TABLE 8.7 Summary of Computations for the Method of Hooke and Jeeves Using Line Search

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	j	$\mathbf{y}_j$	$\mathbf{d}_j$	λ_j	$\mathbf{y}_{j+1}$	$\mathbf{d}$	$\hat{\lambda}$	$\mathbf{y}_3+\hat{\lambda}\mathbf{d}$
1	(0.00, 3.00)	1	(0.00, 3.00)	(1.0, 0.0)	3.13	(3.13, 3.00)	—	—	—
	52.00	2	(3.13, 3.00)	(0.0, 1.0)	−1.44	(3.13, 1.56)	(3.13, −1.44)	−0.10	(2.82, 1.70)
2	(3.13, 1.56)	1	(2.82, 1.70)	(1.0, 0.0)	−0.12	(2.70, 1.70)	—	—	—
	1.63	2	(2.70, 1.70)	(0.0, 1.0)	−0.35	(2.70, 1.35)	(−0.43, −0.21)	1.50	(2.06, 1.04)
3	(2.70, 1.35)	1	(2.06, 1.04)	(1.0, 0.0)	−0.02	(2.04, 1.04)	—	—	—
	0.24	2	(2.04, 1.04)	(0.0, 1.0)	−0.02	(2.04, 1.02)	(−0.66, −0.33)	0.06	(2.00, 1.00)
4	(2.04, 1.02)	1	(2.00, 1.00)	(1.0, 0.0)	0.00	(2.00, 1.00)	—	—	—
	0.000003	2	(2.00, 1.00)	(0.0, 1.0)	0.00	(2.00, 1.00)			
5	(2.00, 1.00)								
	0.00								

Convergence of the Method of Hooke and Jeeves

Suppose that f is differentiable, and let the solution set $\Omega = \{\bar{\mathbf{x}} : \nabla f(\bar{\mathbf{x}}) = \mathbf{0}\}$. Note that each iteration of the method of Hooke and Jeeves consists of an application of the cyclic coordinate method, in addition to a pattern search. Let the cyclic coordinate search be denoted by the map $\mathbf{B}$ and the pattern search be denoted by the map $\mathbf{C}$. Using an argument similar to that of Theorem 7.3.5, it follows that $\mathbf{B}$ is closed. If the minimum of f along any line is unique and letting $\alpha = f$, then $\alpha(\mathbf{y}) < \alpha(\mathbf{x})$ for $\mathbf{x} \notin \Omega$. By definition of $\mathbf{C}$, $\alpha(\mathbf{z}) \leq \alpha(\mathbf{y})$ for $\mathbf{z} \in \mathbf{C}(\mathbf{y})$. Assuming that $\bigwedge = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_1)\}$, where $\mathbf{x}_1$ is the starting point, is compact, convergence of the procedure is established by Theorem 7.3.4.

The Method of Hooke and Jeeves with Discrete Steps

As mentioned earlier, the method of Hooke and Jeeves, as originally proposed, does not perform line search but replaces it with a simple scheme involving functional evaluations. A summary of the method is given below.

Initialization Step Let $\mathbf{d}_1, \ldots, \mathbf{d}_n$ be the coordinate directions. Choose a scalar $\varepsilon > 0$ to be used for terminating the algorithm. Furthermore, choose an initial step size, $\Delta \geq \varepsilon$, and an acceleration factor, $\alpha > 0$. Choose a starting point $\mathbf{x}_1$, let $\mathbf{y}_1 = \mathbf{x}_1$, let $k = j = 1$, and go to the main step.

Main Step

1. If $f(\mathbf{y}_j + \Delta\mathbf{d}_j) < f(\mathbf{y}_j)$, the trial is termed a *success*; let $\mathbf{y}_{j+1} = \mathbf{y}_j + \Delta\mathbf{d}_j$, and go to step 2. If, however, $f(\mathbf{y}_j + \Delta\mathbf{d}_j) \geq f(\mathbf{y}_j)$, the trial is deemed a *failure*. In this case, if $f(\mathbf{y}_j - \Delta\mathbf{d}_j) < f(\mathbf{y}_j)$, let $\mathbf{y}_{j+1} = \mathbf{y}_j - \Delta\mathbf{d}_j$, and go to step 2; if $f(\mathbf{y}_j - \Delta\mathbf{d}_j) \geq f(\mathbf{y}_j)$, let $\mathbf{y}_{j+1} = \mathbf{y}_j$, and go to step 2.
2. If $j < n$, replace j by $j+1$, and repeat step 1. Otherwise, go to step 3 if $f(\mathbf{y}_{n+1}) < f(\mathbf{x}_k)$, and go to step 4 if $f(\mathbf{y}_{n+1}) \geq f(\mathbf{x}_k)$.
3. Let $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$, and let $\mathbf{y}_1 = \mathbf{x}_{k+1} + \alpha(\mathbf{x}_{k+1} - \mathbf{x}_k)$. Replace k by $k+1$, let $j = 1$, and go to step 1.
4. If $\Delta \leq \varepsilon$ stop; $\mathbf{x}_k$ is the solution. Otherwise, replace Δ by $\Delta/2$. Let $\mathbf{y}_1 = \mathbf{x}_k$, $\mathbf{x}_{k+1} = \mathbf{x}_k$, replace k by $k+1$, let $j = 1$, and repeat step 1.

The reader may note that steps 1 and 2 above describe an exploratory search. Furthermore, step 3 is an acceleration step along the direction $\mathbf{x}_{k+1} - \mathbf{x}_k$. Note that a decision whether to accept or reject the acceleration step is not made until after an exploratory search is performed. In step 4, the step size Δ is reduced. The procedure could easily be modified so that different step sizes are

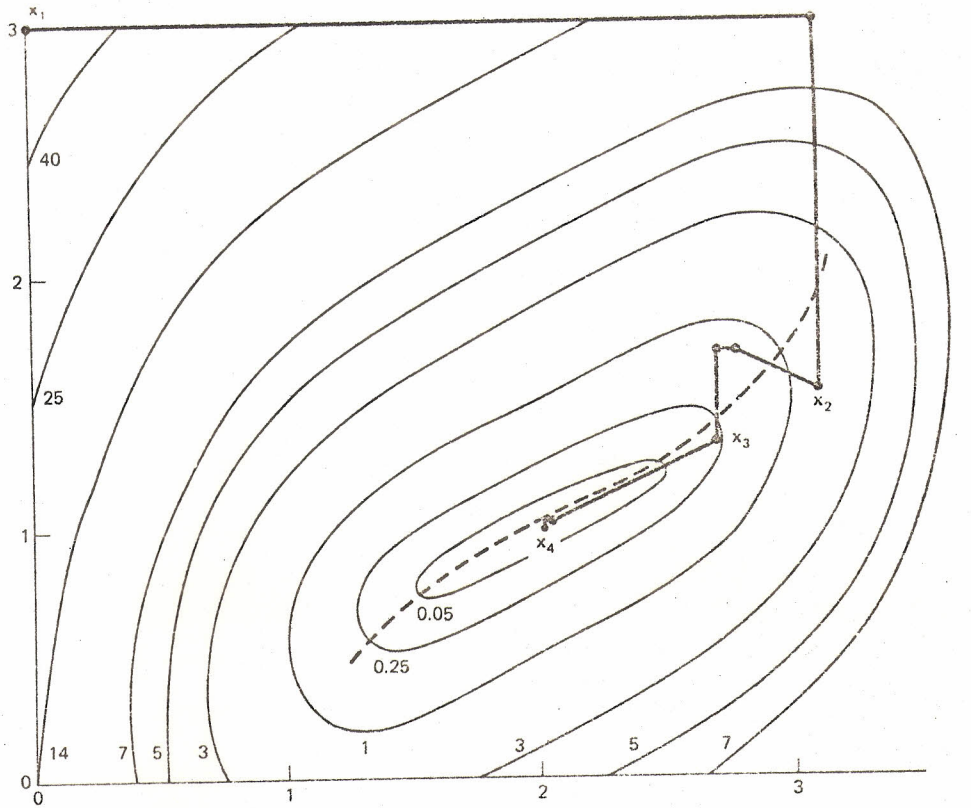

Figure 8.8 Illustration of the method of Hooke and Jeeves using line search.

used along the different directions. This is sometimes adopted for the purpose of scaling.

