Module – 5: Numerical Methods for Unconstrained Optimum Design

Derivative-based search methods also known as gradient-based search methods, are iterative where the same calculations are repeated in each and every iteration. In such approaches, we estimate an initial design and improve it iteratively, until optimality conditions are satisfied. Many numerical methods have been developed for NLP problems.

The unconstrained optimization problems are classified as 1D (one-dimensional) and multidimensional problems, as shown in Fig. 5.1. The 1D problems involve only one design variable and the multidimensional ones involve many design variables. Although 1D problems are important in their own right, multidimensional problems are reduced to 1D as well in their numerical solution process. This happens when a step size needs to be calculated in desired search direction in order to improve the current design. Therefore many methods have been developed for 1D minimization problems because this is one major calculation in optimization methods, for multidimensional problems. One-dimensional search is usually called line search or 1D search.

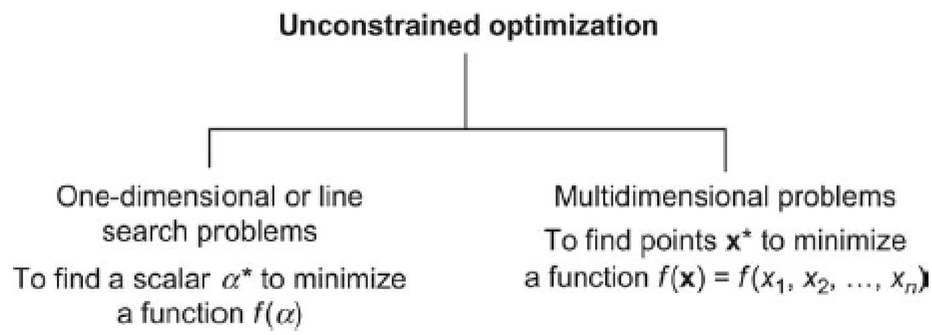

Figure 5.1: Classification of unconstrained optimization problems.

A GENERAL ITERATIVE ALGORITHM

In this section, we first discuss some general concepts related to iterative numerical methodfor optimization and then give a general step-by-step algorithm. Many gradient-based optimizationmethods are described by the following iterative prescription:

Vector form:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \; ; k = 0, 1, 2, \ldots \tag{5.1}$$

Component form:

$$x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}; \; i = 1 \text{ to } n; \; k = 0, 1, 2, \ldots \tag{5.2}$$

Where,

k = superscript representing the iteration number;

i= subscript denoting the design variable number;

$\mathbf{x}(0)$ = starting point;

$\Delta\mathbf{x}(k)$ = change in the current point.

The iterative scheme described in Eq. 5.1or 5.2is continued until the optimalityconditions are satisfied or a termination criterion is met. This iterative scheme is applicableto constrained as well as unconstrained problems. For unconstrained problems, calculationsfor $\Delta\mathbf{x}^{(k)}$ depend on the cost function and its derivatives at the current design point. Forconstrained problems, the constraints must also be considered while computing the changein design $\Delta\mathbf{x}^{(k)}$. Therefore, in addition to the cost function and its derivatives, the constraintfunctions and their derivatives play a role in determining $\Delta\mathbf{x}^{(k)}$. There are several methodsfor calculating $\Delta\mathbf{x}^{(k)}$ for unconstrained and constrained problems. This chapter focuses onmethods for unconstrained optimization problems.

In most methods, the change in design $\Delta\mathbf{x}^{(k)}$ is further decomposed into two parts as:

$$\Delta\mathbf{x}^{(k)}=\alpha_k\mathbf{d}^{(\mathbf{k})}$$

Where,

$\mathbf{d}^{(k)}$ = “desirable” search direction in the design space; and

α_k = *a positive* scalar calledthe *step size* in the search direction.

If the direction $\mathbf{d}^{(k)}$ is any “good,” then the step size must be greater than 0; this will becomeclearer when we relate the search direction to a descent direction for the cost function. Thus,the process of computing $\Delta\mathbf{x}(k)$ involves solving two separate subproblems:

1. the direction-finding subproblem

2. the step length determination subproblem (scale factor for the direction)

The process of moving from one design point to the next, is illustrated in Fig. 5.2

