Module – 6: Numerical Methods for Unconstrained Optimum Design – Indirect Methods

6.1 The gradient of a function, the Steepest Descent

Thus, until now we have assumed that a search direction in the design space is known and we have tackled the problem of step size determination. In this section, and the later one, we will address the question of how to determine the search direction $\mathbf{d}$. The *basic requirement for* $\mathbf{d}$ *is that the cost function be reduced* if we take a small step along $\mathbf{d}$; that is, the descent condition of Eq. (10.9) must be satisfied. This will be called the *descent direction.*

Several methods are available for determining a descent direction for unconstrained optimization problems. The *steepest–descent method* is the simplest, the oldest, and probably the best known numerical method for unconstrained optimization. The philosophy of the method, introduced by Cauchy in 1847, is to find the direction $\mathbf{d}$, at the current iteration, in which the cost function $f(\mathbf{x})$ decreases most rapidly, at least locally. Because of this philosophy, the method is called the *steepest–descent* search technique. Also, properties of the gradient of the cost function are used in the iterative process, which is the reason for its alternate name: the *gradient method.* The steepest–descent method is a *first-order method* since only the gradient of the cost function is calculated and used to evaluate the search direction. In the later chapter, we will discuss *second-order methods* in which the Hessian of the function will also be used in determining the search direction.

The gradient of a scalar function $f(x_1, x_2, \ldots, x_n)$ was defined in chapter: Optimum Design Concepts: Optimality Conditions as the column vector:

$$\mathbf{c} = \nabla f = \left[\frac{\partial f}{\partial x_1}\frac{\partial f}{\partial x_2}\cdots\frac{\partial f}{\partial x_n}\right]^T \tag{10.25}$$

To simplify the notation, we will use vector $\mathbf{c}$ to represent the gradient of the cost function $f(\mathbf{x})$; that is, $c_i = \partial f/\partial x_i$. We will use a superscript to denote the point at which this vector is calculated:

$$\mathbf{c}^{(k)} = \mathbf{c}\left(\mathbf{x}^{(k)}\right) = \left[\frac{\partial f\left(\mathbf{x}^{(k)}\right)}{\partial x_i}\right]^T \tag{10.26}$$

The *gradient vector has several properties* that are used in the steepest–descent method. These will be discussed in the later chapter in more detail. The most important property is that *gradient vector at a point* $\mathbf{x}$ *is in the direction of maximum increase in the cost function.* Thus the direction of maximum decrease is opposite to that, that is, negative of the gradient vector. Any small move in the negative gradient direction will result in maximum local rate of decrease in the cost function. The negative gradient vector thus represents a *direction of steepest descent* for the cost function and is written as

$$\mathbf{d} = -\mathbf{c}, \quad \text{or} \quad d_i = -c_i = -\frac{\partial f}{\partial x_i}; \quad i = 1 \text{ to } n \tag{10.27}$$

Note that since $\mathbf{d} = -\mathbf{c}$, the descent condition of inequality Eq. (10.9) is always satisfied by the steepest–descent direction as

$$(\mathbf{c} \cdot \mathbf{d}) = -\|\mathbf{c}\|^2 < 0 \tag{10.28}$$

Steepest–Descent Algorithm

Equation (10.27) gives a direction of change in the design space for use in Eq. (10.4). Based on the preceding discussion, the *steepest–descent algorithm* is stated as follows:

Step 1: Estimate a starting design $\mathbf{x}^{(0)}$ and set the iteration counter $k = 0$. Select a convergence parameter $\varepsilon > 0$.
Step 2: Calculate the gradient of $f(\mathbf{x})$ at the current point $\mathbf{x}^{(k)}$ as $\mathbf{c}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.
Step 3: Calculate the length of $\mathbf{c}^{(k)}$ as $\|\mathbf{c}^{(k)}\|$. If $\|\mathbf{c}^{(k)}\| < \varepsilon$, then stop the iterative process because $\mathbf{x}^* = \mathbf{x}^{(k)}$ is a local minimum point. Otherwise, continue.
Step 4: Let the search direction at the current point $\mathbf{x}^{(k)}$ be $\mathbf{d}^{(k)} = -\mathbf{c}^{(k)}$.
Step 5: Calculate a step size α_k that minimizes $f(\alpha) = f(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)})$ in the direction $\mathbf{d}^{(k)}$. Any 1D search algorithm may be used to determine α_k.
Step 6: Update the design using Eq. (10.4) as $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{d}^{(k)}$. Set $k = k + 1$, and go to step 2.

