

1.1. METHODS FOR UNIVARIATE FUNCTIONS

1.1.1. Finding the Zero of a Univariate Function

It will be shown that a necessary condition for x^* to be an unconstrained minimizer of a twice-continuously differentiable univariate function $f(x)$ is that $f'(x^*) = 0$. Since x^* is a zero of the derivative function, it can be seen that the problems of univariate minimization and zero-finding are very closely related (although not equivalent). Therefore, we shall first consider the problem of finding a point x^* in a bounded interval such that

$$f(x^*) = 0,$$

where f is a nonlinear function. We restrict ourselves to the case when f changes sign at x^* . The case when f does not change sign is much harder. Note that if f does not change sign then there is a function very similar to f that does not have a zero in the interval. Usually on a computer we cannot expect to solve a problem *precisely*. Instead we find an approximate solution (which in some cases may be exact) to a very similar problem as we now discuss. Clearly we can expect difficulties when a very similar problem does not have a solution.

The first question that should be asked is: what is meant by “finding” a zero? Normally we shall be working in finite precision on some computer, with only a finite number of representable values of x , and a computed value of f . Hence it is unrealistic to expect to find a machine-representable \bar{x} such that $fl(f(\bar{x}))$ is exactly zero, where $fl(f)$ is the computed value of f . We shall often be satisfied if an algorithm provides an interval $[a, b]$ such that

$$f(a)f(b) < 0 \quad \text{and} \quad |a - b| < \delta,$$

where $\delta > 0$ is some “small” tolerance. Since a zero of f must lie in $[a, b]$, any point in the interval can be taken as an estimate of the zero.

A difficult issue when defining any algorithm that generates an infinite sequence is when to stop. In our case one possibility is to terminate when a sufficiently small interval of uncertainty is known. In the above that corresponds to choosing δ . An inherent difficulty is what value to choose for δ . We clearly have a minimum value (what is it?), but such a value is usually much smaller than a sensible value for δ . A perennial difficulty is to determine what is meant by “small”. Another difficulty with the use of an interval is that we know only the sign of $fl(f(a))fl(f(b))$, which may differ from that of $f(a)f(b)$. Consequently, we cannot be sure that a zero does lie in the interval $[a, b]$. If either $f(a)$ or $f(b)$ is small when compared to the error in computing f then the sign is likely to be wrong. Since we are specifically seeking values where f is very small then the possibility of an error is clearly real. The problem with the use of an interval is that the sign may be wrong even when $b - a$ is large since only one of the signs of the function need be incorrect (ironically if both signs are wrong we end up with the correct assumption). An alternative approach is outlined in the next paragraph.

As we have mentioned in general we cannot expect to find the exact solution of a problem when working in finite precision. A useful concept in the numerical solution of problems is that of a neighboring problem. What we can sometimes show is we have the exact solution of a neighboring problem (even this is not always possible for some problems). A neighboring function, say $f_N(x)$, for which an exact zero is known is given by

$$f_N(x) = f(x) - f(\bar{x}),$$

where \bar{x} is an estimate of the solution. Clearly, the closest neighbor will be given by the estimate for which $|f(x)|$ is minimized. In general $f_N(x)$ is unknown since we only know $fl(f(x))$. However, it is usual to have at least an idea of the error made when computing $f(x)$ and this may be small compared to the differences between f at the different estimates. What is of interest is not so much an exact definition of a neighboring problem, but the knowledge that we have the exact solution of a neighboring problem albeit unknown that is sufficiently close to the original problem. In this case if an error bound is known for computing $f(x)$ then we have

$$|f_N(x) - f(x)| \leq |fl(f(\bar{x}))| + E,$$

where E is the bound on the error. In many cases we expect $|fl(f(\bar{x}))| \gg E$ since the error from computation is small compared with the accuracy that we seek. An example of such circumstances is when f is a model of a real world function for which the data is accurate to only 1%. It obviously does not make much sense to find a zero to a very close neighbor of the wrong problem. This example also illustrates why an alternative termination criterion can be useful. By noting how much a 1% perturbation in the data impacts f we can judge how close to make the neighboring function before terminating. In general if we have a measure of the accuracy that the model function approximates the real function we have a basis with which to compare a suitable bound on a neighboring function.

In our first approach of choosing δ we cannot assert that a zero of the neighboring function $fl(f)$ lies in a specific interval.

In order to describe algorithms for zero-finding, we require two preliminary definitions. The smallest interval in which x^* is known to lie is called the *interval of uncertainty*. The zero is said to be *bracketed* in an interval if f changes sign in the interval.

It is usually necessary when discussing or when proving properties of algorithms to “idealize” the algorithm. Nonetheless, we need to be cognizant of the consequence on the behavior and properties of an algorithm when it is applied in practice. It is difficult in most cases to know the precise effect, but some awareness of the *type* of effect is essential even if it cannot be quantified. To illustrate the point consider an algorithm that is guaranteed (in theory) to reduce the interval of uncertainty by a “significant” amount at each iteration. The test for convergence that the interval be reduced below $\delta > 0$ ensures we stop in a

finite number of iterations. However, as we have mentioned, in practice we will know only the sign of $fl(f(a)f(b))$ and if $fl(f(a))$ or $fl(f(b))$ is sufficiently close to zero we cannot be assured the sign is correct. Consequently, if δ is made sufficiently small it is almost inevitable that this will occur. Therefore two consequences of finite precision are that δ should not be chosen to be too “small” and that the interval of uncertainty may be in error by some “small” amount. Without some knowledge of the error made when computing $f(x)$ it is difficult to say what is “small”.