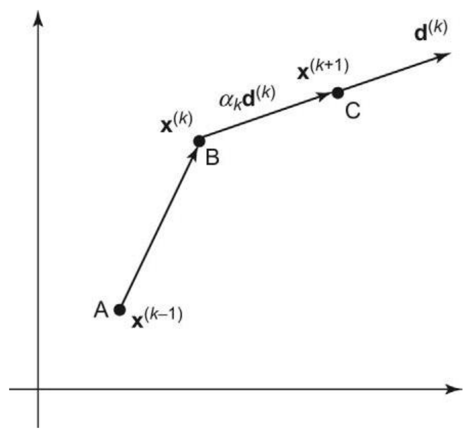

Figure 5.2: Conceptual diagram for iterative steps of an optimization method

In Fig. 5.2, B is the current design point $\mathbf{x}^{(k)}$, $\mathbf{d}^{(k)}$ is the search direction, and $\alpha_k$is a steplength. Therefore, when $\alpha_k\mathbf{d}^{(\mathbf{k})}$is added to the current design $\mathbf{x}^{(k)}$, we reach a new point C inthe design space, $\mathbf{x}^{(k+1)}$. The entire process is repeated from point C. There are many proceduresfor calculating the step size $\alpha_k$and the search direction $\mathbf{d}^{(k)}$. Various combinations ofthese procedures are used to develop different optimization algorithms.

5.1. Direct Method General algorithm for unconstrained minimization methods

The iterative process just described represents an organized search through the designspace for points that represent local minima for the cost function. The process is summarizedas a *general algorithm* that is applicable to both constrained and unconstrained problems:

Step 1: Estimate a reasonable starting design $\mathbf{x}^{(0)}$. Set the iteration counter $k = 0$.
Step 2: Compute a search direction $\mathbf{d}^{(k)}$ at the point $\mathbf{x}^{(k)}$ in the design space. This calculationgenerally requires a cost function value and its gradient for unconstrained problems and,in addition, constraint functions and their gradients for constrained problems.
Step 3: Check for convergence of the algorithm. If it has converged, stop; otherwise, continue.
Step 4: Calculate a positive step size *ak*in the direction $\mathbf{d}^{(k)}$.
Step 5: Update the design as follows, set $k = k + 1$, and go to step 2.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \tag{5.3}$$

In the remaining sections of this chapter, we present some basic methods for calculatingthe step size α_k and the search direction $\mathbf{d}^{(k)}$ for unconstrained optimization problems toimplement the aforementioned general algorithm.

DESCENT DIRECTION AND CONVERGENCE OF ALGORITHMS

The unconstrained minimization problem is defined as finding x to:
minimize

$$f(x) \tag{5.4}$$

Since we want to minimize the cost function, the idea of a descent step is introduced, which simply means that changes in the design at every search step must reduce the cost function value. Convergence of an algorithm and its rate of convergence are also briefly described.

Descent Direction and Descent Step

We have referred to $d^{(k)}$ as a desirable direction of design change in the iterative process. Now we discuss what we mean by a desirable direction. The objective of the iterative optimization process is to reach a minimum point for the cost function $f(x)$.

Let us assume that we are in the kth iteration and have determined that $x^{(k)}$ is not a minimum point; that is, the optimality conditions of Theorem 4.4 are not satisfied. If $x^{(k)}$ is not a minimum point, then we should be able to find another point $x^{(k+1)}$ with a smaller cost function value than the one at $x^{(k)}$. This statement can be expressed mathematically as

$$f(x^{(k+1)}) < f(x^{(k)}) \tag{5.5}$$

Substitute $x^{(k+1)}$ from Eq. (5.3) into the preceding inequality to obtain

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)}) \tag{5.6}$$

Approximating the left side of Eq. (5.6) by the linear Taylor's expansion at the point $x^{(k)}$, we get

$$f(x^{(k)}) + \alpha_k (c^{(k)} \cdot d^{(k)}) < f(x^{(k)}) \tag{5.7}$$

where $c^{(k)} = \nabla f(x^{(k)})$ is the gradient of $f(x)$ at the point $x^{(k)}$; and (a .b) is dot product of the vectors **a** and **b**. Subtracting $f(x^{(k)})$ from both sides of inequality Eq. (5.7), we get $\alpha_k(c^{(k)} \cdot d^{(k)}) < 0$. Since $\alpha_k > 0$, it may be dropped without affecting the inequality. Therefore, we get the condition

$$(c^{(k)} \cdot d^{(k)}) < 0 \tag{5.8}$$

Since $c^{(k)}$ is a known vector (the gradient of the cost function), only unknown in Eq. (5.8) is the search direction $d^{(k)}$. Thus a desirable direction must satisfy inequality in Eq. (5.8).

5.2. Rate of convergence

The unconstrained minimization problem is defined as finding **x** to:

Minimize

$$f(\mathbf{x}) \tag{5.4}$$

Since we want to minimize the cost function, the idea of a *descent step* is introduced, whichsimply means that *changes in the design at every search step must reduce the cost function value*. The convergence of an algorithm and its rate of convergence are also briefly described.

