Lecture 4: Solving Equations: Newton's Method, Bisection, and the Secant Method

Instructor: Professor Amos Ron Scribes: Yunpeng Li, Mark Cowlishaw, Nathanael Fillmore

1 Review of Fixed Point Iterations

In our last lecture we discussed solving equations in one variable. Such an equation can always be written in the form:

$$f(x) = 0 \tag{1}$$

To find numerically a solution r for equation (1), we discussed the method of fixed point iterations. In this method, we rewrite (1) in the form:

$$x = g(x) \tag{2}$$

and use equation (2) to define a sequence $x_0, x_1, x_2, \ldots$, with x_0 an initial approximation of the solution, and the rest of the sequence generated using:

$$x_{n+1} := g(x_n). \tag{3}$$

If we choose a good transformation to equation (2), then the sequence $x_0, x_1, x_2, \ldots$ will converge to the actual solution, r. At the end of the last lecture, we discussed methods to test whether a particular function g will produce a sequence $x_0, x_1, x_2, \ldots$ that converges to r. The test hinges on the value of the derivative of g in the area close to the solution r. Specifically, if we define an interval I around the initial value x_0

$$I = [x_0 - \delta, x_0 + \delta], \tag{4}$$

and we know that the actual solution $r \in I$, then the sequence $x_0, x_1, x_2, \ldots$ will converge to r if for all values $c \in I$ $|g'(c)| \leq \lambda < 1$ and if all entries of the sequence $x_0, x_1, x_2, \ldots$ remain in the interval I. We can make the following two claims, for positive and negative values of the derivative of g:

CLAIM 1.1 (Convergence for Positive Slope). *If $0 \leq g'(c) \leq \lambda < 1$, $\forall c \in I$, and if $r \in I$, then $(x_j)_{j=0}^{\infty}$ converges to r.*

CLAIM 1.2 (Convergence for Negative Slope). *If we only know that $|g'(c)| \leq \lambda < 1$, $\forall c \in I$, and that $r \in I$, then it is possible that the approximations x_j will not stay in the interval, invalidating thereby our error analysis. However, if we require that the condition $|g'(c)| \leq \lambda < 1$ occurs in the larger interval $I' := [x_0 - 2\delta, x_0 + 2\delta]$ (and retain the assumption that r lies in the smaller interval I), then convergence is guaranteed.*

Furthermore, we can distinguish a bad choice of function g:

