TABLE 8.10 (*Contd.*)

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	j	$\mathbf{y}_j$ $f(\mathbf{y}_j)$	Δ_j	$\mathbf{d}_j$	$\mathbf{y}_j+\Delta_j\mathbf{d}_j$ $f(\mathbf{y}_j+\Delta_j\mathbf{d}_j)$
		1	(2.93, 1.20) 1.03	−0.10	(0.89, −0.45)	(2.84, 1.25) 0.61(S)
		2	(2.84, 1.25) 0.61	0.40	(−0.45, −0.89)	(2.66, 0.89) 0.96(F)
		1	(2.84, 1.25) 0.61	−0.20	(0.89, −0.45)	(2.66, 1.34) 0.19(S)
		2	(2.66, 1.34) 0.19	−0.20	(−0.45, −0.89)	(2.75, 1.52) 0.40(F)

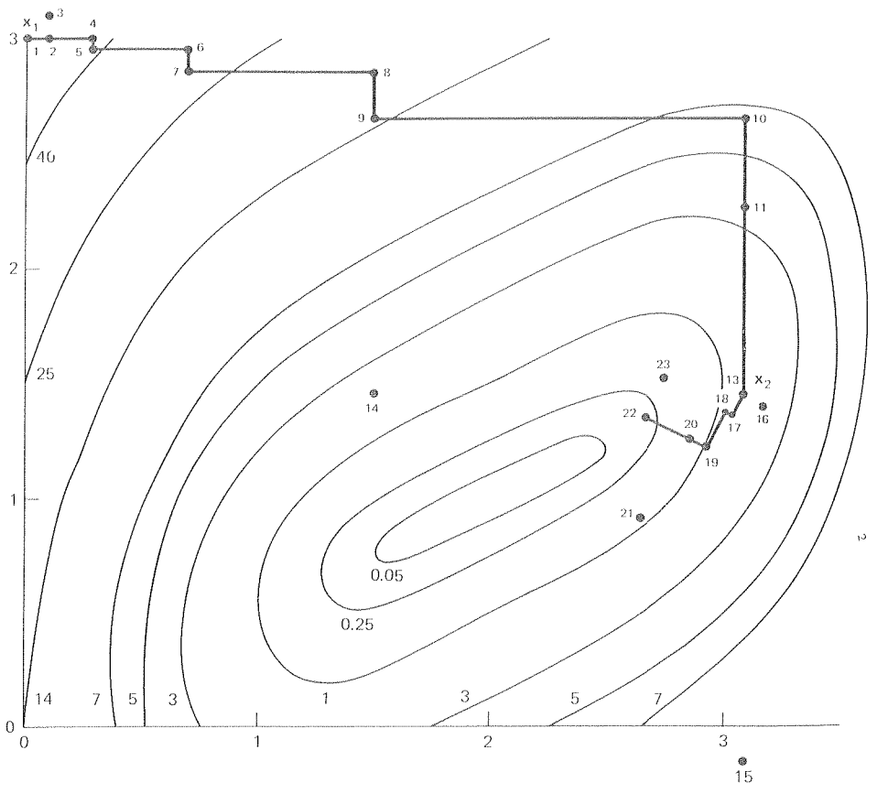

Figure 8.13 Illustration of Rosenbrock's procedure using discrete steps. (The numbers denote the order in which points are generated.)

required. In particular, $(\mathbf{x}_2-\mathbf{x}_1)=\lambda_1\mathbf{d}_1+\lambda_2\mathbf{d}_2$, where $\lambda_1=3.10$ and $\lambda_2=-1.55$. Using (8.9), the reader can easily verify that the new search directions are given by $(0.89, -0.45)$ and $(-0.45, -0.89)$, which are used in the second iteration. The procedure is terminated in the middle of the second iteration.

In Figure 8.13, the progress of Rosenbrock's method is shown, and the points generated are numbered sequentially.