The central idea behind numerical methods of optimization is to search for the optimumpoint in an iterative manner, generating a sequence of designs. It is important to note that thesuccess of a method depends on the guarantee of convergence of the sequence to the optimumpoint. The property of convergence to a local optimum point irrespective of the starting pointis called *global convergence* of the numerical method. It is desirable to employ such convergentnumerical methods because they are more reliable. F or unconstrained problems, a convergentalgorithm must reduce the cost function at each and every iteration until a minimum point isreached. It is important to note that the algorithms converge to a local minimum point only, asopposed to a global minimum, since they use only local information about the cost functionand its derivatives in the search process. Methods to search for global minima are describedin chapter: Global Optimization Concepts and Methods.

Rate of Convergence: In practice, a numerical method may take a large number of iterations to reach an optimumpoint. Therefore, it is important to employ methods having a convergence rate that is faster. An algorithm’s rate of convergence is usually measured by the number of iterations and functionevaluations needed to obtain an acceptable solution. *Rate of convergence is a measure of how fastthe difference between the solution point and its estimates, goes to zero*. Faster algorithms usually use second-order information about the problem functions while calculating the search direction.They are known as *Newton methods*. Many algorithms also approximate second-order informationusing only first-order information. They are known as *quasi-Newton methods.*

5.3. Unimodal and Nonunimodal function

The numerical line search process to calculate a step size is itself iterative, requiring severaliterations before a minimum point for $f\,(\alpha)$ is reached. Many line search techniques are based on comparing function values at several points along the search direction. Usually, wemust make some assumptions on the form of the line search function $f\,(\alpha)$ to compute stepsize by numerical methods. For example, it must be assumed that a minimum exists and thatit is unique in some interval of interest. A function with this property is called the *unimodal function*. Fig. 5.3 shows the graph of such a function that decreases continuously until the

minimum point is reached. Comparing Fig. 5.2 and 5.3, we observe that $f\,(\alpha)$ is a unimodal function in some interval. Therefore, it has a unique minimum point in that interval.*Most 1D search methods* assume the line search function to be a unimodal function in some interval. This may appear to be a severe restriction on the methods; however, it is not. Forfunctions that are not unimodal, we can think of locating only a local minimum point that isclosest to the starting point (ie, closest to α= 0). This is illustrated in Fig. 10.5, where the function$f\,(\alpha)$ is not unimodal for $0 \le \alpha \le \alpha_0$. Points A , B , and C are local minima. However, if werestrict a to lie between 0 and $\bar{\alpha}$there is only one local minimum point A because the function

$f(a)$ is unimodal for $0 \le \alpha \le \bar{\alpha}$. Thus, the assumption of unimodality is not as restrictive as itmay appear.

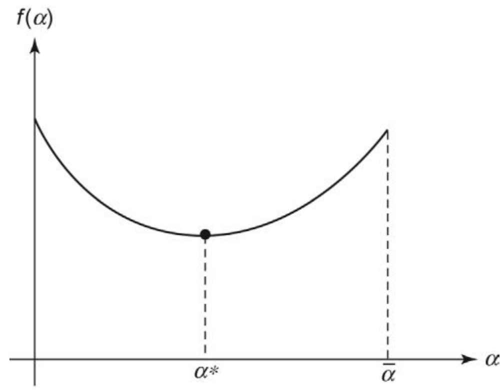

Figure 5.3: Unimodal Function

Most 1D search methods assume the line search function to be a unimodal function in some interval. This may appear to be a severe restriction on the methods; however, it is not. Forfunctions that are not unimodal, we can think of locating only a local minimum point that isclosest to the starting point (i.e., closest to $\alpha = 0$). This is illustrated in Fig. 5.4, where the function$f(\alpha)$ is not unimodal for $0 \leq \alpha \leq \alpha_0$. Points A , B , and C are local minima. However, if werestrict *a* to lie between 0 and $\bar{\alpha}$there is only one local minimum point A because the function$f(\alpha)$ is unimodal for $0 \leq \alpha \leq \bar{\alpha}$.. Thus, the assumption of unimodality is not as restrictive as itmay appear.

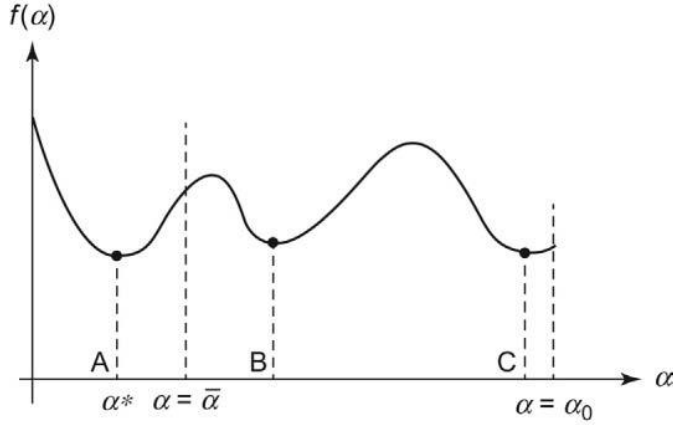

Figure 5.4: Nonunimodal function $f(\alpha)$ for $0 \leq \alpha \leq \alpha_0$ (unimodal for $0 \leq a \leq \bar{\alpha}$).