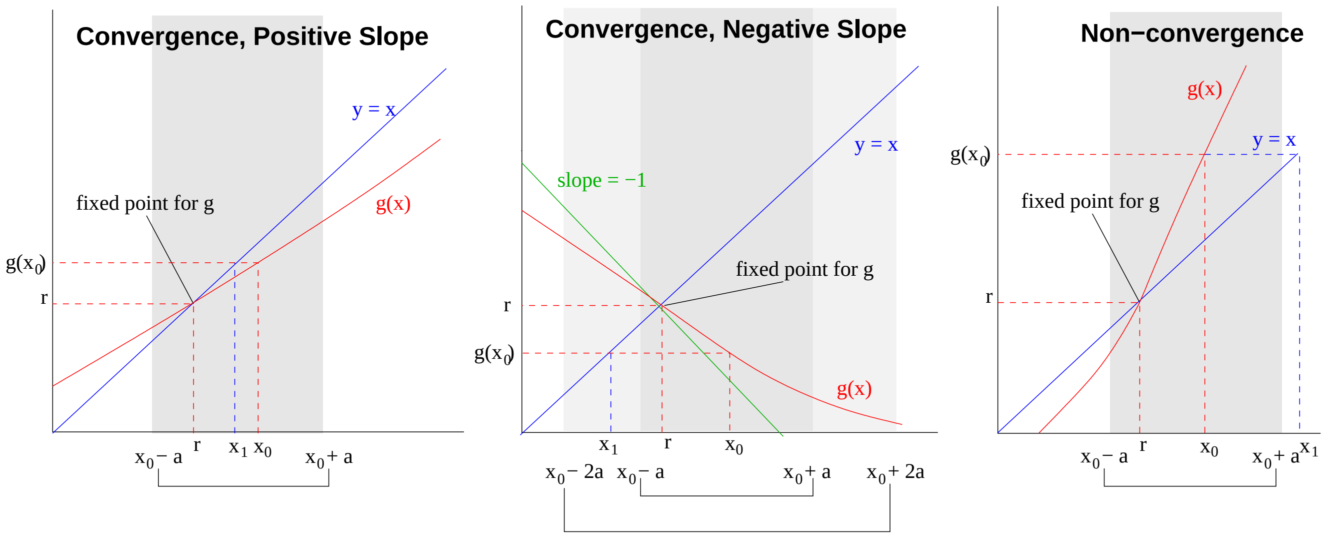

Figure 1: Examples of Convergence and Non-convergence to Fixed Point r

Claim 1.3 (Non-convergence). *If* $|g'(r)| > 1$ *then the sequence* $x_0, x_1, x_2, \ldots$ *will* never *converge to* r.

Examples of each of these possibilities for the slope of g are shown in Figure 1

This analysis leads to the following general procedure for using fixed point iterations:

1. Find an interval I that you know to contain the solution r to the equation.

2. Transform the equation to the form $x = g(x)$. Estimate $g'(x)$ over the interval I.

3. If $g'(x)$ is small ($\ll 1$) and is non-negative over the interval, then use g. Take x_0 to be the center of I.

4. If $|g'(x)|$ is small ($\ll 1$) over the interval, but negative somewhere on the interval, define a new interval $\tilde{I}$ that is double the size of the original interval/ If $|g'(x)|$ is small ($\ll 1$) over this new interval, then use g.

5. Otherwise, choose a new transformation $x = g(x)$ and begin again.

Thus, if we try to solve $x^2 - 5 = 0$, and know that $\sqrt{5} \in [2.2, 2.3]$, then, with $x_0 := 2.25$, we need the condition $|g'| < \lambda < 1$ to hold on the interval $[2.15, 2.35]$. You can reduce the size of the latter interval when: (1) g' is positive, (2) you improve your knowledge on $sqrt5$.

Note that it is sometimes impossible to identify an interval that contains the solution to the equation. In these cases, you may choose an interval where you know that your g has a small slope, and hope that the sequence $x_0, x_1, x_2, \ldots$ converges to a solution. This may work in some cases, but there are no guarantees.

2 Newton's Method

Recall that at the end of the last class, we discussed conditions for quadratic convergence of fixed point iterations. The sequence $x_0, x_1, x_2, \ldots$ defined using

$$x_{n+1} = g(x_n)$$

converges quadratically to r if $g'(r) = 0$, where r is a fixed point of g (that is, $g(r) = r$). Recall that the error of an iteration e_n is defined as:

$$e_n = x_n - r \tag{5}$$

and quadratic convergence occurs when the error of each iteration is bounded by the square of the error in the previous iteration:

$$|e_{n+1}| \leq \lambda_n e_n^2 \tag{6}$$

where λ_n is defined using the Taylor series remainder:

$$\lambda_n = \frac{g''(c_n)}{2} \tag{7}$$

and c_n is some value between x_n and r. Note that, unlike the case of linear convergence, λ_n does not have to be small to ensure fast convergence, the square of the previous error will usually dominate. We have previously seen an example of quadratic convergence in our method for finding numerical solutions to quadratic equations.

Example 2.1. *To solve equations of the form*

$$x^2 + bx = c$$

we used the formula

$$x_{\text{new}} = \frac{x_{\text{old}}^2 + c}{2x_{\text{old}} + b}$$

We can now recognize that this is simply an application of fixed point iterations, with:

$$f(x) = x^2 + bx - c = 0 \tag{8}$$

Which has been simply transformed into:

$$x = g(x) = \frac{x^2 + c}{2x + b}$$

If we take the derivative of g, we see that:

$$\begin{aligned} g'(x) &= \frac{(2x+b)(2x) - 2(x^2+c)}{(2x+b)^2} \\ &= \frac{2x^2 + 2bx - 2c}{(2x+b)^2} \\ &= \frac{2f(x)}{[f'(x)]^2}. \end{aligned}$$

Since $f(r) = 0$, $g'(r) = 0$, too. Note that, without any particular knowledge about the locations of the roots of f, g was nevertheless carefully constructed to have derivative 0 at the roots of f:

$$f(r) = 0 \quad \Longrightarrow \quad g'(r) = 0.$$

This function was constructed using a technique called *Newton's Method*. In Newton's method, given a function

$$f(x) = 0,$$

we construct the function g as follows:

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

For example, remember our method for finding the square root of 5.

