From the first equation of system (5.59) it is clear that for $k=1$ the coefficients in formula (5.62) are:

$$A_1 = -\frac{c_1}{b_1}, \quad F_1 = \frac{f_1}{b_1}. \tag{5.63}$$

Suppose that all the coefficients A_k and F_k have already been computed up to some fixed k, $1 \le k \le n-1$. Substituting the expression $x_k = A_k x_{k+1} + F_k$ into the equation number $k+1$ of system (5.59) we obtain:

$$x_{k+1} = -\frac{c_{k+1}}{b_{k+1} + a_{k+1}A_k} x_{k+2} + \frac{f_{k+1} - a_{k+1}F_k}{b_{k+1} + a_{k+1}A_k}.$$

Therefore, the coefficients A_k and F_k satisfy the following recurrence relations:

$$\begin{gathered} A_{k+1} = -\frac{c_{k+1}}{b_{k+1} + a_{k+1}A_k}, \quad F_{k+1} = \frac{f_{k+1} - a_{k+1}F_k}{b_{k+1} + a_{k+1}A_k}, \\ k = 1, 2, \dots, n-1. \end{gathered} \tag{5.64}$$

As such, the algorithm of solving system (5.59) gets split into two stages. At the first stage, we evaluate the coefficients A_k and F_k for $k = 1, 2, \dots, n$ using formulae (5.63) and (5.64). At the second stage, we solve back for the actual unknowns $x_n, x_{n-1}, \dots, x_1$ using formulae (5.62) for $k = n, n-1, \dots, 1$.

In the literature, one can find several alternative names for the tri-diagonal Gaussian elimination procedure that we have described. Sometimes, the term *marching* is used. The first stage of the algorithm is also referred to as the forward stage or forward marching, when the marching coefficients A_k and F_k are computed. Accordingly, the second stage of the algorithm, when relations (5.62) are applied consecutively in the reverse order is called backward marching.

We will now estimate the computational complexity of the tri-diagonal elimination. At the forward stage, the elimination according to formulae (5.63) and (5.64) requires $\mathscr{O}(n)$ arithmetic operations. At the backward stage, formula (5.62) is applied n times, which also requires $\mathscr{O}(n)$ operations. Altogether, the complexity of the tri-diagonal elimination is $\mathscr{O}(n)$ arithmetic operations. It is clear that no algorithm can be built that would be asymptotically cheaper than $\mathscr{O}(n)$, because the number of unknowns in the system is also $\mathscr{O}(n)$.

Let us additionally note that the tri-diagonal elimination is apparently the only example available in the literature of a direct method with linear complexity, i.e., of a method that produces the exact solution of a linear system at a cost of $\mathscr{O}(n)$ operations. In other words, the computational cost is directly proportional to the dimension of the system. We will later see examples of direct methods that produce the exact solution at a cost of $\mathscr{O}(n \ln n)$ operations, and examples of iterative methods that cost $\mathscr{O}(n)$ operations but only produce an approximate solution. However, no other method of computing the exact solution with a genuinely linear complexity is known.

The algorithm can also be generalized to the case of the banded matrices. Matrices of this type may contain non-zero entries on several neighboring diagonals, including the main diagonal. Normally we would assume that the number m of the non-zero