5.4. Reduction of a single variable

For an optimization problem with several variables, the direction-finding subproblem must be solved first. Then, a step size must be determined by searching for the minimumof the cost function along the search direction. This is always a 1D minimization problem,also called a *line search* problem. To see how the line search will be used in multidimensionalproblems, let us assume for the moment that a search direction $\mathbf{d}^{(k)}$ has been computed. Then, in Eqs. (5.1) and (5.3), scalar $\alpha_k$is the only unknown.

Since the best step size $\alpha_k$is yet unknown, we replace it by αin Eq. (5.3), which is then treated as an unknown in the step size calculation subproblem. Then, using Eqs. (5.1) and (5.3), the cost function $f(\mathbf{x})$ at the new point $\mathbf{x}^{(k+1)}$ is given as $f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)} + \alpha^{(k)})$. Now,since $\mathbf{d}^{(k)}$ is known, the right side becomes a function of the scalar parameter αonly. This processis summarized in the following equations:

***Design update*:**

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \_\mathbf{d}^{(k)} \tag{5.5}$$

***Cost function evaluation*:**

$$f(\mathbf{x}^{(k+1)}) = (f(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}) = \bar{f}(\alpha) \tag{5.6}$$

Where, $\bar{f}(\alpha)$is the new function, with αas the only independent variable (in the sequel, wewill drop the over bar for functions of a single variable). Note that at $\alpha = 0$, $f(0) = f(\mathbf{x}^{(k)})$ fromEq. (5.6), which is the current value of the cost function.

It is important to understand this reduction of a function of n variables to a function of only one variable since this procedure is used in almost all gradient-based optimization methods. It is also important to understand the geometric significance of Eq. (5.6).

5.5. One-dimensional minimization methods

1D Minimization Problem:

If $x^{(k)}$ is not a minimum point, then it is possible to find a descent direction $d^{(k)}$ at the point and reduce the cost function further. Recall that a small move along $d^{(k)}$ reduces the cost function. Therefore, using Eqs. (10.6) and (10.11), the descent condition for the cost function can be expressed as the inequality:

$$f(\alpha) < f(0) \tag{10.12}$$

Since f(a) is a function of single variable (also called the line search function), we can plot f(a) versus a. To satisfy inequality Eq. (10.12), the curve $f(\alpha)$ versus a must have a negative slope at the point $\alpha = 0$. Such a function is shown by the solid curve in Fig. 10.3. It must be understood that if the search direction is that of descent, the graph of f(a) versus a cannot be the one shown by the dashed curve because a small positive a would cause the function f(a) to increase, violating inequality Eq. (10.12). This will also be a contradiction as $d^{(k)}$ is a direction of descent for the cost function.

Therefore, the graph of $f(\alpha)$ versus a must be the solid curve in Fig. 10.3 for all problems. In fact, the slope of the curve $f(\alpha)$ at $\alpha = 0$ is calculated by differentiating Eq. (10.11) as $f'(0) = (c^{(k)}.d^{(k)})$, which is negative, as seen in Eq. (10.9). This discussion shows that if $d^{(k)}$ is a descent direction, then a must always be a positive scalar in Eq. (10.3). Thus, the step size determination subproblem is to find a to:

minimize$f(\alpha)$

Solving this problem gives the step size $\alpha_k = \alpha^*$ for use in Eq. (10.3)

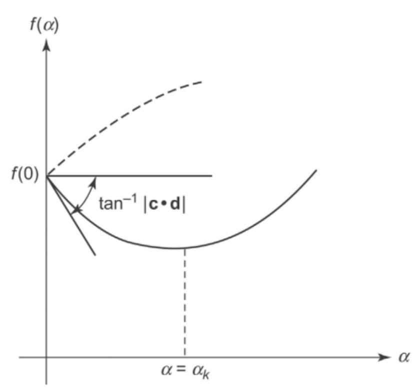

Figure 5.5: Graph of $f(\alpha)$ versusα.

5.6. Equal Interval method

Interval-Reducing Methods

The line search problem, then, is to find α in an interval $0 \le \alpha \le \alpha_o$ at which the function $f(\alpha)$ has a global minimum. However, this statement of the problem requires some modification. Since we are dealing with numerical methods, it is not possible to locate the exact minimum point α^*. In fact, *what we determine is the interval in which the minimum* lies—some lower and upper limits α_l and α_u for α^*. The interval (α_l, α_u) is called the *interval of uncertainty* and is designated as

$$I = \alpha_u - \alpha_l \qquad (10.14)$$

Most numerical methods iteratively reduce this interval of uncertainty until it satisfies a specified tolerance ε, (ie, $I < \varepsilon$). Once this stopping criterion is satisfied, α^* is taken as $0.5(\alpha_l + \alpha_u)$.