The basic idea of the steepest–descent method is quite simple. We start with an initial estimate for the minimum design. The direction of steepest descent is computed at that point. If the direction is nonzero, we move as far as possible along it to reduce the cost function. At the new design point, we calculate the steepest–descent direction again and repeat the entire process. Examples 10.4 and 10.5 illustrate the calculations involved in the steepest–descent method.

Although the method of steepest–descent is quite simple and robust (it is convergent), it has some drawbacks:

1. Even if convergence of the steepest–descent method is guaranteed, a large number of iterations may be required to reach the minimum point.
2. Each iteration of the method is started independently of others, which can be inefficient. Information calculated at the previous iterations is not used.
3. Only first-order information about the function is used at each iteration to determine the search direction. This is one reason that convergence of the method is slow. The rate of convergence of the steepest–descent method depends on the condition number of the Hessian of the cost function at the optimum point. If the condition number is large, the rate of convergence of the method is slow.
4. Practical experience with the steepest–descent method has shown that a substantial decrease in the cost function is achieved in the initial few iterations and then this decreases slows considerably in later iterations.

6.2 Conjugate Gradient (Fletcher-Reeves), Method

Many optimization methods are based on the concept of conjugate gradients; however, we will describe a method attributed to Fletcher and Reeves (1964) in this section. The conjugate gradient method is a very simple and effective modification of the steepest–descent method. It is shown in the later chapter that the steepest–descent directions at two consecutive steps are orthogonal to each other. This tends to slow down the steepest–descent method, although

it is guaranteed to converge to a local minimum point. The conjugate gradient directions are not orthogonal to each other. Rather, these directions tend to cut diagonally through the orthogonal steepest–descent directions. Therefore, they improve the rate of convergence of the steepest–descent method considerably. Actually, the *conjugate gradient directions* $\mathbf{d}^{(i)}$ are orthogonal with respect to a symmetric and positive definite matrix $\mathbf{A}$, that is,

$$\mathbf{d}^{(i)T}\mathbf{A}\mathbf{d}^{(j)} = 0 \quad \text{for all} \quad i \quad \text{and} \quad j, i \neq j \tag{10.29}$$

Conjugate Gradient Algorithm

Step 1: Estimate a starting design as $\mathbf{x}^{(0)}$. Set the iteration counter $k = 0$. Select the convergence parameter ε. Calculate

$$\mathbf{d}^{(0)} = -\mathbf{c}^{(0)} = -\nabla f\left(\mathbf{x}^{(0)}\right) \tag{10.30}$$

Check the stopping criterion. If $\|\mathbf{c}^{(0)}\| < \varepsilon$, then stop. Otherwise, go to step 5 (note that the first iteration of the conjugate gradient and steepest–descent methods is the same).
Step 2: Compute the gradient of the cost function as $\mathbf{c}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.
Step 3: Calculate $\|\mathbf{c}^{(k)}\|$. If $\|\mathbf{c}^{(k)}\| < \varepsilon$, then stop; otherwise continue.
Step 4: Calculate the new conjugate direction as:

$$\mathbf{d}^{(k)} = -\mathbf{c}^{(k)} + \beta_k \mathbf{d}^{(k-1)} \tag{10.31}$$

$$\beta_k = \left(\frac{\left\|\mathbf{c}^{(k)}\right\|}{\left\|\mathbf{c}^{(k-1)}\right\|}\right)^2 = \frac{\left(\mathbf{c}^{(k)} \bullet \mathbf{c}^{(k)}\right)}{\left(\mathbf{c}^{(k-1)} \bullet \mathbf{c}^{(k-1)}\right)} \tag{10.32}$$

Step 5: Compute a step size $\alpha_k = \alpha$ to minimize $f(\alpha) = f(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)})$.
Step 6: Change the design as follows: set $k = k + 1$ and go to step 2.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \tag{10.33}$$

Note that the conjugate direction shown in Eq. (10.31) satisfies the descent condition of inequality Eq. (10.9). This can be shown by substituting $\mathbf{d}^{(k)}$ from Eq. (10.31) into inequality Eq. (10.9) which gives

$$\left(\mathbf{c}^{(k)} \bullet \mathbf{d}^{(k)}\right) = -\left\|\mathbf{c}^{(k)}\right\|^2 + \beta_k\left(\mathbf{c}^{(k)} \bullet \mathbf{d}^{(k-1)}\right) \tag{10.34}$$

Thus if the second term in Eq. (10.34) is negative or zero the descent condition will be satisfied. If the step size termination condition given in Eq. (10.13) is satisfied then $\left(\mathbf{c}^{(k)} \cdot \mathbf{d}^{(k-1)}\right) = 0$. This implies that as long as the step size is determined accurately, the conjugate gradient direction is guaranteed to be that of descent.