8.4.3 Example

Consider the following problem:

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

We solve the problem using the method of Hooke and Jeeves with discrete steps. The parameters α and Δ are chosen as 1.0 and 0.2, respectively. Figure 8.9 shows the path taken by the algorithm starting from (0.0, 3.0). The points generated are numbered sequentially, and the acceleration step that is rejected is shown by the dotted lines. From this particular starting point, the optimal solution is easily reached.

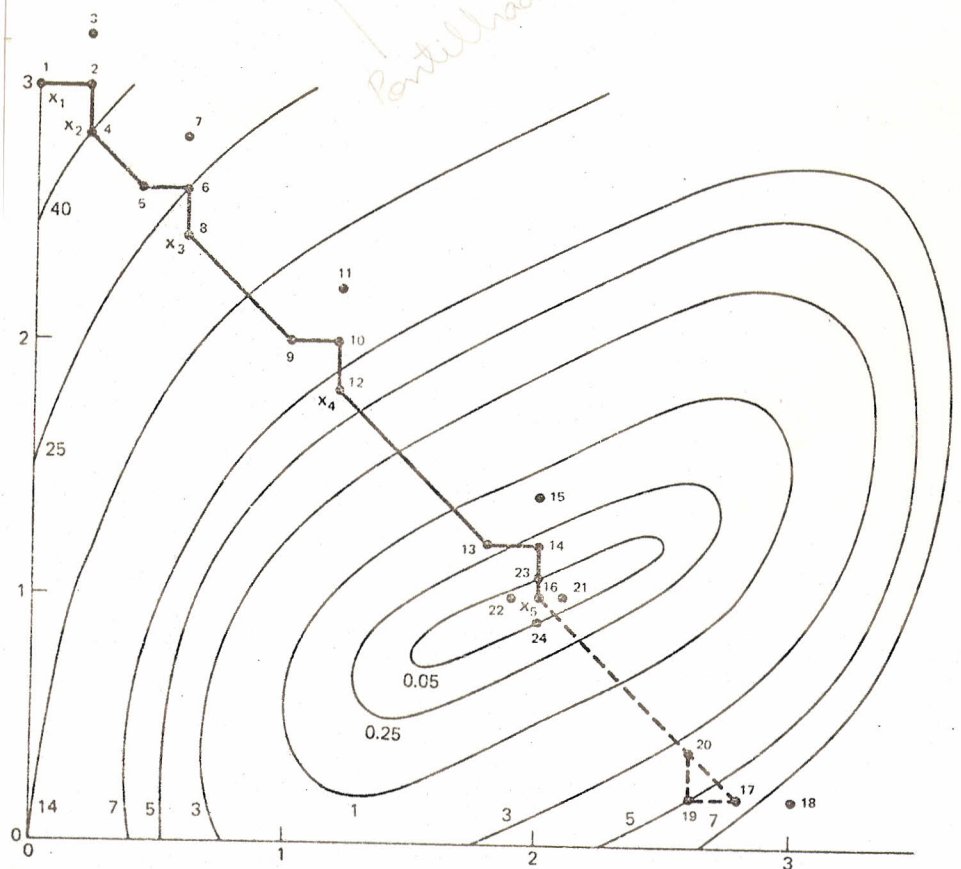

Figure 8.9 Illustration of the method of Hooke and Jeeves using discrete steps starting from (0.0, 3.0). (The numbers denote the order in which points are generated.)

In order to give a more comprehensive illustration, Table 8.8 summarizes the computations starting from the new initial point (2.0, 3.0). Here (S) denotes that the trial is a success and (F) denotes that the trial is a failure. At the first iteration, and at subsequent iterations whenever $f(\mathbf{y}_3)\geq f(\mathbf{x}_k)$, the vector $\mathbf{y}_1$ is taken as $\mathbf{x}_k$. Otherwise, $\mathbf{y}_1=2\mathbf{x}_{k+1}-\mathbf{x}_k$. Note that at the end of iteration $k=10$, the point (1.70, 0.80) is reached with objective value 0.02. The procedure is stopped here with the termination parameter $\varepsilon=0.1$. If a greater degree of accuracy is required, Δ should be reduced to 0.05.

Figure 8.10 illustrates the path taken by the method. The points generated are again numbered sequentially, and dotted lines represent rejected acceleration steps.

The Method of Rosenbrock

As originally proposed, the method of Rosenbrock does not employ line search but rather takes discrete steps along the search directions. We present here a continuous version of the method that utilizes line search. At each iteration, the procedure searches iteratively along n linearly independent and orthogonal directions. When a new point is reached at the end of an iteration, a new set of orthogonal vectors is constructed. In Figure 8.11, the new directions are denoted by $\bar{\mathbf{d}}_1$ and $\bar{\mathbf{d}}_2$.

Construction of the Search Directions

Let $\mathbf{d}_1,\ldots,\mathbf{d}_n$ be linearly independent vectors, each with norm equal to one. Furthermore, suppose that these vectors are mutually orthogonal, that is $\mathbf{d}_i^t\mathbf{d}_j=0$ for $i\neq j$. Starting from the current vector $\mathbf{x}_k$, the objective function f is minimized along each of the directions iteratively, resulting in the point $\mathbf{x}_{k+1}$. In particular, $\mathbf{x}_{k+1}-\mathbf{x}_k=\sum_{j=1}^{n}\lambda_j\mathbf{d}_j$, where λ_j is the distance moved along $\mathbf{d}_j$. The new collection of directions $\bar{\mathbf{d}}_1,\ldots,\bar{\mathbf{d}}_n$ are formed by the Gram-Schmidt procedure as follows:

$$\mathbf{a}_j=\begin{cases}\mathbf{d}_j & \text{if } \lambda_j=0\\ \sum_{i=j}^{n}\lambda_i\mathbf{d}_i & \text{if } \lambda_j\neq 0\end{cases}$$

$$\mathbf{b}_j=\begin{cases}\mathbf{a}_j & j=1\\ \mathbf{a}_j-\sum_{i=1}^{j-1}(\mathbf{a}_j^t\bar{\mathbf{d}}_i)\bar{\mathbf{d}}_i & j\geq 2\end{cases} \qquad (8.9)$$

$$\bar{\mathbf{d}}_j=\frac{\mathbf{b}_j}{\|\mathbf{b}_j\|}$$

TABLE 8.8 Summary of the Computations for the Method of Hooke and Jeeves with Discrete Steps