8.5 Multidimensional Search Using Derivatives

In the previous section, we described several minimization procedures that use only functional evaluations during the course of optimization. We now discuss some methods that use derivatives in determining the search directions. In particular, we discuss the steepest descent method and the method of Newton.

The Method of Steepest Descent

The method of steepest descent is one of the most fundamental procedures for minimizing a differentiable function of several variables. Recall that a vector $\mathbf{d}$ is called a direction of descent of a function f at $\mathbf{x}$ if there exists a $\delta>0$ such that $f(\mathbf{x}+\lambda\mathbf{d})<f(\mathbf{x})$ for all $\lambda\in(0,\delta)$. In particular, if $\lim_{\lambda\to 0^+}[f(\mathbf{x}+\lambda\mathbf{d})-f(\mathbf{x})]/\lambda<0$, then $\mathbf{d}$ is a direction of descent. The method of steepest descent moves along the direction $\mathbf{d}$ with $\|\mathbf{d}\|=1$, which minimizes the above limit. Lemma 8.5.1 below shows that if f is differentiable at $\mathbf{x}$ with a nonzero gradient, then $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$ is indeed the direction of steepest descent. For this reason, in the presence of differentiability, the method of the steepest descent is sometimes called the gradient method.

8.5.1 Lemma

Suppose that $f: E_n\to E_1$ is differentiable at $\mathbf{x}$, and suppose that $\nabla f(\mathbf{x})\neq\mathbf{0}$. Then the optimal solution to the problem to minimize $f'(\mathbf{x};\mathbf{d})$ subject to $\|\mathbf{d}\|\leq 1$ is given by $\bar{\mathbf{d}}=-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$; that is, $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$ is the direction of steepest descent of f at $\mathbf{x}$.

Proof

From differentiability of at $\mathbf{x}$, it follows that

$$f'(\mathbf{x};\mathbf{d})=\lim_{\lambda\to 0^+}\frac{f(\mathbf{x}+\lambda\mathbf{d})-f(\mathbf{x})}{\lambda}=\nabla f(\mathbf{x})^t\mathbf{d}$$

Thus the problem reduces to minimize $\nabla f(\mathbf{x})^t\mathbf{d}$ subject to $\|\mathbf{d}\|\leq 1$. By the Schwartz inequality, for $\|\mathbf{d}\|\leq 1$, we have

$$\nabla f(\mathbf{x})^t\mathbf{d}\geq -\|\nabla f(\mathbf{x})\|\,\|\mathbf{d}\|\geq -\|\nabla f(\mathbf{x})\|.$$

For $\bar{\mathbf{d}} = -\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$, we have $\nabla f(\mathbf{x})^t \bar{\mathbf{d}} = -\|\nabla f(\mathbf{x})\|$. Thus $\bar{\mathbf{d}}$ is the optimal solution, and the proof is complete.

Summary of the Steepest Descent Algorithm

Given a point $\mathbf{x}$, the steepest descent algorithm proceeds by performing a line search along the direction $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$, or equivalently, along the direction $-\nabla f(\mathbf{x})$. A summary of the method is given below.

Initialization Step Let $\varepsilon > 0$ be the termination scalar. Choose a starting point $\mathbf{x}_1$, let $k = 1$, and go to the main step.

Main Step If $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$, stop; otherwise, let $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, and let λ_k be an optimal solution to the problem to minimize $f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ subject to $\lambda \geq 0$. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, replace k by $k+1$, and repeat the main step.

8.5.2 Example

Consider the following problem

$$\text{Minimize} \quad (x_1 - 2)^4 + (x_1 - 2x_2)^2$$

We solve this problem using the method of steepest descent, starting with the point (0.00, 3.00). The summary of the computations are given in Table 8.11. After seven iterations, the point $\mathbf{x}_8 = (2.28, 1.15)^t$ is reached. The algorithm is terminated since $\|\nabla f(\mathbf{x}_8)\| = 0.09$ is small. The progress of the method is shown in Figure 8.14. Note that the minimizing point for this problem is (2.00, 1.00).