The first iteration of the conjugate gradient method is just the steepest–descent iteration. The only difference between the conjugate gradient and steepest–descent methods is in Eq. (10.31). In this equation, the current steepest–descent direction is modified by adding a scaled direction that was used in the previous iteration.

The scale factor is determined by using lengths of the gradient vector at the two iterations, as shown in Eq. (10.32). Thus, the conjugate direction is nothing but a deflected steepest–descent direction. This is a simple modification that requires little additional calculation. However, it is very effective in substantially improving the rate of convergence of the steepest–descent method. Therefore, *the conjugate gradient method should always be preferred over the steepest–descent method.* In the later chapter, an example is discussed that compares the rate of convergence of the steepest–descent, conjugate gradient, and Newton's methods. We will see there that the conjugate gradient method performs quite well compared with the other two.

Convergence of the Conjugate Gradient Method

The conjugate gradient algorithm finds the minimum in n iterations for positive definite quadratic functions having n design variables. For general functions, if the minimum has not been found by then, the iterative process needs to be restarted every $(n + 1)$th iteration for computational stability. That is, set $\mathbf{x}^{(0)} = \mathbf{x}^{(n+1)}$ and restart the process from step 1 of the algorithm. The algorithm is very simple to program and works very well for general unconstrained minimization problems. Example 10.6 illustrates the calculations involved in the conjugate gradient method.

6.3 Step Size Determination – Polynomial Interpolation

The interval-reducing methods described in chapter: Numerical Methods for Unconstrained Optimum Design can require too many function evaluations during line search to determine an appropriate step size. In realistic engineering design problems, function evaluation requires a significant amount of computational effort. Therefore, methods such as golden section search are inefficient for many practical applications. In this section, we present some other line search methods such as polynomial interpolation and inexact line search.

Recall that the step size calculation problem is to find α to:

Minimize

$$f(\alpha) = f\left(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}\right) \tag{11.1}$$

It is assumed that the search direction $\mathbf{d}^{(k)}$ is that of descent at the current point $\mathbf{x}^{(k)}$, that is:

$$\left(\mathbf{c}^{(k)} \cdot \mathbf{d}^{(k)}\right) < 0 \tag{11.2}$$

Differentiating $f(\alpha)$ in Eq. (11.1) with respect to α and using the chain rule of differentiation, we get:

$$f'(\alpha) = \left(\mathbf{c}\left(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}\right) \bullet \mathbf{d}^{(k)}\right); \quad \mathbf{c}\left(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}\right) = \nabla f\left(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}\right) \tag{11.3}$$

where "prime" indicates the first derivative of $f(\alpha)$. Evaluating Eq. (11.3) at $\alpha = 0$, we get:

$$f'(0) = \left(\mathbf{c}^{(k)} \bullet \mathbf{d}^{(k)}\right) < 0 \tag{11.4}$$

Thus the slope of the curve $f(\alpha)$ versus α is negative at $\alpha = 0$, as can be observed in Fig. 10.3. If an exact step size is determined as α_k, then $f'(\alpha_k) = 0$, which gives the following condition from Eq. (11.3), called the *line search termination criterion*:

$$\left(\mathbf{c}^{(k+1)} \bullet \mathbf{d}^{(k)}\right) = 0 \tag{11.5}$$

Instead of evaluating the function at numerous trial points during line search, we can pass a curve through a limited number of points and use the analytical procedure to calculate the step size. Any continuous function on a given interval can be approximated as closely as desired by passing a higher-order polynomial through its data points and then calculating its minimum explicitly. The minimum point of the approximating polynomial is often a good estimate of the exact minimum of the line search function $f(\alpha)$. Thus, polynomial interpolation can be an efficient technique for one-dimensional search. Whereas many polynomial interpolation schemes can be devised, we will present two procedures based on *quadratic interpolation*.