Iteration k	Δ	x_k $f(x_k)$	j	y_j $f(y_j)$	d_j	$y_j+\Delta d_j$ $f(y_j+\Delta d_j)$	$y_j-\Delta d_j$ $f(y_j-\Delta d_j)$
1	0.2	(2.00, 3.00)	1	(2.00, 3.00)	(1.0, 0.0)	(2.20, 3.00)	—
		16.00		16.00		14.44(S)	
			2	(2.20, 3.00)	(0.0, 1.0)	(2.20, 3.20)	(2.20, 2.80)
				14.44		17.64(F)	11.56(S)
2	0.2	(2.20, 2.80)	1	(2.40, 2.60)	(1.0, 0.0)	(2.60, 2.60)	—
		11.56		7.87		6.89(S)	
			2	(2.60, 2.60)	(0.0, 1.0)	(2.60, 2.80)	(2.60, 2.40)
				6.89		9.13(F)	4.97(S)
3	0.2	(2.60, 2.40)	1	(3.00, 2.00)	(1.0, 0.0)	(3.20, 2.00)	(2.80, 2.00)
		4.97		2.00		2.71(F)	1.85(S)
			2	(2.80, 2.00)	(0.0, 1.0)	(2.80, 2.20)	(2.80, 1.80)
				1.85		2.97(F)	1.05(S)
4	0.2	(2.80, 1.80)	1	(3.00, 1.20)	(1.0, 0.0)	(3.20, 1.20)	(2.80, 1.20)
		1.05		1.36		2.71(F)	0.57(S)
			2	(2.80, 1.20)	(0.0, 1.0)	(2.80, 1.40)	—
				0.57		0.41(S)	
5	0.2	(2.80, 1.40)	1	(2.80, 1.00)	(1.0, 0.0)	(3.00, 1.00)	(2.60, 1.00)
		0.41		1.05		2.00(F)	0.49(S)
			2	(2.60, 1.00)	(0.0, 1.0)	(2.60, 1.20)	—
				0.49		0.17(S)	
6	0.2	(2.60, 1.20)	1	(2.40, 1.00)	(1.0, 0.0)	(2.60, 1.00)	(2.20, 1.00)
		0.17		0.19		0.49(F)	0.04(S)
			2	(2.20, 1.00)	(0.0, 1.0)	(2.20, 1.20)	(2.20, 0.80)
				0.04		0.04(F)	0.36(F)
7	0.2	(2.20, 1.00)	1	(1.80, 0.80)	(1.0, 0.0)	(2.00, 0.80)	(1.60, 0.80)
		0.04		0.04		0.16(F)	0.03(S)
			2	(1.60, 0.80)	(0.0, 1.0)	(1.60, 1.00)	(1.60, 0.60)
				0.03		0.19(F)	0.19(F)
8	0.2	(1.60, 0.80)	1	(1.00, 0.60)	(1.0, 0.0)	(1.20, 0.60)	—
		0.03		0.67		0.41(S)	
			2	(1.20, 0.60)	(0.0, 1.0)	(1.20, 0.80)	(1.20, 0.40)
				0.41		0.57(F)	0.57(F)
9	0.1	(1.60, 0.80)	1	(1.60, 0.80)	(1.0, 0.0)	(1.70, 0.80)	—
		0.03		0.03		0.02(S)	
			2	(1.70, 0.80)	(0.0, 1.0)	(1.70, 0.90)	(1.70, 0.70)
				0.02		0.02(F)	0.10(F)
10	0.1	(1.70, 0.80)	1	(1.80, 0.80)	(1.0, 0.0)	(1.90, 0.80)	(1.70, 0.80)
		0.02		0.04		0.09(F)	0.02(S)
			2	(1.70, 0.80)	(0.0, 1.0)	(1.70, 0.90)	(1.70, 0.70)
				0.02		0.02(F)	0.10(F)

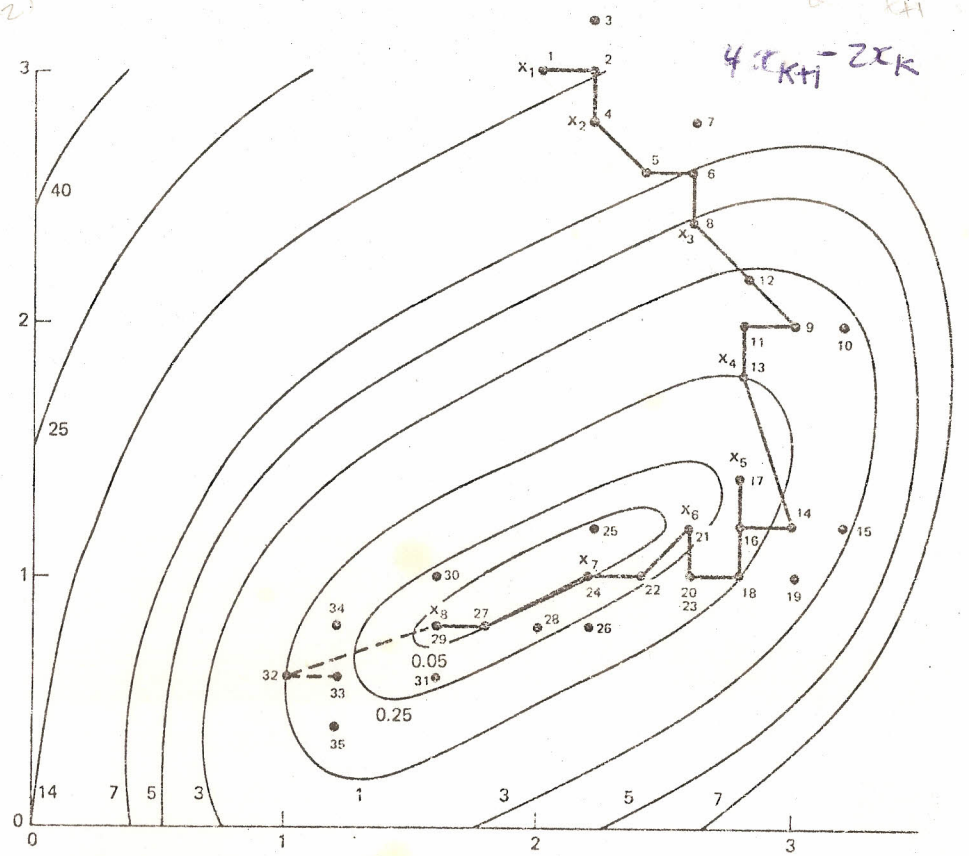

Figure 8.10 Illustration of the method of Hooke and Jeeves using discrete steps starting from (2.0, 3.0). (The numbers denote the order in which points are generated.)

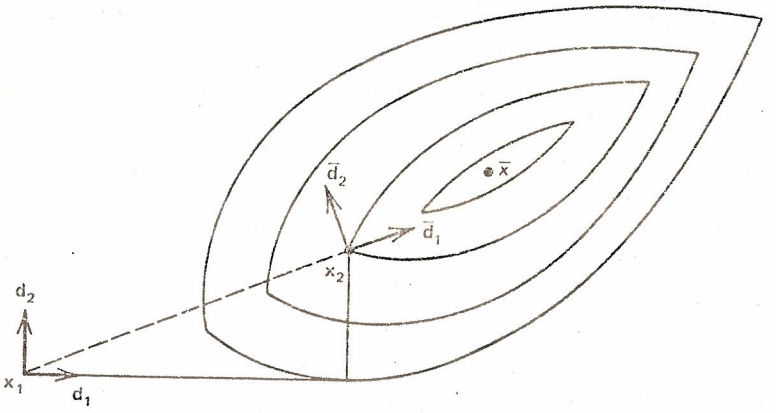

Figure 8.11 Construction of new search directions in Rosenbrock's procedure.

