Chapter 1

Linear Algebra

The source of linear algebra is the solution of systems of linear algebraic equations. Linear algebra is the foundation upon which almost all applied mathematics rests. This is not to say that nonlinear equations are less important; rather, progress in the vastly more complicated nonlinear realm is impossible without a firm grasp of the fundamentals of linear systems. Furthermore, linear algebra underlies the numerical analysis of continuous systems, both linear and nonlinear, which are typically modeled by differential equations. Without a systematic development of the subject from the start, we will be ill equipped to handle the resulting large systems of linear equations involving many (e.g., thousands of) unknowns.

This first chapter is devoted to the systematic development of direct\(^\dagger\) algorithms for solving systems of linear algebraic equations in a finite number of variables. Our primary focus will be the most important situation involving the same number of equations as unknowns, although in Section 1.8 we extend our techniques to completely general linear systems. While the former usually have a unique solution, more general systems more typically have either no solutions, or infinitely many, and so tend to be of less direct physical relevance. Nevertheless, the ability to confidently handle all types of linear systems is a basic prerequisite for the subject.

The basic solution algorithm is known as *Gaussian elimination*, in honor of one of the all-time mathematical greats — the nineteenth century German mathematician Carl Friedrich Gauss. As the father of linear algebra, his name will occur repeatedly throughout this text. Gaussian elimination is quite elementary, but remains one of *the* most important techniques in applied (as well as theoretical) mathematics. Section 1.7 discusses some practical issues and limitations in computer implementations of the Gaussian elimination method for large systems arising in applications.

The systematic development of the subject relies on the fundamental concepts of scalar, vector, and matrix, and we quickly review the basics of matrix arithmetic. Gaussian elimination can be reinterpreted as matrix factorization, the (permuted) *LU* decomposition, which provides additional insight into the solution algorithm. Matrix inverses and determinants are discussed in Sections 1.5 and 1.9, respectively. However, both play a relatively minor role in practical applied mathematics, and so will not assume their more traditional central role in this applications-oriented text.

\(^\dagger\) Indirect algorithms, which are based on iteration, will be the subject of Chapter 10.
1.1. Solution of Linear Systems.

Gaussian elimination is a simple, systematic approach to the solution of systems of linear equations. It is the workhorse of linear algebra, and as such of absolutely fundamental importance in applied mathematics. In this section, we review the method in the most important case in which there are the same number of equations as unknowns. The general situation will be deferred until Section 1.8.

To illustrate, consider an elementary system of three linear equations

\[\begin{align*}
x + 2y + z &= 2, \\
2x + 6y + z &= 7, \\
x + y + 4z &= 3,
\end{align*}\]

in three unknowns \(x, y, z\). Linearity refers to the fact that the unknowns only appear to the first power in the equations. The basic solution method is to systematically employ the following fundamental operation:

**Linear System Operation #1:** Add a multiple of one equation to another equation.

Before continuing, you should convince yourself that this operation does not change the solutions to the system. As a result, our goal is to judiciously apply the operation and so be led to a much simpler linear system that is easy to solve, and, moreover has the same solutions as the original. Any linear system that is derived from the original system by successive application of such operations will be called an equivalent system. By the preceding remark, equivalent linear systems have the same solutions.

The systematic feature is that we successively eliminate the variables in our equations in order of appearance. We begin by eliminating the first variable, \(x\), from the second equation. To this end, we subtract twice the first equation from the second, leading to the equivalent system

\[\begin{align*}
x + 2y + z &= 2, \\
2y - z &= 3, \\
x + y + 4z &= 3. 
\end{align*}\]

Next, we eliminate \(x\) from the third equation by subtracting the first equation from it:

\[\begin{align*}
x + 2y + z &= 2, \\
2y - z &= 3, \\
-y + 3z &= 1. 
\end{align*}\]

The equivalent system (1.3) is already simpler than the original system (1.1). Notice that the second and third equations do not involve \(x\) (by design) and so constitute a system of two linear equations for two unknowns. Moreover, once we have solved this subsystem for \(y\) and \(z\), we can substitute the answer into the first equation, and we need only solve a single linear equation for \(x\).

\[\footnote{Also, there are no product terms like \(xy\) or \(xyz\). The “official” definition of linearity will be deferred until Chapter 7.}\]
We continue on in this fashion, the next phase being the elimination of the second variable \( y \) from the third equation by adding \( \frac{1}{2} \) the second equation to it. The result is

\[
\begin{align*}
x + 2y + z &= 2, \\
2y - z &= 3, \\
\frac{5}{2}z &= \frac{5}{2},
\end{align*}
\]

which is the simple system we are after. It is in what is called \textit{triangular form}, which means that, while the first equation involves all three variables, the second equation only involves the second and third variables, and the last equation only involves the last variable.

Any triangular system can be straightforwardly solved by the method of \textit{Back Substitution}. As the name suggests, we work backwards, solving the last equation first, which requires \( z = 1 \). We substitute this result back into the next to last equation, which becomes \( 2y - 1 = 3 \), with solution \( y = 2 \). We finally substitute these two values for \( y \) and \( z \) into the first equation, which becomes \( x + 5 = 2 \), and so the solution to the triangular system (1.4) is

\[
x = -3, \quad y = 2, \quad z = 1.
\]

Moreover, since we only used our basic operation to pass from (1.1) to the triangular system (1.4), this is also the solution to the original system of linear equations. We note that the system (1.1) has a unique — meaning one and only one — solution, namely (1.5).