Convergence of the Steepest Descent Method

Let $\Omega = \{\bar{\mathbf{x}} : \nabla f(\bar{\mathbf{x}}) = \mathbf{0}\}$, and let f be the descent function. The algorithmic map $\mathbf{A} = \mathbf{MD}$, where $\mathbf{D}(\mathbf{x}) = [\mathbf{x}, \nabla f(\mathbf{x})]$, and $\mathbf{M}$ is the line search map over the closed interval $[0, \infty)$. Assuming that f is continuously differentiable, then $\mathbf{D}$ is continuous. Furthermore, $\mathbf{M}$ is closed by Theorem 8.3.1. Then the algorithmic map $\mathbf{A}$ is closed by Corollary 2 to Theorem 7.3.2. Finally, if $\mathbf{x} \notin \Omega$, then $\nabla f(\mathbf{x})^t \mathbf{d} < 0$, where $\mathbf{d} = -\nabla f(\mathbf{x})$. By Theorem 4.1.2, $\mathbf{d}$ is a descent direction, and hence $f(\mathbf{y}) < f(\mathbf{x})$ for $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. Assuming that the sequence generated by the algorithm is contained in a compact set, then by Theorem 7.2.3, the steepest descent algorithm converges to a point with zero gradient.

Zigzagging of the Steepest Descent Method

The method of steepest descent usually works quite well during early stages of the optimization process. However, as a stationary point is approached, the method usually behaves poorly, where small orthogonal steps are taken. This

TABLE 8.11 **Summary of Computations for the Method of Steepest Descent**

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	$\|\nabla f(\mathbf{x}_k)\|$	$\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$	λ_k	$\mathbf{x}_{k+1}$
1	(0.00, 3.00) 52.00	(−44.00, 24.00)	50.12	(44.00, −24.00)	0.062	(2.70, 1.51)
2	(2.70, 1.51) 0.34	(0.73, 1.28)	1.47	(−0.73, −1.28)	0.24	(2.52, 1.20)
3	(2.52, 1.20) 0.09	(0.80, −0.48)	0.93	(−0.80, 0.48)	0.11	(2.43, 1.25)
4	(2.43, 1.25) 0.04	(0.18, 0.28)	0.33	(−0.18, −0.28)	0.31	(2.37, 1.16)
5	(2.37, 1.16) 0.02	(0.30, −0.20)	0.36	(−0.30, 0.20)	0.12	(2.33, 1.18)
6	(2.33, 1.18) 0.01	(0.08, 0.12)	0.14	(−0.08, −0.12)	0.36	(2.30, 1.14)
7	(2.30, 1.14) 0.009	(0.15, −0.08)	0.17	(−0.15, 0.08)	0.13	(2.28, 1.15)
8	(2.28, 1.15) 0.007	(0.05, 0.08)	0.09			

zigzagging phenomena was encountered in Example 8.5.2 and is illustrated in Figure 8.14, in which zigzagging occurs along the valley shown by the dotted lines.

Zigzagging and poor convergence of the steepest descent algorithm at later stages can be explained by considering the following expression of the function f:

$$f(\mathbf{x}_k + \lambda \mathbf{d}) = f(\mathbf{x}_k) + \lambda \nabla f(\mathbf{x}_k)^t \mathbf{d} + \lambda \|\mathbf{d}\| \alpha(\mathbf{x}_k; \lambda \mathbf{d})$$

where $\alpha(\mathbf{x}_k; \lambda \mathbf{d}) \to 0$ as $\lambda \mathbf{d} \to \mathbf{0}$, and $\mathbf{d} = -\nabla f(\mathbf{x}_k)$. If $\mathbf{x}_k$ is close to a stationary point with zero gradient, and if f is continuously differentiable, then $\|\nabla f(\mathbf{x}_k)\|$ will be small, making the term $\lambda \nabla f(\mathbf{x}_k)^t \mathbf{d} = -\lambda \|\nabla f(\mathbf{x}_k)\|^2$ of a small order of magnitude. Since the steepest descent method employs the linear approximation of f to find a direction of movement, where the term $\lambda \|\mathbf{d}\| \alpha(\mathbf{x}_k; \lambda \mathbf{d})$ is essentially ignored, we should expect that the directions generated at late stages will not be very effective.

