x_i^{k+1}, only after we have computed the entire vector of new values, x^{k+1}.
- Gauss-Seidel: use new guess, x_i^{k+1},as soon as it is available.
- Formal definition: construct x^{k+1} componentwise by solving

$$\begin{aligned} f^1\left(x_1^{k+1}, x_2^k, x_3^k, \cdots, x_n^k\right) &= 0, \\ f^2\left(x_1^{k+1}, x_2^{k+1}, x_3^k, \cdots, x_n^k\right) &= 0, \\ &\vdots \\ f^{n-1}\left(x_1^{k+1}, \cdots, x_{n-2}^{k+1}, x_{n-1}^{k+1}, x_n^k\right) &= 0, \\ f^n\left(x_1^{k+1}, \cdots, x_{n-2}^{k+1}, x_{n-1}^{k+1}, x_n^{k+1}\right) &= 0. \end{aligned} \tag{5.4.4}$$

- Both indexing and ordering matter in GS.
  - Back-substitution on triangular system is GS
  - Strategy: choose indexing and ordering which makes Jacobian nearly triangular

- Features of Gaussian methods:
  - Each step in GJ or GS is a nonlinear equation
    - Usually solved by some iterative method.
    - Economize on effort at each iteration with loose stopping rule.
  - Can apply extrapolation and acceleration methods
  - Can apply ideas at block level - “block GJ, block GS”
    - Find groups of variables and orderings such that Jacobian is nearly block diagonal or block triangular.
    - Example: in {apples, oranges, cheddar cheese, swiss cheese} problem, put cheeses in one block, fruit in the other, and use Newton to solve blocks
  - Convergence is at best linear
    - Discussion of convergence in chapter 3 applies here.
    - Key fact: for any $x^{k+1} = G(x^k)$ the spectral radius of $G_x(x^*)$ is asymptotic linear rate of convergence.

Fixed-Point Iteration

- The simplest iterative method for solving $x = f(x)$ is

$$x^{k+1} = f(x^k) \tag{5.4.8}$$

  called fixed-point iteration; also known as *successive approximation*, *successive substitution*, or *function iteration*.

- Method is sensitive to transformations: Consider

$$x^3 - x - 1 = 0 \tag{5.3.3}$$

  - Rewrite as $x = (x+1)^{1/3}$; then the iteration

$$x_{k+1} = (x_k + 1)^{1/3}. \tag{5.3.4}$$

    converges to a solution of (5.3.3) if $x_0 = 1$.
  - Rewrite (5.3.3) as $x = x^3 - 1$; then the iteration

$$x_{k+1} = x_k^3 - 1 \tag{5.3.5}$$

    diverges to $-\infty$ if $x_0 = 1$.

- Naive implementations of the fixed-point iteration approach often fail.
- However, most algorithms have the form $x_{k+1} = f(x_k)$.
- Aim: construct fixed-point iteration which works.

Contraction Mapping Case of Function Iteration

- For a special class of functions, fixed-point iteration will work well.
- A *differentiable contraction map on* D is any C^1 $f : D \to R^n$ defined on a closed, bounded, convex set $D \subset R^n$ such that
  - $f(D) \subset D$, and
  - $\max_{x \in D} \| J(x) \|_\infty < 1$, $J(x)$ is Jacobian of f.
- *(Contraction mapping theorem)* If f is a differentiable contraction map on D, then
  - $x = f(x)$ has a unique solution, $x^* \in D$;
  - $x^{k+1} = f(x^k)$ converges to x^*; and
  - there is a sequence $\epsilon_k \to 0$ such that

$$\| x^* - x^{k+1} \|_\infty \leq (\| J(x^*) \|_\infty + \epsilon_k) \| x^* - x^k \|_\infty$$

- If $f(x^*) = x^*$, f is Lipschitz at x^*, and $\rho(J(x^*)) < 1$, then for x^0 close to x^*, $x^{k+1} = f(x^k)$ is convergent.

