Lecture 6

Secant Methods*

In this lecture we introduce two additional methods to find numerical solutions of the equation $f(x) = 0$. Both of these methods are based on approximating the function by secant lines just as Newton's method was based on approximating the function by tangent lines.

The Secant Method

The secant method requires two initial points x_0 and x_1 which are both reasonably close to the solution x^*. Preferably the signs of $y_0 = f(x_0)$ and $y_1 = f(x_1)$ should be different. Once x_0 and x_1 are determined the method proceeds by the following formula:

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{y_i - y_{i-1}} y_i \tag{6.1}$$

Example: Suppose $f(x) = x^4 - 5$ for which the true solution is $x^* = \sqrt[4]{5}$. Plotting this function reveals that the solution is at about 1.25. If we let $x_0 = 1$ and $x_1 = 2$ then we know that the root is in between x_0 and x_1. Next we have that $y_0 = f(1) = -4$ and $y_1 = f(2) = 11$. We may then calculate x_2 from the formula (6.1):

$$x_2 = 2 - \frac{2-1}{11-(-4)} 11 = \frac{19}{15} \approx 1.2666....$$

Pluggin $x_2 = 19/15$ into $f(x)$ we obtain $y_2 = f(19/15) \approx -2.425758....$ In the next step we would use $x_1 = 2$ and $x_2 = 19/15$ in the formula (6.1) to find x_3 and so on.

Below is a program for the secant method. Notice that it requires two input guesses x_0 and x_1, but it does not require the derivative to be input.

figure yet to be drawn, alas

Figure 6.1: The secant method.

```
function x = mysecant(f,x0,x1,n)
format long                    % prints more digits
format compact                 % makes the output more compact
% Solves f(x) = 0 by doing n steps of the secant method starting with x0 and x1.
% Inputs: f -- the function, input as an inline function
%         x0 -- starting guess, a number
%         x1 -- second starting geuss
%         n -- the number of steps to do
% Output: x -- the approximate solution
y0 = f(x0);
y1 = f(x1);
for i = 1:n              % Do n times
    x = x1 - (x1-x0)*y1/(y1-y0)    % secant formula.
    y=f(x)                         % y value at the new approximate solution.
% Move numbers to get ready for the next step
    x0=x1;
    y0=y1;
    x1=x;
    y1=y;
end
```

The *Regula Falsi* Method

The *Regula Falsi* method is somewhat a combination of the secant method and bisection method. The idea is to use secant lines to approximate $f(x)$, but choose how to update using the sign of $f(x_n)$.

Just as in the bisection method, we begin with a and b for which $f(a)$ and $f(b)$ have different signs. Then let:

$$x = b - \frac{b-a}{f(b)-f(a)} f(b).$$

Next check the sign of $f(x)$. If it is the same as the sign of $f(a)$ then x becomes the new a. Otherwise let $b = x$.

Convergence

If we can begin with a good choice x_0, then Newton's method will converge to x^* rapidly. The secant method is a little slower than Newton's method and the *Regula Falsi* method is slightly slower than that. Both however are still much faster than the bisection method.

If we do not have a good starting point or interval, then the secant method, just like Newton's method can fail altogether. The *Regula Falsi* method, just like the bisection method always works because it keeps the solution inside a definite interval.

Simulations and Experiments

Although Newton's method converges faster than any other method, there are contexts when it is not convenient, or even impossible. One obvious situation is when it is difficult to calculate a formula for $f'(x)$ even though one knows the formula for $f(x)$. This is often the case when $f(x)$ is not defined explicitly, but implicitly. There are other situations, which are very common in engineering and science, where even a formula for $f(x)$ is not known. This happens when $f(x)$ is the result of experiment or simulation rather than a formula. In such situations, the secant method is usually the best choice.

Exercises

6.1 Use the program `mysecant` on $f(x) = x^3 - 4$. Compare the results with those of Exercises 3.3 and 5.3. Then modify the program to accept a tolerance using a `while` loop. Test the program and turn in the program as well as the work above.

6.2 Write a program `myregfalsi` based on `mybisect`. Use it on $f(x) = x^3 - 4$. Compare the results with those of Exercises 3.3 and 5.3.