Methods based on the preceding philosophy are called *interval-reducing methods*.

Equal–Interval Search

Initial Bracketing of Minimum—Phase I

As mentioned earlier, the basic idea of any interval-reducing method is to successively reduce the interval of uncertainty to a small acceptable value. To clearly describe the idea, we start with a very simple approach called *equal–interval search*. The idea is quite elementary, as illustrated in Fig. 10.6. In the interval $0 \le \alpha \le \bar{\alpha}$, the function $f(\alpha)$ is evaluated at several points using a uniform grid on the α axis in phase I. To do this, we select a small number δ and evaluate the function at the a values of δ, 2δ, 3δ, …, $q\delta$, $(q + 1)\delta$, and so on, as can be seen in Fig. 10.6a. We compare the values of the function at two successive points, say q and $(q + 1)$.

Then, if the function at the point $q\delta$ is larger than that at the subsequent point $(q + 1)\delta$—that is, $f(q\delta) > f((q + 1)\delta)$—the minimum point has not been surpassed yet. However, if the function has started to increase, that is:

$$f(q\delta) < f\big((q+1)\delta\big) \qquad (10.15)$$

then the minimum has been surpassed. Note that once the condition in Eq. (10.15) is satisfied for points $q\delta$ and $(q + 1)\delta$ the minimum can be between either the points $(q - 1)\delta$ and $q\delta$ or the

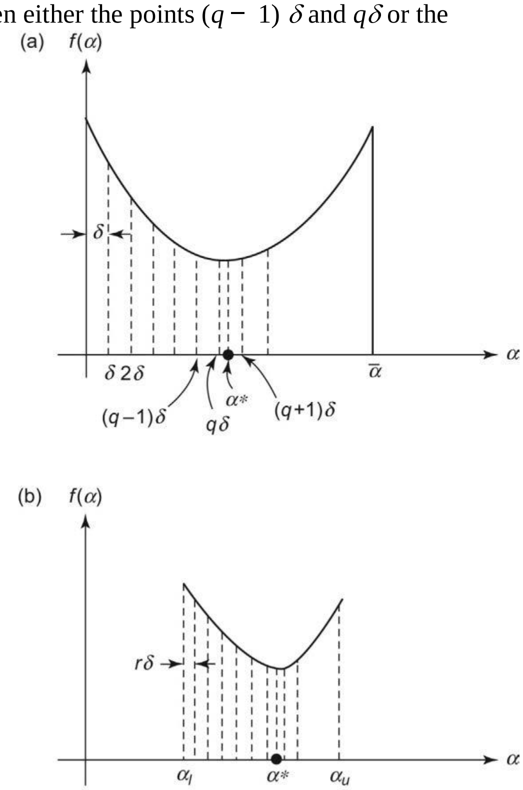

FIGURE 10.6 Equal–interval search process. (a) Phase I: initial bracketing of minimum. (b) Phase II: reducing the interval of uncertainty.

points $q\delta$ and $(q + 1)\ \delta.$. To account for both possibilities, we take the minimum to lie between the points $(q - 1)\ \delta$ and $(q + 1)\ \delta.$. Thus, lower and upper limits for the interval of uncertainty are established and the interval of uncertainty of Eq. (10.14) can be calculated as:

$$\alpha_l = (q - 1)\ \delta,\ \alpha_u = (q + 1)\ \delta,\ I = \alpha_u - \alpha_l = 2\delta \tag{10.16}$$

Reducing the Interval of Uncertainty—Phase II

Establishment of the lower and upper limits on the minimum value of α indicates the end of Phase I. In Phase II, we restart the search process from the lower end of the interval of uncertainty $\alpha = \alpha_l$ with some reduced value for the increment δ, say $r\delta$, where $r \ll 1$. Then the preceding process of Phase I is repeated from $\alpha = \alpha_l$ with the reduced δ, and the minimum is again bracketed. Now the interval of uncertainty I is reduced to $2r\delta$. This is illustrated in Fig. 10.6b. The value of the increment is further reduced to, say, $r_2\delta$, and the process is repeated until the interval of uncertainty is reduced to an acceptable value ε. Note that the method is *convergent* for unimodal functions and can be easily coded into a computer program.