And that, barring a few complications that can crop up from time to time, is all that there is to the method of Gaussian elimination! It is very simple, but its importance cannot be overemphasized. Before discussing the relevant issues, it will help to reformulate our method in a more convenient matrix notation.

\[\text{1.2. Matrices and Vectors.}\]

A \textit{matrix} is a rectangular array of numbers. Thus,

\[
\begin{pmatrix}
1 & 0 & 3 \\
-2 & 4 & 1
\end{pmatrix}, \quad
\begin{pmatrix}
\pi & 0 \\
e & \frac{1}{2} \\
-1 & .83 \\
\sqrt{5} & -\frac{4}{7}
\end{pmatrix}, \quad
\begin{pmatrix}
.2 & -1.6 & .32 \\
0 & 0 \\
1 & 3 & 5
\end{pmatrix},
\]

are all examples of matrices. We use the notation

\[
A = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\end{pmatrix}
\]

for a general matrix of size \( m \times n \) (read “\( m \) by \( n \)”), where \( m \) denotes the number of \textit{rows} in \( A \) and \( n \) denotes the number of \textit{columns}. Thus, the preceding examples of matrices have respective sizes \( 2 \times 3, 4 \times 2, 1 \times 3, 2 \times 1 \) and \( 2 \times 2 \). A matrix is \textit{square} if \( m = n \), i.e., it has the same number of rows as columns. A \textit{column vector} is a \( m \times 1 \) matrix, while a \textit{row vector} is a \( 1 \times n \) matrix.
vector is a $1 \times n$ matrix. As we shall see, column vectors are by far the more important of the two, and the term “vector” without qualification will always mean “column vector”. A $1 \times 1$ matrix, which has but a single entry, is both a row and column vector.

The number that lies in the $i^{th}$ row and the $j^{th}$ column of $A$ is called the $(i,j)$ entry of $A$, and is denoted by $a_{ij}$. The row index always appears first and the column index second. Two matrices are equal, $A = B$, if and only if they have the same size, and all their entries are the same: $a_{ij} = b_{ij}$.

A general linear system of $m$ equations in $n$ unknowns will take the form

$$
a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1, \\
a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n = b_2, \\
\vdots \quad \vdots \quad \vdots \\
a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n = b_m. $$

As such, it has three basic constituents: the $m \times n$ coefficient matrix $A$, with entries $a_{ij}$ as in (1.6), the column vector $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ containing the unknowns, and the column vector $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$ containing right hand sides. For instance, in our previous example (1.1),

the coefficient matrix is $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix}$, the vector of unknowns is $\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, while $\mathbf{b} = \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix}$ contains the right hand sides.

**Remark:** We will consistently use bold face lower case letters to denote vectors, and ordinary capital letters to denote general matrices.

**Matrix Arithmetic**

There are three basic operations in matrix arithmetic: *matrix addition*, *scalar multiplication*, and *matrix multiplication*. First we define *addition* of matrices. You are only allowed to add two matrices of the *same size*, and matrix addition is performed entry by entry.

---

\[ ^{\dagger} \text{In tensor analysis, [2], a sub- and super-script notation is adopted, with } a_{ij}^{k} \text{ denoting the } (i,j) \text{ entry of the matrix } A. \text{ This has certain advantages, but, to avoid possible confusion with } \text{powers, we shall stick with the simpler subscript notation throughout this text.} \]
entry. Therefore, if $A$ and $B$ are $m \times n$ matrices, their sum $C = A + B$ is the $m \times n$ matrix whose entries are given by $c_{ij} = a_{ij} + b_{ij}$ for $i = 1, \ldots, m, j = 1, \ldots, n$. For example,

$$
\begin{pmatrix}
1 & 2 \\
-1 & 0
\end{pmatrix} +
\begin{pmatrix}
3 & -5 \\
2 & 1
\end{pmatrix} =
\begin{pmatrix}
4 & -3 \\
1 & 1
\end{pmatrix}.
$$

When defined, matrix addition is commutative, $A + B = B + A$, and associative, $A + (B + C) = (A + B) + C$, just like ordinary addition.

A scalar is a fancy name for an ordinary number — the term merely distinguishes it from a vector or a matrix. For the time being, we will restrict our attention to real scalars and matrices with real entries, but eventually complex scalars and complex matrices must be dealt with. We will often identify a scalar $c \in \mathbb{R}$ with the $1 \times 1$ matrix $(c)$ in which it is the sole entry. Scalar multiplication takes a scalar $c$ and an $m \times n$ matrix $A$ and computes the $m \times n$ matrix $C = cA$ by multiplying each entry of $A$ by $c$. Thus, $b_{ij} = ca_{ij}$ for $i = 1, \ldots, m, j = 1, \ldots, n$. For example,

$$
3 \begin{pmatrix} 1 & 2 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ -3 & 0 \end{pmatrix}.
$$

Basic properties of scalar multiplication are summarized at the end of this section.

Finally, we define matrix multiplication. First, the product between a row vector $a$ and a column vector $x$ having the same number of entries is the scalar defined by the following rule:

$$
a \cdot x = (a_1, a_2, \ldots, a_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = \sum_{k=1}^{n} a_k x_k. \quad (1.8)
$$

More generally, if $A$ is an $m \times n$ matrix and $B$ is an $n \times p$ matrix, so that the number of columns in $A$ equals the number of rows in $B$, then the matrix product $C = AB$ is defined as the $m \times p$ matrix whose $(i, j)$ entry equals the vector product of the $i$th row of $A$ and the $j$th column of $B$. Therefore,

$$
c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}. \quad (1.9)
$$

Note that our restriction on the sizes of $A$ and $B$ guarantees that the relevant row and column vectors will have the same number of entries, and so their product is defined.

For example, the product of the coefficient matrix $A$ and vector of unknowns $x$ for our original system (1.1) is given by

$$
A \cdot x = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x + 2y + z \\ 2x + 6y + z \\ x + y + 4z \end{pmatrix}.
$$
The result is a column vector whose entries reproduce the left hand sides of the original linear system! As a result, we can rewrite the system

\[ A \mathbf{x} = \mathbf{b} \quad (1.10) \]

as an equality between two column vectors. This result is general; a linear system (1.7) consisting of \( m \) equations in \( n \) unknowns can be written in the matrix form (1.10) where \( A \) is the \( m \times n \) coefficient matrix (1.6), \( \mathbf{x} \) is the \( n \times 1 \) column vectors of unknowns, and \( \mathbf{b} \) is the \( m \times 1 \) column vector containing the right hand sides. This is the reason behind the non-evident definition of matrix multiplication. Component-wise multiplication of matrix entries turns out to be almost completely useless in applications.

