6.8 Prove that the product of two lower triangular matrices is lower triangular and that the inverse of a nonsingular lower triangular matrix is lower triangular.

6.9 Prove that the product **AB** of two square matrices is nonsingular if and only if both **A** and **B** are nonsingular.

6.10 Let $\mathbf{L}_n\mathbf{P}_n \cdots \mathbf{L}_1\mathbf{P}_1\mathbf{A} = \mathbf{U}$ be a triangular factorization of **A**; let matrices $\mathbf{Q}_n, \mathbf{Q}_{n-1}, \ldots, \mathbf{Q}_0$ be defined by $\mathbf{Q}_n = \mathbf{I}$ and $\mathbf{Q}_{k-1} = \mathbf{Q}_k\mathbf{P}_k$ $(k = n, n-1, \ldots, 1)$ and let $\mathbf{L}_k^*$ stand for the matrix $\mathbf{Q}_k\mathbf{L}_k\mathbf{Q}_k^T$. Prove that:

(i) Each $\mathbf{Q}_k$ is a permutation matrix agreeing with **I** in the first k rows and columns.

(ii) $\mathbf{L}_n\mathbf{P}_n \cdots \mathbf{L}_1\mathbf{P}_1 = \mathbf{L}_n^* \cdots \mathbf{L}_1^*\mathbf{Q}_0$.

(iii) Each $\mathbf{L}_k^*$ is a lower triangular eta matrix whose eta column is the kth column.

6.11 An *LU-decomposition* of a matrix **A** consists of a lower triangular matrix **L**, an upper triangular matrix **U**, and a permutation matrix **P** such that **PA** = **LU**. Use the results of problems 6.8, 6.9, and 6.10 to prove that every nonsingular matrix has an LU-decomposition.

6.12 At the end of this chapter, the claim was made that "even very sparse matrices tend to have dense inverses." Artificially constructed examples do not support empirical claims such as this one very convincingly, but here is one anyway:

$$\text{if}\quad \mathbf{A} = \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \end{bmatrix} \quad \text{then}\quad \mathbf{A}^{-1} = \begin{bmatrix} 0.5 & 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 & 0.5 \end{bmatrix}.$$

Compute a triangular factorization of **A** and compare its density with the density of $\mathbf{A}^{-1}$. Generalize this example to arbitrary sizes $n \times n$ such that n is odd. (Why is there no generalization with n even?)

7

The Revised Simplex Method

In each iteration of the simplex method, one basic feasible solution is replaced by another. When the old solution is represented by a dictionary, the new solution is easy to find, and only a small part of the dictionary is actually used for that purpose. This part may be reconstructed directly from the original data and the new solution found without any reference to dictionaries. The resulting implementations of the simplex method are known under the generic name of the *revised simplex method*; the implementation of the simplex method that updates a dictionary in each iteration is known as the *standard simplex method*. We shall see that each iteration of the revised simplex method requires solving two systems of linear equations. Typically, these systems are not solved from scratch; instead, some device facilitating their solution is used and updated in each iteration. The device presented in this chapter is essentially the same as the popular "product form of the inverse" developed by G. B. Dantzig and W. Orchard-Hays (1954).

Each iteration of the revised simplex method may or may not take less time than the corresponding iteration of the standard simplex method. The outcome of this comparison depends not only on the particular implementation of the revised simplex method but also on the nature of the data. We shall see that, on the typical large and sparse LP problems solved in applications, the revised simplex method works faster than the standard simplex method. This is the reason why modern computer programs for solving LP problems always use some form of the revised simplex method.

MATRIX DESCRIPTION OF DICTIONARIES

Our preliminary task is to develop an understanding of the relationship between dictionaries and the original data. For illustration, we shall consider the dictionary

$$\begin{array}{rcl} x_1 &=& 54 - 0.5x_2 - 0.5x_4 - 0.5x_5 + 0.5x_6 \\ x_3 &=& 63 - 0.5x_2 - 0.5x_4 + 0.5x_5 - 1.5x_6 \\ x_7 &=& 15 + 0.5x_2 - 0.5x_4 + 0.5x_5 + 2.5x_6 \\ \hline z &=& 1782 - 2.5x_2 + 1.5x_4 - 3.5x_5 - 8.5x_6 \end{array} \tag{7.1}$$

arising from the problem

$$\begin{array}{ll} \text{maximize} & 19x_1 + 13x_2 + 12x_3 + 17x_4 \\ \text{subject to} & 3x_1 + 2x_2 + x_3 + 2x_4 \le 225 \\ & x_1 + x_2 + x_3 + x_4 \le 117 \\ & 4x_1 + 3x_2 + 3x_3 + 4x_4 \le 420 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{array} \tag{7.2}$$

after two iterations of the standard simplex method. To relate the coefficients in (7.1) to the data in (7.2), we first recall that the top three equations in the dictionary are equivalent to the three equations

$$\begin{array}{l} 3x_1 + 2x_2 + x_3 + 2x_4 + x_5 = 225 \\ x_1 + x_2 + x_3 + x_4 + x_6 = 117 \\ 4x_1 + 3x_2 + 3x_3 + 4x_4 + x_7 = 420. \end{array} \tag{7.3}$$

Hence they arise by solving (7.3) for x_1, x_3, and x_7. In matrix terms, this solution may be described quite compactly. First, we record system (7.3) as $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 4 & 3 & 3 & 4 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 225 \\ 117 \\ 420 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}.$$