1.1.1.1. The method of bisection. This method is based on *systematically reducing the interval of uncertainty by function comparison*. Suppose that an initial interval $[a, b]$ has been specified in which $f(a)f(b) < 0$. We evaluate f at the midpoint of the interval and test its sign. If the function value is zero, the algorithm terminates; otherwise, a new interval of uncertainty is produced by discarding the value of a or b , depending on whether $f(a)$ or $f(b)$ agrees in sign with f at the midpoint. This process is illustrated in Figure 4a (of Practical Optimization).

Bisection is *guaranteed* to find x^* to within any specified tolerance δ if f can be computed to sufficient accuracy so that its sign is correct. A satisfactory interval may always be found by bisection under these conditions in about $\log_2((b-a)/\delta)$ evaluations of f . Bisection can be shown to be the “optimal” algorithm for the class of functions that change sign in $[a, b]$, in the sense that it yields the smallest interval of uncertainty apriori for a *specified* number of function evaluations.

Some indication of the speed of a zero-finding algorithm can be obtained by finding the *rate* at which the interval of uncertainty converges to zero. For the bisection algorithm, the length of this interval converges to zero linearly, with asymptotic error constant $\frac{1}{2}$. Although bisection has the best worst-case performance its performance is always equal to its worst case.

The bisection algorithm is well suited to using the interval of uncertainty as a basis for termination. We may also use as a test for termination that $|fl(f(\bar{x}))| < \epsilon$, where \bar{x} is the trial value for which $|fl(f)|$ is minimized. With such a termination criterion it is no longer possible to determine apriori the number of necessary trials. When a function is such that it is changing rapidly in the neighborhood of the solution then usually rather more trials are required than are required to reduce the interval of uncertainty below ϵ . Conversely, if the function is only slowly varying in the neighborhood of the zero then fewer trials will be required. Of course the nature of the function may not be known. Using this test the algorithm may be lucky on some occasions and terminate after very few trials.

1.1.1.2. Newton’s method. The defect of the bisection algorithm is that no account is taken of the relative magnitudes of the values of f at the various points. If f is known to be *well behaved*, it seems reasonable to use the *values* of f at the end points to determine the next estimate of x^* . A means of utilizing the magnitude of f during the search for a

zero is to *approximate* or *model* f by a function \hat{f} whose zero can be easily calculated. An iterative procedure can then be devised in which the zero of \hat{f} is taken as a new estimate of the zero of f itself.

If f is differentiable, an obvious candidate for \hat{f} is the *tangent line* at x_k , the current estimate of the zero. To compute the zero of the tangent line, we evaluate $f(x_k)$ and $f'(x_k)$; if $f'(x_k)$ is non-zero, the zero of the tangent line is given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (1.1.1)$$

The iterative procedure defined by (1.1.1) is known as *Newton's method for finding a zero*, and is illustrated in Figure 4b. It has a different flavor from bisection in that there is no “interval of uncertainty” *per se*; rather, a new estimate of the zero is computed at each iteration. With a pure Newton's method we have little choice other than to use $|f_l(f)| < \epsilon$ as a test for termination. If an interval of uncertainty is required then it is usually easy to modify the algorithm to generate one.

Some well-known and seemingly unrelated procedures are actually equivalent to Newton's method. For example, with the “divide and average” technique taught to children for computing \sqrt{a} , the new estimate of the square root is given by

$$x_{k+1} = \frac{1}{2} \left(x_k + \frac{a}{x_k} \right);$$

this is precisely the next iterate of Newton's method applied to the function $f(x) = x^2 - a$.

The underlying assumption of Newton's method is that, for the purposes of zero-finding, f is “like” a straight line. One therefore expects Newton's method to work well when f satisfies this assumption. In particular, if $f'(x^*)$ is non-zero, and the starting point x_0 is “close enough” to x^* , Newton's method converges *quadratically* to x^* (see Theorem 1.1.2).

Example 1.1.1. When $\sqrt{4}$ is computed by Newton's method with starting point $x_0 = 2.5$, the errors $\{x_k^2 - 4\}$, $k = 0, \dots, 4$, are 2.25, .2025, 2.439×10^{-3} , 3.717×10^{-7} , and 8.0×10^{-16} (computed with approximately sixteen decimal digits of precision).

The major difficulty with Newton's method is that its spectacular convergence rate is only *local*. If x_0 is not close enough to x^* , the Newton iteration may diverge. For example, the Newton approximation depicted in Figure 4c lies at a completely unreasonable value. In addition, since the Newton iteration is undefined when $f'(x_k)$ is zero, numerical difficulties occur when $f'(x_k)$ is “small”. Newton's method must therefore be used with care, and it is not an all-purpose algorithm for zero-finding. The method is often used when a good initial estimate is known. For example, suppose we wish to compute the zero of some function $f(x, a)$ for a large number of values of a . Such a situation may arise in a loop, where a is also the result of a calculation. It is often the case that the *range* of values for a is known (it could be an angle). We can store a small number of solutions corresponding to a set of

values of a that covers the range of interest. When a zero for some specific value for a , say \bar{a} , is required we can use the zero corresponding to the nearest value to \bar{a} in the tabulated set as the initial point in a Newton iteration. By choosing a suitable number of stored values we can often be assured that Newton's method converges and does so in some known and small number of iterations. This dispenses with the need for a termination criterion. In the case of finding a square root there is no necessity to store any "known" solutions. We always know the integer n such that the unknown zero lies in the interval $[n, n + 1]$.