Quadratic Curve Fitting

Many times it is sufficient to approximate the function $f(\alpha)$ on an interval of uncertainty by a quadratic function. To replace a function in an interval with a quadratic function, we need to

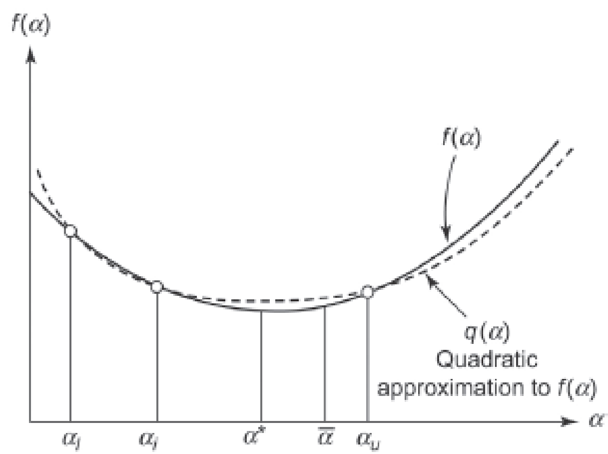

FIGURE 11.1 **Quadratic approximation for a function $f(\alpha)$.**

know the function value at three distinct points to determine the three coefficients of the quadratic polynomial. It must also be assumed that the function $f(\alpha)$ is sufficiently smooth and unimodal, and the initial interval of uncertainty (α_l, α_u) is known. Let α_i be any intermediate point in the interval (α_l, α_u), and let $f(\alpha_l)$, $f(\alpha_i)$, and $f(\alpha_u)$ be the function values at the respective points. Fig. 11.1 shows the function $f(\alpha)$ and the quadratic function $q(\alpha)$ as its approximation in the interval (α_l, α_u). $\bar{\alpha}$ is the minimum point of the quadratic function $q(\alpha)$, whereas α^* is the exact minimum point of $f(\alpha)$. Iteration can be used to improve the estimate $\bar{\alpha}$ for α^*.

Any quadratic function $q(\alpha)$ can be expressed in the general form as:

$$q(\alpha) = a_0 + a_1\alpha + a_2\alpha^2 \tag{11.6}$$

where a_0, a_1, and a_2 are the unknown coefficients. Since the function $q(\alpha)$ must have the same value as the function $f(\alpha)$ at the points α_l, α_i, and α_u, we get three equations in three unknowns a_0, a_1, and a_2 as follows:

$$a_0 + a_1\alpha_l + a_2\alpha_l^2 = f(\alpha_l) \tag{11.7}$$

$$a_0 + a_1\alpha_i + a_2\alpha_i^2 = f(\alpha_i) \tag{11.8}$$

$$a_0 + a_1\alpha_u + a_2\alpha_u^2 = f(\alpha_u) \tag{11.9}$$

6.4 Properties of the Gradient Vector

Property 1

The gradient vector $\mathbf{c}$ of a function $f(x_1, x_2, \ldots, x_n)$ at the given point $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is orthogonal (normal) to the tangent hyperplane for the surface $f(x_1, x_2, \ldots, x_n)$ = constant.

Proof

This is an important property of the gradient vector and is shown graphically in Fig. 11.4. The figure shows the surface $f(\mathbf{x})$ = constant; $\mathbf{x}^*$ is a point on the surface; C is any curve on the surface through the point $\mathbf{x}^*$; $\mathbf{T}$ is a vector tangent to C at the point $\mathbf{x}^*$; $\mathbf{u}$ is any unit vector; and $\mathbf{c}$ is the gradient vector at $\mathbf{x}^*$. According to the above property, vectors $\mathbf{c}$ and $\mathbf{T}$ are normal to each other; that is, their dot product is zero, $(\mathbf{c} \cdot \mathbf{T}) = 0$.