EXAMPLE 2.2. *To find the square root of* 5*, we use the quadratic equation* $x^2 = 5$*, or:*

$$f(x) = x^2 - 5 = 0$$

Using Newton's method, we construct a function $g(x)$*:*

$$x = g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - 5}{2x} = \frac{x^2 + 5}{2x}$$

Recall that the sequence $x_0, x_1, x_2, \ldots$ *defined by* g *converged very quickly to the square root.*

2.1 Details of Newton's Method

We must show that Newton's Method produces a valid transformation of $f(x) = 0$ and exhibits quadratic convergence (for most functions) to the solution.

1. The equation $x = g(x)$ defined by Newton's method is equivalent to the original equation $f(x) = 0$:

   This is elementary algebra, since the equation

   $$x = x - \frac{f(x)}{f'(x)}$$

   can be easily transformed into $f(x) = 0$, by simply subtracting x from both sides, then multiplying both sides by $f'(x)$.

2. Newton's Method converges quadratically to the solution r of $f(x) = 0$:

   To show this, simply compute g':

   $$\begin{aligned} g'(x) &= \frac{d}{dx}\left(x - \frac{f(x)}{f'(x)}\right) \\ &= 1 - \frac{f'(x) \cdot f'(x) - f(x) \cdot f''(x)}{[f'(x)]^2} \\ &= 1 - 1 + \frac{f(x) \cdot f''(x)}{[f'(x)]^2} \\ &= f(x) \cdot \frac{f''(x)}{[f'(x)]^2} \end{aligned}$$

and, since $f(r) = 0$, $g'(r) = 0$, which, as we have shown, produces quadratic convergence, provided that $f'(r) \neq 0$.

It is important to note that the above analysis requires g to be twice differentiable (otherwise, our Taylor series expansion, which led to us to quadratic convergence is not valid). Indeed, g'' is not only necessary for our error analysis: it also appears explicitly in the error formula:

$$e_{n+1} = \frac{g''(c_n)}{2} e_n^2 \tag{10}$$

Note that f' appears in g, hence f''' will appear in g''. So, we require the function f to be three times differentiable for the quadratic convergence of Newton's method. Since g is constructed using the derivative of f, this analysis requires f to be 3 *times differentiable* in the region near the analytic solution r.

2.2 Geometric Interpretation of Newton's Method

Newton's method uses a simple idea to provide a powerful tool for fixed point analysis. The idea is that we can use tangent lines to approximate the behavior of f near a root. The method goes as follows. We start with a point x_0, close to a root of f. The line tangent to f at x_0 will intersect the x-axis at some point $(x_1, 0)$. The x-coordinate of this intersection should be closer to the root of f than x_0 was. The process is shown in Figure 2.2.

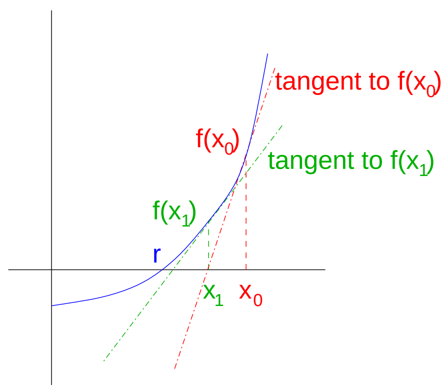

Figure 2: Geometric Interpretation of Newton's Method

This ends our discussion of fixed point iterations. We will now explore some other methods for finding numeric solutions for equations.

3 Bisection

Bisection is a simple method for finding roots that relies on the principle of divide and conquer. The idea is to find an interval containing the root and to split the interval into smaller and smaller sections - in the end, we will have a tiny interval that contains the root, and we may take the midpoint of the interval as our approximation.