To emphasize the fact that only the basic variables x_1, x_3, x_7 are treated as unknowns, we write $\mathbf{Ax}$ as $\mathbf{A}_B\mathbf{x}_B + \mathbf{A}_N\mathbf{x}_N$ with

$$\mathbf{A}_B = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}, \quad \mathbf{A}_N = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 3 & 4 & 0 & 0 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_1 \\ x_3 \\ x_7 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} x_2 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

and then cast the system $\mathbf{Ax} = \mathbf{b}$ in the form

$$\mathbf{A}_B\mathbf{x}_B = \mathbf{b} - \mathbf{A}_N\mathbf{x}_N. \tag{7.4}$$

Since the square matrix $\mathbf{A}_B$ happens to be nonsingular, both sides of (7.4) may be multiplied by $\mathbf{A}_B^{-1}$ on the left. Thus we obtain

$$\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N \tag{7.5}$$

which is a compact record of the top three equations in (7.1). To obtain the fourth equation, we record the objective function z as $\mathbf{cx}$ with

$$\mathbf{c} = [19, 13, 12, 17, 0, 0, 0]$$

or, more suggestively, as $\mathbf{c}_B\mathbf{x}_B + \mathbf{c}_N\mathbf{x}_N$, with

$$\mathbf{c}_B = [19, 12, 0] \quad \text{and} \quad \mathbf{c}_N = [13, 17, 0, 0].$$

Substituting for $\mathbf{x}_B$ from (7.5) we obtain

$$z = \mathbf{c}_B(\mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N) + \mathbf{c}_N\mathbf{x}_N = \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{A}_N)\mathbf{x}_N.$$

Thus, dictionary (7.1) may be recorded in matrix terms as

$$\begin{array}{rcl} \mathbf{x}_B &=& \mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N \\ \hline z &=& \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{A}_N)\mathbf{x}_N \end{array}. \tag{7.6}$$

More generally, consider an arbitrary LP problem in the standard form

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} c_jx_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij}x_j \le b_i & (i = 1, 2, \dots, m) \\ & x_j \ge 0 & (j = 1, 2, \dots, n). \end{array}$$

After the introduction of the slack variables $x_{n+1}, x_{n+2}, \dots, x_{n+m}$, this problem may be recorded as

$$\begin{array}{ll} \text{maximize} & \mathbf{cx} \\ \text{subject to} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0}. \end{array}$$

(The matrix $\mathbf{A}$ has m rows and $n + m$ columns, of which the last m form the identity matrix. The column vector $\mathbf{x}$ has length $n + m$ and the column vector $\mathbf{b}$ has length m. The row vector $\mathbf{c}$ has length $n + m$ and its last m components are zeros.) Each basic feasible solution $\mathbf{x}^*$ of this problem partitions $x_1, x_2, \ldots, x_{n+m}$ into m basic and n nonbasic variables. As in our example, this partition induces a partition of $\mathbf{A}$ into $\mathbf{A}_B$ and $\mathbf{A}_N$, a partition of $\mathbf{x}$ into $\mathbf{x}_B$ and $\mathbf{x}_N$, and a partition of $\mathbf{c}$ into $\mathbf{c}_B$ and $\mathbf{c}_N$. We propose to show that

$$\text{matrix } \mathbf{A}_B \text{ is nonsingular} \tag{7.7}$$

by showing that the system $\mathbf{A}_B\mathbf{x}_B = \mathbf{b}$ has precisely one solution. The existence of a solution is evident: since the basic feasible solution $\mathbf{x}^*$ satisfies $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ and $\mathbf{x}_N^* = \mathbf{0}$, it satisfies $\mathbf{A}_B\mathbf{x}_B^* = \mathbf{A}\mathbf{x}^* - \mathbf{A}_N\mathbf{x}_N^* = \mathbf{b}$. To verify that there are no other solutions, consider an arbitrary vector $\tilde{\mathbf{x}}_B$ such that $\mathbf{A}_B\tilde{\mathbf{x}}_B = \mathbf{b}$ and set $\tilde{\mathbf{x}}_N = \mathbf{0}$. Since the resulting vector $\tilde{\mathbf{x}}$ satisfies $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}_B\tilde{\mathbf{x}}_B + \mathbf{A}_N\tilde{\mathbf{x}}_N = \mathbf{b}$, it must satisfy the top m equations in the dictionary representing $\mathbf{x}^*$. But then $\tilde{\mathbf{x}}_N = \mathbf{0}$ implies $\tilde{\mathbf{x}}_B = \mathbf{x}_B^*$. Thus the proof of (7.7) is completed. Now the arguments given above show that the dictionary representing $\mathbf{x}^*$ has the form (7.6). The matrix $\mathbf{A}_B$ is called the *basis matrix* or (when there is no danger of confusion with the set of basic variables) simply the *basis*. It is customary to denote the basis matrix by $\mathbf{B}$ rather than $\mathbf{A}_B$. We shall bow to this convention and record the dictionary as

$$\begin{aligned} \mathbf{x}_B &= \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}_N\mathbf{x}_N \\ \hline z &= \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A}_N)\mathbf{x}_N. \end{aligned}$$

Of course, $\mathbf{B}^{-1}\mathbf{b}$ is nothing but the vector $\mathbf{x}_B^*$ specifying the current values of the basic variables.

THE REVISED SIMPLEX METHOD

In each iteration of the simplex method, we first choose the entering variable, then find the leaving variable, and finally update the current basic feasible solution. An examination of the way these tasks are carried out in the standard simplex method will lead us to the alternative, the revised simplex method. For illustration, we shall consider the update of the feasible dictionary (7.1) in the standard simplex method. The corresponding iteration of the revised simplex method begins with

$$\mathbf{x}_B^* = \begin{bmatrix} x_1^* \\ x_3^* \\ x_7^* \end{bmatrix} = \begin{bmatrix} 54 \\ 63 \\ 15 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}.$$

