15 Nonlinear Equations and Zero-Finders

This lecture describes several methods for the solution of nonlinear equations. In particular, we will discuss the computation of zeros of nonlinear functions $f(x)$. The latter is a classical topic in scientific computation. Most of our discussion is concerned with the computation of a single zero, denoted by x^*, of a real-valued nonlinear function of a real variable. Extensions to the computation of several zeros and to the determination of complex-valued zeros also will be commented on.

We compute approximations of the desired zero x^* by solving a nonlinear equation by an *iterative method*. Given an initial approximation, x_0, of x^* the iterative method determines a sequence of improved approximations $x_1, x_2, x_3 \ldots$ that converge to x^*. We are interested in conditions on $f(x)$ and x_0 that secure convergence, as well as in the rate of convergence. Ideally, we would like rapid convergence to x^* for any initial approximation x_0. The approximate solutions x_k are referred to as *iterates*.

As we will see, it is often advantageous if the initial iterate x_0 lives in a vicinity of x^*. Such an initial iterate often can be determined by graphing the function $f(x)$. When one has to determine the zeros of a sequence of nonlinear functions, a zero of a nearby function may provide a good choice of x_0.

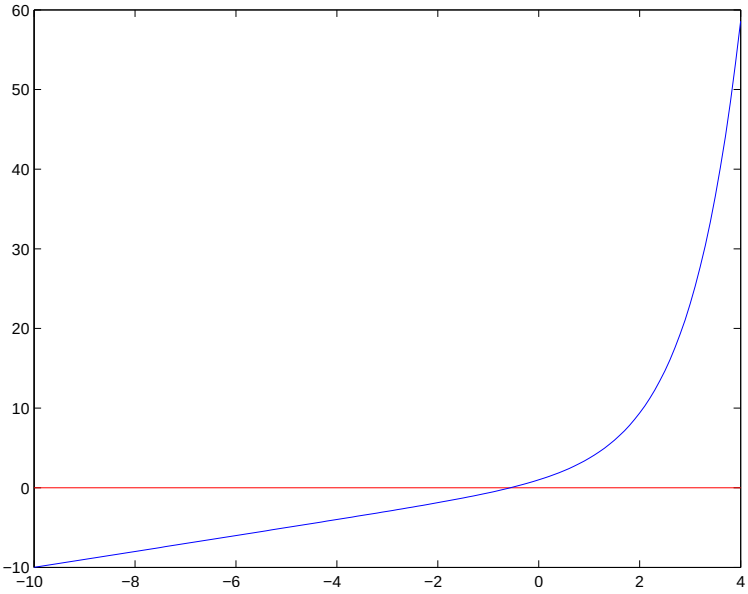

Figure 1: The function (2) of Example 15.1 (blue graph) and a horizontal line (red) passing through the origin.

Example 15.1: Consider the problem of solving the nonlinear equation

$$x = -e^x. \tag{1}$$

This problem is equivalent to finding the zero of the function

$$f_1(x) = e^x + x, \tag{2}$$

which is depicted by the blue graph of Figure 1. The red graph of the figure is the horizontal line $y = 0$. The desired solution x^* is at the intersection of these graphs. Figure 1 shows that the solution x^* is unique, and that it lives in the interval $-2 \leq x \leq 0$. □

Example 15.2: There are many ways to determine a function, whose zero is the solution of equation (1). For instance, multiplying (1) by e^{-x} yields the function

$$f_2(x) = 1 - xe^{-x}. \tag{3}$$

Thus, we can solve equation (1) also by determining a zero of $f_2(x)$. □

Example 15.3: Letting $t = e^x$ in (1) gives the function

$$f_3(t) = t - \ln(-t), \qquad x = \ln(t). \tag{4}$$

We therefore may determine the solution x^* of equation (1) by computing the zero t^* of (4) and them evaluating $x^* = \ln(t^*)$. □

Hence, we can determine the solution x^* of equation (1) by computing the zero of any of the functions f_1, f_2, or f_3. Below we will discuss applications of iterative methods to these functions and compare their performance.

The methods described in this lecture either bracket x^* or approximate $f(x)$ in a neighborhood of x^* by a sequence of linear or quadratic polynomials. The polynomials are designed so that a zero of each successive linear or quadratic polynomial provides an improved approximation of x^*.

