Chapter 4

Nonlinear equations

4.1 Root finding

Consider the problem of solving any nonlinear relation $g(x) = h(x)$ in the real variable x. We rephrase this problem as one of finding the zero (root) of a function, here $f(x) = g(x) - h(x)$. The minimal assumption we need on f, g, h is that they're continuous.

We have at our disposal any number of evaluations of f and its derivative f'.

1. **Method 1: bisection**. The bisection methods starts from two points a_0 and b_0 such that

$$f(a_0) > 0, \qquad \text{and} \qquad f(b_0) < 0.$$

   Because f is continous, there must exist a root in the interval $[a_0, b_0]$. At stage k, assume that we have obtained an interval $[a_k, b_k]$ such that the same sign properties hold: $f(a_k) > 0$ and $f(b_k) < 0$. The bisection method consists in subdividing the interval $[a_k, b_k]$ in two and discard the half in which there may not be a root. Let $m_k = (a_k + b_k)/2$.

   - If $f(m_k) < 0$, then it is the interval $[a_k, m_k]$ which is of interest. We put $a_{k+1} = a_k$ and $b_{k+1} = m_k$.
   - If $f(m_k) > 0$, then it is the interval $[m_k, b_k]$ which is of interest. We put $a_{k+1} = m_k$ and $b_{k+1} = b_k$.
   - If $f(m_k) = 0$, then m_k is a root and we stop the algorithm.

In practice, this iteration is stopped once $f(m_k)$ gets small enough. Let x^* be the unknown root. The error obeys

$$|x^* - m_k| \leq |b_k - a_k| = 2^{-k}|b_0 - a_0|.$$

Every step of the bisection discovers a new correct digit in the binary expansion of x^*.

The advantage of the bisection method is that it is guaranteed to converge to a root, by construction. On the other hand, convergence is rather show compared to the next 2 methods we now present. If there are several roots, the bisection method will converge toward one of them (we may not have no control over which root the method chooses.)

2. **Method 2: Newton-Raphson**. This method is very important: it is the basis of most optimization solvers in science and engineering. Let us first present the Newton-Raphson method for solving a single scalar equation $f(x) = 0$.

   Newton's method fits a tangent line to the point $(x_n, f(x_n))$ on the graph of f, and defines x_{n+1} at the intersection of this tangent line with the x axis. We have

   $$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n),$$

   from which we isolate

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

   For instance, we can find the decimal expansion of $\sqrt{2}$ by finding the positive root of $f(x) = x^2 - 2$. The iteration reads

   $$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{x_n}{2} + \frac{1}{x_n}.$$

   Starting with $x_0 = 1$, we get $x_1 = \frac{3}{2} = 1.5$, $x_2 = \frac{17}{12} = 1.4167...$, $x_3 = \frac{577}{408} = 1.4142157...$ The true value of $\sqrt{2}$ is 1.4142135...

   Convergence is very fast, when it occurs. Assume that f'' is continuous, and that $f'(x) \neq 0$ in some neighborhood of the root x^* (large enough so that all our iterates stay in this neighborhood.) Put $\epsilon_n = x_n - x^*$.

Then we can perform a Taylor expansion of f around x_n, and evaluate it at $x = x^*$:

$$0 = f(x^*) = f(x_n) + (x^* - x_n)f'(x_n) + \frac{1}{2}(x^* - x_n)^2 f''(\xi),$$

for some $\xi \in \text{int}(x_n, x^*)$ (the notation int refers to the interval generated by x_n and x^*, i.e., either $[x_n, x^*]$ or $[x^*, x_n]$.) We also have the equation defining x_{n+1}:

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n).$$

Subtracting those 2 equations, we get

$$0 = -\epsilon_{n+1} f'(x_n) + \frac{1}{2}\epsilon_n^2 f''(\xi),$$

$$\epsilon_{n+1} = \frac{1}{2}\frac{f''(\xi)}{f'(x_n)}\epsilon_n^2.$$

Our assumptions ensure that the ratio $\frac{f''(\xi)}{f'(x_n)}$ exists and converges to some limit $(f''(x^*)/f'(x^*))$ as $n \to \infty$. Hence the sequence is bounded uniformly in n, and we can write

$$|\epsilon_{n+1}| \leq C\epsilon_n^2,$$

where $C > 0$ is some number (which depends on f but not on n.) It follows that

$$|\epsilon_n| \leq \frac{1}{C}(C\epsilon_0)^{2^k}.$$

We say the method "converges quadratically" because the exponent of ϵ_n is 2. The number of correct digits is *squared* at each iteration. In contrast, the bisection method only converges linearly. We also sometimes refer to "linear convergence" as first-order convergence, although the meaning of the expression is completely different from what is was in the previous chapters.

