Many equations cannot be solved exactly/algebraically, and our only recourse will be to apply numerical methods. For example, how can we find a solution to the equation

$$x^{23} + e^{\sin(x)} + \ln(x) = 0?$$

The most elementary technique for root finding is *bisection*.

1 Bisection

Recall the intermediate value theorem from calculus:

Theorem 1.1 (Intermediate Value Theorem)**.** *Let $f(x)$ be a continuous function on $[a, b]$. Then for all $y \in [f(a), f(b)]$ (or $[f(b), f(a)]$) there exists $x \in [a, b]$ such that $f(x) = y$.*

We can turn the IVT into a way to solve equations using the following consequence:

Corollary 1.2. *If $f(x)$ is a continuous function on $[a, b]$ and $f(a)f(b) < 0$, then there exists $x \in [a, b]$ such that $f(x) = 0$.*

We can repeatedly apply this idea to home in on the location of a solution to the equation $f(x) = 0$.

1. Initialize $a_0, b_0 \in [a, b]$ such that $f(a_0)f(b_0) < 0$.
2. Let $m_i = \frac{a_{i-1}+b_{i-1}}{2}$. If $f(m_i)f(a_{i-1}) < 0$, then set $a_i = a_{i-1}$ and $b_i = m_i$. Otherwise, if $f(m_i)f(b_{i-1}) < 0$, then set $a_i = m_i$ and $b_i = b_{i-1}$.
3. Repeat step 2 until a desired stopping criterion.

Question 1.3. How should we decide when to stop?

Some options are:

1. $|b_i - a_i|$ sufficiently small
2. $\frac{|b_i-a_i|}{|m_i|}$ sufficiently small
3. $|f(m_i)|$ sufficiently small
4. i sufficiently large
5. $fl(a_i) = fl(b_i)$

There are various considerations when designing a numerical procedure; two of the most important are correctness and efficiency. It's important to know that the algorithm is correct (in the sense that it will eventually produce the correct answer). In addition to correctness, we also want to produce the correct answer as quickly as possible. The bisection algorithm works because of the intermediate value theorem, so now we turn to how quickly it produces a solution.

Question 1.4. Suppose we're given a continuous function $f(x)$ with $f(-1) < 0$ and $f(2) > 0$. If $f(0) = 0$ with no other zeroes on the interval $[-1, 2]$, how many iterations will the bisection algorithm need in order to terminate using floating point numbers? Assume our stopping condition is that $fl(a_i) = fl(b_i)$.

Question 1.5. Given a function $f(x)$ with a zero in $[a, b]$, how can we bound the error of stopping the bisection algorithm after N steps?

The error at iteration i is bounded above by $\frac{1}{2^i}|b-a|$, so this algorithm converges *linearly* in the size of the input. This can be quite slow (up to 1000 steps...this may not seem like much, but if the function is expensive to evaluate, e.g. each function evaluation takes a few minutes, this can turn a 10 minute computation into a 1000 minute computation) – how can we do better? As usual, there are a few choices:

1. Use a better computer (impractical).
2. Improve the bisection algorithm (homework).
3. Come up with a better algorithm (fixed point and Newton).

Example 1.6. Use bisection to find a solution to the equation $x^2 - 3$ starting from the interval $[2, 3]$.

2 Fixed-point iterations

Definition 2.1. A fixed point of a function g is any value x such that $f(x) = x$.

Solving a single variable equation is an equivalent problem to finding fixed points, for example given a function $f(x)$, let $g(x) = x - f(x)$. Then

$$f(x) = 0 \iff g(x) = x.$$

Bisection attempts to find solutions to $f(x) = 0$ by identifying an interval $[a, b]$ containing a solution and then cutting it in half each time to approximate the desired solution. Another iterative approach is *fixed point iteration*:

1. Choose an initial guess x_0.
2. Repeatedly iterate $x_i = g(x_{i-1})$ until desired condition is reached.

The reason for the name fixed point iteration is the following. Suppose $x_i \to x$ (shorthand for $\lim_{i\to\infty} x_i = x$). Then by continuity of g,

$$g(x_i) \to g(x) \implies g(x) = x$$

so the limit x (if it exists) is a fixed point of g. Compared to bisection, fixed point iteration is much more subtle: can we guarantee that fixed point iteration will converge and if so, how quickly will it find that answer?

Example 2.2. Suppose we want to find a solution to $f(x) = x^2 - 3$ using fixed point iteration, considering the following choices of fixed point formulations:

$$g_1(x) = \frac{3}{x} \qquad g_2(x) = x - (x^2 - 3) \qquad g_3(x) = x - \frac{1}{2}(x^2 - 3) \qquad g_4(x) = x - \frac{1}{2x}(x^2 - 3)$$

with an initial guess of $x = 1.5$. What happens in each case? (g_3 converges slowly, g_4 converges quickly, g_1 oscillates forever, and g_2 diverges.)

