Order of convergence and Newton's method

2/1/2011

We previously saw that an iterative scheme which generates a sequence $x_0, x_1, \ldots, x_k, \ldots$ to approximate an exact value x_* must satisfy:

$$|e_{k+1}| \leq L |e_k|^d \quad , \quad \text{where} \quad e_k = x_k - x_*$$

for convergence. In particular:

- If $d=1$, then we shall also require $L<1$ in order to guarantee that the error e_k is being reduced. In this case, we talk about <u>linear convergence</u>
- If $d>1$ we no longer need $L<1$ for the error to decrease (although we would need to be "close enough" to the solution to guarantee progress). This case is described as <u>superlinear</u> convergence. The case $d=2$ is called <u>quadratic</u> convergence.

We previously saw that Newton's method converges quadratically. More generally, for the fixed point iteration $x_{k+1} = g(x_k)$, <u>if</u> $g'(a) = 0$ (a is the solution) then:

From Taylor's theorem:

$$g(x) = g(a) + g'(a)(x-a) + g''(c)\frac{(x-a)^2}{2}$$

where c is a number between x & a.

Set $x \leftarrow x_n$. Since $g'(a) = 0$, we have

$$g(x_n) = g(a) + g''(c)\frac{(x_n-a)^2}{2}$$

$$|g(x_n) - g(a)| = \left|\frac{g''(c)}{2}\right| |x_n-a|^2$$

$$|x_{n+1} - a| = \left|\frac{g''(c)}{2}\right| |x_k-a|^2 \Rightarrow |e_{n+1}| \leq L |e_n|^2$$

where $L = \max_{\substack{x \text{ between} \\ a \& x_n}} \left|\frac{g''(x)}{2}\right|$

For Newton's Method $g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$. If $f'(a) \neq 0$, then obviously $g'(a) = 0$ and Newton converges quadratically.

So far, we have ignored the case $g'(a) = 0$. This case is typically described as a <u>multiple root</u> because if $f(x)$ is a polynomial and $f(a) = f'(a) = \cdots = f^{(k-1)}(a) = 0$, this means that $(x-a)^k$ (a root with multiplicity $= k$) is a <u>factor</u> of $f(x)$.

Let's now assume that $f(a) = f'(a) = \cdots = f^{(k-1)}(a) = 0$, but $f^{(k)}(a) \neq 0$. At first, it may appear that Newton's method may be inapplicable in this case, because the denominator $(f'(x))^2$ becomes near-zero close to the solution. However, consider:

$$f(x) = (x-1)^3(x+2)$$

Newton's method $g(x) = x - \dfrac{f(x)}{f'(x)} = x - \dfrac{(x-1)^3(x+2)}{3(x-1)^2(x+2) + (x-1)^3}$

$$= x - \frac{(x-1)(x+2)}{4x+5} \quad (\star)$$

← remains non-zero near the solution $x_* = 1$

→ optional reading

In fact we can show that g remains a contraction near a in this case. Remember: $g'(x) = \dfrac{f(x)f''(x)}{[f'(x)]^2}$

Taylor on $f(x)$: $f(x) = \underbrace{f(a)}_{=0} + \underbrace{f'(a)}_{=0}(x-a) + \cdots + \underbrace{f^{(k-1)}(a)}_{=0}\frac{(x-a)^{k-1}}{(k-1)!} + \underbrace{f^{(k)}(c_1)}_{\neq 0}\frac{(x-a)^k}{k!}$

Taylor on $f'(x)$: $f'(x) = \underbrace{f'(a)}_{=0} + \underbrace{f''(a)}_{=0}(x-a) + \cdots + \underbrace{f^{(k-1)}(a)}_{=0}\frac{(x-a)^{k-2}}{(k-2)!} + \underbrace{f^{(k)}(c_2)}_{\neq 0}\frac{(x-a)^{k-1}}{(k-1)!}$

Taylor on $f''(x)$: $f''(x) = f''(a) + \cdots + f^{(k-1)}(a)\frac{(x-a)^{k-3}}{(k-3)!} + f^{(k)}(c_3)\frac{(x-a)^{k-2}}{(k-2)!}$