The entering variable may be any nonbasic variable with a positive coefficient in the last row of the dictionary. As previously observed, the coefficients in this row form the vector $\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A}_N$. If the standard simplex method is used, then this vector is readily available as part of the dictionary; in our example, we have

$$z = \cdots - 2.5x_2 + 1.5x_4 - 3.5x_5 - 8.5x_6. \tag{7.8}$$

If the revised simplex method is used, then the vector $\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A}_N$ is computed in two steps: first we find $\mathbf{y} = \mathbf{c}_B\mathbf{B}^{-1}$ by solving the system $\mathbf{y}\mathbf{B} = \mathbf{c}_B$ and then we calculate $\mathbf{c}_N - \mathbf{y}\mathbf{A}_N$. In our example, we would first solve the system

$$[y_1, y_2, y_3] \cdot \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix} = [19, 12, 0]$$

to find $\mathbf{y} = [y_1, y_2, y_3] = [3.5, 8.5, 0]$ and then we would calculate

$$[13, 17, 0, 0] - [3.5, 8.5, 0] \cdot \begin{bmatrix} 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 3 & 4 & 0 & 0 \end{bmatrix} = [-2.5, 1.5, -3.5, -8.5]$$

to find the vector featured in (7.8). As the only positive component of this vector is its second component, the second component x_4 of the vector $\mathbf{x}_N = [x_2, x_4, x_5, x_6]^T$ enters the basis. Incidentally, note that the components of $\mathbf{c}_N - \mathbf{y}\mathbf{A}_N$ may be calculated individually; if a nonbasic variable x_j corresponds to a component c_j of $\mathbf{c}_N$ and to a column $\mathbf{a}$ of $\mathbf{A}_N$, then the corresponding component of $\mathbf{c}_N - \mathbf{y}\mathbf{A}_N$ equals $c_j - \mathbf{y}\mathbf{a}$. Thus, the entering variable may be any nonbasic variable x_j for which $\mathbf{y}\mathbf{a} < c_j$. The corresponding column $\mathbf{a}$ of $\mathbf{A}$ is called the *entering column*.

To determine the leaving variable, we increase the value t of the entering variable from zero to some positive level, maintaining the values of the remaining nonbasic variables at their zero levels and adjusting the values of the basic variables so as to preserve the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$. As t increases, the values of the basic variables change until a variable whose value is the first to drop to zero leaves the basis. To find the leaving variable and the largest admissible value of t, we have to know how precisely the values of the basic variables change with the changes of t. If the standard simplex method is used, then this information is readily available as part of the dictionary; in our example, we have

$$\begin{aligned} x_1 &= 54 \cdots - 0.5x_4 \cdots & & & x_1 &= 54 - 0.5t \\ x_3 &= 63 \cdots - 0.5x_4 \cdots, & &\text{and so} & x_3 &= 63 - 0.5t. \\ x_7 &= 15 \cdots - 0.5x_4 \cdots & & & x_7 &= 15 - 0.5t \end{aligned} \tag{7.9}$$

More generally, the top m equations of the dictionary read $\mathbf{x}_B = \mathbf{x}_B^* - \mathbf{B}^{-1}\mathbf{A}_N\mathbf{x}_N$, and so $\mathbf{x}_B$ changes from $\mathbf{x}_B^*$ to $\mathbf{x}_B^* - t\mathbf{d}$, with $\mathbf{d}$ standing for the column of $\mathbf{B}^{-1}\mathbf{A}_N$ that corresponds to the entering variable. Note that $\mathbf{d} = \mathbf{B}^{-1}\mathbf{a}$, with $\mathbf{a}$ standing for the entering column. If the revised simplex method is used, then only $\mathbf{x}_B^*$ is readily available, whereas $\mathbf{d}$ is obtained by solving the system $\mathbf{Bd} = \mathbf{a}$. In our example, we would solve the system

$$\begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix} \cdot \mathbf{d} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} \quad \text{to find the vector} \quad \mathbf{d} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

featured in (7.9). We find easily that t can be increased all the way to 30, at which point $54 - 0.5t = 39$, $63 - 0.5t = 48$, $15 - 0.5t = 0$, and x_7 leaves the basis.

So far, the revised simplex method has been requiring computations not needed in the standard simplex method. This trend gets reversed at the end of the iteration: whereas the standard simplex method requires a laborious update of the entire dictionary, no such computations are needed in the revised simplex method. In our example, the revised simplex method merely enters the next iteration with

$$\mathbf{x}_B^* = \begin{bmatrix} x_1^* \\ x_3^* \\ x_4^* \end{bmatrix} = \begin{bmatrix} 39 \\ 48 \\ 30 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 1 \\ 4 & 3 & 4 \end{bmatrix}.$$

Incidentally, note that the order of the columns of $\mathbf{B}$ is unimportant as long as it matches the order of the components of $\mathbf{x}_B^*$: the next iteration could just as well be entered with

$$\mathbf{x}_B^* = \begin{bmatrix} x_3^* \\ x_4^* \\ x_1^* \end{bmatrix} = \begin{bmatrix} 48 \\ 30 \\ 39 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 3 & 4 & 4 \end{bmatrix}.$$

To put it differently, the fact that the variables $x_1, x_2, \ldots, x_{n+m}$ happen to be ordered by their subscripts is just coincidental; the columns of $\mathbf{B}$ may be presented in any other order. An ordered list of the basic variables that specifies the actual order of the m columns of $\mathbf{B}$ is called the *basis heading*. We shall find it convenient to replace the leaving variable by the entering variable in each update of the basis heading: the corresponding update of $\mathbf{B}$ amounts to a replacement of the *leaving column* by the entering column.

Our development of the revised simplex method is summarized in Box 7.1.

An Economic Interpretation of the Revised Simplex Method

The revised simplex method is intimately related to two subjects presented in Chapter 5: the Complementary Slackness Theorem and the economic interpretation of dual variables. To illustrate the relationship, we shall consider a hypothetical furniture-manufacturing company.