5.6. Golden section search method

Golden section search is an improvement over the alternate equal–interval search and is one of the better methods in the class of interval-reducing methods. The basic idea of the method is still the same: Evaluate the function at predetermined points, compare them to bracket the minimum in Phase I, and then converge on the minimum point in Phase II by systematically reducing the interval of uncertainty. The method uses fewer function evaluations to reach the minimum point compared to other similar methods. The number of function evaluations is reduced during both the phases, the initial bracketing phase as well as the interval-reducing phase.

Initial Bracketing of Minimum—Phase I

In the equal–interval methods, the selected increment δ is kept fixed to bracket the minimum initially. This can be an inefficient process if δ happens to be a small number. An alternate procedure is to vary the increment at each step, that is, multiply it by a factor $r > 1$. This way initial bracketing of the minimum is rapid; however, the length of the initial interval of uncertainty is also larger. The *golden section* search procedure is such a *variable-interval search*
method. In it the value of r is not selected arbitrarily. It is selected as the *golden ratio*, which can be derived as 1.618 in several different ways. One derivation is based on the *Fibonacci sequence*, which is defined as:

$$F_0 = 1;\ F_1 = 1;\ F_n = F_{n-1} + F_{n-2},\ n = 2, 3, ... \tag{10.18}$$

Any number of the Fibonacci sequence for $n > 1$ is obtained by adding the previous two numbers, so the sequence is given as 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89…. This sequence has the property.

$$Fn/Fn\text{-}1 \rightarrow 1.618 \text{ as } n \rightarrow \infty \tag{10.19}$$

That is, as n grows larger, the ratio between two successive numbers F_n and F_{n-1} in the Fibonacci sequence reaches a constant value of 1.618 or $(\sqrt{5} + 1)/2$. This golden ratio has many other interesting properties that will be exploited in the 1D search procedure. One property is that 1/1.618 = 0.618.

Fig. 10.8 illustrates the process of initially bracketing the minimum using a sequence of larger increments based on the golden ratio. In the figure, starting at $q = 0$, we evaluate $f(\alpha)$ at $\alpha = \delta$, where $\delta > 0$ is a small number. We check to see if the value $f(\delta)$ is smaller than the value $f(0)$. If it is, we then take an increment of 1.618δ in the step size (i.e. the increment is 1.618 times the previous increment δ). This way we evaluate the function and compare its values at the following points as shown in Fig. 10.8:

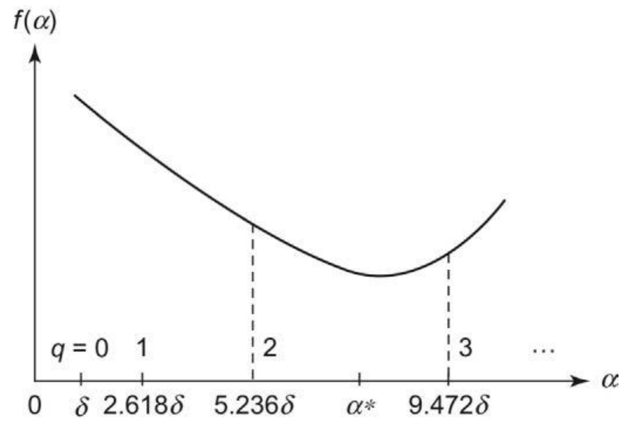

FIGURE 10.8 Initial bracketing of the minimum point in the golden section method.

the previous increment δ). This way we evaluate the function and compare its values at the following points as shown in Fig. 10.8:

$$
\begin{aligned}
&q = 0; \quad \alpha_0 = \delta \\
&q = 1; \quad \alpha_1 = \delta + 1.618\delta = 2.618\delta = \sum_{j=0}^{1} \delta(1.618)^j \\
&q = 2; \quad \alpha_2 = 2.618\delta + 1.618(1.618\delta) = 5.236\delta = \sum_{j=0}^{2} \delta(1.618)^j \\
&q = 3; \quad \alpha_3 = 5.236\delta + 1.618^3\delta = 9.472\delta = \sum_{j=0}^{3} \delta(1.618)^j
\end{aligned}
\tag{10.20}
$$

In general, we continue to evaluate the function at the points:

$$
\alpha_q = \sum_{j=0}^{q} \delta(1.618)^j; \quad q = 0, 1, 2, ... \tag{10.21}
$$

Let us assume that the function at α_{q-1} is smaller than that at the previous point α_{q-2} and the subsequent point α_q, that is:

$$
f(\alpha_{q-1}) < f(\alpha_{q-2}) \quad \text{and} \quad f(\alpha_{q-1}) < f(\alpha_q) \tag{10.22}
$$

Therefore, the minimum point has been surpassed. Actually, the minimum point lies between the previous two intervals, that is, between α_{q-2} and α_q, as in equal–interval search. Therefore, *upper and lower limits on the interval of uncertainty* are:

$$
\alpha_u = \alpha_q = \sum_{j=0}^{q} \delta(1.618)^j; \quad \alpha_l = \alpha_{q-2} = \sum_{j=0}^{q-2} \delta(1.618)^j \tag{10.23}
$$