Now, the bad news. Matrix multiplication is not commutative. For example, \( BA \) may not be defined even when \( AB \) is. Even if both are defined, they may be different sized matrices. For example the product of a row vector \( \mathbf{r} \), a \( 1 \times n \) matrix, and a column vector \( \mathbf{c} \), an \( n \times 1 \) matrix, is a \( 1 \times 1 \) matrix or scalar \( s = \mathbf{r} \mathbf{c} \), whereas the reversed product \( C = \mathbf{c} \mathbf{r} \) is an \( n \times n \) matrix. For example,

\[
\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} -2 & 5 \\ -4 & 11 \end{pmatrix} \neq \begin{pmatrix} 3 & 4 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.
\]

In computing the latter product, don’t forget that we multiply the rows of the first matrix by the columns of the second. Moreover, even if the matrix products \( AB \) and \( BA \) have the same size, which requires both \( A \) and \( B \) to be square matrices, we may still have \( AB \neq BA \). For example,

\[
\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} -2 & 5 \\ -4 & 11 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.
\]

On the other hand, matrix multiplication is associative, so \( A(BC) = (AB)C \) whenever \( A \) has size \( m \times n \), \( B \) has size \( n \times p \) and \( C \) has size \( p \times q \); the result is a matrix of size \( m \times q \). The proof of this fact is left to the reader. Consequently, the one significant difference between matrix algebra and ordinary algebra is that you need to be careful not to change the order of multiplicative factors without proper justification.

Since matrix multiplication multiplies rows times columns, one can compute the columns in a matrix product \( C = AB \) by multiplying the matrix \( A \) by the individual columns of \( B \). The \( k \)th column of \( C \) is equal to the product of \( A \) with the \( k \)th column of \( B \). For example, the two columns of the matrix product

\[
\begin{pmatrix} 1 & -1 & 2 \\ 2 & 0 & -2 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 8 & 6 \end{pmatrix}
\]

are obtained by multiplying the first matrix with the individual columns of the second:

\[
\begin{pmatrix} 1 & -1 & 2 \\ 2 & 0 & -2 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 8 \end{pmatrix}, \quad \begin{pmatrix} 1 & -1 & 2 \\ 2 & 0 & -2 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}.
\]
In general, if we use $b_j$ to denote the $j^{th}$ column of $B$, then
\[
AB = A \begin{pmatrix} b_1 & b_2 & \ldots & b_p \end{pmatrix} = \begin{pmatrix} A b_1 & A b_2 & \ldots & A b_p \end{pmatrix}.
\] (1.11)

There are two important special matrices. The first is the zero matrix of size $m \times n$, denoted $O_{m \times n}$ or just $O$ if the size is clear from context. It forms the additive unit, so $A + O = A = O + A$ for any matrix $A$ of the same size. The role of the multiplicative unit is played by the square identity matrix
\[
I = I_n = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}
\]
of size $n \times n$. The entries of $I$ along the main diagonal (which runs from top left to bottom right) are equal to 1; the off-diagonal entries are all 0. As the reader can check, if $A$ is any $m \times n$ matrix, then $I_m A = A = A I_n$. We will sometimes write the last equation as just $IA = A = AI$; even though the identity matrices can have different sizes, only one size is valid for each matrix product to be defined.

The identity matrix is a particular example of a diagonal matrix. In general, a matrix is diagonal if all its off-diagonal entries are zero: $a_{ij} = 0$ for all $i \neq j$. We will sometimes write $D = \text{diag} \left( c_1, \ldots, c_n \right)$ for the $n \times n$ diagonal matrix with diagonal entries $d_{ii} = c_i$.

Thus, $\text{diag} \left( 1, 3, 0 \right)$ refers to the diagonal matrix
\[
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{pmatrix},
\]
while the $n \times n$ identity matrix can be written as $I_n = \text{diag} \left( 1, 1, \ldots, 1 \right)$.

Let us conclude this section by summarizing the basic properties of matrix arithmetic. In the following table, $A, B, C$ are matrices, $c, d$ scalars, $O$ is a zero matrix, and $I$ is an identity matrix. The matrices are assumed to have the correct sizes so that the indicated operations are defined.

### 1.3. Gaussian Elimination — Regular Case.

With the basic matrix arithmetic operations in hand, let us now return to our primary task. The goal is to develop a systematic method for solving linear systems of equations. While we could continue to work directly with the equations, matrices provide a convenient alternative that begins by merely shortening the amount of writing, but ultimately leads to profound insight into the solution and its structure.

We begin by replacing the system (1.7) by its matrix constituents. It is convenient to ignore the vector of unknowns, and form the augmented matrix
\[
M = (A | b) = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn} & b_n
\end{pmatrix}
\]  
(1.12)
which is an $m \times (n+1)$ matrix obtained by tacking the right hand side vector onto the original coefficient matrix. The extra vertical line is included just to remind us that the last column of this matrix is special. For example, the augmented matrix for the system (1.1), i.e.,

$$
\begin{align*}
x + 2y + z &= 2, \\
2x + 6y + z &= 7, \\
x + y + 4z &= 3,
\end{align*}
$$

is

$$
M = \begin{pmatrix}
1 & 2 & 1 & 2 \\
2 & 6 & 1 & 7 \\
1 & 1 & 4 & 3
\end{pmatrix}.
$$

(1.13)

Note that one can immediately recover the equations in the original linear system from the augmented matrix. Since operations on equations also affect their right hand sides, keeping track of everything is most easily done through the augmented matrix.

For the time being, we will concentrate our efforts on linear systems that have the same number, $n$, of equations as unknowns. The associated coefficient matrix $A$ is square, of size $n \times n$. The corresponding augmented matrix $M = (A | b)$ then has size $n \times (n+1)$.

The matrix operation that assumes the role of Linear System Operation #1 is:

**Elementary Row Operation #1:**

Add a scalar multiple of one row of the augmented matrix to another row.

For example, if we add $-2$ times the first row of the augmented matrix (1.13) to the second row, the result is the row vector

$$
-2(1 2 1 2) + (2 6 1 7) = (0 2 -1 3).
$$
The result can be recognized as the second row of the modified augmented matrix
\[
\begin{pmatrix}
1 & 2 & 1 & 2 \\
0 & 2 & -1 & 3 \\
1 & 1 & 4 & 3
\end{pmatrix}
\] (1.14)
that corresponds to the first equivalent system (1.2). When elementary row operation #1 is performed, it is critical that the result replace the row being added to — not the row being multiplied by the scalar. Notice that the elimination of a variable in an equation — in this case, the first variable in the second equation — amounts to making its entry in the coefficient matrix equal to zero.