As we will learn in the remainder of the chapter, there are some ways to overcome the difficulties of zigzagging by *deflecting* the gradient. Rather than moving along $\mathbf{d} = -\nabla f(\mathbf{x})$, we could move along $\mathbf{d} = -\mathbf{D}\nabla f(\mathbf{x})$ or along $\mathbf{d} = -\nabla f(\mathbf{x}) + \mathbf{g}$, where $\mathbf{D}$ is an appropriate matrix, and $\mathbf{g}$ is an appropriate vector. These correction procedures will be discussed in more detail shortly.

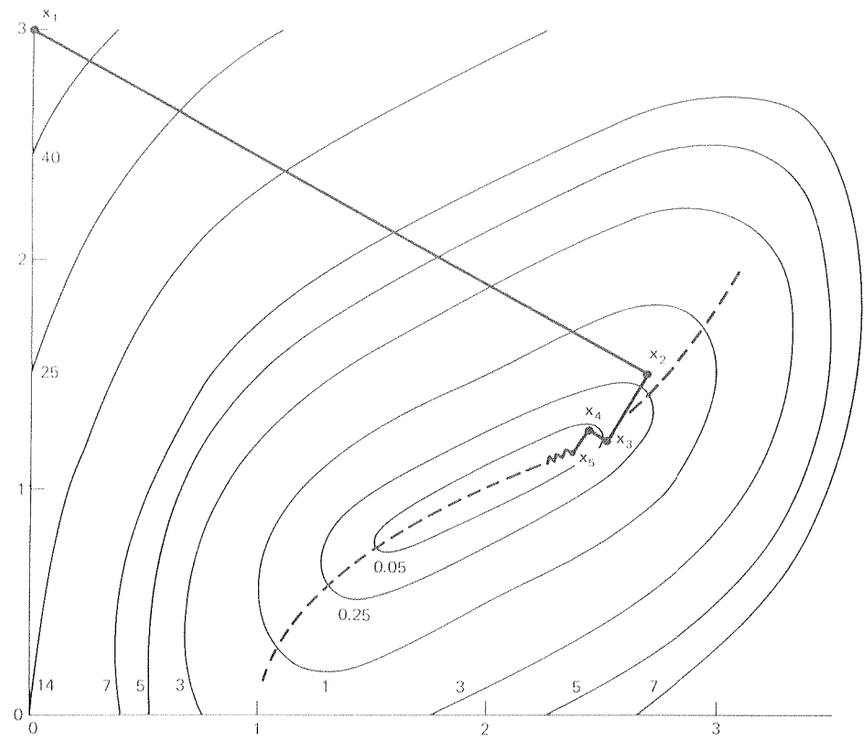

Figure 8.14 Illustration of the method of steepest descent.

The Method of Newton

In Section 8.2 we discussed Newton's method for minimizing a function of a single variable.

The method of Newton is a procedure that deflects the steepest descent direction by premultiplying it by the inverse of the Hessian matrix. This operation is motivated by finding a suitable direction for the quadratic approximation to the function rather than by finding a linear approximation to the function, as in the gradient search. In order to motivate the procedure, consider the following quadratic approximation q at a given point $\mathbf{x}_k$:

$$q(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^t(\mathbf{x}-\mathbf{x}_k) + \tfrac{1}{2}(\mathbf{x}-\mathbf{x}_k)^t\mathbf{H}(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$$

where $\mathbf{H}(\mathbf{x}_k)$ is the Hessian matrix of f at $\mathbf{x}_k$. A necessary condition for a

minimum of the quadratic approximation q is that $\nabla q(\mathbf{x}) = \mathbf{0}$, or $\nabla f(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) = \mathbf{0}$. Assuming that the inverse of $\mathbf{H}(\mathbf{x}_k)$ exists, the successor point $\mathbf{x}_{k+1}$ is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$$