Thus: $$g'(x) = \frac{f^{(k)}(c_1)\frac{(x-a)^k}{k!}\, f^{(k)}(c_3)\frac{(x-a)^{k-2}}{(k-2)!}}{\left[f^{(k)}(c_2)\frac{(x-a)^{k-1}}{(k-1)!}\right]^2}$$

$$\xrightarrow[x\to a]{} \frac{[f^{(k)}(a)]^2}{[f^{(k)}(a)]^2}\,\frac{[(k-1)!]^2}{k!\,(k-2)!} = \frac{k-1}{k}$$

Since $g'(a) = \frac{k-1}{k} < 1$, g is a contraction in an interval $(a-\delta, a+\delta)$.

optional reading ←

However, in all of these cases, convergence is limited to be only linear since

$$g(x_n) = g(a) + g'(a)(x_n - a) \Rightarrow g(x_k) - g(a) = g'(c)(x_k - a)$$

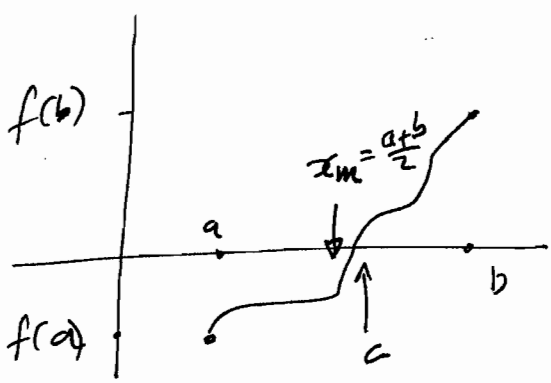
$$\Rightarrow |x_{n+1} - a| \le L\,|x_n - a| \Rightarrow |e_{n+1}| \le L\,|e_k|$$

Caution: If you know (or suspect) $f(x)$ to have a multiple root, then Newton's method is only safe (albeit slow) when we can write an analytic formula for $\frac{f(x)}{f'(x)}$ that is not at risk of division by zero (see (*)). otherwise roundoff error can cause instabilities when ~~subtracting~~ dividing two near-zero quantities (f and f')

Bisection (§2.2)

An alternative method for locating a solution of the nonlinear equation $f(x)=0$, assuming that f is continuous is the bisection method (or Bolzano's bisection method). The name (and the reason why it works) is due to Bolzano's theorem for continuous functions

Thm (Bolzano) If f is continuous in $[a,b]$ and $f(a)f(b)<0$ (i.e. f has different signs at a & b). then, there is a value $c\in[a,b]$ such that $f(c)=0$.



The bisection algoritm attempts to locate c, by checking the 2 sub-intervals (a,x_m), (x_m,b) where x_m is the midpoint $x_m=\frac{a+b}{2}$.

If $f(x_m) = 0$, we have our solution.

In the much more likely case that $f(x_m) \neq 0$, we observe that $f(x_m)$ must have the opposite sign than one of $f(a)$ or $f(b)$.

Thus:

- Either $f(a)f(x_m) < 0$, or
- $f(x_m)f(b) < 0$

We pick whichever of these 2 ~~intervals~~ intervals satisfies this condition, and continue the bisection process with it.

The bisection algorithm is summarized (in pseudocode) as follows:

Set $I_0 = [a_0, b_0]$ where $f(a_0) f(b_0) < 0$

For $k = 0, 1, \ldots, N$

$x^M \leftarrow \frac{a_k + b_k}{2}$

if $x^M = 0$ then x^M is the desired solution

else

if $f(a_k) f(x^M) < 0$ Set $I_{k+1} = [a_{k+1}, b_{k+1}] \leftarrow [a_k, x^M]$

if $f(x^M) f(b_k) < 0$ Set $I_{k+1} = [a_{k+1}, b_{k+1}] \leftarrow [x^M, b_k]$

Report the approximate solution as $x_{approx} = \frac{a_N + b_N}{2}$.