Theorem 1.1.1. (*Local convergence of Newton's method*). Let $f(x)$ be a univariate function that is continuously differentiable everywhere. Assume that $f(x^*) = 0$ for some x^* and that $f'(x^*) \neq 0$. Then there exists an open interval S such that, for any x_0 in S , the Newton iterates

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

are well defined, remain in S and converge to x^* .

Proof. Let α be a fixed constant in $(0, 1)$. Since f' is continuous at x^* and $f'(x^*)$ is non-zero, there is an open interval $S = (x^* - \epsilon, x^* + \epsilon)$ and a positive constant μ such that

$$\frac{1}{|f'(x)|} \leq \mu \quad \text{and} \quad |f'(y) - f'(x)| \leq \frac{\alpha}{\mu} \quad (1.1.2)$$

for every x and y in S . Suppose that $x_k \in S$. Since $f(x^*) = 0$, algebraic manipulation of (1.1.1) gives

$$x_{k+1} - x^* = -\frac{1}{f'(x_k)} (f(x_k) - f(x^*) - f'(x_k)(x_k - x^*)),$$

and hence the first bound in (1.1.2) implies

$$|x_{k+1} - x^*| \leq \mu |f(x_k) - f(x^*) - f'(x_k)(x_k - x^*)|.$$

From the integral mean-value theorem A1 we have

$$f(x_k) - f(x^*) - f'(x_k)(x_k - x^*) = \int_0^1 [f'(x^* + \xi(x_k - x^*)) - f'(x_k)] (x_k - x^*) d\xi.$$

Hence,

$$|x_{k+1} - x^*| \leq \mu \left\{ \max_{0 \leq \xi \leq 1} |f'(x^* + \xi(x_k - x^*)) - f'(x_k)| \right\} |x_k - x^*|. \quad (1.1.3)$$

Thus, the second bound in (1.1.2) gives

$$|x_{k+1} - x^*| \leq \alpha |x_k - x^*|$$

as long as $x_k \in S$. Since $\alpha < 1$, this last inequality implies that if $x_0 \in S$, then $x_k \in S$ for $k = 1, 2, \dots$, and that $\{x_k\}$ converges to x^* . ■

Theorem 1.1.2. (Rate of convergence of Newton's method). If the conditions of Theorem 1.1.1 hold, then the sequence $\{x_k\}$ produced by Newton's method converges q -superlinearly to x^* . Moreover, if $f'(x)$ is Lipschitz continuous in S , where S is defined by Theorem 1.1.1, with

$$|f'(x) - f'(x^*)| \leq \gamma|x - x^*|, \quad x \in S, \quad (1.1.4)$$

for some constant $\gamma > 0$, then the sequence converges q -quadratically to x^* .

Proof. The linear convergence of the sequence $\{x_k\}$ to x^* was established in Theorem 1.1.1. Define

$$\beta_k = \mu \left\{ \max_{0 \leq \xi \leq 1} \left| f'(x^* + \xi(x_k - x^*)) - f'(x_k) \right| \right\}, \quad (1.1.5)$$

and assume that $x_0 \in S$, with μ defined as in Theorem 1.1.1. The continuity of f' and the convergence of $\{x_k\}$ to x^* imply that β_k converges to zero. Since (1.1.3) can be written as

$$|x_{k+1} - x^*| \leq \beta_k |x_k - x^*|,$$

the definition of q -superlinear convergence to x^* is satisfied by $\{x_k\}$. Let ξ^* denote the value of ξ for which the maximum in (1.1.5) is achieved, and let y^* denote $x^* + \xi^*(x_k - x^*)$. Then

$$|f'(y^*) - f'(x_k)| \leq |f'(y^*) - f'(x^*)| + |f'(x^*) - f'(x_k)|. \quad (1.1.6)$$

Applying (1.1.4) in (1.1.6) and (1.1.5) gives

$$\beta_k \leq 2\mu\gamma|x_k - x^*|.$$

Hence, $\{x_k\}$ satisfies the definition of quadratic convergence to x^* . ■

1.1.1.3. Secant and regula falsi methods. A possible objection to Newton's method is that f' is required. In practical problems, f' may be expensive to evaluate when compared to f , or it may be unknown (a subroutine to evaluate f may be available but an analytical definition of f may not be known). A different method is suggested by using the same idea of approximating f by a straight line \hat{f} , but choosing as \hat{f} the straight line that passes through the values of f at the two most recent iterates; in essence, $f'(x_k)$ is replaced in the Newton formula by the finite-difference approximation $(f_k - f_{k-1})/(x_k - x_{k-1})$, where f_k denotes $f(x_k)$. The iterates are then defined by:

$$x_{k+1} = x_k - \left(\frac{x_k - x_{k-1}}{f_k - f_{k-1}} \right) f_k,$$

and the method is called the *secant method*, or the *method of linear interpolation*. As with Newton's method, we compute a new estimate of the zero rather than an interval of uncertainty. The values of f at two points are required to initiate the secant method.

