LECTURE 29: *Newton's Method*

We have studied two bracketing methods for finding zeros of a function, bisection and *regula falsi*. These methods have certain virtues (most importantly, they always converge), but some exploratory evaluations of f might be necessary to determine an initial bracket. Moreover, while these methods appear to converge fairly quickly, when f is expensive to compute (e.g., requiring solution of a differential equation) or many systems must be solved (e.g., to evaluate $\sqrt{A}$ as a zero of $f(x) = x^2 - A$ for many values of A), every objective function evaluation counts. The next few sections describe algorithms that can converge more quickly that bracketing methods — provided we have a sufficiently good initial estimate of the root.

4.2 *Newton's method*

We begin with Newton's method, the most celebrated root-finding algorithm. The algorithm's motivation resembles *regula falsi*: model f with a line, and estimate the root of f by the root of that line. In *regula falsi*, this line interpolated the function values at either end of the root bracket. Newton's method is based purely on local information at the current solution estimate, x_k. Whereas the bracketing methods only required that f be continuous, Newton's method needs $f \in C^1(\mathbb{R})$; to analyze the algorithm, we will further require $f \in C^2(\mathbb{R})$. This latter condition means that f can be expanded in a Taylor series centered at the approximate root x_k:

$$f(x_*) = f(x_k) + f'(x_k)(x_* - x_k) + \tfrac{1}{2} f''(\xi)(x_* - x_k)^2, \tag{4.4}$$

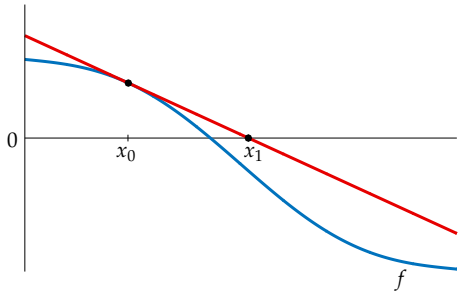
where x_* is the exact solution, $f(x_*) = 0$, and ξ is between x_k and x_*. Ignore the error term in this series, and you have a linear model for f; i.e., $f'(x_k)$ is the slope of the line secant to f at the point x_k. Specifically,

$$0 = f(x_*) \approx f(x_k) + f'(x_k)(x_* - x_k),$$

which implies

$$x_* \approx x_k - \frac{f(x_k)}{f'(x_k)}.$$

We get an iterative method by replacing x_* in this formula with x_{k+1}.



> *Newton's method* updates the approximate root x_k via
>
> $$x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}. \tag{4.5}$$

Newton's method is famous because it often converges very quickly, but that excellent convergence comes at no small cost. For a bad starting guess x_0, it can *diverge* entirely. When it converges, the root it finds can, in some circumstances, depend sensitively on the initial guess: this is a famous source of beautiful fractals. However, for a good x_0, the convergence is usually lightning quick. Let $e_k = x_k - x_*$ be the error at the kth step. Subtract x_* from both sides of the iteration (4.5) to obtain a recurrence for the error,

$$e_{k+1} = e_k - \frac{f(x_k)}{f'(x_k)}. \tag{4.6}$$

The Taylor expansion of $f(x_*)$ about the point x_k given in (4.4) gives

$$0 = f(x_k) - f'(x_k)e_k + \tfrac{1}{2}f''(\xi)e_k^2.$$

Solving this equation for $f(x_k)$ and substituting that formula into the expression (4.6) for e_{k+1} gives

$$\begin{aligned} e_{k+1} &= e_k - \frac{f'(x_k)e_k + \frac{1}{2}f''(\xi)e_k^2}{f'(x_k)} \\ &= -\frac{f''(\xi)}{2f'(x_k)}e_k^2. \end{aligned}$$

When x_k converges to x_*, $\xi \in [x_*, x_k]$ also converges to x_*. Supposing that x_* is a simple root, so that $f'(x_*) \neq 0$, the above analysis suggests that when x_k is near x_*,

$$|e_{k+1}| \leq C|e_k|^2$$

for constant

$$C = \frac{|f''(x_*)|}{2|f'(x_*)|}$$

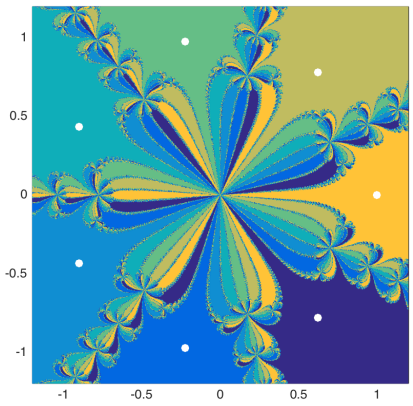
independent of k. Thus we say that if $f'(\star) \neq 0$, then Newton's method *converges quadratically*, roughly meaning each iteration of Newton's method *doubles the number of correct digits* at each iteration. Compare this to bisection, where

$$|e_{k+1}| \leq \tfrac{1}{2}|e_k|,$$

meaning that the error was halved at each step. Significantly, Newton's method will often exhibit a transient period of linear convergence as x_k is gradually improved; once x_k gets close enough to x_*, the behavior transitions to quadratic convergence and full machine precision is attained in just a couple more iterations.