Thus, the initial interval of uncertainty is calculated as:

$$
\begin{aligned}
I = \alpha_u - \alpha_l &= \sum_{j=0}^{q} \delta(1.618)^j - \sum_{j=0}^{q-2} \delta(1.618)^j = \delta(1.618)^{q-1} + \delta(1.618)^q \\
&= \delta(1.618)^{q-1}(1 + 1.618) = 2.618(1.618)^{q-1}\delta
\end{aligned}
\tag{10.24}
$$

Reducing the Interval of Uncertainty—Phase II

The next task is to start reducing the interval of uncertainty by evaluating and comparing functions at some points in the established interval of uncertainty I. *The method uses two values of the function* within the interval I, just as in the alternate equal–interval search shown in Fig. 10.7. H owever, the points a_a and a_b are not located at $I/3$ from either end of the interval of uncertainty. Instead, they are located at a distance of $0.382I$ (or $0.618I$) from either end. The factor 0.382 is related to the golden ratio, as described in the following section. To see how the factor 0.618 is determined, consider two points symmetrically located a distance τI from either end, as shown in Fig. 10.9a—points a_a and a_b are located at distance τI from either end of the interval. Comparing function values at a_a and a_b, either the left (a_l, a_a) or the right (a_b, a_u) portion of the interval is discarded because the minimum cannot lie there.
Let us assume that the right portion is discarded, as shown in Fig. 10.9b, so $a\ 2_l$ and $a\ 2_u$ are the new lower and upper bounds on the minimum. The new interval of uncertainty is $I2 = \tau I$.
There is one point in the new interval at which the function value is known. It is required that this point be located at a distance of $\tau I2$ from the left end; therefore, $\tau I2 = (1 - \tau)I$.
Since $I2 = \tau I$, this gives the equation $\tau^2 + \tau - 1 = 0$. The positive root of this equation is
$\_ = (\square 1 + 5)/2 = 0.618$.
Thus the two points are located at a distance of $0.618I$ or $0.382I$ from either end of the interval. The golden section search can be initiated once the initial interval of uncertainty is known. If the initial bracketing is done using the variable step increment (with a factor of 1.618, which is 1/0.618), then the function value at one of the points a_{q-1} is already known. It turns out that a_{q-1} is automatically the point a_a. This can be seen by multiplying the initial interval I in Eq. (10.24) by 0.382. If the preceding procedure is not used to initially bracket the minimum, then the points a_a and a_b will have to be located at a distance of $0.382I$ from the lower and upper limits for the interval of uncertainty.

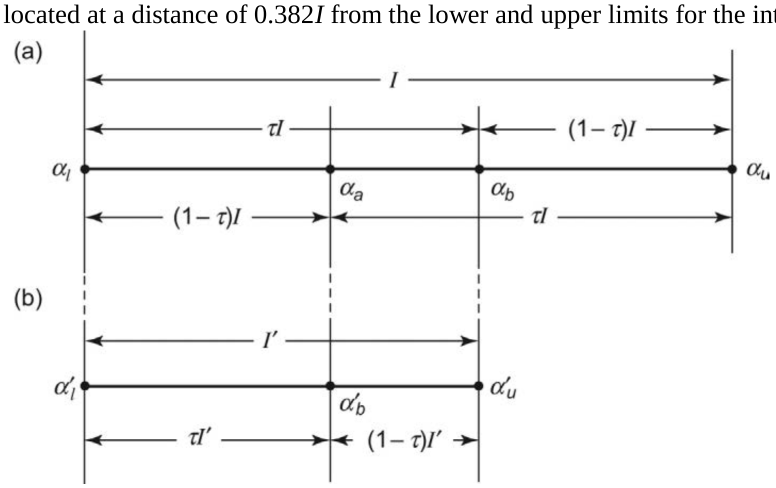

FIGURE 10.9 Golden section partition.

Algorithm for 1D Search by Golden Sections

Find α to minimize $f(\alpha)$.

Step 1: *Phase I.* For a chosen small number δ, calculate $f(0)$, $f(\alpha_0)$, $f(\alpha_1)$, ..., where α_i are given by Eq. (10.21). Let q be the smallest integer to satisfy Eqs. (10.22), where α_q, α_{q-1}, and α_{q-2} are calculated from Eq. (10.21). The upper and lower bounds (α_l and α_u) on α^* (optimum value for α) are given by Eq. (10.23). The interval of uncertainty is given as $I = \alpha_u - \alpha_l$.