Lemma 8.4.4 below shows that the new directions establihsed by the Rosenbrock procedure are indeed linearly independent and orthogonal.

8.4.4 Lemma

Suppose that the vectors $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are linearly independent and mutually orthogonal. Then the directions $\bar{\mathbf{d}}_1, \ldots, \bar{\mathbf{d}}_n$ defined by (8.9) are also linearly independent and mutually orthogonal for any set of $\lambda_1, \ldots, \lambda_n$. Furthermore, if $\lambda_j = 0$ then $\bar{\mathbf{d}}_j = \mathbf{d}_j$.

Proof

We first show that $\mathbf{a}_1, \ldots, \mathbf{a}_n$ are linearly independent. Suppose that $\sum_{j=1}^n \mu_j \mathbf{a}_j = \mathbf{0}$. Let $I = \{j : \lambda_j = 0\}$, and let $J(j) = \{i : i \notin I, i \leq j\}$. Noting (8.9), we get

$$\mathbf{0} = \sum_{j=1}^n \mu_j \mathbf{a}_j = \sum_{j \in I} \mu_j \mathbf{d}_j + \sum_{j \notin I} \mu_j \left(\sum_{i=j}^n \lambda_i \mathbf{d}_i \right)$$

$$= \sum_{j \in I} \mu_j \mathbf{d}_j + \sum_{j \notin I} \left(\lambda_j \sum_{i \in J(j)} \mu_i \right) \mathbf{d}_j$$

Since $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are linearly independent, $\mu_j = 0$ for $j \in I$ and $\lambda_j \sum_{i \in J(j)} \mu_i = 0$ for $j \notin I$. But $\lambda_j \neq 0$ for $j \notin I$, and hence $\sum_{i \in J(j)} \mu_i = 0$. Therefore $\mu_1 = \ldots = \mu_n = 0$, and hence $\mathbf{a}_1, \ldots, \mathbf{a}_n$ are linearly independent.

To show that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent, we use the following induction argument. Since $\mathbf{b}_1 = \mathbf{a}_1 \neq \mathbf{0}$, it suffices to show that if $\mathbf{b}_1, \ldots, \mathbf{b}_k$ are linearly independent, then $\mathbf{b}_1, \ldots, \mathbf{b}_k, \mathbf{b}_{k+1}$ are also linearly independent. Suppose that $\sum_{j=1}^{k+1} \alpha_j \mathbf{b}_j = \mathbf{0}$. Using the definition of $\mathbf{b}_{k+1}$ in (8.9), we get

$$\mathbf{0} = \sum_{j=1}^k \alpha_j \mathbf{b}_j + \alpha_{k+1} \mathbf{b}_{k+1} = \sum_{j=1}^k \left[\alpha_j - \frac{\alpha_{k+1} (\mathbf{a}_{k+1}^t \bar{\mathbf{d}}_j)}{\|\mathbf{b}_j\|} \right] \mathbf{b}_j + \alpha_{k+1} \mathbf{a}_{k+1} \qquad (8.10)$$

From (8.9), it follows that each vector $\mathbf{b}_j$ is a linear combination of $\mathbf{a}_1, \ldots, \mathbf{a}_j$. Since $\mathbf{a}_1, \ldots, \mathbf{a}_{k+1}$ are linearly independent, it follows from (8.10) that $\alpha_{k+1} = 0$. Since $\mathbf{b}_1, \ldots, \mathbf{b}_k$ are assumed linearly independent by the induction hypotheses, from (8.10) we get $\alpha_j - \alpha_{k+1} (\mathbf{a}_{k+1}^t \bar{\mathbf{d}}_j) / \|\mathbf{b}_j\| = 0$ for $j = 1, \ldots, k$. Since $\alpha_{k+1} = 0$, $\alpha_j = 0$ for each j. This shows that $\mathbf{b}_1, \ldots, \mathbf{b}_{k+1}$ are linearly independent. By definition of $\bar{\mathbf{d}}_j$, linear independence of $\bar{\mathbf{d}}_1, \ldots, \bar{\mathbf{d}}_n$ is immediate.

Now we show orthogonality of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and hence orthogonality of $\bar{\mathbf{d}}_1, \ldots, \bar{\mathbf{d}}_n$. From (8.9) $\mathbf{b}_1^t \mathbf{b}_2 = 0$, and thus it sufficies to show that if $\mathbf{b}_1, \ldots, \mathbf{b}_k$ are mutually orthogonal, then $\mathbf{b}_1, \ldots, \mathbf{b}_k, \mathbf{b}_{k+1}$ are also mutually orthogonal. From (8.10) and noting that $\mathbf{b}_j^t \bar{\mathbf{d}}_i = 0$ for $i \neq j$, it follows that

$$\mathbf{b}_j^t \mathbf{b}_{k+1} = \mathbf{b}_j^t \left[\mathbf{a}_{k+1} - \sum_{i=1}^k (\mathbf{a}_{k+1}^t \bar{\mathbf{d}}_i) \bar{\mathbf{d}}_i \right]$$

$$= \mathbf{b}_j^t \mathbf{a}_{k+1} - (\mathbf{a}_{k+1}^t \bar{\mathbf{d}}_j) \mathbf{b}_j^t \bar{\mathbf{d}}_j = 0$$

Thus, $\mathbf{b}_1, \ldots, \mathbf{b}_{k+1}$ are mutually orthogonal.

To complete the proof, we show that $\bar{\mathbf{d}}_j = \mathbf{d}_j$ if $\lambda_j = 0$. From (8.9), if $\lambda_j = 0$, we get

$$\mathbf{b}_j = \mathbf{d}_j - \sum_{i=1}^{j-1} \frac{1}{\|\mathbf{b}_i\|} (\mathbf{d}_j^t \mathbf{b}_i) \bar{\mathbf{d}}_i \qquad (8.11)$$

Note that $\mathbf{b}_i$ is a linear combination of $\mathbf{a}_1, \ldots, \mathbf{a}_i$, so that $\mathbf{b}_i = \sum_{r=1}^i \beta_{ir} \mathbf{a}_r$. From (8.9), it thus follows that

$$\mathbf{b}_i = \sum_{r \in R} \beta_{ir} \mathbf{d}_r + \sum_{r \in \bar{R}} \beta_{ir} \left(\sum_{s=r}^n \lambda_s \mathbf{d}_s \right) \qquad (8.12)$$

where $R = \{r : r \leq i, \lambda_r = 0\}$ and $\bar{R} = \{r : r \leq i, \lambda_r \neq 0\}$. Consider $i < j$ and note that $\mathbf{d}_j^t \mathbf{d}_\nu = 0$ for $\nu \neq j$. For $r \in R$, $r \leq i < j$ and hence $\mathbf{d}_j^t \mathbf{d}_r = 0$. For $r \notin R$, $\mathbf{d}_j^t (\sum_{s=r}^n \lambda_s \mathbf{d}_s) = \lambda_j \mathbf{d}_j^t \mathbf{d}_j = \lambda_j$. By assumption, $\lambda_j = 0$, and thus multiplying (8.12) by $\mathbf{d}_j^t$ we get $\mathbf{d}_j^t \mathbf{b}_i = 0$ for $i < j$. From (8.11) it follows that $\mathbf{b}_j = \mathbf{d}_j$ and hence $\bar{\mathbf{d}}_j = \mathbf{d}_j$. This completes the proof.