The above equation gives the recursive form of the points generated by Newton's method for the multidimensional case. Assuming that $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$, that $\mathbf{H}(\bar{\mathbf{x}})$ is positive definite at a local minimum $\bar{\mathbf{x}}$, and that f is twice continuously differentiable, it follows that $\mathbf{H}(\mathbf{x}_k)$ is positive definite at points close to $\bar{\mathbf{x}}$, and hence the successor point $\mathbf{x}_{k+1}$ is well defined.

8.5.3 Example

Consider the following problem:

Minimize $\quad (x_1-2)^4 + (x_1-2x_2)^2$

The summary of the computations using Newton's method is given in Table 8.12. At each iteration $\mathbf{x}_{k+1}$ is given by $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$. After six iterations, the point $\mathbf{x}_7 = (1.83, 0.91)^t$ is reached. At this point $\|\nabla f(\mathbf{x}_7)\| = 0.04$, and the procedure is terminated. The points generated by the method are shown in Figure 8.15.

In the above example, the value of the objective function decreased at each iteration. However, this will not generally be the case, so f cannot be used as a descent function. Theorem 8.5.4 below indicates that Newton's method indeed converges, provided we start from a point close enough to the optimal point.

Convergence of the Method of Newton

In general, the points generated by the method of Newton may not converge. The reason for this is that $\mathbf{H}(\mathbf{x}_k)$ may be singular, so that $\mathbf{x}_{k+1}$ is not well defined. Even if $\mathbf{H}(\mathbf{x}_k)^{-1}$ exists, $f(\mathbf{x}_{k+1})$ is not necessarily less than $f(\mathbf{x}_k)$. However, if the starting point is close enough to a point $\bar{\mathbf{x}}$ such that $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$ and $\mathbf{H}(\bar{\mathbf{x}})$ is of full rank, then the method of Newton is well defined and converges to $\bar{\mathbf{x}}$. This is proved in Theorem 8.5.4 below by showing that all the assumptions of Theorem 7.2.3 hold, where the descent function α is given by $\alpha(\mathbf{x}) = \|\mathbf{x}-\bar{\mathbf{x}}\|$.

8.5.4 Theorem

Let $f: E_n \to E_1$ be twice continuously differentiable. Consider Newton's algorithm defined by the map $\mathbf{A}(\mathbf{x}) = \mathbf{x} - \mathbf{H}(\mathbf{x})^{-1}\nabla f(\mathbf{x})$. Let $\bar{\mathbf{x}}$ be such that $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$

TABLE 8.12 **Summary of Computations for the Method of Newton**

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	$\mathbf{H}(\mathbf{x}_k)$	$\mathbf{H}(\mathbf{x}_k)^{-1}$	$-\mathbf{H}(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$	$\mathbf{x}_{k+1}$
1	(0.00, 3.00) 52.00	(−44.0, 24.0)	$\begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{384}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 50.0 \end{bmatrix}$	(0.67, −2.67)	(0.67, 0.33)
2	(0.67, 0.33) 3.13	(−9.39, −0.04)	$\begin{bmatrix} 23.23 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{169.84}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 23.23 \end{bmatrix}$	(0.44, 0.23)	(1.11, 0.56)
3	(1.11, 0.56) 0.63	(−2.84, −0.04)	$\begin{bmatrix} 11.50 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{76}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 11.50 \end{bmatrix}$	(0.30, 0.14)	(1.41, 0.70)
4	(1.41, 0.70) 0.12	(−0.80, −0.04)	$\begin{bmatrix} 6.18 & 4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{33.44}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 6.18 \end{bmatrix}$	(0.20, 0.10)	(1.61, 0.80)
5	(1.61, 0.80) 0.02	(−0.22, −0.04)	$\begin{bmatrix} 3.83 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{14.64}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 3.83 \end{bmatrix}$	(0.13, 0.07)	(1.74, 0.87)
6	(1.74, 0.87) 0.005	(−0.07, 0.00)	$\begin{bmatrix} 2.81 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{6.48}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 2.81 \end{bmatrix}$	(0.09, 0.04)	(1.83, 0.91)
7	(1.83, 0.91) 0.0009	(0.0003, −0.04)				