- A bookcase requires three hours of work, one unit of metal, and four units of wood, and it brings in a net profit of $19.

simplex multipliers

BOX 7.1 An Iteration of the Revised Simplex Method

Step 1. Solve the system $\mathbf{yB} = \mathbf{c}_B$.

Step 2. Choose an entering column. This may be any column $\mathbf{a}$ of $\mathbf{A}_N$ such that $\mathbf{ya}$ is less than the corresponding component of $\mathbf{c}_N$. If there is no such column, then the current solution is optimal.

Step 3. Solve the system $\mathbf{Bd} = \mathbf{a}$.

Step 4. Find the largest t such that $\mathbf{x}_B^* - t\mathbf{d} \geq \mathbf{0}$. If there is no such t, then the problem is unbounded; otherwise, at least one component of $\mathbf{x}_B^* - t\mathbf{d}$ equals zero and the corresponding variable is leaving the basis.

Step 5. Set the value of the entering variable at t and replace the values $\mathbf{x}_B^*$ of the basic variables by $\mathbf{x}_B^* - t\mathbf{d}$. Replace the leaving column of $\mathbf{B}$ by the entering column and, in the basis heading, replace the leaving variable by the entering variable.

- A desk requires two hours of work, one unit of metal, and three units of wood, and it brings in a net profit of $13.
- A chair requires one hour of work, one unit of metal, and three units of wood, and it brings in a net profit of $12.
- A bedframe requires two hours of work, one unit of metal, and four units of wood, and it brings in a net profit of $17.
- Only 225 hours of labor, 117 units of metal, and 420 units of wood are available per day.

Note that the problem of maximizing the total net profit of the company, under the assumption that all the furniture can be sold, is nothing but our old example (7.2).

Now suppose that a program of making 54 bookcases and 63 chairs per day has been proposed to the company. To find out if this program is optimal, we may appeal to the Complementary Slackness Theorem (the version presented as Theorem 5.3): a feasible solution $\mathbf{x}^*$ is optimal if and only if there are numbers $y_1, y_2, \ldots, y_m$ that satisfy a certain system of equations and a certain system of inequalities. In this particular example, $\mathbf{x}^*$ is the basic feasible solution with

$$\mathbf{x}_B^* = \begin{bmatrix} x_1^* \\ x_3^* \\ x_7^* \end{bmatrix} = \begin{bmatrix} 54 \\ 63 \\ 15 \end{bmatrix}$$

the system of equations is

$$\begin{aligned} 3y_1 + y_2 + 4y_3 &= 19 \\ y_1 + y_2 + 3y_3 &= 12 \\ y_3 &= 0 \end{aligned} \tag{7.10}$$

and the system of inequalities is

$$\begin{aligned} 2y_1 + y_2 + 3y_3 &\geq 13 \\ 2y_1 + y_2 + 4y_3 &\geq 17 \\ y_1 &\geq 0 \\ y_2 &\geq 0. \end{aligned} \tag{7.11}$$

Note that (7.10) is nothing but $\mathbf{yB} = \mathbf{c}_B$, that (7.11) is nothing but $\mathbf{yA}_N \geq \mathbf{c}_N$, and that this observation generalizes: if $\mathbf{x}^*$ is a nondegenerate basic feasible solution, then the system of equations featured in the Complementary Slackness Theorem is nothing but $\mathbf{yB} = \mathbf{c}_B$ and the system of inequalities is nothing but $\mathbf{yA}_N \geq \mathbf{c}_N$. But $\mathbf{yB} = \mathbf{c}_B$ is precisely the system of equations solved in step 1 of an iteration of the revised simplex method and $\mathbf{yA}_N \geq \mathbf{c}_N$ is the system of inequalities considered in step 2. Thus the first two steps in each iteration of the revised simplex method may be seen as checking the current feasible solution $\mathbf{x}^*$ for optimality by the Complementary Slackness Theorem. (Actually, this statement is not quite correct: if $\mathbf{x}^*$ happens to be degenerate, then the system of equations featured in the Complementary Slackness Theorem consists of fewer than m equations and forms a proper subsystem of $\mathbf{yB} = \mathbf{c}_B$.)

Furthermore, these first two steps may be given an economic interpretation along the lines described in Chapter 5. Solving system (7.10), or $\mathbf{yB} = \mathbf{c}_B$ in general, may be interpreted as assigning temporary *shadow prices* to the *resources* (time, metal, and wood) in such a way that the total shadow price of the resources consumed by each of the three *basic activities* (making bookcases, making chairs, and leaving wood unused) matches the net profit returned by this activity. Thus, the solution $y_1 = 3.5$, $y_2 = 8.5$, $y_3 = 0$ of (7.10) appraises time at \$3.50/hour, metal at \$8.50/unit, and wood at \$0/unit. Evaluating the left-hand side of (7.11), or $\mathbf{yA}_N$ in general, may be interpreted as finding the total shadow price of the resources consumed by each of the *nonbasic activities* (making desks, making bedframes, leaving working time unused, and leaving metal unused); this operation is sometimes referred to as *pricing out* the nonbasic activities. If none of these activities pays back more than it consumes (that is, if $\mathbf{c}_N \leq \mathbf{yA}_N$), then the current program is optimal. (A converse of this implication, although guaranteed by the Complementary Slackness Theorem whenever $\mathbf{x}^*$ is nondegenerate, does not hold in general: a degenerate $\mathbf{x}^*$ may be optimal even if some of the inequalities in $\mathbf{c}_N \leq \mathbf{yA}_N$ are violated. In that case, the simplex method will go through a few degenerate iterations without changing $\mathbf{x}^*$ until it comes up with a basis that yields a vector $\mathbf{y}$ with $\mathbf{yA}_N \geq \mathbf{c}_N$.) In our example, making bedframes does pay back more (\$17) than it consumes (time worth \$7, metal worth \$8.50, and wood worth nothing under the current pricing scheme).