From the above theorem, if $\lambda_j = 0$, then the new direction $\bar{\mathbf{d}}_j$ is equal to the old direction $\mathbf{d}_j$. Hence, we only need to compute new directions for those indices with $\lambda_j \neq 0$.

Summary of the Method of Rosenbrock Using Line Search

Now we give a summary of Rosenbrock's method using line search for minimizing a function f of several variables. As we will shortly show, if f is differentiable, then the method converges to a point with zero gradient.

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

Main Step

1. Let λ_j be an optimal solution to the problem to minimize $f(\mathbf{y}_j + \lambda \mathbf{d}_j)$ subject to $\lambda \in E_1$, and let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$. If $j < n$, replace j by $j+1$, and repeat step 1. Otherwise, go to step 2.
2. Let $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$. If $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$, then stop; otherwise, let $\mathbf{y}_1 = \mathbf{x}_{k+1}$, replace k by $k+1$, let $j = 1$, and go to step 3.
3. Form a new set of linearly independent orthogonal search directions by (8.9). Denote these new directions by $\mathbf{d}_1, \ldots, \mathbf{d}_n$, and repeat step 1.

8.4.5 Example

Consider the following problem:

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

TABLE 8.9 **Summary of Computations for the Method of Rosenbrock Using Line Search**

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	j	$\mathbf{y}_j$ $f(\mathbf{y}_j)$	$\mathbf{d}_j$	λ_j	$\mathbf{y}_{j+1}$ $f(\mathbf{y}_{j+1})$
1	(0.00, 3.00)	1	(0.00, 3.00)	(1.00, 0.00)	3.13	(3.13, 3.00)
	52.00		52.00			9.87
		2	(3.13, 3.00)	(0.00, 1.00)	−1.44	(3.13, 1.56)
			9.87			1.63
2	(3.13, 1.56)	1	(3.13, 1.56)	(0.91, −0.42)	−0.34	(2.82, 1.70)
	1.63		1.63			0.79
		2	(2.82, 1.70)	(−0.42, −0.91)	0.51	(2.61, 1.24)
			0.79			0.16
3	(2.61, 1.24)	1	(2.61, 1.24)	(−0.85, −0.52)	0.38	(2.29, 1.04)
	0.16		0.16			0.05
		2	(2.29, 1.04)	(0.52, −0.85)	−0.10	(2.24, 1.13)
			0.05			0.004
4	(2.24, 1.13)	1	(2.24, 1.13)	(−0.96, −0.28)	0.04	(2.20, 1.12)
	0.004		0.004			0.003
		2	(2.20, 1.12)	(0.28, −0.96)	0.02	(2.21, 1.10)
			0.003			0.002

We solve this problem by the method of Rosenbrock using line search. Table 8.9 summarizes the computations starting from the point (0.00, 3.00). The point $\mathbf{y}_2$ is obtained by optimizing the function along the direction $\mathbf{d}_1$ starting from $\mathbf{y}_1$, and $\mathbf{y}_3$ is obtained by optimizing the function along the direction $\mathbf{d}_2$ starting from $\mathbf{y}_2$. After the first iteration, we have $\lambda_1 = 3.13$ and $\lambda_2 = -1.44$. Using (8.9), the new search directions are (0.91, −0.42) and (−0.42, −0.91). After four iterations, the point (2.21, 1.10) is reached, and the corresponding objective function value is 0.002. We now have $\|\mathbf{x}_4 - \mathbf{x}_5\| = 0.15$, and the procedure is stopped.

In Figure 8.12, the progress of the method is shown. It may be interesting to compare this figure with Figure 8.13, which is given later for the method of Rosenbrock using discrete steps.

Convergence of the Method of Rosenbrock

Note that, according to Lemma 8.4.4, the search directions employed by the method are linearly independent, mutually orthogonal, and each has norm 1. Thus, at any given iteration, the matrix $\mathbf{D}$ denoting the search directions satisfies $\mathbf{D}'\mathbf{D} = \mathbf{I}$. Thus, $\det[\mathbf{D}] = 1$, and hence assumption (1) of Theorem 7.3.5

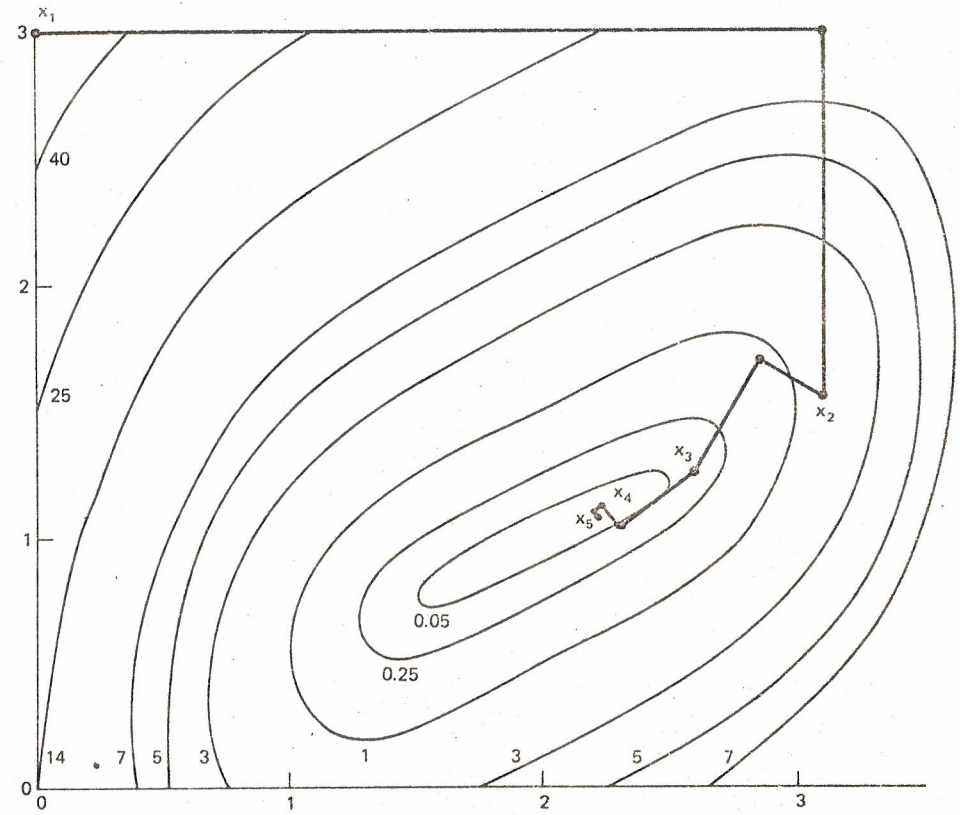

Figure 8.12 Illustration of the method of Rosenbrock using line search.

holds. By this theorem, it then follows that the method of Rosenbrock using line search converges to a stationary point if the following assumptions are true.

1. The minimum of f along any line in E_n is unique.
2. The sequence of points generated by the algorithm is contained in a compact subset of E_n.