To begin, we need to find an interval $[a, b]$ in which the sign of $f(a)$ and $f(b)$ differ. If f is continuous, we know that f must be zero at least once on the interval. An example of this situation is shown in Figure 3.

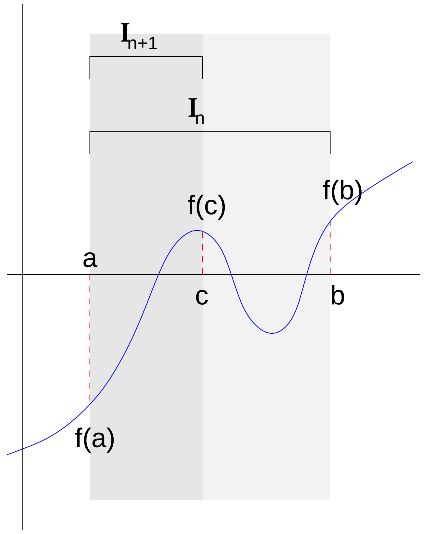

Figure 3: Intervals in the Bisection Method

On each iteration, we calculate the midpoint c of the interval, and examine the sign of $f(c)$. We will then form a new interval with c as an endpoint. If $f(a)$ and $f(c)$ have differing signs, then $[a, c]$ is the new interval, otherwise $[c, b]$ is the new interval. Note that, if $f(c) = 0$, we have just found the root. More formally, the process proceeds as follows:

3.1 The Bisection Process

Given a continuous function f, we will find a root of f $(f(x) = 0)$. At the beginning, we need to find an initial interval $I_0 = [a_0, b_0]$ in which $f(a_0)$ and $f(b_0)$ have opposite signs $(f(a_0) \cdot f(b_0) < 0)$. Then we repeat the following steps for each iteration :

1. Calculate the midpoint c_n:

$$c_n = \frac{a_n + b_n}{2}$$

2. Define the new interval I_{n+1} as:

$$I_{n+1} = \begin{cases} [a_n, c_n], & \text{if } f(a_n) \cdot f(c_n) < 0, \\ [c_n, b_n], & \text{otherwise.} \end{cases} \tag{11}$$

We iterate until achieving a desired level of accuracy. Then we take the midpoint of the last interval as our approximation to the root.

3.2 Order of Convergence

Note that, at each iteration, the size of the interval is halved:

$$|I_{n+1}| = \frac{1}{2} \cdot |I_n|$$

Since we will eventually take the midpoint of the interval as our final approximation, the error at any step is at most half the size of the f we *define* the error to be

$$e_n := |I_n|/2$$

(which is not the actual error, but a truly good way to assess the size of the error), we get:

$$e_{n+1} = \frac{1}{2} \cdot e_n$$

Thus, bisection has linear convergence. This error measure does not compare precisely to the error measure we used for fixed point iterations, but it is close enough for comparison.

3.3 Cost

Bisection requires one evaluation of the function f at each iteration.

3.4 Robustness

What is the robustness of the bisection method? A drawback is that we need to find a "good" initial interval $I_0 = [a, b]$, so that $f(a)f(b) < 0$ (we also require f to be continuous on $[a, b]$, but this is really mild). Good news: once we have such an interval, the algorithm is guaranteed to converge - so it is extremely robust.

Note that we do need to assume that the function f is continuous, but we *always* need to make this assumption, regardless of the root-finding algorithm. Our algorithms always require that the behavior of the function f near a root is similar to its behavior at the root. If this assumption is violated, then we really can't say anything. For example, consider the discontinuous function

$$f(x) = \begin{cases} x & x \notin \{-1, 0\} \\ 0 & x = -1 \\ 1 & x = 0 \end{cases}$$

No root-finding algorithm will be able to find the unique root at $x = -1$.

3.5 Comparison with Fixed Point Iterations

Bisection guarantees linear convergence at a rate of 1/2 for any continuous function and requires only one function evaluation per iteration (we have to evaluate $f(c_n)$ each time, but $f(a_n), f(b_n)$ should not be recomputed). Bisection doesn't even require full function evaluations, it simply requires that we can determine the sign of a function at a particular point.