Where the Complementary Slackness Theorem leaves off, the revised simplex method continues: it attempts to construct an improved program by *substituting* the profitable *entering activity* (making bedframes) for a suitable mix of the basic activities. The mix, d_i units of each basic activity i per unit of the entering activity, must consume resources at the same rate as the entering activity itself. In our example, this requirement gives rise to the system

$$\begin{aligned} 3d_1 + d_3 \quad &= 2 \\ d_1 + d_3 \quad &= 1 \\ 4d_1 + 3d_3 + d_7 &= 4 \end{aligned} \tag{7.12}$$

and the solution of this system, $d_1 = 0.5$, $d_3 = 0.5$, $d_7 = 0.5$, specifies the concentrations d_i of the constituents i in the mix: each bedframe will be substituted for half a bookcase plus half a chair plus half a unit of unused wood. [Of course, (7.12) is nothing but the system $\mathbf{Bd} = \mathbf{a}$ solved in step 3 of the iteration.] Since the substitution raises the company's profit (by \$1.50 per bedframe), the largest admissible number t of chairs should be substituted for $0.5t$ bookcases plus $0.5t$ chairs plus $0.5t$ units of unused wood; since only 15 units of unused wood are available, the value of t is limited to 30. (This is step 4 of the iteration, where the largest admissible value of t is determined.) The resulting improved program calls for 39 bookcases, 48 chairs, and 30 bedframes to be made every day. The three new basic activities are making bookcases, making chairs, and making bedframes; the old basic activity of leaving wood unused has just become nonbasic. (This is step 5, where the substitution is actually carried out.)

Along similar lines, each iteration of the revised simplex method may be interpreted in economic terms of pricing and substitution. The interpretation becomes a little less intuitive when some of the numbers $y_1, y_2, \ldots, y_m$ or $d_1, d_2, \ldots, d_m$ come out negative, but it may be justified even in those cases.

Eta Factorization of the Basis

The efficiency of the revised simplex method hinges on the ease of implementing steps 1 and 3 of each iteration. Typically, the systems $\mathbf{yB} = \mathbf{c}_B$ and $\mathbf{Bd} = \mathbf{a}$ are not solved from scratch; instead, some device is used to facilitate their solutions and is updated at the end of each iteration. Thus our description of the revised simplex method encompasses a whole class of implementations, each depending on the choice of device that facilitates solutions of the two systems. We are about to describe the simplest of these devices, almost the same (see problem 7.13) as the popular "product form of the inverse" developed by G. B. Dantzig and W. Orchard-Hays (1954). A class of devices that are more efficient, but also more complicated, will be presented in Chapter 24.

Let $\mathbf{B}_k$ denote the basis matrix obtained after k iterations of the simplex method, so that each $\mathbf{B}_k$ differs from the preceding $\mathbf{B}_{k-1}$ in only one column. Consider a fixed

k and say that it is the pth column in which $\mathbf{B}_k$ differs from $\mathbf{B}_{k-1}$. Now the pth column of $\mathbf{B}_k$ is the entering column $\mathbf{a}$ selected in step 2 of the kth iteration and appearing as the right-hand side in the system $\mathbf{B}_{k-1}\mathbf{d} = \mathbf{a}$, which is solved in step 3 of the same iteration. Hence

$$\mathbf{B}_k = \mathbf{B}_{k-1}\mathbf{E}_k \tag{7.13}$$

with $\mathbf{E}_k$ standing for the identity matrix whose pth column is replaced by $\mathbf{d}$. (To verify this matrix equation, we need only compare its two sides column by column, keeping in mind that the jth column of $\mathbf{B}_{k-1}\mathbf{E}_k$ equals $\mathbf{B}_{k-1}$ multiplied by the jth column of $\mathbf{E}_k$ on the right.) For instance,

$$\begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 1 \\ 4 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & 0.5 \\ & 1 & 0.5 \\ & & 0.5 \end{bmatrix}$$

in the example just used. The importance of equation (7.13) for the revised simplex method is paramount: no matter what device is used to solve the two systems $\mathbf{y}\mathbf{B}_{k-1} = \mathbf{c}_B$ and $\mathbf{B}_{k-1}\mathbf{d} = \mathbf{a}$, its update invariably relies on the fact that $\mathbf{B}_k = \mathbf{B}_{k-1}\mathbf{E}_k$ with the eta matrix $\mathbf{E}_k$ readily available.

When the initial basis consists of the slack variables, we have $\mathbf{B}_0 = \mathbf{I}$ and successive applications of (7.13) yield $\mathbf{B}_1 = \mathbf{E}_1$, $\mathbf{B}_2 = \mathbf{E}_1\mathbf{E}_2$, $\mathbf{B}_3 = \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3$, and so on. Thus we have

$$\mathbf{B}_k = \mathbf{E}_1\mathbf{E}_2 \cdots \mathbf{E}_k.$$

This *eta factorization* of $\mathbf{B}_k$ suggests a convenient way of solving the two systems of equations: the system $\mathbf{y}\mathbf{B}_k = \mathbf{c}_B$ may be seen as

$$(((\mathbf{y}\mathbf{E}_1)\mathbf{E}_2)\cdots)\mathbf{E}_k = \mathbf{c}_B$$

and the system $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ may be seen as

$$\mathbf{E}_1(\mathbf{E}_2(\cdots(\mathbf{E}_k\mathbf{d}))) = \mathbf{a}.$$

For instance, $\mathbf{y}\mathbf{B}_4 = \mathbf{c}_B$ may be solved by solving the sequence of systems

$$\mathbf{u}\mathbf{E}_4 = \mathbf{c}_B, \quad \mathbf{v}\mathbf{E}_3 = \mathbf{u}, \quad \mathbf{w}\mathbf{E}_2 = \mathbf{v}, \quad \text{and} \quad \mathbf{y}\mathbf{E}_1 = \mathbf{w}$$

