MS554 Lecture Notes: Solving Non-linear Equations
Lecture 6; Sept. 17, 2001

Objectives:

- The problem; solve x such that $f(x) = a$ can always be re-written to solve x such that $g(x) = 0$, and can therefore always be treated as a "root-finding problem".
- Roots can be found analytically for some problems (linear, quadratic, etc.).
- Roots for non-linear problems can be found, to within a desired confidence, using a variety of iterative methods.
- Methods covered: bisection, newton's, secant, false position, fixed-point iteration.
- **Materials:** Hornberger and Wiberg, Chapter 2; Gerald and Wheatley, Chapter 1

Bisection Method

Suppose you want to find some x such that $f(x) = 0$, and you know that that for two values, a and b $f(a) > 0$ and $f(b) < 0$. If f(x) is continuous, then there must be some x between a and b that is a root of f(x). The bisection method searches for this root iteratively, using $x_0 = (a+b)/2$ as the first guess. If $f(x_0) > 0$, you knows that x_0 is closer to the root than a, so a is discarded, and the next guess is $x_1 = (x_0 + b)/2$. Conversely, if $f(x_0) < 0$, you knows that x_0 is closer to the root than b, so b is discarded, and the next guess is $x_1 = (a + x_0)$. The process continues until either $|f(x_n)| < \epsilon$ or $|x_n - x_{n+1}| < \epsilon$, where ϵ is the desired level of tolerance. In psuedo-code:

$$
\begin{aligned}
& f(a) < 0;\; f(b) > 0;\; n = 0; \\
& x_0 = \frac{a+b}{2}; \\
& do\ while\ abs\,(f(x_n)) > \epsilon \\
& if\ f(x_n) < 0;\; a = x_n \\
& else\ if\ f(x_n) > 0;\; b = x_n \\
& x_{n+1} = \frac{a+b}{2} \\
& n = n + 1 \\
& enddo
\end{aligned}
\tag{1}
$$

The Bisection method has two advantages; first is that a solution is guaranteed as long as f(x) is continuous. Secondly, the error in the solution can be estimated by $|x_n - x| \leq \frac{|x_n + x_{n-1}|}{2}$. If the original interval was [a, b], then the error after n iterations is guaranteed to be bounded by $\frac{|a-b|}{2^n}$. The disadvantage is that the bisection method converges relatively slowly compared to methods that use information about f'(x). The bisection method can miss a root within an interval if either the function doesn't cross zero (that is the root is also the local minimum or maximum of the function), or if more than one root are within the interval. The bisection method is often used to get a starting point for methods that converge more quickly.

Newton's Method

Newton's method is also an iterative method. It uses the value of the derivative of $f(x_n)$, $f'(x_n)$, to choose the next guess (x_{n+1}) for the root. It converges more quickly than the bisection method. Recall the Taylor series expansion;

$$f(x_{n+1}) \;=\; f(x_n) \;+\; (x_{n+1} - x_n)f'(x_n) \;+\; \frac{(x_{n+1} - x_n)^2}{2} f''(x_n) \;+\; \cdots. \tag{2}$$

In this case, we want to chose x_{n+1} so that $f(x_{n+1}) \approx 0$, to approximate the root. Rearranging Equation 2 to solve for x_{n+1} such that $f(x_{n+1}) = 0$ gives

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{3}$$

The iterations proceed until $|f(x_{n+1})| < \epsilon$, where ϵ is the tolerance for error.

If you start with an initial guess, x_n, that is *near enough to* a root, Newton's method will converge quickly. It is not guaranteed to converge, however, and situations can occur where it can send iterations far from a root. Using Newton's method requires that you can evaluate the first derivative of f(x). Newton's method requires 2 function evaluations per iteration (f(x) and f'(x)); and this should be taken into account when its efficiency is compared to other methods.

Secant Method

The secant method is similar to Newton's method, except it uses the function values at successive guesses, $f(x_{n-1})$ and $f(x_n)$, to estimate f'(x). Unlike Newton's method, it can be used when the df/dx can not be evaluated analytically. It uses a first-order, forward difference to estimate df/dx:

$$f'(x_n) \;\approx\; \Delta f/\Delta x \;=\; \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \tag{4}$$

This estimate of f'(x) is then used to choose the next guess, x_{n+1}, so that $f(x_{n+1}) \approx 0$.

$$x_{n+1} \;=\; x_n \;-\; f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \tag{5}$$

Like Newton's method, the iterations proceed until $|f(x_{n+1})| < \epsilon$, where ϵ is the tolerance for error.

Except for the fact that the secant method can be used when f'(x) does not have an analytical solution; it has the same pros and cons as Newton's method. It converges quickly provided that the initial guesses (x_0 and x_1) are *close enough* to the root, x. It is not guaranteed to converge, however, and the search algorithm can produce guesses that are outside of the neighborhood of the root.