Convergence is ensured as soon as the starting point x_0 is close enough to the (unknown) root x^*, in the sense that $|C\epsilon_0| < 1$, so that $(C\epsilon_0)^{2^k} \to 0$ as $k \to \infty$. If the condition $|C\epsilon_0| < 1$ is not satisfied, Newton's method may very well diverge. For instance, we expect problems when the derivative is very small: following the tangent can take us to a region very far away from the root. An example of a function $f(x)$ for

which Newton's method diverges is atan(x), when x_0 is chosen to be too far from the origin.

On the plus side, Newton's method is fast. On the minus side, Newton's method only converges to a root only when you're already quite close to it.

3. **Method 3: the secant method**.

   If we do not know the derivative, we cannot set up Newton's method, but we can approximate it by replacing the derivative by (let $f_n = f(x_n)$)

$$f[x_{n-1}, x_n] = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}.$$

   Hence we define x_{n+1} by

$$x_{n+1} = x_n - \frac{f_n}{f[x_{n-1}, x_n]}.$$

   The geometrical idea is to replace the tangent line at x_n by the secant line supported by x_{n-1} and x_n. The secant method requires two points x_0 and x_1 as starting guesses.

   Notice that at each step, only one evaluation of f is necessary, because $f(x_{n-1})$ is already known from the previous iteration. If we were to form a finite difference approximation of the derivative with a very small grid step h, we may be more accurate but that requires two evaluations of f rather than one.

   Let us check the convergence properties of the secant method. The line joining the two points (x_{n-1}, f_{n-1}) and (x_n, f_n) is the degree-1 interpolant in the interval $[x_{n-1}, x_n]$:

$$p(x) = f_n + f[x_{n-1}, x_n](x - x_n).$$

   Outside of this interval, it is an extrapolant. Regardless of whether $x \in [x_{n-1}, x_n]$ or not, the difference between p and f is known from a theorem we saw in the previous chapter:

$$f(x) - p(x) = \frac{1}{2} f''(\xi)(x - x_n)(x - x_{n-1}),$$

where ξ is in the smallest interval containing x, x_{n-1}, and x_n. Evaluating this relation at the root $x = x^*$, we get

$$0 = f_n + f[x_{n-1}, x_n](x^* - x_n) + \frac{1}{2}f''(\xi)(x^* - x_n)(x^* - x_{n-1}).$$

On the other hand the definition of x_{n+1} gives

$$0 = f_n + f[x_{n-1}, x_n](x_{n+1} - x_n).$$

Subtracting these two equations we get

$$\epsilon_{n+1} = \frac{1}{2}\frac{f''(\xi)}{f[x_{n-1}, x_n]}\epsilon_n\epsilon_{n-1}.$$

Again, thanks to the same assumptions on f as in Newton's method, the ratio $\frac{f''(\xi)}{f[x_{n-1},x_n]}$ has a finite limit as $n \to \infty$, hence is bounded by some number $C > 0$. We get

$$|\epsilon_{n+1}| \leq C|\epsilon_n||\epsilon_{n-1}|.$$

The decay of ϵ_n is somewhere between first (linear) and second (quadratic) order. To obtain a more precise rate of decay, we guess that the inequality above should be reducible to the form $|\epsilon_n| \leq C|\epsilon_{n-1}|^p$ for some p. Using this equation and $|\epsilon_{n+1}| \leq C|\epsilon_n|^p$ above, we get

$$|\epsilon_{n-1}|^{p^2} \leq C|\epsilon_{n-1}|^p|\epsilon_{n-1}|.$$

The exponents match on the left and the right provided $p^2 = p + 1$, which has for positive solution

$$p = \frac{1 + \sqrt{5}}{2}. \qquad \text{(a number sometimes called the golden ratio).}$$

We check that $p = 1.618...$, a number between 1 and 2, Hence the secant method is faster than bisection, but slower than Newton's method. The secant method inherits the problem of Newton's method: it only converges when the starting guesses x_0 and x_1 are sufficiently close to the root.

We can also set up Newton's method in several dimensions. A system of nonlinear equations is of the form

$$f_i(x_1, \ldots, x_n) = 0, \qquad i = 1, \ldots, n.$$

We take the same number of equations and unknowns, so that we may be in a situation where there is one solution (rather than a continuum of solutions or no solution at all.) Whether the system has zero, one or several solutions is still a question that needs to be addressed separately. The shorthand notation for the system is $\mathbf{f}(\mathbf{x}) = 0$

By analogy with the 1D case we perform a Taylor expansion about $\mathbf{x}_n$:

$$0 = \mathbf{f}(\mathbf{x}^*) = \mathbf{f}(\mathbf{x}_n) + \nabla\mathbf{f}(\mathbf{x}_n)(\mathbf{x}^* - \mathbf{x_n}) + O(\|\mathbf{x}^* - \mathbf{x}_n\|^2).$$

With indices this equation is written as

$$0 = f_i(\mathbf{x}^*) = f_i(\mathbf{x}_n) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(\mathbf{x}_n)(x_{j,n} - x_j^*) + O(\sum_j (x_{j,n} - x_j^*)^2).$$