To prove this property, we take any curve C on the surface $f(x_1, x_2, \ldots, x_n)$ = constant, as was shown in Fig. 11.4. Let the curve pass through the point $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$. Moreover,

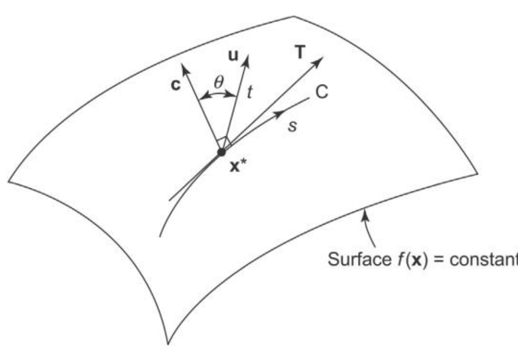

FIGURE 11.4 **Gradient vector for the surface *f* (x) = constant at the point x*.**

let s be a parameter along C. Then a unit tangent vector **T** along C at the point **x*** is given as:

$$\mathbf{T} = \left[\frac{\partial x_1}{\partial s} \frac{\partial x_2}{\partial s} \cdots \frac{\partial x_n}{\partial s} \right]^T \tag{a}$$

Since $f(\mathbf{x})$ = constant, the derivative of f along curve C is zero; that is, $df/ds = 0$ (the directional derivative of f in the direction s). Or, using the chain rule of differentiation, we get:

$$\frac{df}{ds} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial s} + \cdots + \frac{\partial f}{\partial x_n}\frac{\partial x_n}{\partial s} = 0 \tag{b}$$

Writing Eq. (b) in the vector form after identifying $\partial f/\partial x_i$ and $\partial x_i/\partial s$ (from Eq. (a)) as components of the gradient and the unit tangent vectors, we obtain $(\mathbf{c} \cdot \mathbf{T}) = 0$, or $\mathbf{c}^T\mathbf{T} = 0$. Since the dot product of the gradient vector **c** with the tangential vector **T** is zero, the vectors are normal to each other. But **T** is any tangent vector at **x***, and so **c** is orthogonal to the tangent hyperplane for the surface $f(\mathbf{x})$ = constant at point **x***.

Property 2

The second property is that the gradient represents a direction of maximum rate of increase for the function $f(\mathbf{x})$ at the given point **x***.

Proof

To show this, let **u** be a unit vector in any direction that is not tangential to the surface. This is shown in Fig. 11.4. Let t be a parameter along **u**. The derivative of $f(\mathbf{x})$ in the direction **u** at the point **x*** (ie, the directional derivative of f) is given as:

$$\frac{df}{dt} = \lim_{\varepsilon \to 0} \frac{f(\mathbf{x} + \varepsilon\mathbf{u}) - f(\mathbf{x})}{\varepsilon} \tag{c}$$

where ε is a small number and t is a parameter along **u**. Using Taylor's expansion, we have:

$$f(\mathbf{x} + \varepsilon\mathbf{u}) = f(\mathbf{x}) + \varepsilon\left[u_1\frac{\partial f}{\partial x_1} + u_2\frac{\partial f}{\partial x_2} + \cdots + u_n\frac{\partial f}{\partial x_n} \right] + o\left(\varepsilon^2\right) \tag{d}$$

where u_i are components of the unit vector **u** and $o(\varepsilon^2)$ are terms of order ε^2. Rewriting the foregoing equation:

$$f(\mathbf{x} + \varepsilon\mathbf{u}) - f(\mathbf{x}) = \varepsilon\sum_{i=1}^{n} u_i\frac{\partial f}{\partial x_i} + o\left(\varepsilon^2\right) \tag{e}$$

Substituting Eq. (e) into (c) and taking the indicated limit, we get:

$$\frac{df}{dt} = \sum_{i=1}^{n} u_i\frac{\partial f}{\partial x_i} = (\mathbf{c} \cdot \mathbf{u}) = \mathbf{c}^T\mathbf{u} \tag{f}$$

Using the definition of the dot product in Eq. (f), we get:

$$\frac{df}{dt} = \|\mathbf{c}\|\ \|\mathbf{u}\|\cos\theta \tag{g}$$

where θ is the angle between the **c** and **u** vectors. The right side of Eq. (g) will have extreme values when $\theta = 0$ or 180 degrees. When $\theta = 0$ degrees, vector **u** is along **c** and $\cos\theta = 1$. Therefore, from Eq. (g), df/dt represents the maximum rate of increase for $f(\mathbf{x})$ when $\theta = 0$ degrees. Similarly, when $\theta = 180$ degrees, vector **u** points in the negative **c** direction. From Eq. (g), then, df/dt represents the maximum rate of decrease for $f(\mathbf{x})$ when $\theta = 180$ degrees.

According to the foregoing property of the gradient vector, if we need to move away from the surface $f(\mathbf{x})$ = constant, the function increases most rapidly along the gradient vector compared with a move in any other direction. In Fig. 11.4, a small move along the direction **c** will result in a larger increase in the function, compared with a similar move along the direction **u**. Of course, any small move along the direction **T** results in no change in the function since **T** is tangent to the surface.