False Position Method

Also called *linear interpolation*, and *regula falsa*, the false position method blends the secant and bisection methods. Like the secant method, it chooses the next estimate, x_2 based on a the y-intercept of the line that connects the two estimates x_0 and x_1. The estimates x_0 and x_1 have to satisfy $f(x_0) \times f(x_1) < 0$, however (i.e. they bracket a root for a continuous function). The function value of $f(x_2)$ is evaluated, and either x_0 or x_1 is replaced with x_2, so that the root is still bracketed, in a manner similar to the bisection method. This can converge more quickly than the bisection method, but is guaranteed to find a root. In psuedo-code:

$$
\begin{aligned}
& f(a) < 0; \; f(b) > 0; \\
& n = 0 \\
& x_0 = \frac{a+b}{2}; \\
& do\ while\ abs\,(f(x_n)) > \epsilon \\
& if\ f(x_n) < 0; a = x_n \\
& else\ if\ f(x_n) > 0; b = x_n \\
& x_{n+1} \;=\; a - f(a)\frac{b-a}{f(b)-f(a)} \\
& n = n+1 \\
& enddo
\end{aligned}
\tag{6}
$$

Fixed-point Iteration

This is a clever method for iterating to a solution. The problem of "root-finding", i.e. find x such that f(x) = 0, is rewritten to the form $x = g(x)$, for some g(x). There are usually many ways that g(x) can be defined for a given f(x). For example, if $f(x) = ax^3 + bx^2 + cx + d = 0$, then $x = g(x)$ could be written either as $x = -(ax^3 + bx^2 + d)/c$, or $x = \sqrt{-(ax^3 + cx + d)/b}$. Once g(x) is chosen, an initial guess, x_0 is evaluated to give $g(x_0)$. Successive guesses of the root, x_{n+1} are taken to equal $g(x_n)$, until $|x_{n+1} - x_n| < \epsilon$; where ϵ is the tolerance for error.

Fixed-point iteration is guaranteed to converge when $|g'(x)| < 1$ (see below), and will converge monotonically when $0 < g'(x) < 1$. It will converge in an oscillating fashion when $-1 < g'(x) < 0$, and can diverge when $|g'(x)| > 1$.

Hornberger and Wiberg's notes contain an example of using fixed-point iteration to solve for wavelength in intermediate depths.

Convergence of Iterative Methods

The convergence rate of an iterative method refers to the rate at which successive estimates approach the true answer. If we define the true root of f(x)=0 to be $x = x_r$, then the error in the n^{th} estimate is $x_n - x_r = e_n$. The convergence rate is defined to be p such that $e_{n+1} \approx K(e_n)^p$, where K is some constant. If $p \approx 1$, the method is said to be *linear*, if $p \approx 2$, the method is said to have *quadratic* convergence, etc.

Gerald and Wheatley provide examples of how to analyze convergence rates of the fixed-point iteration, newton's, and secant methods.

Convergence of Fixed-point Iteration

Remember that for fixed-point iteration; we take as our $n + 1^{th}$ guess the evaluation of $g(x_n)$: $x_{n+1} = g(x_n)$. For the true root, $f(x_r) = 0$ and $x_r = g(x_r)$. The error, $e_{n+1} = x_{n+1} - x_r$, and this can be rewritten:

$$\begin{aligned}
e_{n+1} &= x_{n+1} - x_r \\
&= g(x_n) - g(x_r) \\
&= \frac{g(x_n) - g(x_r)}{x_n - x_r}(x_n - x_r) \\
&= \frac{g(x_n) - g(x_r)}{x_n - x_r} e_n. \\
&= g'(\zeta) e_n
\end{aligned} \tag{7}$$

If both g(x) and g'(x) are continuous near the root, there must be some value ζ such that $e_{n+1} = |g'(\zeta)||e_n|$. This explains why the method converges if $|g'(x)| < 1$, and can diverge if $|g'(x)| > 1$. The order of convergence will depend on the properties of $g'(x)$ near the root.

Convergence of Newton's Method

We can write Newton's method as a fixed-point iteration:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = g(x_n). \tag{8}$$

From the previous discussion, we know that we need to evaluate $g'(x_n)$ to analyze the convergence; $e_{n+1} = |g'(\zeta)||e_n|$. For $g(x) = x - \frac{f(x)}{f'(x)}$, we get

$$g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}. \tag{9}$$

So, if $|\frac{f(x)f''(x)}{(f'(x))^2}| < 1$, Newton's method will converge.

We get additional insight from evaluating the behavior of g(x) near the root using a Taylor series expansion:

$$g(x_n) = g(x_r) + g'(x_r)(x_n - x_r) + g''(\zeta)\frac{(x_n - x_r)^2}{2}. \tag{10}$$

From equation 9, we know that $g'(x_r) = \frac{f(x_r)f''(x_r)}{(f'(x_r))^2}$, but because $f(x_r) = 0$, $g'(x_r) = 0$. That means that we can simplify equation 10 to be

$$\begin{aligned} g(x_n) &= g(x_r) + g''(\zeta)\frac{(x_n - x_r)^2}{2} \\ g(x_n) - g(x_r) &= g''(\zeta)\frac{(x_n - x_r)^2}{2} \\ x_{n+1} - x_r &= \frac{g''(\zeta)}{2}e_n^2 \\ e_{n+1} &= \frac{g''(\zeta)}{2}e_n^2. \end{aligned} \tag{11}$$

Newton's method is therefore *quadratic*, the error in each iteration is proportional to the square of the error in the previous step. This means that the number of significant digits in the estimate is doubled with each iteration.