15.1 Bisection

Figure 1 shows that the zero x^* of the function (2) lives in the interval $[a, b] = [-2, 0]$, and that $f(a) < 0$ and $f(b) > 0$. The bisection method determines a sequence of intervals of decreasing length, each one guaranteed to contain x^*. The method proceeds as follows: Denote the midpoint of the interval $[a, b]$ by m. Determine the sign of $f(m)$. If $f(a)f(m) < 0$, then x^* lives in the interval $[a, m]$ and we let $b := m$; otherwise x^* is in the interval $[m, b]$ and we let $a := m$. We now have a new interval $[a, b]$ of half the length of the original interval, and such that $f(a)f(b) < 0$. We repeat the computations until a sufficiently small interval that contains x^* has been determined. In the rare event that $f(m) = 0$, we have found x^* and can terminate the computations. We refer to this event as *lucky termination*.

The following Matlab/Octave code describes the computations. The user-supplied parameter delta specifies the desired accuracy of the computed approximation of x^*. The computations are terminated when an interval of length smaller than delta which contains x^* has been found. The bisection method requires an initial interval $[a, b]$, such that $f(a)f(b) < 0$. Such an interval can be found, e.g., by plotting the function. The following bisection code ignores the possibility of lucky termination.

```
while b-a>=delta
   m=(a+b)/2;
   if f(m)f(a)<0
      b=m;
   else
      a=m;
   end
end
```

Bisection is a good way to determine the approximate location of a zero of a real-valued nonlinear function of a real variable. However, when this zero is requested with high accuracy, typically many function evaluations are required. Note that bisection only uses the sign of the function at the evaluation points. Other zero-finders discussed below use the values of the function at selected points also. Using the function values makes it possible to determine an accurate approximation of the zero with fewer function evaluations. Keeping the number of function evaluations small can be important when the evaluation of the function is time-consuming.

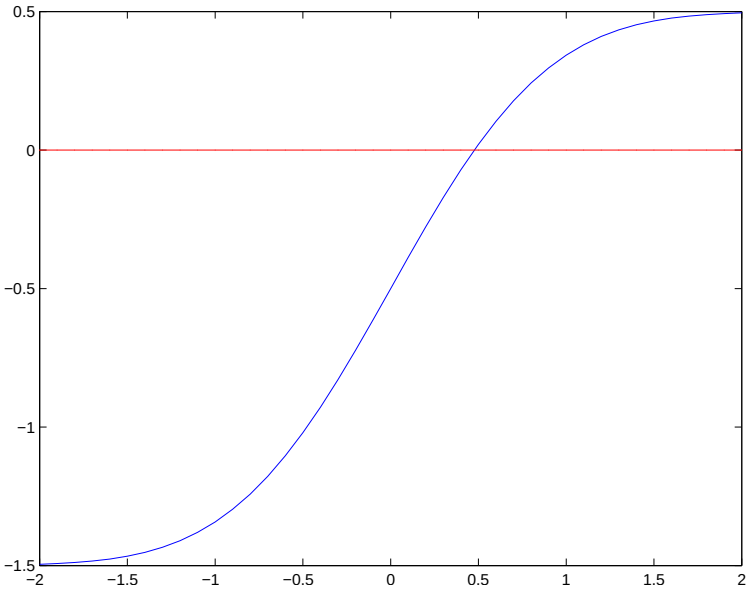

Figure 2: The function (5) of Example 15.4 (blue graph) and a horizontal line (red) passing through the origin.

Example 15.4: Determine the value $x > 0$ for which the error function

$$x \to \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$$

achieves the value $1/2$. Thus, we would like to compute the zero of

$$f(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt - \frac{1}{2}. \tag{5}$$

This function is shown in Figure 2. Each function evaluation requires the computation of an integral. This has to be done by a numerical method and can be fairly time consuming if high accuracy is desired. We therefore would like to use a method that requires few function evaluations. □

15.2 Bisection enhanced by linear interpolation

We describe a modification of bisection, that also uses the function values of $f(x)$. Let an interval $[a, b]$ be known, such that $f(x)$ is defined there and $f(a)f(b) < 0$. Consider the straight line through the points