(Watch the subscript n which indicates the n-th iterate while the subscript j indicates the j-th component.) The next iterate $\mathbf{x}_{n+1}$ is defined by neglecting the quadratic error and isolating $\mathbf{x}^*$. A linear system of equations needs to be solved: the Jacobian matrix $\nabla\mathbf{f}(\mathbf{x}_n)$ is inverted and we get

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\nabla\mathbf{f}(\mathbf{x}_n)]^{-1}\,\mathbf{f}(\mathbf{x}_n).$$

The geometrical interpretation of this equation is that we can fit the tangent plane to each of the surfaces $y = f_i(x_1, \ldots, x_n)$ in $\mathbb{R}^{n+1}$, find the line at the intersection of all these planes, and check where this line intersects the (hyper)plane $y = 0$.

Newton's method is still quadratically convergent in multiple dimensions, and special care must still be taken to make sure that we start close enough to a root.

Example 10.

$$x_1^2 + x_2^2 = 1, \qquad x_2 = \sin(x_1).$$

Write this as a root-finding problem: $f_1 = f_2 = 0$ *with*

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 1, \qquad f_2(x_1, x_2) = x_2 - sin(x_1).$$

The Jacobian matrix is

$$J = \nabla \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 & 2x_2 \\ -\cos(x_1) & 1 \end{pmatrix}.$$

Use the formula for the inverse of a 2-by-2 matrix:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

to obtain

$$J^{-1} = \frac{1}{2x_1 + 2x_2 \cos(x_1)} \begin{pmatrix} 1 & -2x_2 \\ \cos(x_1) & 2x_1 \end{pmatrix}.$$

The Newton iteration is therefore

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - J^{-1} \begin{pmatrix} x_{1,n}^2 + x_{2,n}^2 - 1 \\ x_{2,n} - \sin x_{1,n} \end{pmatrix}.$$

4.2 Optimization problems

Another important recurring problem in science and engineering is that of finding a minimum or a maximum of a function $F(x)$. A point x^* is a local minimum when $F(y) \geq F(x^*)$ for all y in a neighborhood of x^*. It is a global minimum when $F(y) \geq F(x^*)$ for all y. We write

$$\min_x F(x)$$

for the minimum value $F(x^*)$. We then call x^* the argument of the minimum. Maximizing $F(x)$ instead is the same as minimizing $-F(x)$, so it suffices to talk about minimization.

When $F(x)$ is smooth, and x is allowed to run over all real numbers (not restricted to an interval or other set), then it suffices to solve $F'(x) = 0$ (and check that $F''(x) > 0$) in order to find a local minimum. Hence it suffices to apply Newton's method or any other root-finding method to the function $f(x) = F'(x)$. We obtain

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

In multiple dimensions, we minimize a scalar function $F(x_1, \ldots, x_n)$. The optimality condition, obeyed at the minimum $x_1^*, \ldots x_n^*$, is that all the partial derivatives of F vanish, i.e.,

$$\nabla F(x_1^*, \ldots x_n^*) = 0.$$

Newton's method, also called Newton descent, follows from considering these equations as a nonlinear system $f_i(x_1, \ldots, x_n) = 0$ with $f_i = \frac{\partial F}{\partial x_i}$. We get

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left[\nabla\nabla F(\mathbf{x}_n)\right]^{-1} \nabla F(\mathbf{x}_n).$$

The matrix $\nabla\nabla F$ of second partial derivatives of F is called the Hessian. In index notation,

$$(\nabla\nabla F)_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}.$$

Compare Newton's method with simple gradient descent:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \nabla F(\mathbf{x}_n),$$

for some sufficiently small scalar α. Gradient descent is slower but typically converges from a larger set of initial guesses than Newton's method.

Example 11. *Consider*

$$F(x_1, x_2) = x_1^2 + (\log x_2)^2.$$

This function has a unique minimum for $x_1 \in \mathbb{R}$ and $x_2 > 0$. We compute

$$\nabla F(x_1, x_2) = \begin{pmatrix} 2x_1 \\ \frac{2 \log x_2}{x_2} \end{pmatrix}$$

and

$$\nabla\nabla F(x_1, x_2) = \begin{pmatrix} 2 & 0 \\ 0 & \frac{2 - 2\log x_2}{x_2^2} \end{pmatrix}.$$

Newton's iteration is therefore

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{x_{2,n}^2}{2 - 2\log x_{2,n}} \end{pmatrix} \begin{pmatrix} 2x_{1,n} \\ 2\frac{\log x_{2,n}}{x_{2,n}} \end{pmatrix}.$$

Notice that x_1 goes in one step to zero, because a quadratic function is exactly minimized by Newton's method.

MIT OpenCourseWare
http://ocw.mit.edu

18.330 Introduction to Numerical Analysis
Spring 2012

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.