Now we consider the convergence of the secant method. To simplify the presentation, we use the notation

$$f[x_k, x_{k-1}] = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}},$$

so that $f[x_k, x_{k-1}]$ denotes the divided-difference approximation to $f'(x_k)$ when $x_k \neq x_{k-1}$.

Theorem 1.1.3. (*Local convergence of the secant method*). *Let $f(x)$ be a univariate function that is continuously differentiable everywhere. Assume that $f(x^*) = 0$ for some x^* and that $f'(x^*) \neq 0$. Then there exists an open interval S such that, for any distinct $x_0, x_1 \in S$, the secant iterates*

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_{k-1}]} \quad (1.1.7)$$

are well defined, remain in S and converge to x^ .*

Proof. Let δ be a fixed constant in $(0, \frac{1}{3})$. Since f' is continuous at x^* and $f'(x^*)$ is non-zero, there is an open interval $S = (x^* - \epsilon, x^* + \epsilon)$ and a positive constant μ such that

$$\frac{1}{|f'(x)|} \leq \mu \quad \text{and} \quad |f'(y) - f'(x)| \leq \frac{\delta}{\mu} \quad (1.1.8)$$

for every x and y in S . Suppose that $x_k, x_{k-1} \in S$ and that $x_k \neq x_{k-1}$. Using the fact that $f(x^*) = 0$, algebraic manipulation of (1.1.7) gives

$$\begin{aligned} x_{k+1} - x^* &= -\frac{1}{f[x_k, x_{k-1}]} \left(f(x_k) - f(x^*) - f[x_k, x_{k-1}](x_k - x^*) \right), \\ &= -\frac{1}{f[x_k, x_{k-1}]} \left((f(x_k) - f(x^*) - f'(x_k)(x_k - x^*)) \right. \\ &\quad \left. - (f[x_k, x_{k-1}] - f'(x_k))(x_k - x^*) \right). \end{aligned} \quad (1.1.9)$$

From (A.1) and the second bound in (1.1.8), we obtain

$$|f(x_k) - f(x^*) - f'(x_k)(x_k - x^*)| \leq \frac{\delta}{\mu} |x_k - x^*|. \quad (1.1.10)$$

We now derive an upper bound on $|f[x_k, x_{k-1}] - f'(x_k)|$. Using (A.1), we have

$$f(x_{k-1}) - f(x_k) - f'(x_k)(x_{k-1} - x_k) = \int_0^1 (f'(x_k + \xi(x_{k-1} - x_k)) - f'(x_k))(x_{k-1} - x_k) d\xi.$$

The definition of $f[x_k, x_{k-1}]$ then gives

$$\left| f[x_k, x_{k-1}] - f'(x_k) \right| \leq \max_{0 \leq \xi \leq 1} \left| f'(x_k + \xi(x_{k-1} - x_k)) - f'(x_k) \right|,$$

so that

$$\left| f[x_k, x_{k-1}] - f'(x_k) \right| \leq \frac{\delta}{\mu}. \quad (1.1.11)$$

Finally, we derive a lower bound on $|f[x_k, x_{k-1}]|$, using (1.1.11):

$$\left|f[x_k, x_{k-1}]\right| \geq \left|\left|f'(x_k)\right| - \left|f'(x_k) - f[x_k, x_{k-1}]\right|\right| \geq \frac{(1-\delta)}{\mu}. \quad (1.1.12)$$

Using (1.1.10), (1.1.11) and (1.1.12) in (1.1.9), we obtain the overall bound:

$$|x_{k+1} - x^*| \leq \frac{2\delta}{(1-\delta)} |x_k - x^*|. \quad (1.1.13)$$

By assumption $x_0, x_1 \in S$ and are distinct. It follows by induction from (1.1.13) and $\delta < \frac{1}{3}$ that $x_k, x_{k-1} \in S$ and are distinct for $k = 2, \dots$, and that $\{x_k\}$ converges to x^* . ■

If $f'(x^*)$ is non-zero and x_0 and x_1 are sufficiently close to x^* , the secant method can be shown to have a superlinear convergence rate $r \approx 1.6180$, and thus it can be a rapidly convergent method. For example, when the secant method is applied to Example 1.1.1 with $x_0 = 1$, $x_1 = 2.5$, the sequence of errors $\{x_k^2 - 4\}$ for $k = 2, \dots, 7$, is -5.51×10^{-1} , -6.53×10^{-2} , 2.44×10^{-3} , -1.00×10^{-5} , -1.53×10^{-9} , and 8.88×10^{-16} (computed with approximately sixteen decimal digits of precision).