$(a, f(a))$ and $(b, f(b))$. It can be expressed as the linear function

$$p(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a); \tag{6}$$

cf. Example 11.4. This polynomial is a linear approximation of $f(x)$ on the interval $[a, b]$. Its zero, which we denote by z, satisfies

$$f(a) + \frac{f(b) - f(a)}{b - a}(z - a) = 0,$$

i.e.,

$$z = a - \frac{b - a}{f(b) - f(a)} f(a). \tag{7}$$

When $f(a)$ is close to zero and $f(b)$ is not, the zero z of $p(x)$ is close to a, and generally a better approximation of a zero of $f(x)$ than the midpoint m between a and b. Similarly, when $f(b)$ is nearly zero but $f(a)$ is not, the point z is close to b, and typically a better approximation of a zero of $f(x)$ than m. This suggests that it may be advantageous to replace the midpoint m in the bisection method by z. Thus, if $f(a)f(z) < 0$, then we set $b = z$, otherwise we set $a = z$. This gives a new shorter interval that contains a zero. The computations are repeated until this interval is sufficiently short. This method is known as *regula falsi* and can be traced back to the 3rd century BC.

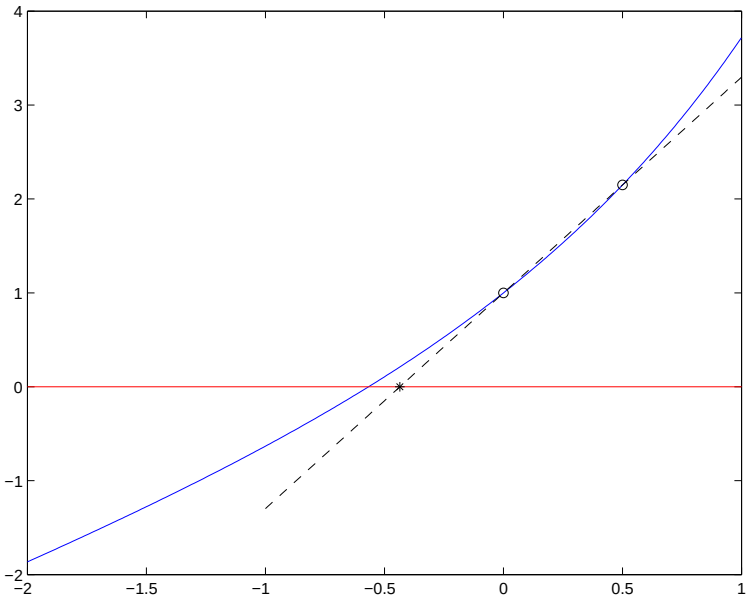

Figure 3: The function $f(x)$ of Example 15.1 defined by (2) (blue graph), a horizontal line (red) passing through the origin, the points $(x_k, f(x_k)) = (0, f(0))$ and $(x_{k-1}, f(x_{k-1})) = (0.5, f(0.5))$ marked by o, the secant through these points (i.e., the polynomial (8)) marked by a dashed line, and the new approximation x_{k+1} given by (9) marked by $*$.

15.3 Linear interpolation using the secant

The enhanced bisection method is faster than standard bisection, however, it also typically requires a fairly large number of function evaluations to determine the location of a zero to high accuracy. There are other ways of using linear interpolation, which yield faster convergence. However, these methods do not bracket the zero. The first one of these methods we will discuss is the *secant method*. This method also is based on the formulas (6) and (7). Assume that we have two approximations, x_{k-1} and x_k, of x^* and would like to determine a new approximation, x_{k+1}. The approximations x_{k-1} and x_k are not required to bracket x^*. Substituting $a := x_k$ and $b := x_{k-1}$ into (6) yields the linear polynomial

$$p(x) = f(x_k) + \frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k}(x - x_k). \tag{8}$$

The graph of $p(x)$ is the *secant* to the graph of $f(x)$ through the points $(x_k, f(x_k))$ and $(x_{k-1}, f(x_{k-1})$; see Figure 3

Let x_{k+1} denote the zero of (8). Analogously to (7), we have

$$x_{k+1} = x_k - \frac{x_{k-1} - x_k}{f(x_{k-1}) - f(x_k)} f(x_k). \tag{9}$$

The important properties of the secant method include:

- When the method converges, convergence typically is much faster than for any variant of bisection. Generally, the method converges when the x_k and x_{k-1} are close enough to x^*. Therefore the secant method often is applied in conjunction with bisection; the latter method is used to determine approximations x_{k-1} and x_k close enough to x^* to secure convergence of the secant method. It is not always clear when such approximations have been found. When the secant method does not converge, one has to return to bisection to determine improved approximations of x^*.

