3.3 Roots of an equation

In physics, we often encounter situations in which we need to find the possible value of x that ensures the equation $f(x) = 0$, where $f(x)$ can either be an explicit or an implicit function of x. If such a value exists, we call it a *root* or *zero* of the equation. In this section, we will discuss only single-variable problems and leave the discussion of multivariable cases to Chapter 5, after we have gained some basic knowledge of matrix operations.

Bisection method

If we know that there is a root $x = x_r$ in the region $[a, b]$ for $f(x) = 0$, we can use the *bisection method* to find it within a required accuracy. The bisection method is the most intuitive method, and the idea is very simple. Because there is a root in the region, $f(a)f(b) < 0$. We can divide the region into two equal parts with $x_0 = (a + b)/2$. Then we have either $f(a)f(x_0) < 0$ or $f(x_0)f(b) < 0$. If $f(a)f(x_0) < 0$, the next trial value is $x_1 = (a + x_0)/2$; otherwise, $x_1 = (x_0 + b)/2$. This procedure is repeated until the improvement on x_k or $|f(x_k)|$ is less than the given tolerance δ.

Let us take $f(x) = e^x \ln x - x^2$ as an example to illustrate how the bisection method works. We know that when $x = 1$, $f(x) = -1$ and when $x = 2$, $f(x) = e^2 \ln 2 - 4 \approx 1$. So there is at least one value $x_r \in [1, 2]$ that would make $f(x_r) = 0$. In the neighborhood of x_r, we have $f(x_r + \delta) > 0$ and $f(x_r - \delta) < 0$.

The Newton method

This method is based on linear approximation of a smooth function around its root. We can formally expand the function $f(x_r) = 0$ in the neighborhood of the root x_r through the Taylor expansion

$$f(x_r) \simeq f(x) + (x_r - x)f'(x) + \cdots = 0, \tag{3.47}$$

where x can be viewed as a trial value for the root of x_r at the kth step and the approximate value of the next step x_{k+1} can be derived from

$$f(x_{k+1}) = f(x_k) + (x_{k+1} - x_k)f'(x_k) \simeq 0, \tag{3.48}$$

that is,

$$x_{k+1} = x_k + \Delta x_k = x_k - f_k/f'_k, \tag{3.49}$$

with $k = 0, 1, \ldots$. Here we have used the notation $f_k = f(x_k)$. The above iterative scheme is known as the *Newton method*. It is also referred to as the *Newton–Raphson method* in the literature. The above equation is equivalent to approximating the root by drawing a tangent to the curve at the point x_k and taking x_{k+1} as the tangent's intercept on the x axis. This step is repeated toward the root, as illustrated in Fig. 3.1. To see how this method works in a program, we again take the function $f(x) = e^x \ln x - x^2$ as an example. The following program is an implementation of the Newton method.

Secant method

In many cases, especially when $f(x)$ has an implicit dependence on x, an analytic expression for the first-order derivative needed in the Newton method may not exist or may be very difficult to obtain. We have to find an alternative scheme to achieve a similar algorithm. One way to do this is to replace f_k' in Eq. (3.49) with the two-point formula for the first-order derivative, which gives

$$x_{k+1} = x_k - (x_k - x_{k-1}) f_k / (f_k - f_{k-1}). \tag{3.50}$$

This iterative scheme is commonly known as the *secant method*, or the *discrete Newton method*. The disadvantage of the method is that we need two points in order to start the search process. The advantage of the method is that $f(x)$ can now be implicitly given without the need for the first-order derivative. We can still use the function $f(x) = e^x \ln x - x^2$ as an example, in order to make a comparison.

3.4 Extremes of a function

An associated problem to finding the root of an equation is finding the maxima and/or minima of a function. Examples of such situations in physics occur when considering the equilibrium position of an object, the potential surface of a field, and the optimized structures of molecules and small clusters. Here we consider mainly a function of a single variable, $g = g(x)$, and just touch on the multivariable case of $g = g(x_1, x_2, \ldots, x_I)$ with the steepest-descent method. Other schemes for the multivariable cases are left to later chapters.

Knowing the solution of a nonlinear equation $f(x) = 0$, we can develop numerical schemes to obtain minima or maxima of a function $g(x)$. We know that an extreme of $g(x)$ occurs at the point with

$$f(x) = \frac{dg(x)}{dx} = 0, \tag{3.51}$$

which is a minimum (maximum) if $f'(x) = g''(x)$ is greater (less) than zero. So all the root-search schemes discussed so far can be generalized here to search for the extremes of a single-variable function.