Convergence: At every iteration we can define the "approximation" x_k as the midpoint $x_k = \frac{a_k + b_k}{2}$ of I_k. Since the actual solution $f(a) = 0$ satisfies $a \in I_k$, we have

$$|x_k - a| \leq \frac{1}{2} |I_k|$$

If $|I_0| = L$, we have (length) $|I_k| = \left(\frac{1}{2}\right)^k L$ thus

$$|x_k - a| \leq \left(\frac{1}{2}\right)^k L$$

This is a bit different than the previous definition where $|e_{k+1}| \leq h |e_k|^d$

For example $|e_n|$ is not strictly guaranteed to decrease at every iteration. Instead we have

$$E_{k+1} \leq \frac{1}{2} E_k \qquad (**)$$

where $|e_i| = |x_i - a| \leq E_i \quad \forall i$

This is still regarded as linear convergence, by virtue of (**). The bisection method is very robust, and despite having only linear convergence it can be used to find an appoximation within any desired error tolerance. It is often used to localize a good initial guess for fixed pt. iterations. Note that bisection is not a F.P.I itself. ~~Every iterat~~

The <u>secant method</u> is designed to mimic Newton's method, but relaxes the condition that we need to know an analytic expression for $f'(x)$. It operates as follows:

- We start the iteration not only with 1 initial guess but also with a second, improved approximation. This 2nd value can be obtained, for example, using the bisection method. Denote these 2 values by x_0, x_1

- At the k-th step, approximate:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

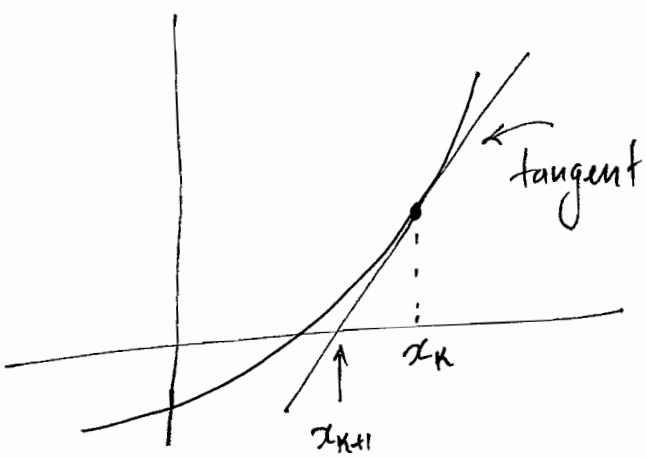
Remember, since $f'(x_k) = \lim_{y \to x_k} \frac{f(x_k) - f(y)}{x_k - y}$, as the iterates get closer to one another (while approaching the sdution) this approximation becomes more and more accurate.

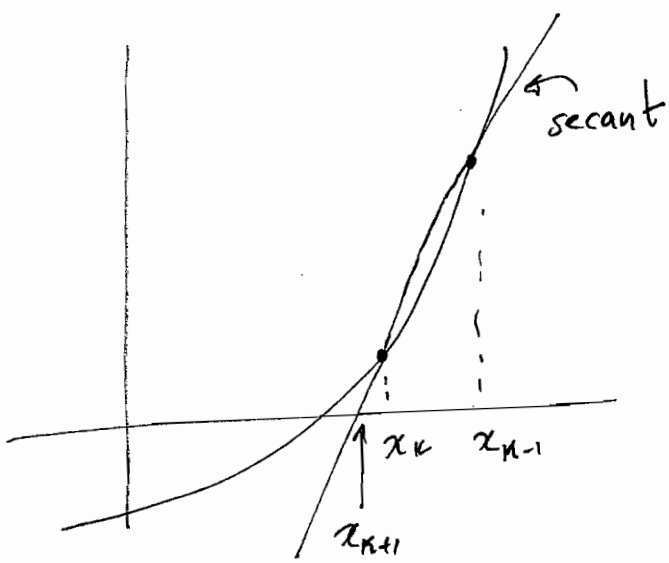
We then replace the approximation for $f'(x_k)$ in Newton's method $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ to obtain:

Secant Method:

$$x_{k+1} = x_k - \frac{f(x_k)}{\dfrac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}}$$

Geometrically, Newton's method approximates $f(x)$ at each step by the tangent line, while this last method approximates f by the secant line





We can show that, once we are "close enough" to the solution, the error e_k in secant method satisfies:

$$|e_{k+1}| \le C \cdot |e_k|^d \quad , \text{ where } d = \frac{1+\sqrt{5}}{2} \approx 1.6$$

Thus secant method provides superlinear convergence. In practice, it may need a few more iterations (50% more?) than Newton, but each iteration is cheaper since no derivatives need to be evaluated.