We shall call the (1, 1) entry of the coefficient matrix the first pivot. The precise definition of pivot will become clear as we continue; the one key requirement is that a pivot be nonzero. Eliminating the first variable \(x\) from the second and third equations amounts to making all the matrix entries in the column below the pivot equal to zero. We have already done this with the (2, 1) entry in (1.14). To make the (3, 1) entry equal to zero, we subtract the first row from the last row. The resulting augmented matrix is
\[
\begin{pmatrix}
1 & 2 & 1 & 2 \\
0 & 2 & -1 & 3 \\
0 & -1 & 3 & 1
\end{pmatrix},
\]
which corresponds to the system (1.3). The second pivot is the (2, 2) entry of this matrix, which is 2, and is the coefficient of the second variable in the second equation. Again, the pivot must be nonzero. We use the elementary row operation of adding \(\frac{1}{2}\) of the second row to the third row to make the entry below the second pivot equal to 0; the result is the augmented matrix
\[
\begin{pmatrix}
1 & 2 & 1 & 2 \\
0 & 2 & -1 & 3 \\
0 & 0 & 5/2 & 5/2
\end{pmatrix},
\]
that corresponds to the triangular system (1.4). We write the final augmented matrix as
\[
N = (U | c), \quad \text{where} \quad U = \begin{pmatrix}
1 & 2 & 1 \\
0 & 2 & -1 \\
0 & 0 & 5/2
\end{pmatrix}, \quad \text{c} = \begin{pmatrix}
2 \\
3 \\
5/2
\end{pmatrix}.
\]
The corresponding linear system has vector form
\[
U \mathbf{x} = \mathbf{c}. \quad (1.15)
\]
Its coefficient matrix \(U\) is upper triangular, which means that all its entries below the main diagonal are zero: \(u_{ij} = 0\) whenever \(i > j\). The three nonzero entries on its diagonal, 1, 2, \(\frac{5}{2}\), including the last one in the (3, 3) slot are the three pivots. Once the system has been reduced to triangular form (1.15), we can easily solve it, as discussed earlier, by back substitution.
Gaussian Elimination — Regular Case

start
for \( j = 1 \) to \( n \)
    if \( m_{jj} = 0 \), stop; print “A is not regular”
    else for \( i = j + 1 \) to \( n \)
        set \( l_{ij} = m_{ij} / m_{jj} \)
        add \(-l_{ij}\) times row \( j \) of \( M \) to row \( i \) of \( M \)
    next \( i \)
next \( j \)
end

The preceding algorithm for solving a linear system is known as regular Gaussian elimination. A square matrix \( A \) will be called regular\(^\dagger\) if the algorithm successfully reduces it to upper triangular form \( U \) with all non-zero pivots on the diagonal. In other words, for regular matrices, we identify each successive nonzero entry in a diagonal position as the current pivot. We then use the pivot row to make all the entries in the column below the pivot equal to zero through elementary row operations of Type #1. A system whose coefficient matrix is regular is solved by first reducing the augmented matrix to upper triangular form and then solving the resulting triangular system by back substitution.

Let us state this algorithm in the form of a program, written in a general “pseudocode” that can be easily translated into any specific language, e.g., C++, FORTRAN, JAVA, MAPLE, MATHEMATICA or MATLAB. We use a single letter \( M = (m_{ij}) \) to denote the current augmented matrix at each stage in the computation, and initialize \( M = (A | b) \). Note that the entries of \( M \) will change as the algorithm progresses. The final output of the program, assuming \( A \) is regular, is the augmented matrix \( M = (U | c) \), where \( U \) is the upper triangular matrix \( U \) whose diagonal entries \( u_{ii} \) are the pivots and \( c \) is the vector of right hand sides obtained after performing the elementary row operations.

Elementary Matrices

A key observation is that elementary row operations can, in fact, be realized by matrix multiplication.

**Definition 1.1.** The elementary matrix \( E \) associated with an elementary row operation for matrices with \( m \) rows is the matrix obtained by applying the row operation to the \( m \times m \) identity matrix \( I_m \).

\(^\dagger\) Strangely, there is no commonly accepted term for these kinds of matrices. Our proposed adjective “regular” will prove to be quite useful in the sequel.
For example, applying the elementary row operation that adds \(-2\) times the first row to the second row to the 3\(\times\)3 identity matrix \(I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\) results in the corresponding elementary matrix \(E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\). We claim that, if \(A\) is any 3-rowed matrix, then multiplying \(E_1 A\) has the same effect as the given elementary row operation. For example,

\[
E_1 A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 1 & 1 & 4 \end{pmatrix},
\]

which you may recognize as the first elementary row operation we used to solve the illustrative example. Indeed, if we set

\[
E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix},
\]

then multiplication by \(E_1\) will subtract twice the first row from the second row, multiplication by \(E_2\) will subtract the first row from the third row, and multiplication by \(E_3\) will add \(\frac{1}{2}\) the second row to the third row — precisely the row operations used to place our original system in triangular form. Therefore, performing them in the correct order (and using the associativity of matrix multiplication), we conclude that when

\[
A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad \text{then} \quad E_3 E_2 E_1 A = U = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & \frac{5}{2} \end{pmatrix}.
\]

The reader should check this by directly multiplying the indicated matrices.

In general, then, the elementary matrix \(E\) of size \(m \times m\) will have all 1’s on the diagonal, a nonzero entry \(c\) in position \((i, j)\), for some \(i \neq j\), and all other entries equal to zero. If \(A\) is any \(m \times n\) matrix, then the matrix product \(EA\) is equal to the matrix obtained from \(A\) by the elementary row operation adding \(c\) times row \(j\) to row \(i\). (Note the reversal of order of \(i\) and \(j\).)