In bisection, the condition for whether a given interval contains a zero is relatively simple to check: $f(a)f(b) < 0$. On the other hand, it turns out to be more difficult to know whether a function $g(x)$ has a fixed point. Before we begin, recall the *mean value theorem* from calculus:

Theorem 2.3 (Mean Value Theorem)**.** *Let g be a differentiable function on $[a, b]$. There exists $\xi \in [a, b]$ such that*

$$\frac{g(b) - b(a)}{b - a} = g'(\xi).$$

Theorem 2.4. *Let g be a differentiable function on the interval $[a, b]$ such that $g(x) \in [a, b]$ for all $x \in [a, b]$ and $|g'| < 1$ on $[a, b]$. Then for any $x_0 \in [a, b]$, the sequence $x_i = g(x_{i-1})$ converges to the unique fixed point x in $[a, b]$.*

Proof. First, suppose that p, q are two fixed points of g in $[a, b]$. If $p \neq q$, then the mean value theorem says that there is $\xi \in [a, b]$ such that

$$1 = \frac{p - q}{p - q} = \frac{g(q) - g(p)}{q - p} = g'(\xi).$$

contradicting that $|g'| < 1$ on $[a, b]$.

Next, if $g(a) = a$ or $g(b) = b$, then we have our fixed point. If not, then $g(a) > a$ and $g(b) < b$, so if we define $f(x) = g(x) - x$, we have $f(a) = g(a) - a > 0$ and $f(b) = g(b) - b < 0$, so the by the same argument as bisection there is some zero $f(x) = 0$ which is exactly a fixed point of g.

Let x be the unique fixed point. Then $|x_i - x| = |g(x_{i-1}) - g(x)|$ by definition, and furthermore by the mean value theorem

$$\frac{|g(x_{i-1}) - g(x)|}{|x_{i-1} - x|} \leq |g'(\xi_i)| < k \implies |x_i - x| = |g(x_{i-1}) - g(x)| < k|x_{i-1} - x|.$$

Repeated applications of this idea gives

$$|x_i - x| < k^i |x_0 - x|$$

so we obtain convergence since $k < 1$. □

The proof here shows that the speed of convergence depends heavily on the value of the derivative. While in some sense, bisection "always" works, we've just seen that fixed point iteration heavily depends on the choice of starting value and fixed point formulation (in the sense that it may be very difficult to guarantee beforehand that the conditions of theorem 1.8 are satisfied). Next time, we'll see a good way to choose the fixed point formula known as *Newton's method*.

3 Newton's method

As we saw, the error of fixed point iteration is controlled by the size of $|g'|$ near the solution. We can try to take advantage of this knowledge by choosing a fixed point formulation for $f(x) = 0$ whose derivative is very small, especially near its fixed point. One way to do this is by setting

$$g(x) = x - h(x)f(x)$$

and then differentiating and solving for $h(x)$.

Question 3.1. What should we choose for $h(x)$ above?

A better approach is to use Taylor's theorem to approximate $f(x) = 0$ by $f(x_n)$ where x_n is "sufficiently close" to x. For this, we'll need to assume that $f(x)$ is twice-differentiable. Then its first order Taylor approximation near x_n with error term is

$$0 = f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(\xi_n)}{2}(x - x_n)^2.$$

Assuming $x - x_n$ is sufficiently small, then $(x - x_n)^2$ will be even smaller, and we get a good approximation by dropping the error term:

$$0 = f(x) \approx f(x_n) + f'(x_n)(x - x_n) \implies x \approx x_n - \frac{f(x_n)}{f'(x_n)}.$$

Using this approximation, we get the fixed point formulation

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

which we then use to perform fixed point iteration as usual.

Question 3.2. How can we think of Newton's method as linear approximation?

The derivation described above assumed that $(x_n - x)$ was "sufficiently small" so that we could be justified in dropping the error term. Let's analyze how well this choice of fixed point formulation works.

Theorem 3.3. *Let f be twice differentiable with $f(x) = 0$ and $f'(x) \neq 0$. Then there is $\delta > 0$ such that Newton's method converges for any initial choice $x_0 \in [x - \delta, x + \delta]$.*

This proof is an exercise in carefully applying calculus definitions, so I'll skip it. In order to analyze the effectiveness of Newton's method, we first introduce a useful concept in error analysis.

Definition 3.4. Suppose $(x_n) \to x$ with $x_n \neq x$. If $\lambda, \alpha > 0$ such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - x|}{|x_n - x|^\alpha} = \lambda$$

then (x_n) is said to converge to x with order α and asymptotic error constant λ. Two special cases are linear convergence ($\alpha = 1$ and $\lambda < 1$) and quadratic convergence ($\alpha = 2$).

Question 3.5. Compare linearly and quadratically convergent sequences both with an asymptotic error constant of .5, supposing both sequences have a starting distance of 1 from the desired value.

For the linearly convergent sequence, we'll get $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$. For the quadratically convergent sequence, we'll get $\frac{1}{2}, \frac{1}{8}, \frac{1}{2^7}, \ldots$. This means that quadratic convergence (or higher order methods) can achieve much quicker convergence given the correct starting conditions. Thus, it's really worth it to look for higher order convergence of numerical methods whenever we can. Conversely, if we start with a point too far away, then the higher order methods will not work. Thus, higher order methods are like high risk high reward.