- The secant method is not guaranteed to converge. However, geometric reasoning can provide simple sufficient conditions for convergence. For instance, if $f(x)$ is continuous, increasing, and convex, and if both x_k and x_{k-1} are larger than x^*, then x_{k+1} will be an improved approximation of x^* with $x_{k+1} > x^*$. This result can be seen by graphing $f(x)$ and the secant; see Figure 3. Similarly, if $f(x)$ is continuous, decreasing, and convex, and if both x_k and x_{k-1} are smaller than x^*, then x_{k+1} will be an improved approximation of x^* with $x_{k+1} < x^*$.

- When the x_k converge to x^* as k increases, the distance between consecutive approximations, $x_k - x_{k-1}$, converges to zero as well. For x_k and x_{k-1} very close but distinct, it can be difficult to evaluate the quotient
$$\frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k} \tag{10}$$
in (8) accurately. Poor accuracy may limit how well we are be able to approximate x^*. The iterations should be terminated before round-off errors or other errors in the function values $f(x_j)$ give rise to large errors in the computed value of the quotient (10). Moreover, numerical difficulties may arise when $f(x_k)$ is very close to $f(x_{k-1})$; in fact, x_{k+1} cannot be evaluated when $f(x_k) = f(x_{k-1})$.

Assume that we have computed a sequence of approximations $x_1, x_2, \ldots, x_k$ of x^* and have terminated the computations because $|f(x_k)|$ is "tiny." We would like to know how well x_k approximates x^*. Assume

that $f(x)$ has a continuous derivative in a neighborhood of x^*, which contains x_k. The mean value theorem then provides a bound for the error in x_k. We have

$$|f(x_k)| = |f(x_k) - f(x^*)| = |f'(\tilde{x})|\,|x_k - x^*|, \tag{11}$$

where $\tilde{x}$ lives in the interval with endpoints x_k and x^*. It follows from (11) that if $|f'(\tilde{x})|$ is tiny as well, then $|x_k - x^*|$ is not guaranteed to be small. Thus, without information about $f'(x)$ in a neighborhood of x^*, we cannot conclude that x_k is an accurate approximation of x^* from the fact that $|f(x_k)|$ is small. We only can conclude that x_k is close to x^* when $|f(x_k)|$ is small and $f'(x)$ is large in a neighborhood of x^*.

The secant method often is applied when the derivative of $f(x)$ is cumbersome to evaluate. This can be the case, for instance, when $f(x)$ is the product, quotient, or composition of several functions. For functions, which allow evaluation of the derivative for a reasonable cost, the method of the following section is preferable.

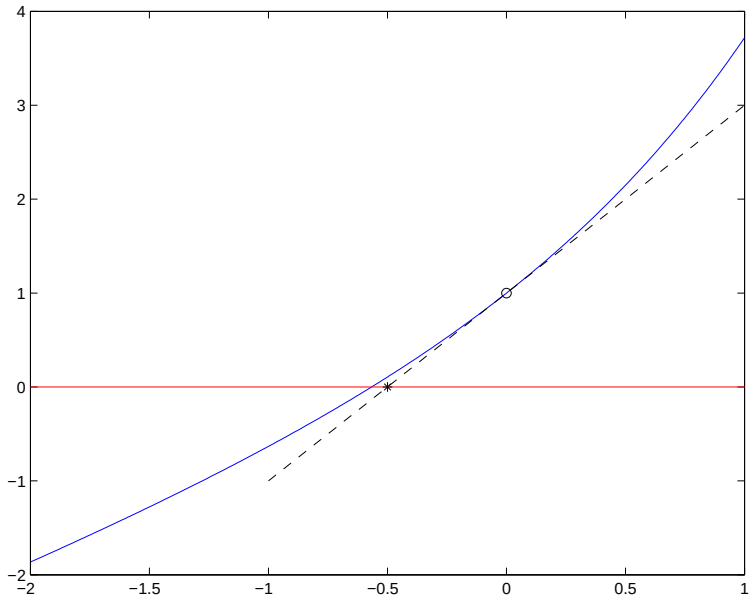

Figure 4: The function $f(x)$ of Example 15.1 defined by (2) (blue graph), a horizontal line (red) passing through the origin, the point $(x_k, f(x_k) = (0, f(0))$ marked by o, the tangent through this point (i.e., the polynomial (8)) marked by a dashed line, and the new approximation x_{k+1} given by (12) marked by $*$.