One difficulty with the secant method is that the iterates may diverge if the straight line approximation is an *extrapolation* (i.e., when f_k and f_{k-1} have the same sign). If f_k and f_{k-1} were always of opposite sign, the predicted zero would necessarily lie strictly between x_{k-1} and x_k . This suggests a modification of the secant method—the *method of false position*, or *regula falsi*, in which x_{k+1} replaces either x_k or x_{k-1} , depending on which corresponding function value agrees in sign with f_{k+1} . The two initial points x_0 and x_1 must be chosen such that $f_0 f_1 < 0$. In this way the zero remains bracketed, and there is no danger from extrapolation. Unfortunately, although this modification of the secant method guarantees convergence, its rate of convergence is only linear, and the asymptotic error constant can be arbitrarily close to unity. In Figure 4d we show an example where x_0 will never be discarded, since $f_i < 0$ for $i \geq 1$. For the convergence rate to be superlinear we must have the divided difference converge to $f'(x^*)$. When one of the points used is always a poor approximation to x^* it follows the divided difference is a poor approximation to $f'(x^*)$ and hence we may get slow convergence. In general the method of false position may be much less efficient than bisection. This illustrates that care is necessary when attempting to ensure convergence, in order not to destroy the rapid rate of convergence of the original method.

1.1.1.4. Rational interpolation and higher-order methods. Other procedures for well-behaved functions can be developed by constructing approximations based on the values of f and its derivatives at any number of known points. However, higher-order schemes based upon polynomial interpolation have the disadvantage that one particular zero of the polynomial must be selected for the next iterate. Also in general the zeros of the polynomial

must themselves be found by iteration. There are some approximating functions without such drawbacks. For example, the rational interpolation function of the form

$$\hat{f} = \frac{x - c}{d_0 + d_1x + d_2x^2}.$$

The values of c , d_0 , d_1 , and d_2 are chosen so that the function value and derivatives of \hat{f} agree with those of f at two points. If derivatives are not available, a rational interpolation function \hat{f} without the x^2 term can be computed from the values of f at three points.

1.1.1.5. Safeguarded zero-finding algorithms. The bisection algorithm can be viewed as a technique for constructing a set of intervals $\{I_j\}$, each containing x^* , such that the interval I_j lies wholly within I_{j-1} , and the length of each interval is strictly smaller than that of its predecessor. Mathematically this process can be expressed as: *given I_0 such that $x^* \in I_0$, for $j = 1, \dots$, find $\{I_j\}$ such that $I_j \subset I_{j-1}$ and $x^* \in I_j$.* Note that, given an interval I_j , knowledge of the sign of f at a single interior point (say, u) in I_{j-1} enables a new interval I_j to be found, assuming that the sign of f is known at the end points of I_{j-1} . Methods that generate a set of nested intervals in this manner are known as *bracketing methods*.

With the bisection algorithm, u is obtained simply by bisecting I_{j-1} . However, we may just as easily compute u using Newton's method, linear approximation or rational approximation. Furthermore, the approximating function need not be based on the values of f at the end points of the interval of uncertainty, but rather could use the "best" points found so far. For example, in Figure 4d the points x_1 and x_2 could be used for linear interpolation, even though the interval of uncertainty is $[x_2, x_0]$.

The best methods available for zero-finding are the so-called *safeguarded* procedures. The idea is to combine a guaranteed, reliable method (such as bisection) with a fast-convergent method (such as linear or rational interpolation), to yield an algorithm that will converge rapidly if f is well-behaved, but is not much less efficient than the guaranteed method in the worst case.

A safeguarded linear interpolation method might include the following logic at each iteration. An interval of uncertainty is known, say $[a, b]$. The two "best" points found so far are used to obtain a point u by linear interpolation. Without safeguards, the next step would involve evaluating $f(u)$, and discarding one of the old points in order to form a new set of points for the linear fit. However, a safeguarded procedure ensures that u is a "reasonable" point before evaluating f at u .

The definition of "reasonable" is complicated to define if an algorithm is to be efficient. However, there are some obvious requirements. Firstly, u must lie in $[a, b]$. Secondly, even if u is in $[a, b]$ it should be rejected if it is too close to a or b . (The sequence of interpolants may be converging to an end point.)

Finally, safeguards are necessary to ensure that successive iterates are not "too close". It can happen that u is numerically indistinguishable from the previous best point, even when

u is far from optimal. If an extrapolation is performed with nearly equal iterates, rounding error may cause the predicted point to be poorly defined and hence meaningless. The most common technique used to prevent this phenomenon is to specify a “small” distance δ , and to “take a step of magnitude δ ” whenever a newly generated point is too close to the current best point. The step is taken in the direction that will result in the largest reduction of the interval of uncertainty. The use of this technique will usually cause the final two steps of any safeguarded algorithm to be of length δ , and x^* will therefore lie in an interval of length 2δ .

When a pure interval-reducing strategy is employed it is clear that bisection is the “best”. When an interval-reducing *step* is required it is not clear that the bisection step is necessarily best. If the two best points remain the same after the interval-reducing step then we shall be forced in the next iteration to take another interval-reducing step. We have an additional consideration, which is to try and replace a point used in the interpolation. This may not be easy to do so we also need to consider that we may take only interval-reducing steps. A good strategy needs to take into account the nature of the current best points and those defining the interval of uncertainty. For example, when one of the points that forms the interval is poor compared to the best points then a bisection step is likely to also yield a poor point. Under such circumstances it is better to choose an interval-reducing step that is smaller than the bisection step.