Rosenbrock's Method with Discrete Steps

As mentioned earlier, the method proposed by Rosenbrock avoids line search. Instead, functional values are made at specific points. furthermore, an acceleration feature is incorporated by suitably increasing or decreasing the step lengths as the method proceeds. A summary of the method is given below.

Initialization Step Let $\varepsilon > 0$ be the termination scalar, let $\alpha > 1$ be the expansion factor, and let $\beta \in (-1, 0)$ be the contraction factor. Choose $\mathbf{d}_1, \ldots, \mathbf{d}_n$ as the coordinate directions, and let $\bar{\Delta}_1, \ldots, \bar{\Delta}_n > 0$ be the initial step sizes along these directions. Choose a starting point $\mathbf{x}_1$, let $\mathbf{y}_1 = \mathbf{x}_1$, $k = j = 1$, let $\Delta_j = \bar{\Delta}_j$ for each j, and go to the main step.

Main Step

1. If $f(\mathbf{y}_j + \Delta_j \mathbf{d}_j) < f(\mathbf{y}_j)$, the jth trial is deemed a *success*, $\mathbf{y}_{j+1} = \mathbf{y}_j + \Delta_j \mathbf{d}_j$, and Δ_j is replaced by $\alpha \Delta_j$. If, on the other hand, $f(\mathbf{y}_j + \Delta_j \mathbf{d}_j) \geq f(\mathbf{y}_j)$, the trial is considered a *failure*, $\mathbf{y}_{j+1} = \mathbf{y}_j$, and Δ_j is replaced by $\beta \Delta_j$. If $j < n$, replace j by $j+1$, and repeat step 1. Otherwise, if $j = n$, go to step 2.
2. If $f(\mathbf{y}_{n+1}) < f(\mathbf{y}_1)$, that is, if any of the n trials of step 1were successful, let $\mathbf{y}_1 = \mathbf{y}_{n+1}$, set $j = 1$, and repeat step 1. Now consider the case, $f(\mathbf{y}_{n+1}) = f(\mathbf{y}_1)$, that is, when each of the last n trials of step 1 was a failure. If $f(\mathbf{y}_{n+1}) < f(\mathbf{x}_k)$, that is, if at least one successful trial was encountered in step 1 during iteration k, go to step 3. If $f(\mathbf{y}_{n+1}) = f(\mathbf{x}_k)$, that is, if no successful trial is encountered, stop with $\mathbf{x}_k$ as an estimate of the optimal solution if $|\Delta_j| \leq \varepsilon$ for each j; otherwise, let $\mathbf{y}_1 = \mathbf{y}_{n+1}$, let $j = 1$, and go to step 1.
3. Let $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$. If $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$, stop with $\mathbf{x}_{k+1}$ as an estimate of the optimal solution. Otherwise, compute $\lambda_1, \ldots, \lambda_n$ from the relationship $\mathbf{x}_{k+1} - \mathbf{x}_k = \sum_{j=1}^{n} \lambda_j \mathbf{d}_j$, form a new set of search directions by (8.9) and denote these directions by $\mathbf{d}_1, \ldots, \mathbf{d}_n$, let $\Delta_j = \bar{\Delta}_j$ for each j, let $\mathbf{y}_1 = \mathbf{x}_{k+1}$, replace k by $k+1$, let $j = 1$, and repeat step 1.

Note that discrete steps are taken along the n search directions in step 1. If a success occurs along $\mathbf{d}_j$, then Δ_j is replaced by $\alpha \Delta_j$, and if a failure occurs along $\mathbf{d}_j$, then Δ_j is replaced by $\beta \Delta_j$ Since $\beta < 0$, a failure results in reversing the jth search direction during the next pass through step 1. Note that step 1 is repeated until a failure occurs along each of the search directions, in which case a new set of search directions is formed by the Gram-Schmidt procedure.

8.4.6 Example

Consider the following problem:

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

We solve this problem by the method of Rosenbrock using discrete steps with $\bar{\Delta}_1 = \bar{\Delta}_2 = 0.1$, $\alpha = 2.0$, and $\beta = -0.5$. Table 8.10 summarizes the computations starting from (0.00, 3.00), where (S) denotes a success and (F) denotes a failure. Note that within each iteration the directions $\mathbf{d}_1$ and $\mathbf{d}_2$ are fixed. After seven pases through step 1 of Rosenbrock's method, we moved from $\mathbf{x}_1' = (0.00, 3.00)$ to $\mathbf{x}_2' = (3.10, 1.45)$. At this point, a change of directions is

TABLE 8.10 **Summary of Computations for Rosenbrock's Method Using Discrete Steps**

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	(0.00, 3.00)	1	(0.00, 3.00)	0.10	(1.00, 0.00)	(0.10, 3.00)
	52.00		52.00			47.84(S)
		2	(0.10, 3.00)	0.10	(0.00, 1.00)	(0.10, 3.10)
			47.84			50.24(F)
		1	(0.10, 3.00)	0.20	(1.00, 0.00)	(0.30, 3.00)
			47.84			40.84(S)
		2	(0.30, 3.00)	−0.05	(0.00, 1.00)	(0.30, 2.95)
			40.84			39.71(S)
		1	(0.30, 2.95)	0.40	(1.00, 0.00)	(0.70, 2.95)
			39.71			29.90(S)
		2	(0.70, 2.95)	−0.10	(0.00, 1.00)	(0.70, 2.85)
			29.90			27.86(S)
		1	(0.70, 2.85)	0.80	(1.00, 0.00)	(1.50, 2.85)
			27.86			17.70(S)
		2	(1.50, 2.85)	−0.20	(0.00, 1.00)	(1.50, 2.65)
			17.70			14.50(S)
		1	(1.50, 2.65)	1.60	(1.00, 0.00)	(3.10, 2.65)
			14.50			6.30(S)
		2	(3.10, 2.65)	−0.40	(0.00, 1.00)	(3.10, 2.25)
			6.30			3.42(S)
		1	(3.10, 2.25)	3.20	(1.00, 0.00)	(6.30, 2.25)
			3.42			345.12(F)
		2	(3.10, 2.25)	−0.80	(0.00, 1.00)	(3.10, 1.45)
			3.42			1.50(S)
		1	(3.10, 1.45)	−1.60	(1.00, 0.00)	(1.50, 1.45)
			1.50			2.02(F)
		2	(3.10, 1.45)	−1.60	(0.00, 1.00)	(3.10, −0.15)
			1.50			13.02(F)
2	(3.10, 1.45)	1	(3.10, 1.45)	0.10	(0.89, −0.45)	(3.19, 1.41)
	1.50		1.50			2.14(F)
		2	(3.10, 1.45)	0.10	(−0.45, −0.89)	(3.06, 1.36)
			1.50			1.38(S)
		1	(3.06, 1.36)	−0.05	(0.89, −0.45)	(3.02, 1.38)
			1.38			1.15(S)
		2	(3.02, 1.38)	0.20	(−0.45, −0.89)	(2.93, 1.20)
			1.15			1.03(S)

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)	−0.10	(0.89, −0.45)	(2.84, 1.25)
			1.03			0.61(S)
		2	(2.84, 1.25)	0.40	(−0.45, −0.89)	(2.66, 0.89)
			0.61			0.96(F)
		1	(2.84, 1.25)	−0.20	(0.89, −0.45)	(2.66, 1.34)
			0.61			0.19(S)
		2	(2.66, 1.34)	−0.20	(−0.45, −0.89)	(2.75, 1.52)
			0.19			0.40(F)