15.4 Linear interpolation using the tangent

When x_{k-1} and x_k are close, the quotient (10) approximates the derivative $f'(x_k)$. Replacing the quotient by this derivative in (9) yields *Newton's method*:

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

This method also can be derived by using a Taylor expansion of $f(x^*)$ at x_k. Let $x^* = x_k + h$. Then

$$0 = f(x^*) = f(x_k + h) = f(x_k) + hf'(x_k) + \frac{h^2}{2}f''(x_k) + \frac{h^3}{6}f'''(x_k) + \dots \ . \tag{13}$$

If $|h|$ is small, then $|h|^j$ is much smaller for $j \geq 2$. This suggests that we may ignore all terms with factors h^j for $j \geq 2$ in the Taylor expansion. We obtain the equation

$$0 = f(x_k) + hf'(x_k),$$

Solving this equation for h and letting $x_{k+1} = x_k + h$ gives (12). Geometrically, Newton's method is obtained by moving the point $(x_{k-1}, f(x_{k-1}))$ towards $(x_k, f(x_k))$ in Figure 3. Then the secant in that figure is replaced by the tangent through $(x_k, f(x_k)$; see Figure 4. Comparing Figures 3 and 4 shows Newton's method to give a better approximation of the desired zero than the secant method.

Let x_{k+1} denote the zero of (8). When the iterates x_k are sufficiently close to the zero x^* and $f'(x)$ is continuous in a neighborhood of x^*, Newton's method converges faster than the secant method. It is typically difficult to establish whether x_k is close enough to x^* for Newton's method to converge, without computing the next iterate x_{k+1}. Geometric reasoning provides sufficient conditions for convergence, similarly as for the secant method. For instance, Newton's method converges when $f(x)$ is increasing and convex, and x_k is larger than x^*. Then the next iterate satisfies $x^* < x_{k+1} < x_k$; see Figure 4 for an illustration.

15.5 Other zero-finders

All methods discussed, except for basic bisection, approximate the function $f(x)$ by a sequence of linear functions. One may wonder whether approximation of $f(x)$ by higher degree polynomials would be beneficial. It can be, but the methods become more complicated.

Example 15.5: The starting point for our derivation of the secant method is the linear polynomial (8). After two or more iterations, three nodes x_k, x_{k-1}, x_{k-2} and associated function values $f(x_k), f(x_{k-1}), f(x_{k-2})$ are available. They can be used to determine the interpolating polynomial $p(x)$ of degree at most two. This polynomial typically has two distinct zeros. It is not obvious which one of these zeros provides the best approximation of x^*, however, it is often reasonable to choose the zero closest to the available approximation of x^* as our next approximation. Convergence properties of this method when not all the iterates x_{k-2}, x_{k-1}, and x_k are very close to x^* are poorly understood. Moreover, the quadratic polynomial may have complex zeros even though $f(x)$ is real-valued and x^* is real. □

Example 15.6: Newton's method can be generalized by including more terms in the Taylor series in the local approximation of $f(x)$. For instance, including three terms yields the quadratic equation

$$0 = f(x_k) + hf'(x_k) + \frac{h^2}{2}f''(x_k)$$

for h. This equation has the solutions h_1 and h_2. Assume that the h_j are real and that $|h_1| \leq |h_2|$. We then use h_1 as a correction of x_k, i.e., $x_{k+1} = x_k + h_1$. The main drawback of this method is that in many applications $f''(x_k)$ is tedious to evaluate. Moreover, the h_j may be complex. □

We have only discussed the most popular iterative methods. Many more methods are available. Given this multitude of methods, which one should be applied? Methods that approximate $f(x)$ by a linear polynomial give rapid convergence if the graph of $f(x)$ is close to a straight line in a large neighborhood of x^*.. Similarly, methods that approximate $f(x)$ by a quadratic polynomial yield fast convergence if the graph of $f(x)$ looks like a parabola. Hence, the choice of method can be based on the look of the graph of $f(x)$ around the

desired zero. Also the choice of a particular transformation of the original equations, say (1), to a function, e.g., 2), (3), and (4), whose zero we would like to compute should be based on the same consideration. Thus, we would like the graph of the function $f(x)$ be nearly linear or nearly a parabola.

15.6 Zeros of polynomials

The classical application of zero-finders is to determine zeros of polynomials. A polynomial of degree n,

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0, \qquad a_n \neq 0, \tag{14}$$

has precisely n zeros in the complex plane. This result is known as the fundamental theorem of algebra. The coefficients a_j may be real or complex numbers. Multiple zeros are counted according to their multiplicity.