1.1.1.6. Equivalent problems.

A first step before applying any algorithm is to see whether the problem may be transformed into one for which the algorithms we have at our disposal are better suited. For example, an equivalent problem to $f(x) - a = 0$ is

$$\bar{f}(x) = \frac{1}{f(x)} - \frac{1}{a} = 0.$$

The function \bar{f} may be more like a linear function than $f(x)$. The guiding light is to transform the problem into one that is closer to finding the zero of a linear function. If $f(x)$ has a singularity in the interval of interest or one that is close to the zero then transforming the problem may be of great value. Although Newton’s method may ultimately converge at a quadratic rate on the original problem the interval in which quadratic convergence manifests itself will be very small.

1.1.2. Univariate Minimization

Now we consider the minimization of a univariate function $f(x)$ over some bounded interval $[a, b]$.

Definition 1.1. A point x^* is a local minimizer of $f(x)$ if there is a neighborhood $N(x^*)$ such that

$$f(x^*) \leq f(x) \quad \text{for all } x \in N(x^*), \quad x \neq x^*. \quad (1.1.14)$$

A minimizer is known as a strong or strict minimizer if (1.1.14) holds with strict inequality, i.e.,

$$f(x^*) < f(x) \quad \text{for all } x \in N(x^*), \quad x \neq x^*.$$

A similar definition defines a local maximizer. Note that finding a maximizer of a function f is equivalent to finding a minimizer of the function $-f$.

Definition 1.2. A point x^* is a global minimizer of $f(x)$ for $x \in [a, b]$ if

$$f(x^*) \leq f(x) \quad \text{for all } x \in [a, b].$$

We shall be almost exclusively concerned with determining local minimizers.

In the case of finding a zero of a function it is self evident whether a point is indeed a zero of f . Unfortunately this is not the case of a minimizer. In order to determine a point is a minimizer it is necessary (and usually sufficient) to examine higher derivatives at the point in question. When such derivatives do not exist we cannot in general make such a distinction. In the next lemma and theorem we establish what are known as the first-order and second-order conditions on a local minimizer. The difficulty of distinguishing whether a point is a global minimizer is that information at a single point is not sufficient.

Lemma 1.1.1. Let $f \in C^1$ in an open interval D and let $x \in D$. If $f'(x) \neq 0$, then for any s such that $sf'(x) < 0$, there is a constant α ($\alpha > 0$) for which $f(x + \sigma s) < f(x)$, for all $\sigma \in (0, \alpha)$.

Proof. The continuity of $f'(x)$ ensures the existence of a positive number α such that $x + \alpha s \in D$ and $sf'(x + \sigma s) < 0$ for all $\sigma \in (0, \alpha)$. The result then follows immediately from the Integral Mean-Value Theorem, since

$$f(x + \sigma s) - f(x) = s \int_0^\sigma f'(x + \xi s) d\xi < 0.$$

■

Theorem 1.1.4. (Necessary conditions for a local univariate minimum). If $f(x) \in C^2$ in an open interval D , then $x^* \in D$ is a local minimizer of $f(x)$ only if

$$f'(x^*) = 0 \quad \text{and} \quad f''(x^*) \geq 0.$$

Proof. Using Lemma 1.1.1, we see that $f'(x^*)$ must vanish, since otherwise any sufficiently small neighborhood of x^* would include points with lower values of $f(x)$. The second-order necessary condition is also proved by contradiction. Form (A.2) of the second-order Integral Mean-Value theorem gives

$$f(x^* + s) = f(x^*) + sf'(x^*) + \frac{1}{2}s^2 f''(x^*) + s^2 \int_0^1 (f''(x^* + \xi s) - f''(x^*)) (1 - \xi) d\xi.$$

Since $f'(x^*) = 0$, this expression may be rearranged when $s \neq 0$ to give

$$\frac{f(x^* + s) - f(x^*)}{s^2} = \frac{1}{2}f''(x^*) + \int_0^1 (f''(x^* + \xi s) - f''(x^*)) (1 - \xi) d\xi. \quad (1.1.15)$$

If $f''(x^*) < 0$ the continuity of f'' implies that, for all sufficiently small $|s|$, the integral on the right-hand side of (1.1.15) will be so small that the right-hand side is strictly negative. Hence, $f(x^*) > f(x^* + s)$ for all sufficiently small $|s|$, which contradicts that x^* is a minimizer of f . Consequently, x^* cannot be a local minimizer if $f''(x^*) < 0$. ■

Theorem 1.1.5. (Sufficient conditions for a local univariate minimum). Let $f(x) \in C^2$ in an open interval D . If $f'(x^*) = 0$ and $f''(x^*) > 0$, then $x^* \in D$ is a strong local minimizer of f .

Proof. Since $f''(x)$ is continuous, for any specified $\lambda > 0$ there exists a positive ϵ such that, for every s such that $|s| < \epsilon$,

$$|f''(x^* + s) - f''(x^*)| < \frac{1}{2}\lambda.$$

Let $\lambda = f''(x^*)$.