The elementary row operation that undoes adding \(c\) times row \(j\) to row \(i\) is the inverse row operation that subtracts \(c\) (or, equivalently, adds \(-c\)) times row \(j\) from row \(i\). The corresponding inverse elementary matrix again has 1’s along the diagonal and \(-c\) in the \((i, j)\) slot. Let us denote the inverses of the particular elementary matrices (1.16) by \(L_i\), so that, according to our general rule,

\[
L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad L_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix}.
\]

Note that the product

\[
L_i E_i = I
\]

(1.19)
is the $3 \times 3$ identity matrix, reflecting the fact that these are inverse operations. (A more thorough discussion of matrix inverses will be postponed until the following section.)

The product of the latter three elementary matrices is equal to

\[
L = L_1 L_2 L_3 = \begin{pmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
1 & -\frac{1}{2} & 1
\end{pmatrix}.
\]

The matrix $L$ is called a \textit{special lower triangular} matrix, where “lower triangular” means that all the entries above the main diagonal are 0, while “special” indicates that all the entries on the diagonal are equal to 1. Observe that the entries of $L$ below the diagonal are the same as the corresponding nonzero entries in the $L_i$. This is a general fact, that holds when the lower triangular elementary matrices are multiplied in the correct order. (For instance, the product $L_3 L_2 L_1$ is not so easily predicted.) More generally, the following elementary consequence of the laws of matrix multiplication will be used extensively.

\textbf{Lemma 1.2.} If $L$ and $\hat{L}$ are lower triangular matrices of the same size, so is their product $L \hat{L}$. If they are both special lower triangular, so is their product. Similarly, if $U, \hat{U}$ are (special) upper triangular matrices, so is their product $U \hat{U}$.

\textit{The LU Factorization}

We have almost arrived at our first important result. Consider the product of the matrices $L$ and $U$ in (1.17), (1.20). Using equation (1.19), along with the basic property of the identity matrix $I$ and associativity of matrix multiplication, we conclude that

\[
LU = (L_1 L_2 L_3)(E_3 E_2 E_1 A) = L_1 L_2 (L_3 E_3) E_2 E_1 A = L_1 L_2 I E_2 E_1 A \\
= L_1 (L_2 E_2) E_1 A = L_1 I E_1 A = L_1 E_1 A = I A = A.
\]

In other words, we have factorized the coefficient matrix $A = LU$ into a product of a special lower triangular matrix $L$ and an upper triangular matrix $U$ with the nonzero pivots on its main diagonal. The same holds true for almost all square coefficient matrices.

\textbf{Theorem 1.3.} A matrix $A$ is regular if and only if it can be factorized

\[
A = LU,
\]

where $L$ is a special lower triangular matrix, having all 1’s on the diagonal, and $U$ is upper triangular with nonzero diagonal entries, which are its pivots. The nonzero off-diagonal entries $l_{ij}$ for $i > j$ appearing in $L$ prescribe the elementary row operations that bring $A$ into upper triangular form; namely, one subtracts $l_{ij}$ times row $j$ from row $i$ at the appropriate step of the Gaussian elimination process.

\textbf{Example 1.4.} Let us compute the $LU$ factorization of the matrix $A = \begin{pmatrix}
2 & 1 & 1 \\
4 & 5 & 2 \\
2 & -2 & 0
\end{pmatrix}$.

Applying the Gaussian elimination algorithm, we begin by subtracting twice the first row from the second row, and then subtract the first row from the third. The result is the
matrix \( \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 0 \\ 0 & -3 & -1 \end{pmatrix} \). The next step adds the second row to the third row, leading to the upper triangular matrix \( U = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix} \), with its diagonal entries 2, 3, \(-1\) indicating the pivots. The corresponding lower triangular matrix is \( L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \), whose entries below the diagonal are the *negatives* of the multiples we used during the elimination procedure. Namely, the (2,1) entry of \( L \) indicates that we added \(-2\) times the first row to the second row; the (3,1) entry indicates that we added \(-1\) times the first row to the third; and, finally, the (3,2) entry indicates that we added the second row to the third row during the algorithm. The reader might wish to verify the factorization \( A = LU \), or, explicitly,

\[
\begin{pmatrix} 2 & 1 & 1 \\ 4 & 5 & 2 \\ 2 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix} .
\]

Forward and Back Substitution

Once we know the \( LU \) factorization of a regular matrix \( A \), we are able to solve any associated linear system \( A \mathbf{x} = \mathbf{b} \) in two stages:

1. First solve the lower triangular system

\[
L \mathbf{c} = \mathbf{b}
\]

for the vector \( \mathbf{c} \) by *forward substitution*. This is the same as back substitution, except one solves the equations for the variables in the direct order — from first to last. Explicitly,

\[
c_1 = b_1, \quad c_i = b_i - \sum_{j=1}^{i} l_{ij} c_j, \quad \text{for} \quad i = 2, 3, \ldots, n,
\]

noting that the previously computed values of \( c_1, \ldots, c_{i-1} \) are used to determine \( c_i \).

2. Second, solve the resulting upper triangular system

\[
U \mathbf{x} = \mathbf{c}
\]

by *back substitution*. Explicitly, the values of the unknowns

\[
x_n = \frac{c_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left( c_i - \sum_{j=i+1}^{n} u_{ij} x_j \right), \quad \text{for} \quad i = n-1, \ldots, 2, 1,
\]

are successively computed, but now in reverse order.

Note that this algorithm does indeed solve the original system, since if

\[
U \mathbf{x} = \mathbf{c} \quad \text{and} \quad L \mathbf{c} = \mathbf{b}, \quad \text{then} \quad A \mathbf{x} = LU \mathbf{x} = L \mathbf{c} = \mathbf{b}.
\]

2/25/04
Once we have found the \( LU \) factorization of the coefficient matrix \( A \), the Forward and Back Substitution processes quickly produce the solution, and are easy to program on a computer.

