be calculated directly, which yields:

$$\boldsymbol{L} = \boldsymbol{T}_1^{-1}\boldsymbol{T}_2^{-1}\dots\boldsymbol{T}_{n-1}^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ t_{2,1} & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ t_{k,1} & t_{k,2} & \dots & 1 & 0 & \dots & 0 \\ t_{k+1,1} & t_{k+1,2} & \dots & t_{k+1,k} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \dots & t_{n,k} & t_{n,k+1} & \dots & 1 \end{bmatrix}. \tag{5.75}$$

Consequently, the matrix $\boldsymbol{L}$ is indeed a lower triangular matrix (with all its diagonal entries equal to one), and the factorization formula (5.73) holds.

The $\boldsymbol{LU}$ factorization of the matrix $\boldsymbol{A}$ allows us to analyze the computational complexity of the Gaussian elimination algorithm as it applies to solving multiple linear systems that have the same matrix $\boldsymbol{A}$ but different right-hand sides. The cost of obtaining the factorization itself, i.e., that of computing the matrix $\boldsymbol{U}$, is cubic: $\mathscr{O}(n^3)$ arithmetic operations. This factorization obviously stays the same when the right-hand side changes. For a given right-hand side $\boldsymbol{f}$, we need to solve the system $\boldsymbol{LUx} = \boldsymbol{f}$. This amounts to first solving the system $\boldsymbol{Lg} = \boldsymbol{f}$ with a lower triangular matrix $\boldsymbol{L}$ of (5.75) and obtaining $\boldsymbol{g} = \boldsymbol{L}^{-1}\boldsymbol{f} = \boldsymbol{T}_{n-1}\boldsymbol{T}_{n-2}\dots\boldsymbol{T}_1\boldsymbol{f}$, and then solving the system $\boldsymbol{Ux} = \boldsymbol{g}$ with an upper triangular matrix $\boldsymbol{U}$ of (5.72) and obtaining $\boldsymbol{x}$. The cost of either solution is $\mathscr{O}(n^2)$ operations. Consequently, once the $\boldsymbol{LU}$ factorization has been built, each additional right-hand side can be accommodated at a quadratic cost.

In particular, consider the problem of finding the inverse matrix $\boldsymbol{A}^{-1}$ using Gaussian elimination. By definition, $\boldsymbol{AA}^{-1} = \boldsymbol{I}$. In other words, each column of the matrix $\boldsymbol{A}^{-1}$ is the solution to the system $\boldsymbol{Ax} = \boldsymbol{f}$ with the right-hand side $\boldsymbol{f}$ equal to the corresponding column of the identity matrix $\boldsymbol{I}$. Altogether, there are n columns, each adding an $\mathscr{O}(n^2)$ solution cost to the $\mathscr{O}(n^3)$ initial cost of the $\boldsymbol{LU}$ factorization that is performed only once ahead of time. We therefore conclude that the overall cost of computing $\boldsymbol{A}^{-1}$ using Gaussian elimination is also cubic: $\mathscr{O}(n^3)$ operations.

Finally, let us note that for a given matrix $\boldsymbol{A}$, its $\boldsymbol{LU}$ factorization is, generally speaking, not unique. The procedure that we have described yields a particular form of the $\boldsymbol{LU}$ factorization (5.73) defined by an additional constraint that all diagonal entries of the matrix $\boldsymbol{L}$ of (5.75) be equal to one. Instead, we could have required, for example, that the diagonal entries of $\boldsymbol{U}$ be equal to one. Then, the matrices $\boldsymbol{T}_k$ of (5.71) get replaced by:

$$\tilde{\boldsymbol{T}}_k = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \dots & t_{k,k} & 0 & \dots & 0 \\ 0 & \dots & -t_{k+1,k} & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -t_{n,k} & 0 & \dots & 1 \end{bmatrix}, \quad \text{where} \quad t_{k,k} = \frac{1}{a_{kk}^{(k-1)}}, \tag{5.76}$$