Step 2: *Phase II.* Compute $f(\alpha_b)$, where $\alpha_b = \alpha_l + 0.618I$. Note that, at the first iteration, $\alpha_a = \alpha_l + 0.382I = \alpha_{q-1}$, so $f(\alpha_a)$ is already known.

Step 3: Compare $f(\alpha_a)$ and $f(\alpha_b)$, and go to (1), (2), or (3).

1. If $f(\alpha_a) < f(\alpha_b)$, then minimum point α^* lies between α_l and α_b, that is, $\alpha_l \le \alpha^* \le \alpha_b$. The new limits for the reduced interval of uncertainty are $\alpha'_l = \alpha_l$ and $\alpha'_u = \alpha_b$. Also, $\alpha'_b = \alpha_a$. Compute $f(\alpha'_a)$, where $\alpha'_a = \alpha'_l + 0.382(\alpha'_u - \alpha'_l)$ and go to step 4.
2. If $f(\alpha_a) > f(\alpha_b)$, then minimum point α^* lies between α_a and α_u, that is, $\alpha_a \le \alpha^* \le \alpha_u$. Similar to the procedure in step 3(1), let $\alpha'_l = \alpha_a$ and $\alpha'_u = \alpha_u$, so that $\alpha'_a = \alpha_b$. Compute $f(\alpha'_b)$, where $\alpha'_b = \alpha'_l + 0.618(\alpha'_u - \alpha'_l)$ and go to step 4.
3. If $f(\alpha_a) = f(\alpha_b)$, let $\alpha_l = \alpha_a$ and $\alpha_u = \alpha_b$ and return to step 2.

Step 4: If the new interval of uncertainty $I' = \alpha'_u - \alpha'_l$ is small enough to satisfy a stopping criterion (ie, $I' < \varepsilon$), let $\alpha^* = (\alpha'_u + \alpha'_l)/2$ and stop. Otherwise, delete the primes on α'_l, α'_a, and α'_b and return to step 3.

Problems:

Determine whether the given direction at the point is that of descent for the following functions (show all of the calculations).

10.2 $f(\mathbf{x}) = 3x_1^2 + 2x_1 + 2x_2^2 + 7$; $\mathbf{d} = (-1, 1)$ at $\mathbf{x} = (2, 1)$

10.3 $f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 4$; $\mathbf{d} = (2, 1)$ at $\mathbf{x} = (1, 1)$

10.4 $f(\mathbf{x}) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3$; $\mathbf{d} = (-3, 10, -12)$ at $\mathbf{x} = (1, 2, 3)$

10.5 $f(\mathbf{x}) = 0.1x_1^2 + x_2^2 - 10$; $\mathbf{d} = (1, 2)$ at $\mathbf{x} = (4, 1)$

10.6 $f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$; $\mathbf{d} = (2, 3)$ at $\mathbf{x} = (4, 3)$

10.16 *Find* the minimum of the function $f(\alpha) = 7\alpha^2 - 20\alpha + 22$ using the equal–interval search method within an accuracy of 0.001. Use $\delta = 0.05$.

10.17 For the function $f(\alpha) = 7\alpha^2 - 20\alpha + 22$, use the golden section method to find the minimum with an accuracy of 0.005 (final interval of uncertainty should be less than 0.005). Use $\delta = 0.05$.

10.19 Consider the function $f(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3$. Verify whether the vector $\mathbf{d} = (-12, -40, -48)$ at the point (2, 4, 10) is a descent direction for f. What is the slope of the function at the given point? Find an optimum step size along $\mathbf{d}$ by any numerical method.

10.20 Consider the function $f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 4$. At the point (1, 1), let a search direction be defined as $\mathbf{d} = (1, 2)$. Express f as a function of one variable at the given point along $\mathbf{d}$. Find an optimum step size along $\mathbf{d}$ analytically.

For the following functions, direction of change at a point is given. Derive the function of one variable (line search function) that can be used to determine optimum step size (show all calculations).

10.21 $f(\mathbf{x}) = 0.1x_1^2 + x_2^2 - 10$; $\mathbf{d} = (-1, -2)$ at $\mathbf{x} = (5, 1)$

10.22 $f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$; $\mathbf{d} = (-4, -6)$ at $\mathbf{x} = (4, 4)$
10.23 $f(\mathbf{x}) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$; $\mathbf{d} = (-162, 40)$ at $\mathbf{x} = (2, 2)$
10.24 $f(\mathbf{x}) = (x_1 - 2)^2 + x_2^2$; $\mathbf{d} = (2, -2)$ at $\mathbf{x} = (1, 1)$
10.25 $f(\mathbf{x}) = 0.5x_1^2 + x_2^2 - x_1x_2 - 7x_1 - 7x_2$; $\mathbf{d} = (7, 6)$ at $\mathbf{x} = (1, 1)$