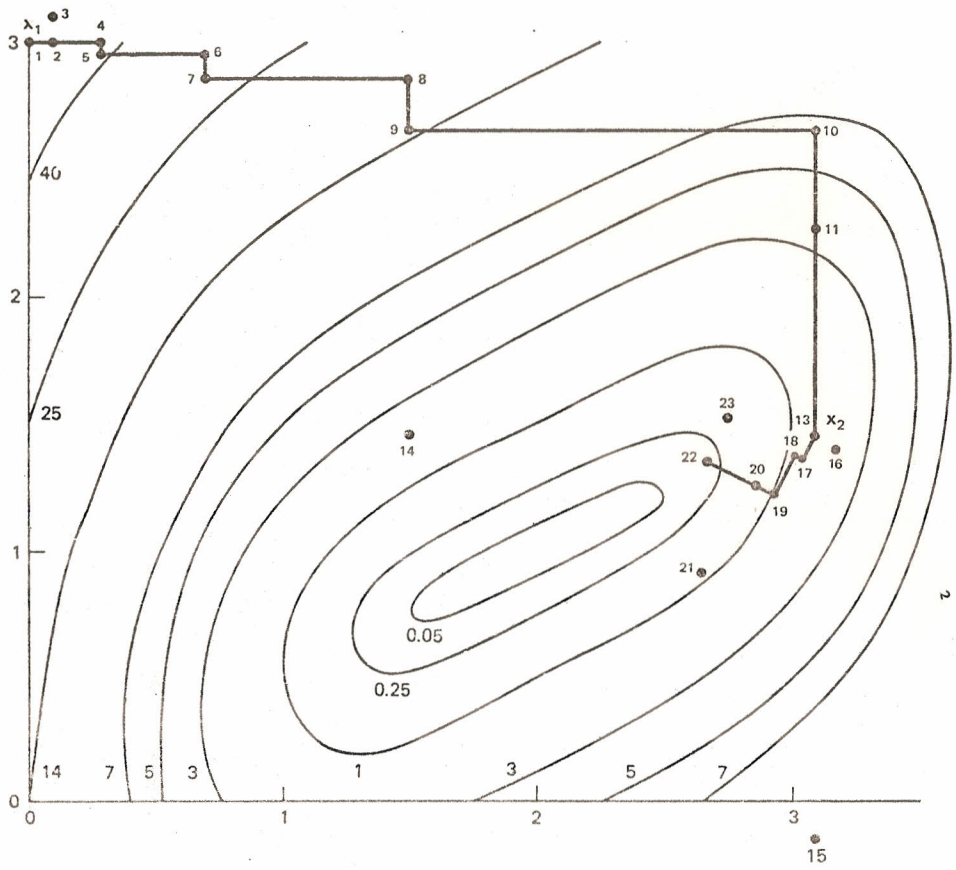

Figure 8.13 Illustration of Rosenbrock's procedure using discrete steps. (The numbers denote the order in which points are generated.)

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 f at $\mathbf{x}$, it follows that

$$f'(\mathbf{x};\mathbf{d})=\lim_{\lambda\to 0^+}\frac{f(\mathbf{x}+\lambda\mathbf{d})-f(\mathbf{x})}{\lambda}=\nabla f(\mathbf{x})'\mathbf{d}$$

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

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

For $\bar{\mathbf{d}} = -\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$, we have $\nabla f(\mathbf{x})' \bar{\mathbf{d}} = -\|\nabla f(\mathbf{x})\|$. 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

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 $\mathbf{x}_8 = (2.28, 1.15)'$ 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 $\nabla f(\mathbf{x})'\mathbf{d} < 0$, 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:

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

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 $\lambda \nabla f(\mathbf{x}_k)'\mathbf{d} = -\lambda \|\nabla f(\mathbf{x}_k)\|^2$ 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.

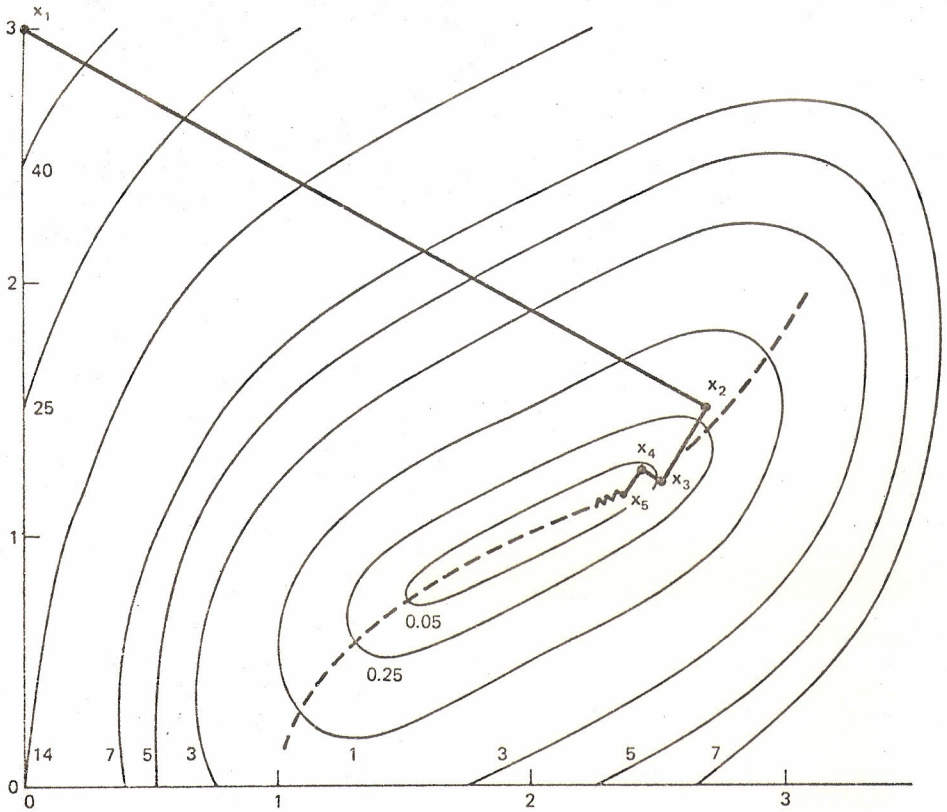

Figure 8.14 Illustration of the method of steepest descent.

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 $\mathbf{x}_k$:

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

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 $\mathbf{x}_7 = (1.83, 0.91)'$ 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 $\nabla f(\mathbf{x})'\mathbf{d} = -\nabla f(\mathbf{x})'\mathbf{B}\nabla f(\mathbf{x}) < 0$ 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 $\mathbf{d}_i'\mathbf{H}\mathbf{d}_j = 0$ 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:

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

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

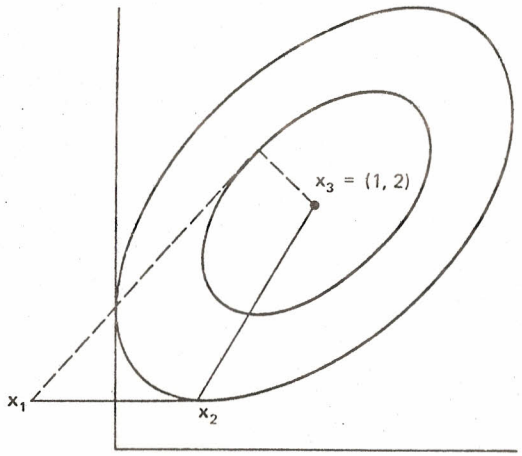

Figure 8.16 Illustration of conjugate directions.