Property 3

The maximum rate of change in $f(\mathbf{x})$ at any point $\mathbf{x}^*$ is the magnitude of the gradient vector.

Proof

Since **u** is a unit vector, the maximum value of df/dt from Eq. (g) is given as:

$$max\left|\frac{df}{dt}\right| = \|\mathbf{c}\| \tag{h}$$

since the maximum value of $\cos\theta$ is 1 when $\theta = 0$ degrees. However, for $\theta = 0$ degrees, **u** is in the direction of the gradient vector. Therefore, the magnitude of the gradient represents the maximum rate of change for the function $f(\mathbf{x})$.

These properties show that the gradient vector at any point $\mathbf{x}^*$ represents a direction of maximum increase in the function $f(\mathbf{x})$ and the rate of increase is the magnitude of the vector. The gradient is therefore called a direction of *steepest ascent* for the function $f(\mathbf{x})$ and the negative of the gradient is called the *direction of steepest descent*. Example 11.3 verifies the properties of the gradient vector.

EXAMPLE 11.3 VERIFICATION OF THE PROPERTIES OF THE GRADIENT VECTOR

Verify the properties of the gradient vector for the following function for the point $\mathbf{x}^{(0)} = (0.6, 4)$.

$$f(\mathbf{x}) = 25x_1^2 + x_2^2 \tag{a}$$

Solution

Fig. 11.5 shows in the $x_1 - x_2$ plane the contours of values 25 and 100 for the function f. The value of the function at (0.6, 4) is $f(0.6, 4) = 25$. The gradient of the function at (0.6, 4) is given as:

$$\mathbf{c} = \nabla f(0.6, 4) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (50x_1, 2x_2) = (30, 8) \tag{b}$$

$$\|\mathbf{c}\| = \sqrt{30 \times 30 + 8 \times 8} = 31.04835 \tag{c}$$

Therefore, a unit vector along the gradient is given as:

$$\mathbf{C} = \frac{\mathbf{c}}{\|\mathbf{c}\|} = (0.966235, 0.257663) \tag{d}$$

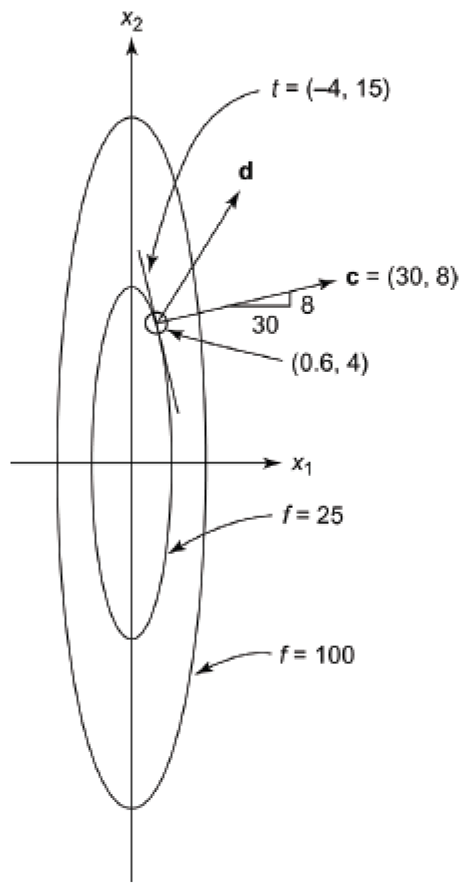

FIGURE 11.5 **Contours of the function $f = 25x_1^2 + x_2^2$ for $f = 25$ and 100.**

Using the given function, a vector tangent to the curve at the point (0.6, 4) is given as:

$$\mathbf{t} = (-4, 15) \tag{e}$$

This vector is obtained by differentiating the equation for the following curve at the point (0.6, 4) with respect to the parameter s along the curve:

$$25x_1^2 + x_2^2 = 25 \tag{f}$$

Differentiating this equation with respect to s at the point (0.6, 4) gives:

$$25 \times 2x_1 \frac{\partial x_1}{\partial s} + 2x_2 \frac{\partial x_2}{\partial s} = 0, \quad \text{or} \quad \frac{\partial x_1}{\partial s} = -\left(\frac{4}{15}\right)\frac{\partial x_2}{\partial s} \tag{g}$$