**Example 1.5.** With the \( LU \) decomposition

\[
\begin{pmatrix}
2 & 1 & 1 \\
4 & 5 & 2 \\
2 & -2 & 0
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
1 & -1 & 1
\end{pmatrix}\begin{pmatrix}
2 & 1 & 1 \\
0 & 3 & 0 \\
0 & 0 & -1
\end{pmatrix}
\]

found in Example 1.4, we can readily solve any linear system with the given coefficient matrix by Forward and Back Substitution. For instance, to find the solution to

\[
\begin{pmatrix}
2 & 1 & 1 \\
4 & 5 & 2 \\
2 & -2 & 0
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
= \begin{pmatrix}
1 \\
2 \\
1
\end{pmatrix},
\]

we first solve the lower triangular system

\[
\begin{pmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
1 & -1 & 1
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= \begin{pmatrix}
1 \\
2 \\
2
\end{pmatrix},
\]

or, explicitly,

\[
\begin{align*}
a &= 1, \\
2a + b &= 2, \\
a - b + c &= 2.
\end{align*}
\]

The first equation says \( a = 1 \); substituting into the second, we find \( b = 0 \); the final equation gives \( c = 1 \). We then solve the upper triangular system

\[
\begin{pmatrix}
2 & 1 & 1 \\
0 & 3 & 0 \\
0 & 0 & -1
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
= \begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= \begin{pmatrix}
1 \\
0 \\
1
\end{pmatrix},
\]

which is

\[
\begin{align*}
2x + y + z &= 1, \\
3y &= 0, \\
-z &= 1.
\end{align*}
\]

In turn, we find \( z = -1 \), then \( y = 0 \), and then \( x = 1 \), which is the unique solution to the original system.

Of course, if we are not given the \( LU \) factorization in advance, we can just use direct Gaussian elimination on the augmented matrix. Forward and Back Substitution is useful if one has already computed the factorization by solving for a particular right hand side \( b \), but then later wants to know the solutions corresponding to alternative \( b \)'s.

### 1.4. Pivoting and Permutations.

The method of Gaussian elimination presented so far applies only to regular matrices. But not every square matrix is regular; a simple class of examples are matrices whose upper left entry is zero, and so cannot serve as the first pivot. More generally, the regular elimination algorithm cannot proceed whenever a zero entry appears in the current pivot spot on the diagonal. Zero can never serve as a pivot, since we cannot use it to eliminate any nonzero entries in the column below it. What then to do? The answer requires revisiting the source of our algorithm.
Let us consider, as a specific example, the linear system

\begin{align*}
3y + z &= 2, \\
2x + 6y + z &= 7, \\
x + 4z &= 3.
\end{align*}

(1.26)

The augmented coefficient matrix is

\[
\begin{pmatrix}
0 & 3 & 1 & | & 2 \\
2 & 6 & 1 & | & 7 \\
1 & 0 & 4 & | & 3
\end{pmatrix}.
\]

In this case, the (1,1) entry is 0, and is not a legitimate pivot. The problem, of course, is that the first variable \( x \) does not appear in the first equation, and so we cannot use it to eliminate \( x \) in the other two equations. But this “problem” is actually a bonus — we already have an equation with only two variables in it, and so we only need to eliminate \( x \) from one of the other two equations. To be systematic, we rewrite the system in a different order,

\begin{align*}
2x + 6y + z &= 7, \\
3y + z &= 2, \\
x + 4z &= 3,
\end{align*}

by interchanging the first two equations. In other words, we employ

**Linear System Operation #2:** Interchange two equations.

Clearly this operation does not change the solution, and so produces an equivalent system. In our case, the resulting augmented coefficient matrix is

\[
\begin{pmatrix}
2 & 6 & 1 & | & 7 \\
0 & 3 & 1 & | & 2 \\
1 & 0 & 4 & | & 3
\end{pmatrix},
\]

and is obtained from the original by performing the second type of row operation:

**Elementary Row Operation #2:** Interchange two rows of the matrix.

The new nonzero upper left entry, 2, can now serve as the first pivot, and we may continue to apply elementary row operations of Type #1 to reduce our matrix to upper triangular form. For this particular example, we eliminate the remaining nonzero entry in the first column by subtracting \( \frac{1}{2} \) the first row from the last:

\[
\begin{pmatrix}
2 & 6 & 1 & | & 7 \\
0 & 3 & 1 & | & 2 \\
0 & -3 & \frac{7}{2} & | & -1 \frac{1}{2}
\end{pmatrix}.
\]

The (2,2) entry serves as the next pivot. To eliminate the nonzero entry below it, we add the second to the third row:

\[
\begin{pmatrix}
2 & 6 & 1 & | & 7 \\
0 & 3 & 1 & | & 2 \\
0 & 0 & \frac{9}{2} & | & \frac{3}{2}
\end{pmatrix}.
\]
We have now placed the system in upper triangular form, with the three pivots, \(2, 3, \frac{9}{2}\) along the diagonal. Back substitution produces the solution \(x = \frac{5}{3}, y = \frac{5}{9}, z = \frac{1}{3}\).

The row interchange that is required when a zero shows up on the diagonal in pivot position is known as pivoting. Later, in Section 1.7, we shall discuss practical reasons for pivoting even when a diagonal entry is nonzero. The coefficient matrices for which the Gaussian elimination algorithm with pivoting produces the solution are of fundamental importance.

**Definition 1.6.** A square matrix is called nonsingular if it can be reduced to upper triangular form with all non-zero elements on the diagonal by elementary row operations of Types 1 and 2. Conversely, a square matrix that cannot be reduced to upper triangular form because at some stage in the elimination procedure the diagonal entry and all the entries below it are zero is called singular.

Every regular matrix is nonsingular, but, as we just saw, nonsingular matrices are more general. Uniqueness of solutions is the key defining characteristic of nonsingularity.

**Theorem 1.7.** A linear system \(A \mathbf{x} = \mathbf{b}\) has a unique solution for every choice of right hand side \(\mathbf{b}\) if and only if its coefficient matrix \(A\) is square and nonsingular.

We are able to prove the “if” part of this theorem, since nonsingularity implies reduction to an equivalent upper triangular form that has the same solutions as the original system. The unique solution to the system is found by back substitution. The “only if” part will be proved in Section 1.8.