However, at each step of updating the value of x, we need to make a judgment as to whether $g(x_{k+1})$ is increasing (decreasing) if we are searching for a maximum (minimum) of the function. If it is, we accept the update. If it is not, we reverse the update; that is, instead of using $x_{k+1} = x_k + \Delta x_k$, we use $x_{k+1} = x_k - \Delta x_k$. With the Newton method, the increment is $\Delta x_k = -f_k/f'_k$, and with the secant method, the increment is $\Delta x_k = -(x_k - x_{k-1})f_k/(f_k - f_{k-1})$.

Let us illustrate these ideas with a simple example of finding the bond length of the diatomic molecule NaCl from the interaction potential between the two ions (Na^+ and Cl^- in this case). Assuming that the interaction potential is $V(r)$ when the two ions are separated by a distance r, the bond length r_{eq} is the equilibrium distance when $V(r)$ is at its minimum. We can model the interaction potential between Na^+ and Cl^- as

$$V(r) = -\frac{e^2}{4\pi\epsilon_0 r} + V_0 e^{-r/r_0}, \tag{3.52}$$

where e is the charge of a proton, ϵ_0 is the electric permittivity of vacuum, and V_0 and r_0 are parameters of this effective interaction. The first term in Eq. (3.52) comes from the Coulomb interaction between the two ions, but the second term

is the result of the electron distribution in the system. We will use $V_0 = 1.09 \times 10^3$ eV, which is taken from the experimental value for solid NaCl (Kittel, 1995), and $r_0 = 0.330$ Å, which is a little larger than the corresponding parameter for solid NaCl ($r_0 = 0.321$ Å), because there is less charge screening in an isolated molecule. At equilibrium, the force between the two ions,

$$f(r) = -\frac{dV(r)}{dr} = -\frac{e^2}{4\pi\epsilon_0 r^2} + \frac{V_0}{r_0} e^{-r/r_0}, \tag{3.53}$$

is zero. Therefore, we search for the root of $f(x) = dg(x)/dx = 0$, with $g(x) = -V(x)$. We will force the algorithm to move toward the minimum of $V(r)$. The following program is an implementation of the algorithm with the secant method to find the bond length of NaCl.

The bond length obtained from the above program is $r_{eq} = 2.36$ Å. We have used $e^2/4\pi\epsilon_0 = 14.4$ Å eV for convenience. The method for searching for the minimum is modified slightly from the secant method used in the preceding section in order to force the search to move toward the minimum of $g(x)$. We will still obtain the same result as with the secant method used in the earlier example for this simple problem, because there is only one minimum of $V(x)$ around the point where we start the search. The other minimum of $V(x)$ is at $x = 0$ which is also a singularity. For a more general $g(x)$, modifications introduced in the above program are necessary.

Another relevant issue is that in many cases we do not have the explicit function $f(x) = g'(x)$ if $g(x)$ is not an explicit function of x. However, we can construct the first-order and second-order derivatives numerically from the two-point or three-point formulas, for example.

In the example program above, the search process is forced to move along the direction of descending the function $g(x)$ when looking for a minimum. In other words, for $x_{k+1} = x_k + \Delta x_k$, the increment Δx_k has the sign opposite to $g'(x_k)$. Based on this observation, an update scheme can be formulated as

$$x_{k+1} = x_k + \Delta x_k = x_k - ag'(x_k), \tag{3.54}$$

with a being a positive, small, and adjustable parameter. This scheme can be generalized to the multivariable case as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k = \mathbf{x}_k - a\nabla g(\mathbf{x}_k)/|\nabla g(\mathbf{x}_k)|, \tag{3.55}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_l)$ and $\nabla g(\mathbf{x}) = (\partial g/\partial x_1, \partial g/\partial x_2, \ldots, \partial g/\partial x_l)$.

Note that step $\Delta\mathbf{x}_k$ here is scaled by $|\nabla g(\mathbf{x}_k)|$ and is forced to move toward the direction of the steepest descent. This is why this method is known as the *steepest-descent method*. The following program is an implementation of such a scheme to search for the minimum of the function $g(x, y) = (x-1)^2 e^{-y^2} + y(y+2)e^{-2x^2}$ around $x = 0.1$ and $y = -1$.