Using form (A.2) of the second-order Integral Mean-Value theorem as in Lemma 1.1.1, we obtain

$$\begin{aligned} \frac{f(x^* + s) - f(x^*)}{s^2} &= \frac{1}{2}f''(x^*) + \int_0^1 (f''(x^* + \xi s) - f''(x^*)) (1 - \xi) d\xi \\ &\geq \frac{1}{2}\lambda - \frac{1}{2} \max_{0 \leq \xi \leq 1} |f''(x^* + \xi s) - f''(x^*)| \\ &> \frac{1}{4}\lambda. \end{aligned}$$

Therefore, $f(x) > f(x^*)$ for all x such that $|x - x^*| < \epsilon$ and $x \neq x^*$. ■

Theorem 1.1.6. (Necessary and sufficient conditions for a local univariate minimum). Let $f(x) \in C^\infty$ in an open interval D . Let p denote the integer such that

$$\frac{d^r f(x^*)}{dx^r} = 0, \quad r = 1, 2, \dots, p-1 \quad \text{and} \quad \frac{d^p f(x^*)}{dx^p} = \theta \neq 0$$

then x^* is a local minimizer of f if and only if p is even and $\theta > 0$.

It is possible no such p exists. Under such circumstances one cannot distinguish a minimizer by examining derivatives at x^* . It will be seen later that generalizing the necessary and sufficient conditions to n -dimensions proved extremely difficult. Many eminent mathematicians, including Lagrange, got the conditions wrong.

A point for which $f' = 0$ is known as a *stationary point*. It follows from the first-order necessary condition that a minimizer is a stationary point. However, in general we cannot

determine a minimizer by using an algorithm to find a zero of f' since other points such as a maximizer are also a zero of f' . In order to distinguish a minimizer from other stationary points it is necessary to examine higher derivatives. In some cases such derivatives may not be known and we are unable to make the distinction. It may be thought that under such circumstances a good strategy would be to attempt to find a minimizer by using an algorithm to determine a zero of f' . Even in these circumstances it is in general an ill advised strategy since it may result in determining a stationary point at which f is higher than at the initial estimate. The methods used to solve the problem of univariate minimization in a bounded interval are analogous to those for zero-finding. However, our definition of “best” is not based on the magnitude of “ $|f'|$ ”, but on f .

If $f \notin C^1$ then f' may not exist at x^* , and in such circumstances the special relationship with zero-finding no longer applies. In general we shall not study algorithms for this class of problem. However, interval-reduction methods based only on function values still work for this case.

To develop an interval-reduction procedure for minimization when only function values are available, we need to define a condition that ensures there is a minimizer in a given interval. For this purpose, we introduce the concept of *unimodality*, of which there are several definitions in the literature. One practical definition is: $f(x)$ is unimodal in $[a, b]$ if there exists a unique $x^* \in [a, b]$ such that, given any $x_1, x_2 \in [a, b]$ for which $x_1 < x_2$:

$$\text{if } x_2 < x^* \text{ then } f(x_1) > f(x_2); \quad \text{if } x_1 > x^* \text{ then } f(x_1) < f(x_2).$$

If f is known to be unimodal in $[a, b]$, it is possible to reduce the interval of uncertainty by comparing the values of f at *two* interior points. For example, in Figure 4e (Practical Optimization), the minimizer must lie in the reduced interval $[x_1, b]$ or $[a, x_2]$, depending on the relative values of $f(x_1)$ and $f(x_2)$.

The basic approach adopted is given an interval of uncertainty we evaluate the function within the interval. The need to assume the function is unimodal arises because for an arbitrary function evaluating the function within the interval does not necessarily reduce the known interval (contrast this with the zero-finding algorithm). Suppose we have three points and the middle point has a lower function value than the end points. We can assert a minimizer lies between the end points and as such we know an interval of uncertainty. If we now evaluate the function in the interval then provided the new function value does not equal the function value at the other interior point, we will identify a new and smaller interval of uncertainty. In general we would be unlucky if it was an identical function value. However, unless we assume something like the property of unimodality we cannot be assured it will not happen repeatedly. In practice we will not usually know if a function is unimodal. However, in a neighborhood of a minimizer all $f \in C^2$ are unimodal. Therefore, our concern on general functions is only whether for disparate values of x we repeatedly obtain the same value of f .

1.1.2.1. Fibonacci search. The “optimum” strategy (i.e., the strategy that yields the maximum reduction in the interval of uncertainty for a given number of function evaluations) is termed *Fibonacci search*. It is based upon the Fibonacci numbers $\{F_i\}$, which satisfy

$$F_k = F_{k-1} + F_{k-2}, \quad F_0 = F_1 = 1.$$

The first few values of the sequence are 1,1,2,3,5,8 and 13. An alternative way of looking at the issue is to ask the question: Given $N + 1$ function evaluations what is the largest interval one can start with and end up with an interval of uncertainty of one? The answer is F_N and the evaluation points are F_0, F_1, \dots, F_N if the minimizer is in $[0, 1]$. Since F_0 and F_1 are both equal to 1 the latter is replaced by $1 + \delta$, where δ is very small. We illustrate this process by considering the case where just two function evaluations are performed. If the original interval is $[0, F_2]$ ($F_2 = 2$), f is evaluated at two points corresponding to F_0 and F_1 , i.e. at $x_1 = 1$ and $x_2 = 1 + \delta$. Regardless of the function values $f(x_1)$ and $f(x_2)$, the new interval of uncertainty is thereby reduced to arbitrarily close to half its original length (see the first diagram in Figure 4f).

