Math128a

Spring 2006

Accelerating Convergence

Patch's Supplementary Notes

1 Overview

Suppose that $x_{k+1} = g(x_k)$ generates a sequence of iterates that converges with order 1 (i.e., slowly) to the fixed point α. We want to approximate α by one of these iterates, but because the convergence is slow, we'll have to do a lot of work (that is, perform a lot of iterations) to get an accurate approximation. Given only $g(x)$ and $g'(x)$, we'd like to construct a new function $h(x)$ with iterates that converge faster to α. We'll outline three ways of seeing that such an $h(x)$ is given by

$$h(x) = \frac{g(x) - xg'(x)}{1 - g'(x)} \tag{1}$$

2 Using Geometry

Consider the two plots of $g(x)$ given in Figure 1. The usual fixed point iteration using $x_{k+1} = g(x_k)$ is shown at left, while at right, information about the slope of g is used to obvious advantage. Instead of moving horizontally, we move along tangent lines to g until we hit the line $y = x$, analogously to how roots are found using Newton's method[1]. If x_k is the x-coordinate of $\tilde{a}$ and x_{k+1} is the x-coordinate of $\tilde{b}$, then the slope

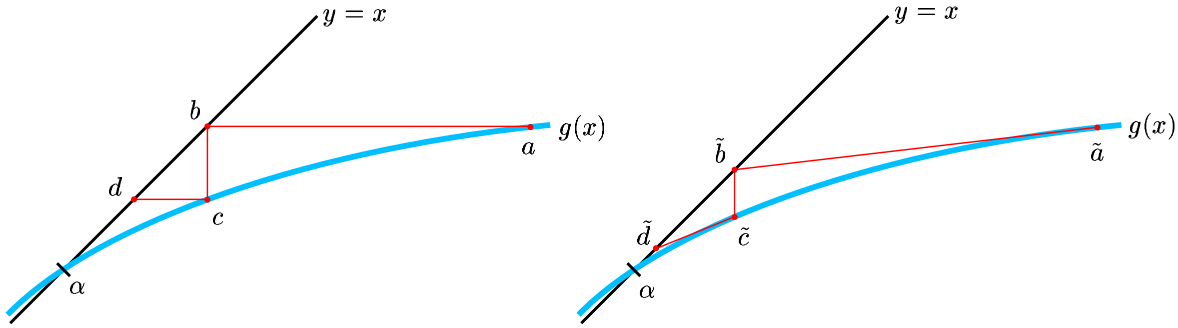

Figure 1: Two different ways of moving towards α.

$g'(x_k)$ of the line connecting $\tilde{a}$ to $\tilde{b}$ is its rise $g(x_k) - x_{k+1}$ over its run $x_k - x_{k+1}$, giving

$$g'(x_k) = \frac{g(x_k) - x_{k+1}}{x_k - x_{k+1}} \tag{2}$$

Solving for x_{k+1} we obtain

$$x_{k+1} = \frac{g(x_k) - x_k g'(x_k)}{1 - g'(x_k)} \tag{3}$$

Figure 1 suggests that this new rule for finding x_{k+1} gives faster convergence than $x_{k+1} = g(x_k)$.

[1] Indeed, note that $h(x)$ constructed using (1) from $g(x) = x - f(x)$ is exactly Newton's method applied to $f(x)$.

3 Linearizing $g(x)$

Our standing assumption is that $g(x)$ generates a sequence of iterates that converges to α. When these iterates get close enough α, $g(x)$ will *look*[2] a lot like its linearization $g(\alpha) + g'(\alpha)(x - \alpha)$, and so $x_{k+1} = g(x_k) \approx g(\alpha) + g'(\alpha)(x_k - \alpha)$. Solving for α, we obtain

$$\alpha \approx \frac{g(x_k) - x_k g'(\alpha)}{1 - g'(\alpha)} \tag{4}$$

If x_k is close to α, then $\epsilon_k = x_k - \alpha$ is small. If g also doesn't have large higher order derivatives, then we can neglect everything except the first term in the following Taylor Series expansion of g' at α

$$g'(x_k) = g'(\alpha + \epsilon_k) = g'(\alpha) + \epsilon_k g''(\alpha) + \frac{\epsilon_k^2}{2} g'''(\alpha) + \cdots \tag{5}$$

That is, we can approximate $g'(\alpha)$ by $g'(x_k)$ in (4), obtaining the following approximation for α

$$\alpha \approx \frac{g(x_k) - x_k g'(x_k)}{1 - g'(x_k)} \tag{6}$$

This result motivates us to use the right hand side of (6) as a new rule for finding x_{k+1}.