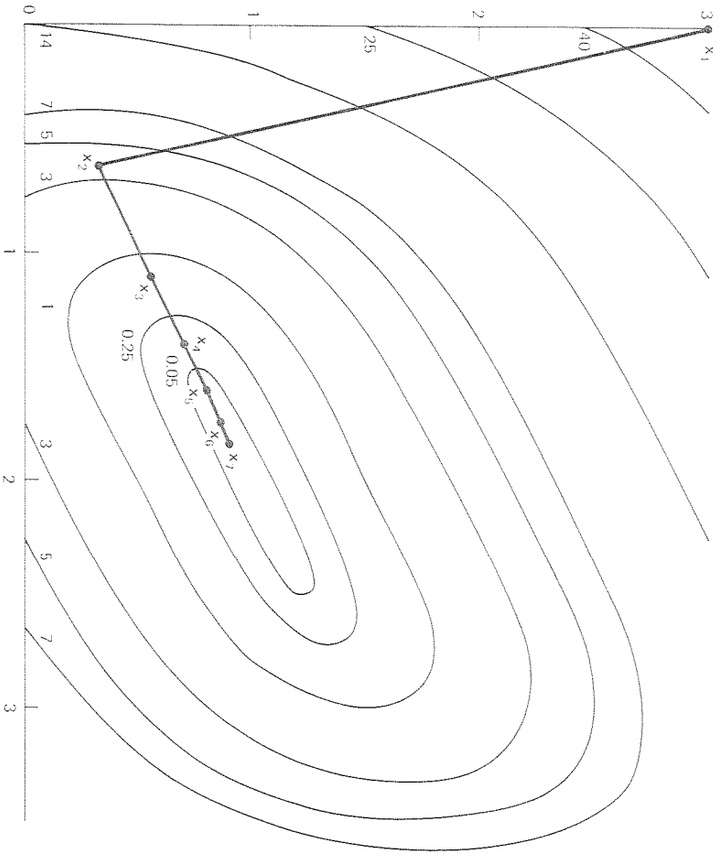

Figure 8.15 Illustration of the method of Newton.

and $\mathbf{H}(\bar{\mathbf{x}})^{-1}$ exists. Let the starting point $\mathbf{x}_1$ be sufficiently close to $\bar{\mathbf{x}}$ so that there exists $k_1, k_2 > 0$ with $k_1 k_2 < 1$ such that

1. $\|\mathbf{H}(\mathbf{x})^{-1}\|^{\dagger} \le k_1$
2. $\dfrac{\|\nabla f(\bar{\mathbf{x}}) - \nabla f(\mathbf{x}) - \mathbf{H}(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})\|}{\|\bar{\mathbf{x}} - \mathbf{x}\|} \le k_2$

for each $\mathbf{x}$ satisfying $\|\mathbf{x} - \bar{\mathbf{x}}\| \le \|\mathbf{x}_1 - \bar{\mathbf{x}}\|$. Then the algorithm converges to $\bar{\mathbf{x}}$.