Then the vector $\mathbf{t}$ tangent to the curve is obtained as $(\partial x_1/\partial s, \partial x_2/\partial s)$. The unit tangent vector is calculated as:

$$\mathrm{T} = \frac{\mathbf{t}}{\|\mathbf{t}\|} = (-0.257663, 0.966235) \tag{h}$$

Property 1

If the gradient is normal to the tangent, then $(\mathbf{C} \cdot \mathbf{T}) = 0$. This is indeed true for the preceding data. We can also use the condition that if two lines are orthogonal, then $m_1 m_2 = -1$, where m_1 and m_2 are the slopes of the two lines (this result can be proved using the rotational transformation of coordinates through 90 degrees). To calculate the slope of the tangent, we use the equation for the curve $25x_1^2 + x_2^2 = 25$, or $x_2 = 5\sqrt{1 - x_1^2}$. Therefore, the slope of the tangent at the point (0.6, 4) is given as:

$$m_1 = \frac{dx_2}{dx_1} = \frac{-5x_1}{\sqrt{1 - x_1^2}} = -\frac{15}{4} \tag{i}$$

This slope is also obtained directly from the tangent vector $\mathbf{t} = (-4, 15)$. The slope of the gradient vector $\mathbf{c} = (30, 8)$ is $m_2 = \frac{8}{30} = \frac{4}{15}$. Thus, $m_1 m_2$ is indeed -1, and the two lines are normal to each other.

Property 2

Consider any arbitrary direction $\mathbf{d} = (0.501034,\ 0.865430)$ at the point (0.6,4), as shown in Fig. 11.5. If $\mathbf{C}$ is the direction of steepest ascent, then the function should increase more rapidly along $\mathbf{C}$ than along $\mathbf{d}$. Let us choose a step size $\alpha = 0.1$ and calculate two points, one along $\mathbf{C}$ and the other along $\mathbf{d}$:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha\mathbf{C} = \begin{bmatrix} 0.6 \\ 4.0 \end{bmatrix} + 0.1 \begin{bmatrix} 0.966235 \\ 0.257633 \end{bmatrix} = \begin{bmatrix} 0.6966235 \\ 4.0257663 \end{bmatrix} \tag{j}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(0)} + \alpha\mathbf{d} = \begin{bmatrix} 0.6 \\ 4.0 \end{bmatrix} + 0.1 \begin{bmatrix} 0.501034 \\ 0.865430 \end{bmatrix} = \begin{bmatrix} 0.6501034 \\ 4.0865430 \end{bmatrix} \tag{k}$$

Now we calculate the function at these points and compare their values: $f(\mathbf{x}^{(1)}) = 28.3389$, $f(\mathbf{x}^{(2)}) = 27.2657$. Since $f(\mathbf{x}^{(1)}) > f(\mathbf{x}^{(2)})$, the function increases more rapidly along C than along $\mathbf{d}$.

Property 3

If the magnitude of the gradient vector represents the maximum rate of change in $f(\mathbf{x})$, then $(\mathbf{c} \cdot \mathbf{c}) > (\mathbf{c} \cdot \mathbf{d})$, $(\mathbf{c} \cdot \mathbf{c}) = 964.0$, and $(\mathbf{c} \cdot \mathbf{d}) = 21.9545$. Therefore, the gradient vector satisfies this property also.

Note that the last two properties are valid only in a local sense—that is, only in a small neighborhood of the point at which the gradient is evaluated.

Problems:

For the following problems, complete two iterations of the steepest–descent method starting from the given design point.

10.52 $f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$; starting design (1, 1)

10.53 $f(x_1, x_2) = 12.096x_1^2 + 21.504x_2^2 - 1.7321x_1 - x_2$; starting design (1, 1)

10.54 $f(x_1, x_2) = 6.983x_1^2 + 12.415x_2^2 - x_1$; starting design (2, 1)

10.55 $f(x_1, x_2) = 12.096x_1^2 + 21.504x_2^2 - x_2$; starting design (1, 2)

10.56 $f(x_1, x_2) = 25x_1^2 + 20x_2^2 - 2x_1 - x_2$; starting design (3, 1)

For the following problems, complete two iterations of the conjugate gradient method.

10.68 Exercise 10.52

10.69 Exercise 10.53

10.70 Exercise 10.54

10.71 Exercise 10.55

10.72 Exercise 10.56