Why would we use fixed point iterations instead of bisection for a particular function f?

- If f is triply differentiable near the root, we may be able to use Newton's method, which converges quadratically.

- If f is differentiable near the root, we may be able to find a function $g(x)$ that converges linearly at a rate faster than 1/2.

Thus, if we are dealing with functions that are not multiply differentiable, or for which we don't have much information, bisection may be a better method for finding roots than fixed point iterations.

3.6 Comparison with Trisection

We might think that if the bisection method is good, the following trisection method will be even better. Begin with an initial interval $I_0 = [a_0, b_0]$ such that $f(a_0)f(b_0) < 0$. Repeat:

1. Calculate the two 1/3-midpoints:

$$c_n^{(1)} = \frac{2a_n + b_n}{3}, \quad c_n^{(2)} = \frac{a_n + 2b_n)}{3}$$

2. Define the new interval I_{n+1} as:

$$I_{n+1} = \begin{cases} [a_n, c_n^{(1)}] & \text{if } f(a_n) \cdot f(c_n^{(1)}) < 0 \\ [c_n^{(1)}, c_n^{(2)}] & \text{if } f(c_n^{(1)}) \cdot f(c_n^{(2)}) < 0 \\ [c_n^{(2)}, b_n] & \text{if } f(c_n^{(2)}) \cdot f(b_n) < 0 \end{cases}$$

[If more than one case holds, choose between them arbitrarily.]

Here $|I_{n+1}| = \frac{1}{3}|I_n|$, so $e_{n+1} = \frac{1}{3}e_n$. At first glance, this error bound looks better than the error bound we get for bisection, where $e_{n+1} = \frac{1}{2}e_n$. However, trisection has twice as much cost per iteration as bisection does: trisection requires two function evaluations per iteration, while bisection requires only one. So for a cost of two function evaluations, bisection guarantees error reduction of $\frac{1}{4}$, while trisection guarantees error reduction of only $\frac{1}{3}$.

This analysis shows that we need to consider both speed and cost when we evaluate numerical algorithms.

4 Secant Method

The last method we will discuss for finding the roots of equations is the secant method. One can think of the secant method as a poor man's version of Newton's method. Where Newton's method uses the derivative of the function, the secant method uses a numeric estimate of the derivative.

Recall that, in Newton's method, we defined a sequence $x_0, x_1, x_2, \ldots$ using an initial guess x_0 and the formula for its successors:

$$x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)} \tag{12}$$

However, f may not be given in closed form, so that evaluating the derivative may be impossible. Recall that we can estimate the derivative of f at x_n using the familiar formula for the slope of secant lines to a curve:

$$f'(x_n) \approx \frac{f(x_n + h) - f(x_n)}{h}$$

where h is small. But what is a good value for h? In the secant method, we use the difference between consecutive values of x_n for h, so that the approximation of the derivative should improve as the sequence $x_0, x_1, x_2, \ldots$ converges to r.

$$f'(x_n) \approx \frac{f_(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \tag{13}$$

See figure 4 for an illustration. Substituting this estimate for the derivative into equation 12 yields:

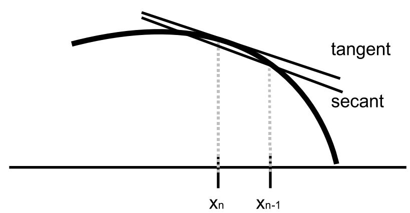

Figure 4: Approximation of tangent by the secant.

$$x_{n+1} := x_n - \frac{f(x_n) \cdot [x_n - x_{n-1}]}{f(x_n) - f(x_n - 1)} \tag{14}$$

Note that this is *not* a fixed point method, since x_{n+1} is based on the values of f at x_{n-1}, while fixed point algorithms do not need even to know the value of x_{n-1}. Also note that only one new function evaluation is required at each step (since previous function values can be stored). Since Newton's method requires an evaluation of the function and its derivative (which is likely at least as difficult as the function evaluation), each iteration of the secant method has roughly half the cost of an iteration of Newton's method.

In the next lecture, we will discuss the convergence of the secant method.