Proof

Let the solution set $\Omega = \{\bar{\mathbf{x}}\}$, and let $X = \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\| \le \|\mathbf{x}_1 - \bar{\mathbf{x}}\|\}$. We prove convergence by using Theorem 7.2.3. Note that X is compact and that the map $\mathbf{A}$

† Let $\mathbf{A}$ be an $n \times n$ matrix. The norm of $\mathbf{A}$, denoted by $\|\mathbf{A}\|$, is defined by $\|\mathbf{A}\| = \text{maximum}_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|/\|\mathbf{x}\|$. Thus, for any vector $\mathbf{z}$, $\|\mathbf{A}\mathbf{z}\| \le \|\mathbf{A}\| \|\mathbf{z}\|$.

is closed on X. We now show that $\alpha(\mathbf{x}) = \|\mathbf{x} - \bar{\mathbf{x}}\|$ is indeed a descent function. Now let $\mathbf{x} \in X$, and suppose that $\mathbf{x} \neq \bar{\mathbf{x}}$. Let $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. Then, by definition of $\mathbf{A}$ and since $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$, we get

$$\begin{aligned}\mathbf{y} - \bar{\mathbf{x}} &= (\mathbf{x} - \bar{\mathbf{x}}) - \mathbf{H}(\mathbf{x})^{-1}[\nabla f(\mathbf{x}) - \nabla f(\bar{\mathbf{x}})] \\ &= \mathbf{H}(\mathbf{x})^{-1}[\nabla f(\bar{\mathbf{x}}) - \nabla f(\mathbf{x}) - \mathbf{H}(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})]\end{aligned}$$

Noting (1) and (2), it then follows that

$$\begin{aligned}\|\mathbf{y} - \bar{\mathbf{x}}\| &= \|\mathbf{H}(\mathbf{x})^{-1}[\nabla f(\bar{\mathbf{x}}) - \nabla f(\mathbf{x}) - \mathbf{H}(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})]\| \\ &\leq \|\mathbf{H}(\mathbf{x})^{-1}\| \, \|\nabla f(\bar{\mathbf{x}}) - \nabla f(\mathbf{x}) - \mathbf{H}(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})\| \\ &\leq k_1 k_2 \|\mathbf{x} - \bar{\mathbf{x}}\| \\ &< \|\mathbf{x} - \bar{\mathbf{x}}\|\end{aligned}$$

This shows that α is indeed a descent function. By the corollary to Theorem 7.2.3, the result follows.

Modification of Newton's Method

We now discuss a modification of Newton's method that guarantees convergence regardless of the starting point. Given $\mathbf{x}$, consider the direction $\mathbf{d} = -\mathbf{B}\nabla f(\mathbf{x})$, where $\mathbf{B}$ is a symmetric positive definite matrix to be determined later. The successor point is $\mathbf{y} = \mathbf{x} + \hat{\lambda}\mathbf{d}$, where $\hat{\lambda}$ is an optimal solution to the problem to minimize $f(\mathbf{x} + \lambda\mathbf{d})$ subject to $\lambda \geq 0$.

We now specify the matrix $\mathbf{B}$ as $(\varepsilon\mathbf{I} + \mathbf{H})^{-1}$, where $\mathbf{H} = \mathbf{H}(\mathbf{x})$. The scalar $\varepsilon \geq 0$ is determined as follows. Fix $\delta > 0$, and let $\varepsilon \geq 0$ be the smallest scalar that would make all the eigenvalues of the matrix $(\varepsilon\mathbf{I} + \mathbf{H})$ greater or equal to δ. Since the eigenvalues of $\varepsilon\mathbf{I} + \mathbf{H}$ are all positive, $\varepsilon\mathbf{I} + \mathbf{H}$ is positive definite and invertible. In particular, $\mathbf{B} = (\varepsilon\mathbf{I} + \mathbf{H})^{-1}$ is also positive definite. Since the eigenvalues of a matrix depend continuously on its elements, ε is a continuous function of $\mathbf{x}$, and hence the point-to-point map $\mathbf{D}: E_n \to E_n \times E_n$ defined by $\mathbf{D}(\mathbf{x}) = (\mathbf{x}, \mathbf{d})$ is continuous. Thus the algorithmic map $\mathbf{A} = \mathbf{MD}$, where $\mathbf{M}$ is the usual line search map over $\{\lambda : \lambda \geq 0\}$.