The revised version of the Gaussian Elimination algorithm, valid for all nonsingular coefficient matrices, is implemented by the accompanying program. The starting point is the augmented matrix \(M = (A | \mathbf{b})\) representing the linear system \(A \mathbf{x} = \mathbf{b}\). After successful termination of the program, the result is an augmented matrix in upper triangular form \(M = (U | \mathbf{c})\) representing the equivalent linear system \(U \mathbf{x} = \mathbf{c}\). One then uses Back Substitution to determine the solution \(\mathbf{x}\) to the linear system.
Permutation Matrices

As with the first type of elementary row operation, row interchanges can be accomplished by multiplication by a second type of elementary matrix. Again, the elementary matrix is found by applying the row operation in question to the identity matrix of the appropriate size. For instance, interchanging rows 1 and 2 of the $3 \times 3$ identity matrix produces the elementary interchange matrix

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$ 

As the reader can check, the effect of multiplying a 3 rowed matrix $A$ on the left by $P$, producing $PA$, is the same as interchanging the first two rows of $A$. For instance,

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \end{pmatrix}.$$

Multiple row interchanges are accomplished by combining such elementary interchange matrices. Each such combination of row interchanges corresponds to a unique permutation matrix.

Definition 1.8. A permutation matrix is a matrix obtained from the identity matrix by any combination of row interchanges.

In particular, applying a row interchange to a permutation matrix produces another permutation matrix. The following result is easily established.

Lemma 1.9. A matrix $P$ is a permutation matrix if and only if each row of $P$ contains all 0 entries except for a single 1, and, in addition, each column of $P$ also contains all 0 entries except for a single 1.

In general, if a permutation matrix $P$ has a 1 in position $(i, j)$, then the effect of multiplication by $P$ is to move the $j$th row of $A$ into the $i$th row of the product $PA$.

Example 1.10. There are six different $3 \times 3$ permutation matrices, namely

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$ (1.27)

These have the following effects: if $A$ is a matrix with row vectors $r_1, r_2, r_3$, then multiplication on the left by each of the six permutation matrices produces

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \begin{pmatrix} r_2 \\ r_1 \\ r_3 \end{pmatrix}, \begin{pmatrix} r_3 \\ r_2 \\ r_1 \end{pmatrix}, \begin{pmatrix} r_1 \\ r_3 \\ r_2 \end{pmatrix}, \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \begin{pmatrix} r_3 \\ r_1 \\ r_2 \end{pmatrix},$$

respectively. Thus, the first permutation matrix, which is the identity, does nothing. The second, third and fourth represent row interchanges. The last two are non-elementary permutations; each can be realized as a pair of row interchanges.
An elementary combinatorial argument proves that there are a total of
\[
  n! = n(n-1)(n-2) \cdots 3 \cdot 2 \cdot 1 \quad (1.28)
\]
different permutation matrices of size \(n \times n\). Moreover, the product \(P = P_1P_2\) of any two permutation matrices is also a permutation matrix. An important point is that multiplication of permutation matrices is \textit{noncommutative} — the order in which one permutes makes a difference. Switching the first and second rows, and then switching the second and third rows \textit{does not} have the same effect as first switching the second and third rows and then switching the first and second rows!

\section*{The Permuted \textit{LU} Factorization}

As we now know, any nonsingular matrix \(A\) can be reduced to upper triangular form by elementary row operations of types \#1 and \#2. The row interchanges merely reorder the equations. If one performs all of the required row interchanges in advance, then the elimination algorithm can proceed without requiring any further pivoting. Thus, the matrix obtained by permuting the rows of \(A\) in the prescribed manner is regular. In other words, if \(A\) is a nonsingular matrix, then there is a permutation matrix \(P\) such that the product \(PA\) is regular, and hence admits an \textit{LU} factorization. As a result, we deduce the general permuted \textit{LU} factorization

\[
  PA = LU, \quad (1.29)
\]
where \(P\) is a permutation matrix, \(L\) is special lower triangular, and \(U\) is upper triangular with the pivots on the diagonal. For instance, in the preceding example, we permuted the first and second rows, and hence equation (1.29) has the explicit form

\[
\begin{pmatrix}
  0 & 1 & 0 \\
  1 & 0 & 0 \\
  0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  0 & 2 & 1 \\
  2 & 6 & 1 \\
  1 & 1 & 4
\end{pmatrix}
= \begin{pmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  \frac{1}{2} & -1 & 1
\end{pmatrix}
\begin{pmatrix}
  2 & 6 & 1 \\
  0 & 2 & 1 \\
  0 & 0 & \frac{1}{2}
\end{pmatrix}. \quad (1.30)
\]

As a result of these considerations, we have established the following generalization of Theorem 1.3.

\section*{Theorem 1.11.} \textit{Let} \(A\) \textit{be an} \(n \times n\) \textit{matrix. Then the following conditions are equivalent:}
\begin{enumerate}[(i)]
  \item \(A\) is nonsingular.
  \item \(A\) has \(n\) nonzero pivots.
  \item \(A\) admits a permuted \textit{LU} factorization: \(PA = LU\).
\end{enumerate}

One should be aware of a couple of practical complications. First, to implement the permutation \(P\) of the rows that makes \(A\) regular, one needs to be clairvoyant: it is not always clear in advance when and where a required row interchange will crop up. Second, any row interchange performed during the course of the Gaussian Elimination algorithm will affect the lower triangular matrix \(L\), and precomputed entries must be permuted accordingly; an example appears in Exercise 1.

Once the permuted \textit{LU} factorization is established, the solution to the original system \(Ax = b\) is obtained by using the same Forward and Back Substitution algorithm presented
above. Explicitly, we first multiply the system \( A \mathbf{x} = \mathbf{b} \) by the permutation matrix, leading to
\[
P A \mathbf{x} = P \mathbf{b} \equiv \mathbf{\hat{b}},
\]
whose right hand side \( \mathbf{\hat{b}} \) has been obtained by permuting the entries of \( \mathbf{b} \) in the same fashion as the rows of \( A \). We then solve the two systems
\[
L \mathbf{c} = \mathbf{\hat{b}}, \quad \text{and} \quad U \mathbf{x} = \mathbf{c},
\]
by, respectively, Forward and Back Substitution as before.