(so that $\mathbf{y}\mathbf{B}_4 = \mathbf{y}\mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4 = \mathbf{w}\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4 = \mathbf{v}\mathbf{E}_3\mathbf{E}_4 = \mathbf{u}\mathbf{E}_4 = \mathbf{c}_B$ as desired) and $\mathbf{B}_4\mathbf{d} = \mathbf{a}$ may be solved by solving the sequence of systems

$$\mathbf{E}_1\mathbf{u} = \mathbf{a}, \quad \mathbf{E}_2\mathbf{v} = \mathbf{u}, \quad \mathbf{E}_3\mathbf{w} = \mathbf{v}, \quad \text{and} \quad \mathbf{E}_4\mathbf{d} = \mathbf{w}$$

(so that $\mathbf{B}_4\mathbf{d} = \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4\mathbf{d} = \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{w} = \mathbf{E}_1\mathbf{E}_2\mathbf{v} = \mathbf{E}_1\mathbf{u} = \mathbf{a}$ as desired). At first, this way of solving $\mathbf{y}\mathbf{B}_k = \mathbf{c}_B$ and $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ may seem rather awkward: in order to solve one system of linear equations, we resort to solving k systems. Note, however, that systems such as $\mathbf{v}\mathbf{E}_i = \mathbf{u}$ or $\mathbf{E}_i\mathbf{v} = \mathbf{u}$ are extremely easy to solve: if the eta column of $\mathbf{E}_i$ has s nonzero entries, then only $s - 1$ multiplications, $s - 1$ additions, and one division are required. Before discussing the efficiency of this scheme any further, let us illustrate it with an example.

The example is again problem (7.2),

$$\text{maximize} \quad \mathbf{c}\mathbf{x} \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}$$

with

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 4 & 3 & 3 & 4 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 225 \\ 117 \\ 420 \end{bmatrix}, \quad \mathbf{c} = [19, 13, 12, 17, 0, 0, 0].$$

As usual, we let the slack variables form the initial basis, so that $\mathbf{B}_0 = \mathbf{I}$ and

$$\mathbf{x}_B^* = \begin{bmatrix} x_5^* \\ x_6^* \\ x_7^* \end{bmatrix} = \begin{bmatrix} 225 \\ 117 \\ 420 \end{bmatrix}.$$

The first iteration of the revised simplex method begins.

Step 1. The system $\mathbf{y}\mathbf{B}_0 = \mathbf{c}_B$ reduces to $\mathbf{y} = [0, 0, 0]$.

Step 2. Since $c_3 - \mathbf{y}\begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} = 12$, we may let x_3 enter the basis.

Step 3. The system $\mathbf{B}_0\mathbf{d} = \mathbf{a}$ reduces to

$$\mathbf{d} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}.$$

Step 4. The largest t such that $225 - t \geq 0$, $117 - t \geq 0$, $420 - 3t \geq 0$ is $t = 117$. Since $117 - t = 0$, the leaving variable is x_6.

Step 5. Now we have

$$\begin{bmatrix} x_5^* \\ x_3^* \\ x_7^* \end{bmatrix} = \begin{bmatrix} 225 - t \\ t \\ 420 - 3t \end{bmatrix} = \begin{bmatrix} 108 \\ 117 \\ 69 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \mathbf{E}_1 = \begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix}.$$

The second iteration begins.

Step 1. Solving the system $\mathbf{y}\mathbf{B}_1 = \mathbf{c}_B$, which reads

$$\mathbf{y}\begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix} = [0, 12, 0], \quad \text{we find } \mathbf{y} = [0, 12, 0].$$

Step 2. Since $c_1 - \mathbf{y}\begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix} = 7$, we may let x_1 enter the basis.

Step 3. Solving the system $\mathbf{B}_1\mathbf{d} = \mathbf{a}$, which reads

$$\begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix} \cdot \mathbf{d} = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}, \quad \text{we find} \quad \mathbf{d} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}.$$

Step 4. The largest t such that $108 - 2t \geq 0$, $117 - t \geq 0$, $69 - t \geq 0$ is $t = 54$. Since $108 - 2t = 0$, the leaving variable is x_5.

Step 5. Now we have

$$\begin{bmatrix} x_1^* \\ x_3^* \\ x_7^* \end{bmatrix} = \begin{bmatrix} t \\ 117 - t \\ 69 - t \end{bmatrix} = \begin{bmatrix} 54 \\ 63 \\ 15 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_2 = \mathbf{E}_1\mathbf{E}_2 \quad \text{with} \quad \mathbf{E}_2 = \begin{bmatrix} 2 & & \\ 1 & 1 & \\ 1 & & 1 \end{bmatrix}.$$

The third iteration begins.

Step 1. We shall solve the system $\mathbf{y}\mathbf{B}_2 = \mathbf{c}_B$ as $(\mathbf{y}\mathbf{E}_1)\mathbf{E}_2 = \mathbf{c}_B$. Solving the system $\mathbf{u}\mathbf{E}_2 = \mathbf{c}_B$, which reads

$$\mathbf{u}\begin{bmatrix} 2 & & \\ 1 & 1 & \\ 1 & & 1 \end{bmatrix} = [19, 12, 0], \quad \text{we find} \quad \mathbf{u} = [3.5, 12, 0].$$

Solving the system $\mathbf{y}\mathbf{E}_1 = \mathbf{u}$, which reads

$$\mathbf{y}\begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix} = [3.5, 12, 0], \quad \text{we find} \quad \mathbf{y} = [3.5, 8.5, 0].$$

Step 2. Since $c_4 - \mathbf{y}\begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} = 1.5$, we may let x_4 enter the basis.