We saw before that when $0 < g'(x) < 1$, fixed point iteration converges linearly with asymptotic error constant $\max_x |g'(x)|$. If we refine our techniques, we can show that Newton's method converges quadratically in good situations. However, note that we need to be careful with quadratic convergence: it's only good if we start "close enough" to the answer – it can fail catastrophically otherwise.

From the fixed point perspective, we have chosen a function with $g'(x) = 0$. Since f is twice differentiable, g' will be continuous, and hence also small near x, unless $f'(x) = 0$ as well. Using this idea, we'll formulate an error bound for Newton's method below.

Theorem 3.6. *Let x be a fixed point of g with $g'(x) = 0$, g'' continuous, and $|g''| < M$ on an open interval containing x. Then there is δ such that for any $x_0 \in [x - \delta, x + \delta]$, Newton's method converges quadratically with asymptotic error constant at most $\frac{M}{2}$.*

Proof. Similarly to before, we investigate the Newton iterations $g(x_n)$ using a first order Taylor approximation near x with error term.

$$g(x_n) = g(x) + g'(x)(x_n - x) + \frac{g''(\xi_n)}{2}(x_n - x)^2$$

for some ξ_n between x_n and x. Our assumptions were that $g(x) = x$ and $g'(x) = 0$, so this simplifies to

$$x_{n+1} = g(x_n) = x + \frac{g''(\xi_n)}{2}(x - x_n)^2.$$

We know from our previous theorem that Newton iterations converge under these assumptions, meaning that $x_n \to x$. In this case, since ξ_n lies between x_n and x, then ξ_n also converges to x, so for n large enough, we get that $\xi_n \in I$, so rearranging the above gives

$$\frac{|x_{n+1} - x|}{|x - x_n|^2} = \frac{|g''(\xi_n)|}{2} \leq \frac{M}{2}$$

for n sufficiently large. □

The conditions required for Newton's method to guarantee success can be fairly difficult to verify. In practice, it may be easier to "shoot first and ask questions later." It's also a great example of a situation where making more assumptions (in this case, having a second derivative) helps us get better results.

Question 3.7. What goes wrong if $f(x)$ has a repeated root at x?

3.1 Higher dimension Newton's method

Fixed point iteration in general, and in particular Newton's method, has a nice generalization to "square" systems of equations. (The reason we need to work with square systems is that these are precisely the systems where we can expect to have finitely many solutions.) Consider the system of equations

$$f_i(x_1, \ldots, x_m) = 0 \qquad 1 \le i \le m$$

which we'll write in vector notation (using boldface to denote vectors) as

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1, \ldots, x_m) \\ \vdots \\ f_m(x_1, \ldots, x_m) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}.$$

Although it takes a bit of work to verify the details, Newton's method for higher dimensions is given by the iteration formula

$$\mathbf{x_{n+1}} = \mathbf{x_n} - (D_f(\mathbf{x_n}))^{-1}\mathbf{f}(\mathbf{x_n})$$

where $D_f(\mathbf{x_n})$ denotes the *Jacobian matrix* of $\mathbf{f}$ evaluated at $\mathbf{x_n}$

$$(D_f(\mathbf{x}))_{i,j} = \frac{\partial f_i}{\partial x_j}(x_1, \ldots, x_n).$$

This formula is entirely analogous to the one-dimensional case of Newton's method (and if $m = 1$ they're exactly the same).

4 Generalization of Newton's method

Even though Newton's method works very well when it does work, it's quite sensitive to a few factors.

1. Choice of initial value x_0 – perform a few iterations of another algorithm such as bisection to get a good starting point for Newton's method.

2. Requires evaluation of first derivative – if this is too expensive, consider secant approximation of first derivative.

3. Requires that $f'(x) \neq 0$, i.e. we only have simple zeroes – can modify functional form of fixed point formulation to handle this.

4. There are ways to accelerate convergence of otherwise linearly convergent sequences. (The idea of these is to leverage the existing computations in a clever way to accelerate convergence rather than computing new things.)

5. For polynomials in particular, the derivatives are particularly nice (they are also polynomials) so there are specialized methods to determine zeroes of polynomials based on Newton's method.

While many of these modifications are for specialized situations, one of them deserves special mention. Often, evaluation of the derivative of a given function $f(x)$ may be difficult or even impossible. For example, $f(x)$ may involve solving a system of linear equations or a differential equation depending on x and thus may not yield a nice formula description which we can differentiate. In these cases, the best we will be able to do is try to come up with some approximation of the derivative, and the simplest approximation for the derivative is the secant approximation:

$$f'(x) \approx \frac{f(b) - f(a)}{b - a}$$

for a sufficiently small interval (a, b) containing x. Note the similarity of this to the mean value theorem. The success of the secant variation of Newton's method depends heavily on the validity of this approximation, but in practice this can be a very useful variation of Newton's method when the derivative is difficult to evaluate.