Optimization of Quadratic Functions: Finite Convergence

The above example demonstrates that a quadratic function can be minimized in, at most, n steps, provided that we search along conjugate directions of the Hessian matrix. This result is generally true for quadratic functions, as shown by Theorem 8.6.3 below. This, coupled with the fact that a general function can be closely represented by its quadratic approximation in the vicinity of the optimal point, makes the notion of conjugacy very useful for optimizing both quadratic and nonquadratic functions.

8.6.3 Theorem

Let $f(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is an $n \times n$ symmetric matrix. Let $\mathbf{d}_1, \ldots, \mathbf{d}_n$ be $\mathbf{H}$-conjugate, and let $\mathbf{x}_1$ be an arbitrary starting point. For $k = 1, \ldots, n$, let λ_k be an optimal solution to the problem to minimize $f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ subject to $\lambda \in E_1$, and let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$. Then for $k = 1, \ldots, n$, we must have

1. $\nabla f(\mathbf{x}_{k+1})'\mathbf{d}_j = 0 \quad$ for $j = 1, \ldots, k$
2. $\nabla f(\mathbf{x}_1)'\mathbf{d}_k = \nabla f(\mathbf{x}_k)'\mathbf{d}_k$
3. $\mathbf{x}_{k+1}$ is an optimal solution to the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} - \mathbf{x}_1 \in L(\mathbf{d}_1, \ldots, \mathbf{d}_k)$, where $L(\mathbf{d}_1, \ldots, \mathbf{d}_k)$ is the linear subspace formed by $\mathbf{d}_1, \ldots, \mathbf{d}_k$, that is, $L(\mathbf{d}_1, \ldots, \mathbf{d}_k) = \{\sum_{j=1}^{k} \mu_j \mathbf{d}_j : \mu_j \in E_1 \text{ for each } j\}$. In particular, $\mathbf{x}_{n+1}$ is a minimum point of f over E_n.

Proof

To prove part (1), first note that $f(\mathbf{x}_j + \lambda \mathbf{d}_j)$ achieves a minimum at λ_j only if $\nabla f(\mathbf{x}_j + \lambda_j \mathbf{d}_j)'\mathbf{d}_j = 0$, that is, $\nabla f(\mathbf{x}_{j+1})'\mathbf{d}_j = 0$. Thus, part (1) holds for $j = k$. For

$j < k$, note that

$$\nabla f(\mathbf{x}_{k+1}) = \mathbf{c} + \mathbf{H}\mathbf{x}_{k+1} = \mathbf{c} + \mathbf{H}\mathbf{x}_{j+1} + \mathbf{H}\left(\sum_{i=j+1}^{k} \lambda_i \mathbf{d}_i\right)$$

$$= \nabla f(\mathbf{x}_{j+1}) + \mathbf{H}\left(\sum_{i=j+1}^{k} \lambda_i \mathbf{d}_i\right) \tag{8.13}$$

By conjugacy, $\mathbf{d}_i'\mathbf{H}\mathbf{d}_j = 0$ for $i = j+1, \ldots, k$. Thus, from (8.13) it follows that $\nabla f(\mathbf{x}_{k+1})'\mathbf{d}_j = 0$, and part (1) holds.

Replacing k by $k-1$ and letting $j = 0$ in (8.13), we get

$$\nabla f(\mathbf{x}_k) = \nabla f(\mathbf{x}_1) + \mathbf{H}\left(\sum_{i=1}^{k-1} \lambda_i \mathbf{d}_i\right) \qquad \text{for } k \geq 2$$

Multiplying by $\mathbf{d}_k'$ and noting that $\mathbf{d}_k'\mathbf{H}\mathbf{d}_i = 0$ for $i = 1, \ldots, k-1$ shows that part (2) holds for $k \geq 2$. Part (2) holds true trivially for $k = 1$.

To show part (3), since $\mathbf{d}_i'\mathbf{H}\mathbf{d}_j = 0$ for $i \neq j$, we get

$$f(\mathbf{x}_{k+1}) = f[\mathbf{x}_1 + (\mathbf{x}_{k+1} - \mathbf{x}_1)] = f\left(\mathbf{x}_1 + \sum_{j=1}^{k} \lambda_j \mathbf{d}_j\right)$$

$$= f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1)'\left(\sum_{j=1}^{k} \lambda_j \mathbf{d}_j\right) + \frac{1}{2}\sum_{j=1}^{k} \lambda_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \tag{8.14}$$

Now suppose that $\mathbf{x} - \mathbf{x}_1 \in L(\mathbf{d}_1, \ldots, \mathbf{d}_k)$, so that $\mathbf{x}$ could be written as $\mathbf{x}_1 + \sum_{j=1}^{k} \mu_j \mathbf{d}_j$. As in (8.14), we get

$$f(\mathbf{x}) = f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1)'\left(\sum_{j=1}^{k} \mu_j \mathbf{d}_j\right) + \frac{1}{2}\sum_{j=1}^{k} \mu_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \tag{8.15}$$

To complete the proof, we need to show that $f(\mathbf{x}) \geq f(\mathbf{x}_{k+1})$. By contradiction, suppose that $f(\mathbf{x}) < f(\mathbf{x}_{k+1})$. Then by (8.14) and (8.15), we must have

$$\nabla f(\mathbf{x}_1)'\left(\sum_{j=1}^{k} \mu_j \mathbf{d}_j\right) + \frac{1}{2}\sum_{j=1}^{k} \mu_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j < \nabla f(\mathbf{x}_1)'\left(\sum_{j=1}^{k} \lambda_j \mathbf{d}_j\right) + \frac{1}{2}\sum_{j=1}^{k} \lambda_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \tag{8.16}$$

By definition of λ_j, note that $f(\mathbf{x}_j + \lambda_j \mathbf{d}_j) \leq f(\mathbf{x}_j + \mu_j \mathbf{d}_j)$ for each j. Therefore

$$f(\mathbf{x}_j) + \lambda_j \nabla f(\mathbf{x}_j)'\mathbf{d}_j + \tfrac{1}{2}\lambda_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \leq f(\mathbf{x}_j) + \mu_j \nabla f(\mathbf{x}_j)'\mathbf{d}_j + \tfrac{1}{2}\mu_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j$$

By part 2, $\nabla f(\mathbf{x}_j)'\mathbf{d}_j = \nabla f(\mathbf{x}_1)'\mathbf{d}_j$ and substituting in the above inequality, we get

$$\lambda_j \nabla f(\mathbf{x}_1)'\mathbf{d}_j + \tfrac{1}{2}\lambda_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \leq \mu_j \nabla f(\mathbf{x}_1)'\mathbf{d}_j + \tfrac{1}{2}\mu_j^2 \mathbf{d}_j'\mathbf{H}\mathbf{d}_j \tag{8.17}$$

Summing (8.17) for $j = 1, \ldots, k$ contradicts (8.16). Thus $\mathbf{x}_{k+1}$ is a minimum point over the manifold $\mathbf{x}_1 + L(\mathbf{d}_1, \ldots, \mathbf{d}_k)$. In particular, since $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are