Step 3. We shall solve the system $\mathbf{B}_2\mathbf{d} = \mathbf{a}$ as $\mathbf{E}_1(\mathbf{E}_2\mathbf{d}) = \mathbf{a}$. Solving the system $\mathbf{E}_1\mathbf{u} = \mathbf{a}$, which reads

$$\begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}, \quad \text{we find} \quad \mathbf{u} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Solving the system $\mathbf{E}_2\mathbf{d} = \mathbf{u}$, which reads

$$\begin{bmatrix} 2 & & \\ 1 & 1 & \\ 1 & & 1 \end{bmatrix} \cdot \mathbf{d} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \text{we find} \quad \mathbf{d} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}.$$

Step 4. The largest t such that $54 - 0.5t \geq 0$, $63 - 0.5t \geq 0$, $15 - 0.5t \geq 0$ is $t = 30$. Since $15 - 0.5t = 0$, the leaving variable is x_7.

Step 5. Now we have

$$\begin{bmatrix} x_1^* \\ x_3^* \\ x_4^* \end{bmatrix} = \begin{bmatrix} 54 - 0.5t \\ 63 - 0.5t \\ t \end{bmatrix} = \begin{bmatrix} 39 \\ 48 \\ 30 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_3 = \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3 \quad \text{with} \quad \mathbf{E}_3 = \begin{bmatrix} 1 & & 0.5 \\ & 1 & 0.5 \\ & & 0.5 \end{bmatrix}.$$

The fourth iteration begins.

Step 1. We shall solve the system $\mathbf{y}\mathbf{B}_3 = \mathbf{c}_B$ as $((\mathbf{y}\mathbf{E}_1)\mathbf{E}_2)\mathbf{E}_3 = \mathbf{c}_B$. Solving the system $\mathbf{u}\mathbf{E}_3 = \mathbf{c}_B$, which reads

$$\mathbf{u}\begin{bmatrix} 1 & & 0.5 \\ & 1 & 0.5 \\ & & 0.5 \end{bmatrix} = [19, 12, 17], \quad \text{we find} \quad \mathbf{u} = [19, 12, 3].$$

Solving the system $\mathbf{v}\mathbf{E}_2 = \mathbf{u}$, which reads

$$\mathbf{v}\begin{bmatrix} 2 & & \\ 1 & 1 & \\ 1 & & 1 \end{bmatrix} = [19, 12, 3], \quad \text{we find} \quad \mathbf{v} = [2, 12, 3].$$

Solving the system $\mathbf{y}\mathbf{E}_1 = \mathbf{v}$, which reads

$$\mathbf{y}\begin{bmatrix} 1 & 1 & \\ & 1 & \\ & 3 & 1 \end{bmatrix} = [2, 12, 3], \quad \text{we find} \quad \mathbf{y} = [2, 1, 3].$$

Step 2. Since

$$\mathbf{c}_N - \mathbf{y}\mathbf{A}_N = [13, 0, 0, 0] - [2, 1, 3] \cdot \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix} = [-1, -2, -1, -3]$$

we find no candidate for entering the basis. Hence the current solution is optimal.

Even though $\mathbf{B}_0 = \mathbf{I}$ whenever the initial basis consists of the slack variables, the case of an arbitrary $\mathbf{B}_0$ is worth considering. In this case, the identity $\mathbf{B}_k = \mathbf{E}_1\mathbf{E}_2 \cdots \mathbf{E}_k$ generalizes into

$$\mathbf{B}_k = \mathbf{B}_0\mathbf{E}_1\mathbf{E}_2 \cdots \mathbf{E}_k$$

and the two systems $\mathbf{y}\mathbf{B}_k = \mathbf{c}_B$, $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ may be solved as $(((\mathbf{y}\mathbf{B}_0)\mathbf{E}_1) \cdots)\mathbf{E}_k = \mathbf{c}_B$ and $\mathbf{B}_0(\mathbf{E}_1(\cdots(\mathbf{E}_k\mathbf{d}))) = \mathbf{a}$, respectively. Now a triangular factorization

$$\mathbf{L}_m\mathbf{P}_m \cdots \mathbf{L}_1\mathbf{P}_1\mathbf{B}_0 = \mathbf{U}$$

of the initial basis $\mathbf{B}_0$ may be computed before the first iteration and then used again and again in conjunction with the growing sequence $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_k$. Note that

$$\mathbf{U} = \mathbf{U}_m\mathbf{U}_{m-1} \cdots \mathbf{U}_1$$

with each $\mathbf{U}_j$ standing for the eta matrix obtained when the jth column of $\mathbf{I}$ is replaced by the jth column of $\mathbf{U}$ (a verification of this claim is left for problem 7.6), and so

$$\mathbf{L}_m\mathbf{P}_m \cdots \mathbf{L}_1\mathbf{P}_1\mathbf{B}_k = \mathbf{U}_m\mathbf{U}_{m-1} \cdots \mathbf{U}_1\mathbf{E}_1\mathbf{E}_2 \cdots \mathbf{E}_k.$$

In this notation, the system $\mathbf{yB}_k = \mathbf{c}_B$ may be solved by first solving $(((\mathbf{yU}_m)\mathbf{U}_{m-1}) \cdots)\mathbf{E}_k = \mathbf{c}_B$ and then replacing $\mathbf{y}$ by $((\mathbf{yL}_m\mathbf{P}_m) \cdots)\mathbf{L}_1\mathbf{P}_1$. The details of this procedure may be spelled out as follows.