Figure 4f depicts the case when three function evaluations are allowed. Here, the original interval of uncertainty is considered to be $[0, F_3]$ ($F_3 = 3$), and the first two points are placed at F_1 and F_2 (i.e., $x_1 = 1$ and $x_2 = 2$). Depending on the values of $f(x_1)$ and $f(x_2)$, the reduced interval will either be $[0, 2]$, or $[1, 3]$. Since the function value at the midpoint of the reduced interval is already available, only one additional evaluation is required to obtain a final interval of length $1 + \delta$.

When N function evaluations are allowed, a Fibonacci search procedure will essentially produce a final interval of uncertainty of length $1/F_N$ times the length of the original interval.

1.1.2.2. Golden section search. A disadvantage of Fibonacci search is that it cannot be adapted to the case when the termination criterion requires that the function values in the final interval of uncertainty differ by less than a specified amount.

A procedure that does not require the *a priori* selection of the final interval of uncertainty and is almost as efficient as Fibonacci search is *golden section search*. It can be shown that

$$\lim_{k \rightarrow \infty} \frac{F_{k-1}}{F_k} = \frac{2}{1 + \sqrt{5}} \equiv \tau \approx .6180,$$

where τ satisfies the quadratic equation $\tau^2 + \tau - 1 = 0$. With a golden section search procedure, if the initial interval is considered to be $[0, 1]$, the two points are placed at τ and $1 - \tau$ (i.e., at approximately .6180 and .3820). No matter how the interval is reduced, one of the old points will then be in the correct position with respect to the new interval; golden section search may thus be viewed as the limiting case of Fibonacci search. Figure 4g illustrates the configuration of these points.

With golden section search, there is a constant reduction of the interval of uncertainty by the factor τ at every step, and the length of the interval of uncertainty converges linearly to zero.

1.1.2.3. Polynomial interpolation. Golden section search and Fibonacci search are similar in the sense that each permits only two possible choices for the next interval of uncertainty. Moreover, the interval chosen is based solely on a comparative test on the last computed value of f . As in the zero-finding case, more efficient procedures can be developed for smooth functions by utilizing known values of f to define the next iterate. In particular, f can be approximated by a simple function \hat{f} whose minimizer is easy to compute, and the minimizer of \hat{f} can be used iteratively as an estimate of the minimizer of f . Since a general straight line has no minimum, we consider approximating f by a parabola (quadratic), of the form:

$$\hat{f} = \frac{1}{2}ax^2 + bx + c.$$

If $a > 0$, \hat{f} has a minimum at x^* such that $ax^* + b = 0$.

Three independent pieces of information are required in order to construct the quadratic approximation \hat{f} . For example, at a particular point x_k , given $f(x_k)$, $f'(x_k)$ and $f''(x_k)$, \hat{f} can be defined by the first three terms of the Taylor-series expansion, i.e.,

$$\hat{f}(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2.$$

If $f''(x_k)$ is non-zero, \hat{f} has a stationary point at x_{k+1} such that

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

or equivalently

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \quad (1.1.16)$$

The formula (1.1.16) is equivalent to Newton's method (1.1.1) applied to finding a zero of $f'(x)$. Note that in order to determine the minimizer of \hat{f} it is not necessary to know $f(x_k)$. However, when we combine the above procedure with safeguarding techniques it is essential that $f(x_k)$ be known.

A quadratic approximation \hat{f} can also be fitted to three function values. When first, but not second, derivatives are known, the values of f and f' at two points provide four independent pieces of data, and \hat{f} may be taken as a cubic polynomial.

Methods based upon polynomial fitting can be shown to have superlinear convergence under suitable conditions; for example, the rate of convergence for parabolic fitting with three function values is approximately 1.324, and the cubic fitting technique has quadratic convergence. However, because these methods are based on a well-behaved simple model, they are subject to the same difficulties as the rapidly converging zero-finding techniques. If \hat{f} is not an accurate representation of the behavior of f , the minimizer of \hat{f} may be a poor estimate—for example, the predicted minimizer may lie outside the initial interval of uncertainty, or \hat{f} may be unbounded below. In order to avoid such difficulties, polynomial fitting strategies can be modified so that the minimizer is always bracketed. For example,

in cubic fitting, the gradient can be required to be of opposite sign at the two points, in which case the minimizer of the fitted cubic will lie in the original interval. Unfortunately, as with the similarly motivated modification of the secant method, this attempt to improve robustness destroys the superlinear convergence rate. With such a method, the convergence rate is linear, and can be arbitrarily slow.

1.1.2.4. Safeguarded polynomial interpolation. As in zero-finding, the best general methods are the so-called *safeguarded* procedures. In this case, polynomial interpolation can be combined with bisection (when derivatives are known) or golden section search (when only function values are available). A safeguarded method based upon parabolic interpolation with three function values requires an interval of uncertainty, say $[a, b]$, and the three “best” points found so far. Suppose that u is the step, computed by parabolic interpolation, from the best point x . Let \bar{u} denote the “approximate” golden section step

$$\bar{u} = \begin{cases} \beta(a - x) & \text{if } x \geq \frac{1}{2}(a + b); \\ \beta(b - x) & \text{if } x < \frac{1}{2}(a + b), \end{cases}$$

where $\beta = 1 - \frac{1}{2}(\sqrt{5} - 1) = 0.382$. During the next iteration \bar{u} will be used instead of u if $|u| > |\bar{u}|$. Safeguards are also included to prevent iterates from being too close (see Section 1.1.1.5).