4 A Telescoping Sum

Before beginning, we recall that $g(x)$ is assumed to generate iterates that converge with order 1 to α. Starting at x_k, we use $x_{k+1} = g(x_k)$ to produce n new iterates. Observe that

$$x_{k+n} = x_k + (x_{k+1} - x_k) + (x_{k+2} - x_{k+1}) + \cdots + (x_{k+n} - x_{k+n-1}) \tag{7}$$

and also that

$$(x_{k+1} - x_k) = (x_{k+1} - \alpha + \alpha - x_k) = (\epsilon_{k+1} - \epsilon_k) \tag{8}$$

where $\epsilon_k = x_k - \alpha$, which gives us

$$x_{k+n} = x_k + (\epsilon_{k+1} - \epsilon_k) + (\epsilon_{k+2} - \epsilon_{k+1}) + \cdots + (\epsilon_{k+n} - \epsilon_{k+n-1}) \tag{9}$$

Because the convergence of the iterates to α is first order, $\epsilon_{k+1} \approx \lambda \epsilon_k$ where $\lambda = g'(\alpha) \neq 0$. Noting that $(\epsilon_{k+n} - \epsilon_{k+n-1}) \approx \lambda^{n-1}(\epsilon_{k+1} - \epsilon_k)$, we write (9) as

$$x_{k+n} \approx x_k + (x_{k+1} - x_k)\left(1 + \lambda + \lambda^2 + \cdots + \lambda^{n-1} = \frac{\lambda^n - 1}{\lambda - 1}\right) \tag{10}$$

where we've used (8) as well as the geometric series formula. Now consider the limit as $n \longrightarrow \infty$. We're assuming convergence, and so $x_{k+n} \longrightarrow \alpha$. Also, let's assume that $|\lambda = g'(\alpha)| < 1$. This follows from the sufficient (but unnecessary[3]) condition for the existence of a unique fixed point on $[a, b]$; that g is continuous on $[a, b]$, that $g(x) \in [a, b]$ when $x \in [a, b]$, that g' is continuous on (a, b) and that $|g'| < \gamma$ on (a, b) for some $0 < \gamma < 1$. From $|\lambda = g'(\alpha)| < 1$ it follows that $\lambda^n \longrightarrow 0$ leaving us with

$$\alpha \approx x_k + \frac{g(x_k) - x_k}{g'(\alpha) - 1} = \frac{g(x_k) - x_k g'(\alpha)}{1 - g'(\alpha)} \tag{11}$$

where we've used $x_{k+1} = g(x_k)$ and $\lambda = g'(\alpha)$. As explained in the previous section, we can approximate $g'(\alpha)$ by $g'(x_k)$, thus obtaining the following new rule for finding x_{k+1}

$$x_{k+1} = \frac{g(x_k) - x_k g'(x_k)}{1 - g'(x_k)} \tag{12}$$

[2] I find it satisfying to actually *look* at images of the functions being considered. Without pictures, we are merely solving for α using the first terms in the Taylor Series expansion $g(x_k) = g(\alpha) + g'(\alpha)(x_k - \alpha) + \frac{1}{2}g''(\alpha)(x_k - \alpha)^2 + \cdots$

[3] Consider $g(x)$ given implicitly by $x = g + g^3$, which has a unique fixed point at 0 that all fixed point iterates converge to regardless of where they start. Clearly $g'(0) = 1$.

5 Is Convergence Faster for $h(x)$?

We've seen three ways of motivating the $h(x)$ in (1), and so at this point we suspect that it'll give faster convergence to α than $g(x)$. It turns out that confirming this suspicion is straightforward. First, only a little work is necessary to check that $h(\alpha) = \alpha$, establishing that α is indeed a fixed point of h. A slightly longer calculation shows that $h'(\alpha) = 0$. The Taylor Series expansion for h centered at α is therefore given by

$$h(x) = \alpha + h''(x)\frac{(x-\alpha)^2}{2} + h'''(x)\frac{(x-\alpha)^2}{3!} + \cdots, \qquad (13)$$

and so neglecting higher order terms for the usual reasons, we obtain $(h(x_k) - \alpha) = O((x_k - \alpha)^2)$, which is the second order convergence we were hoping for.

6 A Practical Example

In the table below, we compare the iterations of $g(x) = 1 + x - \frac{x^2}{2}$ to those of $h(x) = \frac{1}{x} + \frac{x}{2}$, which is the function corresponding to g given by (1). Both of these functions have a fixed point at $\sqrt{2}$. Starting with $x_1 = 0.5$ and $y_1 = 0.5$, we find the following:

k	$x_k = g(x_{k-1})$	$x_k - \sqrt{2}$	$y_k = h(y_{k-1})$	$y_k - \sqrt{2}$
2	1.37500000000000	-0.03921356237310	2.25000000000000	0.83578643762690
3	1.42968750000000	0.01547393762690	1.56944444444444	0.15523088207135
4	1.40768432617188	-0.00652923620122	1.42189036381514	0.00767680144205
5	1.41689674509689	0.00268318272380	1.41423428594007	2.072356697824240e-05
6	1.41309855196381	-0.00111501040929	1.41421356252493	1.518369874276004e-10
7	1.41467479318270	0.00046123080961	1.41421356237309	-2.220446049250313e-16
8	1.41402240794944	-0.00019115442365	1.41421356237309	-2.220446049250313e-16
9	1.41429272285787	0.00007916048478	1.41421356237309	-2.220446049250313e-16
10	1.41418076989350	-0.00003279247959	1.41421356237309	-2.220446049250313e-16

Notice that after only 6 iterations, the iterates generated by $y_k = h(y_{k-1})$ stop getting more accurate. This is because Matlab is unable to represent $\sqrt{2}$ with any greater precision! In only 6 steps we've run into the fact that the computer approximates the real line as a collection of discrete points.