1. Set $i = k$ and $\mathbf{y} = \mathbf{c}_B$.
2. If $i \geq 1$, then set $\mathbf{v} = \mathbf{y}$, replace $\mathbf{y}$ by the solution of $\mathbf{yE}_i = \mathbf{v}$, replace i by $i - 1$, and repeat this step.
3. Set $j = 1$.
4. If $j \leq m$, then set $\mathbf{v} = \mathbf{y}$, replace $\mathbf{y}$ by the solution of $\mathbf{yU}_j = \mathbf{v}$, replace j by $j + 1$, and repeat this step.
5. Set $j = m$.
6. If $j \geq 1$, then replace $\mathbf{y}$ by $\mathbf{yL}_j\mathbf{P}_j$, replace j by $j - 1$, and repeat this step.

Similarly, the system $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ may be solved as $\mathbf{U}_m(\mathbf{U}_{m-1}(\cdots(\mathbf{E}_k\mathbf{d}))) = (\mathbf{L}_m\mathbf{P}_m(\cdots(\mathbf{L}_1\mathbf{P}_1\mathbf{a})))$; the details of this procedure may be spelled out as follows.

1. Set $j = 1$ and $\mathbf{d} = \mathbf{a}$.
2. If $j \leq m$, then replace $\mathbf{d}$ by $\mathbf{L}_j\mathbf{P}_j\mathbf{d}$, replace j by $j + 1$, and repeat this step.
3. Set $j = m$.
4. If $j \geq 1$, then set $\mathbf{v} = \mathbf{d}$, replace $\mathbf{d}$ by the solution of $\mathbf{U}_j\mathbf{d} = \mathbf{v}$, replace j by $j - 1$, and repeat this step.
5. Set $i = 1$.
6. If $i \leq k$, then set $\mathbf{v} = \mathbf{d}$, replace $\mathbf{d}$ by the solution of $\mathbf{E}_i\mathbf{d} = \mathbf{v}$, replace i by $i + 1$, and repeat this step.

To store each $\mathbf{E}_i$, we need only store its eta column and record the position of this column in the matrix. Furthermore, if the eta columns are sufficiently sparse, then they may be stored in the "packed form" mentioned in Chapter 6, so that only the nonzero entries are stored and their positions in the column recorded. The same remark applies to the triangular eta matrices $\mathbf{L}_j$ and $\mathbf{U}_j$. Each of the permutation matrices $\mathbf{P}_j$, obtained by interchanging some row of $\mathbf{I}$ with the jth row, may be represented by a single pointer specifying the interchanged row. A sequential file storing the matrices

$$\mathbf{P}_1, \mathbf{L}_1, \mathbf{P}_2, \mathbf{L}_2, \ldots, \mathbf{P}_m, \mathbf{L}_m, \mathbf{U}_m, \mathbf{U}_{m-1}, \ldots, \mathbf{U}_1, \mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_k$$

in this fashion is called the *eta file*. This file is scanned backward, from $\mathbf{E}_k$ to $\mathbf{P}_1$, in solving the system $\mathbf{yB}_k = \mathbf{c}_B$, and it is solved forward, from $\mathbf{P}_1$ to $\mathbf{E}_k$, in solving the system $\mathbf{B}_k\mathbf{d} = \mathbf{a}$. For this reason, the procedure for solving $\mathbf{yB}_k = \mathbf{c}_B$ is sometimes

referred to as the *backward transformation*, or BTRAN, and the procedure for solving $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ is referred to as the *forward transformation*, or FTRAN. Note that the backward and the forward scans alternate and that each new item $\mathbf{E}_{k+1}$ is added to the open end of the file after the file has been scanned forward all the way to $\mathbf{E}_k$ and before the next scan backward to $\mathbf{P}_1$ begins. (The reader should be warned that the term *eta file* is usually employed in connection with the "product form of the inverse," in which case it refers to a different file; see problem 7.13.)

Refactorizations

Since the eta file grows with each iteration, BTRAN and FTRAN become progressively more and more laborious; eventually, they could even take longer than solving the two systems $\mathbf{yB}_k = \mathbf{c}_B$ and $\mathbf{B}_k\mathbf{d} = \mathbf{a}$ from scratch. Such counterproductive uses of the eta file may be avoided by discarding the whole file from time to time and treating the current $\mathbf{B}_k$ as a new $\mathbf{B}_0$: compute a fresh triangular factorization of this matrix, and let a new sequence $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \ldots$ grow from that point on. These periodic *refactorizations* of the basis keep the overall time spent on executions of steps 1 and 3 within acceptable limits. (For historical reasons, refactorizations are sometimes referred to as "reinversions.")

How often should the basis be refactorized? If T_0 stands for the time spent on the refactorization, if T_k stands for the time spent on BTRAN and FTRAN in the kth iteration after refactorization, and if the basis is refactorized after r iterations, then the average time per execution of steps 1 and 3, including an appropriate share of the overhead T_0, comes to

$$T_r^* = \frac{1}{r}\sum_{k=1}^{r} T_k. \tag{7.14}$$

Obviously, r should be chosen so as to minimize T_r^*. A trivial way of doing so relies on the observation that $T_1^*, T_2^*, T_3^*, \ldots$ first decrease (as the overhead T_0 gets distributed over more and more iterations) and then they begin to grow (as the length of the eta file begins to take over). Thus, we need only keep track of the cumulative total $T_0 + T_1 + \cdots + T_k$ and refactorize as soon as this quantity divided by k stops decreasing. (A rigorous proof of this claim, relying only on the natural assumption that $T_1 \leq T_2 \leq T_3 \leq \cdots$ is left for problem 7.7.)

In solving large sparse problems arising from applications, the basis is refactorized quite frequently, often after every twenty iterations or so. An exact analysis of the reasons behind these frequent refactorizations is both impossible and unnecessary: impossible, since the relevant statistics vary unpredictably from one problem to the next and unnecessary, since there is no point in a theoretical justification of a policy whose practical success has been firmly established. All the same, the insight provided by an inexact analysis is better than no insight at all. For this reason, we are going to present a few observations concerning the behavior of the large sparse problems encountered in practice.