Stopping Rule Problems for Multivariate Systems

- Use ideas from chapter 1
- First, use a rule for stopping.
  - If we want $\| x^k - x^* \| < \epsilon$, we continue until $\| x^{k+1} - x^k \| \leq (1-\beta)\epsilon$ where $\beta = \rho(G_x(x^*))$.
  - Sometimes we know β, as with some contraction mappings
  - Otherwise, estimate β with

$$\hat{\beta} = \left(\frac{\| x^k - x^{k+1} \|}{\| x^{k-L} - x^{k+1} \|} \right)^{1/L}$$

    for some L.
- Second, check that $f(x^k)$ is close to zero.
  - Require that $\| f(x^k) \| \leq \delta$ for some small δ.
  - You should have each component of f small
  - Be careful about units; check should be unit-free
- δ and ϵ should not be less than square root of error in computing f.

Newton's Method for Multivariate Equations

- Sequential linear approximations:
  - Replace f with a linear approximation at x^k
  - Solve linear approximation for x^{k+1}
- Formally:
  - Newton approx around x^k is $f(x) \doteq f(x^k) + J(x^k)\,(x - x^k)$.
  - Zero of approx is

$$x^{k+1} = x^k - J(x^k)^{-1}\, f(x^k) \tag{5.5.1}$$

- Convergence: If $f(x^*) = 0$, $\det(J(x^*)) \neq 0$ and $J(x)$ is Lipschitz near x^*, then for x^0 near x^*, the sequence defined by (5.5.1) satisfies

$$\lim_{k\to\infty} \frac{\| \, x^{k+1} - x^* \, \|}{\| \, x^k - x^* \, \|^2} < \infty \tag{5.5.2}$$

- Problems with Newton method
  - Jacobian, $J(x)$, may be expensive to compute (but not if you use automatic differentiation)
  - May not converge
  - Should really be called the Newton-Raphson-Fourier-Simpson method
- Solutions
  - Broyden approximates $J(x)$
  - Powell hybrid improves likelihood of convergence.
  - Homotopy methods will converge

Secant Method (Broyden)

- Jacobian, $J(x)$, is costly to compute
  - Analytic expressions are difficult to compute
  - Finite-difference approximations require n^2 evaluations of f.
- In R, we used the secant; can we do this for R^n?
- Broyden method
  - Start with initial Jacobian guess, A_0
  - Use A_k to compute the Newton step, s^k: $A_k s^k = -f(x^k)$
  - Set $x^{k+1} = x^k + s^k$.
  - Choose A_{k+1} to be
    * close to A_k
    * consistent with secant equation $f(x^{k+1}) - f(x^k) = A_{k+1}s^k$
    * for any direction q orthogonal to s^k, want $A_{k+1}q = A_k q$, i.e., no change in directions orthogonal to Newton step

– Broyden update is

$$A_{k+1} = A_k + \frac{(y_k - A_k s^k)\,(s^k)^\top}{(s^k)^\top s^k}$$

$$y_k \equiv f(x^{k+1}) - f(x^k)$$

– Stop iteration when $f(x^k)$ is close to zero, or when s^k is small.

– Convergence: There exists $\epsilon > 0$ such that if $\| x^0 - x^* \| < \epsilon$ and $\| A_0 - J(x^*) \| < \epsilon$, then the Broyden method converges superlinearly.

– Key properties of Broyden versus Newton

  * Convergence asserted only x^k, not A_k
  * Need good initial guess for A_0
  * Each iteration of the Broyden method is cheap to compute
  * Broyden method will need more iterations than Newton's method.
  * For large systems, Broyden dominates

Use Least Squares To Improve Chances of Convergence

- Nonlinear Equations as an optimization problem
  - Any solution to $f(x) = 0$ is a *global* solution of

$$0 = \min_{x} \sum_{i=1}^{n} f^{i}(x)^{2} \equiv SSR(x) \tag{5.6.1}$$

  - Benefits of (5.6.1)
    * Can use optimization procedures
    * Will always converge to something
    * May give a good initial guess for any solver
  - Problems with (5.6.1):
    * Hessian is generally ill-conditioned; roughly equals the square of the condition number of $J(x)$
    * (5.6.1) may have many local minima
- Powell's Hybrid Method
  - Do Newton, except check if Newton step reduces the value of $SSR(x)$
  - If not, then switch to least squares
  - Powell (1970) implemented procedure which avoids some conditioning problems of naive scheme.