Example 15.7: The polynomial

$$p(x) = x^6 - 2x^5 - x^4 + 4x^3 - 5x^2 + 6x - 3 \tag{15}$$

has 6 zeros: $x = \pm\sqrt{3}$, $x = \pm i$, where i is the imaginary unit, and the zero $x = 1$ of multiplicity two. □

The secant and Newton methods can be used to compute real and complex zeros of a polynomial with real or complex coefficients, however, bisection cannot. The polynomial (15) is real-valued and has a real-valued derivative for x real. It follows that the Newton and secant corrections of a real initial iterate x_0 are real. Thus, when using a real initial iterate these method only can determine the real zeros $\{1, \pm\sqrt{3}\}$ of $p(x)$. Complex zeros can be found only by using a complex-valued initial iterate x_0.

Having determined one zero of the polynomial (15), we would like to avoid that our zero-finder computes the same zero again when searching for another zero. One can achieve this by dividing out the zeros found. Assume that we have found the zero $x = \sqrt{3}$. We then apply the zero finder to the function

$$q(x) = \frac{p(x)}{x - \sqrt{3}} = x^5 - (2 - \sqrt{3})x^4 + (2 - 2\sqrt{3})x^3 + (-2 + 2\sqrt{3})x^2 + (-1 + 2\sqrt{3})x + \sqrt{3}. \tag{16}$$

Dividing out the computed zero is referred to as *deflation*. Deflation secures that the same zero is not computed twice. We note that deflation has to be carried out with some care, because in the imperfect world of floating-point computations, we are likely to deflate approximations of the actual zero. For instance, neither the zero $x = \sqrt{3}$ nor the coefficients in the right-hand side (16) can be represented exactly. Hence, our representation of the right-hand side polynomial in (16) has not the same zeros as $p(x)$.

Finally, we remark that the Newton and secant methods convergence slower towards zeros of high multiplicity than towards zeros of multiplicity one. The reason for this is that not only $p(x_k)$, but also $p'(x_k)$, converge to zero when the iterate x_k approaches a multiple zero.

This section illustrates that the conceptually simple task of computing all zeros of a polynomial requires quite sophisticated software. Nevertheless, public domain software for the accurate computation of all zeros of polynomials of degree up to a few thousand is available; see [1] for a description. The code carries out some computations in higher precision arithmetic in order to produce accurate zeros.

Exercises

Exercise 15.1. Determine the zero of the function (2) by the bisection and regula falsi methods and compare rates of convergence. How much faster does the length of the interval for regula falsi converge to zero than for bisection?

Exercise 15.2: Determine the zero of the function (2) by the secant and Newton methods and bisection and compare the rates of convergence. How fast to the iterates generated by the Newton and secant methods converge to x^*?

Exercise 15.3. Compare Newton's method applied to computing the zero of the functions (2), (3), and (4). Is any one of these formulations preferable? If this is the case, then explain.

Exercise 15.4. Use Newton's method to solve the problem of Example 15.4. Hint: Use the Matlab/Octave function erf.

Exercise 15.5. Use Newton's method to compute the $a^{1/3}$. Can one prescribe an initial iterate that yields convergence for any real a? Explain. How can you determine when to terminate the iterations?

Exercise 15.6. We will estimate the mass m of the planet Jupiter. Jupiter accounts for more than half the mass of all the planets combined. Let us simplify things by imagining that our solar system consists only of the Sun and Jupiter. It won't be far from the truth. The sun and the planet interact gravitationally, each orbiting the common center of mass. The sun orbits the center of mass in a tight circle of radius $r = 6.9707 \cdot 10^8 m$ with a uniform velocity $v = 11.7 m/s$. The period of rotation of the system is given as $T = 4332.71$ days. The mass of the sun is $M = 1.989 \cdot 10^{30} kg$. Using Newton's law of gravity and a few other equations we find that

$$\sigma x^3 = (1 + x)^2, \tag{17}$$

where $\sigma = \frac{GM}{rv^2}$ and $x = \frac{m}{M}$. The constant $G = 6.67 \cdot 10^{-11} \frac{Nm^2}{kg^2}$ is known as the universal gravitation constant.

- Solve (17) by Newton's method.
- Use your computer program for Newton's method to estimate the mass m of Jupiter within a tolerance of 10^{-16}.

References

[1] D. A. Bini, G. Fiorentino, L. Gemignani, and B. Meini, *Effective fast algorithms for polynomial spectral factorization*, Numer. Algorithms, 34 (2003), pp. 217–227.