Methods for finding roots of functions. Implied Volatility.

1 Introduction.

This document addresses the following problem. Assume that we have a function $f : \mathbb{R} \to \mathbb{R}$ which perhaps has a very complicated form. We wish to look at the equation $f(x) = 0$. If f would be invertible it would be easy to find the roots as the set $\{x : x = f^{-1}(0)\}$. We also assume that we can calculate fast $f(x)$ for all x.

More specifically, this is needed in the context when f is the value given by the Black Scholes formula. Suppose that we look at a Call option whose value is denoted by $C_{BS} = C_{BS}(S_0, K, T, r, \sigma)$. Note that S_0 and r are values that are known at the current time (stock value and the short term interest rate). K and T are strike price and maturity (in years) which characterize each option. Therefore the unknown in this context is the volatility σ. What one does is: observe the bid-ask spread, (denoted B and A) for the specific option value (characterized by K, T above) from the market.

Then one considers the function

$$f(x) = C_{BS}(S_0, K, T, r, x) - (B + A)/2.$$

The value x for which the above function is zero is called the volatility implied by the marked or simply the *implied volatility*[1]. Note that this value is the root of the function f. Thus finding the implied volatility is the same as finding the roots of a function. Bellow I present several methods of finding roots of a function (only the bisection is presented in detail the others just for reference). You may use any method you wish in your calculation[2].

[1] Note that the value $(B+A)/2$ is just a convention, one could also use them separately and obtain two implied volatility values and infer that the real implied volatility value is somewhere inside the interval.

[2] Even the stupid shooting method presented in Hull's book

2 The Bisection method.

The bisection method is a very simple method. It is based on the fact that if the function f has a single root at x_0 then the function changes the sign from negative to positive. Suppose there exist a neighborhood of x_0 which contains only that one root x_0. Take any 2 points a and b in the neighborhood. Then since the function changes sign at x_0 if x_0 is between a and b the values $f(a)$ and $f(b)$ will have opposite signs. Therefore, the product $f(a)f(b)$ will be negative. However, if x_0 is not between a and b then $f(a)$ and $f(b)$ will have the same sign (be it negative or positive) and the product $f(a)f(b)$ will be positive. This is the main idea of the bisection method.

The algorithm starts by finding two points a and b such that the product $f(a)f(b)$ is negative. In the case of implied volatility the suggested starting points are 0 and 1.

Pseudo-Algorithm:

Step 1: Check if the distance between a and b (i.e., $b-a$) is less than some tolerance level ε. If Yes STOP report either point you want (a or b). If No step further.

Step 2: Calculate $f(\frac{a+b}{2})$. Evaluate $f(a)f(\frac{a+b}{2})$ and $f(\frac{a+b}{2})f(b)$

Step 3: If $f(a)f(\frac{a+b}{2}) < 0$ make $b \leftarrow \frac{a+b}{2}$. Go to Step 1

Step 4: If $f(\frac{a+b}{2})f(b) < 0$ make $a \leftarrow \frac{a+b}{2}$. Go to Step 1

Think about what this algorithm does. Obviously if there are more than one root in the starting interval $[a, b]$ or the root is a multiple root, the algorithm fails.

There are many improvements but this is not a course about that. However, I will mention a few. (Some of these ideas are from the Wikipedia web site).

Newton method. This method assumes the function f has a continuous derivative. It also assumes that the derivative $f'(x)$ can be calculated easily (fast) at any point x. Newton's method may not converge if you start too far away from a root. However, if it does converge, it is faster than the bisection

method. Newton's method is also important because it readily generalizes to higher-dimensional problems[3].

More specifically, suppose that we have an approximation for the root as x_n. Then we can get the next approximating value for the root using a simple tangent argument (omitted here):

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

This formula works in any situation, however if the starting point x_0 is far away from the root it may take a long time (or not at all) to converge.

Another issue with this method is the derivative. Notice that in (1) we need to know the derivative value at x_n to calculate the next point. In most cases we cannot compute this value so instead we need to estimate it. One could use finite difference methods (explained later in this class) for the estimation. This provides the next method called the Secant method.

Secant Method. For this method just replace the derivative in (1) like this:

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

Note that this implies that we have to pick two starting points x_0 and x_1.

Muller Method. This is similar with the secant method except that one uses a second order approximation (using second derivative). The formulas are a bit less elegant than the few presented so far so we omit them.

Inverse quadratic interpolation. Uses second order approximation for the inverse function f^{-1}. Once again the formulas are omitted.

Brent's method Finally, the most powerful deterministic algorithm combines three of the methods presented thus far to find the root faster than all three. The three basic algorithms used are bisection, secant and inverse quadratic interpolation methods.

[3]The bisection method does not generalize.

Besides these, there exist new *random* algorithms where perturbations are introduced for a more certain convergence. These are better known for finding extreme points of functions (the best known one is called simulated annealing), but there are some used to find roots of functions as well.

At Stevens there is an entire program in this area (optimization and calculating things faster).