Let $\Omega = \{\bar{\mathbf{x}} : \nabla f(\bar{\mathbf{x}}) = \mathbf{0}\}$, and let $\mathbf{x} \notin \Omega$. Since $\mathbf{B}$ is positive definite, $\mathbf{d} = -\mathbf{B}\nabla f(\mathbf{x}) \neq \mathbf{0}$, and by Theorem 8.3.1, it follows that $\mathbf{M}$ is closed at $(\mathbf{x}, \mathbf{d})$. Furthermore, since $\mathbf{D}$ is a continuous function, by Corollary 2 to Theorem 7.3.2, $\mathbf{A} = \mathbf{MD}$ is closed over the complement of Ω.

In order to invoke Theorem 7.2.3, we need to specify a continuous descent function. Suppose that $\mathbf{x} \notin \Omega$, and let $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. Note that $\nabla f(\mathbf{x})^t\mathbf{d} = -\nabla f(\mathbf{x})^t\mathbf{B}\nabla f(\mathbf{x}) < 0$ since $\mathbf{B}$ is positive definite and $\nabla f(\mathbf{x}) \neq \mathbf{0}$. Thus, $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$, and by Theorem 4.1.2, $f(\mathbf{y}) < f(\mathbf{x})$. Therefore, f is indeed a descent function. Assuming that the sequence generated by the algorithm is contained in a compact set, by Theorem 7.2.3, it follows that the algorithm converges.

It should be noted that if the smallest eigenvalue of $\mathbf{H}(\bar{\mathbf{x}})$ is greater than or equal to δ, then as the points $\{\mathbf{x}_k\}$ generated by the algorithm approach $\bar{\mathbf{x}}$, ε_k will be equal to zero. Thus $\mathbf{d}_k = -\mathbf{H}(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$, and the algorithm reduces to that of Newton.

8.6 Methods Using Conjugate Directions

In this section we discuss several procedures that are based on the important concept of conjugacy. Some of these procedures use derivatives, while others only use functional evaluations. The notion of conjugacy defined below is very useful in unconstrained optimization. In particular, if the objective function is quadratic, then by searching along conjugate directions, the minimum point can be obtained in, at most, n steps.

8.6.1 Definition

Let $\mathbf{H}$ be an $n \times n$ symmetric matrix. The vectors $\mathbf{d}_1, \ldots, \mathbf{d}_k$ are called *$\mathbf{H}$-conjugate* or simply *conjugate* if they are linearly independent and if $\mathbf{d}_i^t\mathbf{H}\mathbf{d}_j = 0$ for $i \neq j$.

The following example illustrates the notion of conjugacy and highlights the significance of optimizing along conjugate directions for quadratic functions.

8.6.2 Example

Consider the following problem:

$$\text{Minimize} \quad -12x_2 + 4x_1^2 + 4x_2^2 - 4x_1x_2$$

Note that the Hessian matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$$

We now generate two conjugate directions $\mathbf{d}_1$ and $\mathbf{d}_2$. Suppose we choose $\mathbf{d}_1^t = (1, 0)$. Then $\mathbf{d}_2^t = (a, b)$ must satisfy $0 = \mathbf{d}_1^t\mathbf{H}\mathbf{d}_2 = 8a - 4b$. In particular, we may choose $a = 1$ and $b = 2$ so that $\mathbf{d}_2^t = (1, 2)$. It may be noted that the conjugate directions are not unique.

If we minimize the objective function f starting from $\mathbf{x}_1^t = (-\frac{1}{2}, 1)$ along the direction $\mathbf{d}_1$, we get the point $\mathbf{x}_2^t = (\frac{1}{2}, 1)$. Now, starting from $\mathbf{x}_2$ and minimizing along $\mathbf{d}_2$, we get $\mathbf{x}_3^t = (1, 2)$. Note that $\mathbf{x}_3$ is the minimum point.

The contours of the objective function and the path taken to reach the optimal point are shown in Figure 8.16. The reader can easily verify that, starting from any point and minimizing along $\mathbf{d}_1$ and $\mathbf{d}_2$, the optimal point is reached in, at most, two steps.