Example 4.7. One way to compute $\sqrt{2}$ is to find the zero of

$$f(x) = x^2 - 2.$$



Newton's method for finding zeros of $f(x) = x^7 - 1$ in the complex plane. This function has seven zeros the complex plane, the principal roots of unity (shown as white dots in the plot). The color encodes the convergence of Newton's method: for each point $x_0 \in \mathbb{C}$, iterate Newton's method until convergence. The color corresponds to which of the seven roots that x_0 converged. In some regions, small changes to x_0 get attracted to vastly different roots.

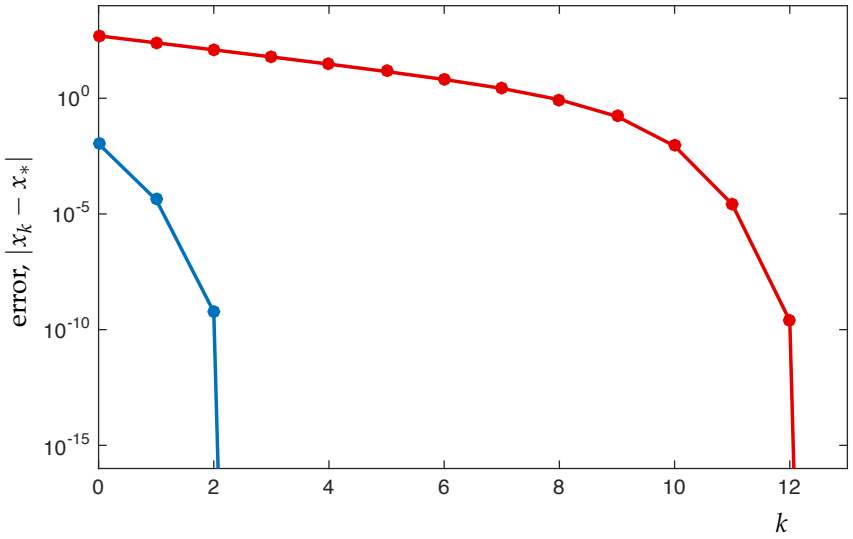


Figure 4.4: The convergence of Newton's method for $f(x) = x^2 - 2$. If x_0 is sufficiently close to the zero $x_* = \sqrt{2}$, the method converges very rapidly. For a poor x_0, the convergence is slow until x_k is sufficiently close. (The situation could be worse: in many cases, bad initial guesses lead to the divergence of Newton's method.)

Figure 4.4 shows convergence in just a few steps for the realistic starting guess $x_0 = 1.25$. The plot also shows the convergence behavior for the (entirely ridiculous) starting guess $x_0 = 1000$, to illustrate a linear phase of convergence as the iterate gradually approaches the region of quadratic convergence. Once x_k is sufficiently close to x_*, convergence proceeds very quickly.

Table 4.1 shows the iterates for $x_0 = 1000$, computed exact arithmetic in Mathematica, and displayed here to more than eighty digits. This is a bit excessive: in the floating point arithmetic we have used

k	x_k
0	1000.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
1	500.0010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
2	250.0024999960000079999840000319999360001279997440005119989760020479959040081919836160 3
3	125.0052499580004679945840630552651285659801482359562239344169580047744668579946389648 4
4	62.5106246430170331488869135840332046452975994432574456663116460063101739147830976134 1
5	31.2713096020621945559642235877170054837456580184233208653636523657827808040615382736 4
6	15.6676329948683664003075552710028165206510015971032445945258154376740347992183401224 8
7	7.8976423478563580671905136093423623811696836517416702511646103416077762821736496011 1
8	4.0754412405194989208879857338706713335299114996130926715933398019154830807536096186 2
9	2.2830928243925538398630669035817794614433923363437778160605553848163720075955537623 6
10	1.5795487524060153652754700172749893512746398177638901618897579136393958626586032325 1
11	1.4228665795786682509120968385630981830931092942876392816289093467384703623818499269 3
12	1.4142398735915306231936461644112003518252948934786012671639574689639269004077455837 5
13	1.4142135626178485126558900035917439663220762854896890824239894439161543633562536005 6
14	1.4142135623730950488228680777571711822141811472942311663725480437703133244040615571 6
15	1.4142135623730950488016887242096980785698304670594999486043964007946076509385830519 0
16	1.4142135623730950488016887242096980785696718753769480731766797379907324784621070477 4
exact	1.4142135623730950488016887242096980785696718753769480731766797379907324784621070388503875 3 . . .

Table 4.1: Convergence of Newton's method for $f(x) = x^2 - 2$ for a poor initial guess. The first 9 iterations each roughly cut he error in half; later iterations then start to double the correct digits (blue, underlined).

all semester, we can only expect to get 15 or 16 digits of accuracy in the best case. It is worth looking at all these digits to get a better appreciation of the quadratic convergence. Once we are in the quadratic regime, notice the characteristic doubling of the number of correct digits (in blue and underlined) at each iteration.