**Example 1.12.** Suppose we wish to solve
\[
\begin{pmatrix}
0 & 2 & 1 \\
2 & 6 & 1 \\
1 & 1 & 4
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
=
\begin{pmatrix}
1 \\
-2 \\
0
\end{pmatrix}.
\]
In view of the \( PA = LU \) factorization established in (1.30), we need only solve the two auxiliary systems (1.32) by Forward and Back Substitution, respectively. The lower triangular system is
\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\frac{1}{2} & -1 & 1
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
=
\begin{pmatrix}
-2 \\
1 \\
0
\end{pmatrix} =
\begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
-2 \\
0
\end{pmatrix},
\]
with solution \( a = -2, \ b = 1, \ c = 2 \). The resulting upper triangular system is
\[
\begin{pmatrix}
2 & 6 & 1 \\
0 & 2 & 1 \\
0 & 0 & \frac{9}{2}
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
=
\begin{pmatrix}
-2 \\
1 \\
37
\end{pmatrix} =
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}.
\]
The solution, which is also the solution to the original system, is obtained by back substitution, with \( z = \frac{4}{9}, \ y = \frac{5}{18}, \ x = -\frac{37}{18} \).

### 1.5. Matrix Inverses.

The inverse of a matrix is analogous to the reciprocal \( a^{-1} = 1/a \) of a scalar, which is the \( 1 \times 1 \) case. We already introduced the inverses of matrices corresponding to elementary row operations. In this section, we will analyze inverses of general square matrices. We begin with the formal definition.

**Definition 1.13.** Let \( A \) be a square matrix of size \( n \times n \). An \( n \times n \) matrix \( X \) is called the inverse of \( A \) if it satisfies
\[
XA = I = AX,
\]
where \( I = I_n \) is the \( n \times n \) identity matrix. The inverse is commonly denoted by \( X = A^{-1} \).

**Remark:** Noncommutativity of matrix multiplication requires that we impose both conditions in (1.33) in order to properly define an inverse to the matrix \( A \). The first condition \( XA = I \) says that \( X \) is a left inverse, while the second \( AX = I \) requires that \( X \) also be a right inverse, in order that it fully qualify as a bona fide inverse of \( A \).
Example 1.14. Since
\[
\begin{pmatrix}
1 & 2 & -1 \\
3 & 4 & -1 \\
-2 & 2 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 \\
3 & 4 & -5 \\
-2 & 2 & 1
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 \\
3 & 4 & -5 \\
-2 & 2 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 2 & -1 \\
0 & 1 & 0 \\
4 & 6 & -7
\end{pmatrix},
\]
we conclude that when \(A = \begin{pmatrix}
1 & 2 & -1 \\
3 & 1 & 2 \\
-2 & 2 & 1
\end{pmatrix}\) then \(A^{-1} = \begin{pmatrix}
1 & 4 & -5 \\
1 & 1 & -1 \\
4 & 6 & -7
\end{pmatrix}\). Note that the entries of \(A^{-1}\) do not follow any easily discernable pattern in terms of the entries of \(A\).

Not every square matrix has an inverse. Indeed, not every scalar has an inverse — the one counterexample being \(a = 0\). There is no general concept of inverse for rectangular matrices.

Example 1.15. Let us compute the inverse \(X = \begin{pmatrix} x & y \\ z & w \end{pmatrix}\) of a general \(2 \times 2\) matrix \(A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\). The right inverse condition
\[
AX = \begin{pmatrix}
ax + bz & ay + bw \\
 cx + dz & cy + dw
\end{pmatrix} = \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix} = I
\]
holds if and only if \(x, y, z, w\) satisfy the linear system
\[
\begin{align*}
ax + bz &= 1, \\
ay + bw &= 0, \\
 cx + dz &= 0, \\
cy + dw &= 0.
\end{align*}
\]
Solving by Gaussian elimination (or directly), we find
\[
x = \frac{d}{ad - bc}, \quad y = -\frac{b}{ad - bc}, \quad z = -\frac{c}{ad - bc}, \quad w = \frac{a}{ad - bc},
\]
provided the common denominator \(ad - bc \neq 0\) does not vanish. Therefore, the matrix
\[
X = \frac{1}{ad - bc} \begin{pmatrix}
d & -b \\
-c & a
\end{pmatrix}
\]
forms a right inverse to \(A\). However, a short computation shows that it also defines a left inverse:
\[
XA = \begin{pmatrix}
ax + yc & xb + yd \\
 za + wc & zb + wd
\end{pmatrix} = \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix} = I,
\]
and hence \(X = A^{-1}\) is the inverse to \(A\).

The denominator appearing in the preceding formulae has a special name; it is called the determinant of the \(2 \times 2\) matrix \(A\), and denoted
\[
\det \begin{pmatrix}
a & b \\
c & d
\end{pmatrix} = ad - bc. \tag{1.34}
\]
Thus, the determinant of a $2 \times 2$ matrix is the product of the diagonal entries minus the product of the off-diagonal entries. (Determinants of larger square matrices will be discussed in Section 1.9.) Thus, the $2 \times 2$ matrix $A$ is invertible, with

$$A^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}, \quad (1.35)$$

if and only if $\det A \neq 0$. For example, if $A = \begin{pmatrix} 1 & 3 \\ -2 & 4 \end{pmatrix}$, then $\det A = 2 \neq 0$. We conclude that $A$ has an inverse, which, by (1.35), is

$$A^{-1} = \frac{1}{2} \begin{pmatrix} -4 & -3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} -2 & -\frac{3}{2} \\ 1 & \frac{1}{2} \end{pmatrix}.$$ 

The following key result will be established later in this chapter.

**Theorem 1.16.** A square matrix $A$ has an inverse if and only if it is nonsingular.

Consequently, an $n \times n$ matrix will have an inverse if and only if it can be reduced to upper triangular form with $n$ nonzero pivots on the diagonal by a combination of elementary row operations. Indeed, “invertible” is often used as a synonym for “nonsingular”. All other matrices are singular and do not have an inverse as defined above. Before attempting to prove this fundamental result, we need to first become familiar with some elementary properties of matrix inverses.

**Lemma 1.17.** The inverse of a square matrix, if it exists, is unique.

*Proof:* If $X$ and $Y$ both satisfy (1.33), so $XA = I = AX$ and $YA = I = AY$, then, by associativity, $X = XI = X(AY) = (XA)Y = 1Y = Y$, and hence $X = Y$. *Q.E.D.*

Inverting a matrix twice gets us back to where we started.

**Lemma 1.18.** If $A$ is invertible, then $A^{-1}$ is also invertible and $(A^{-1})^{-1} = A$.

*Proof:* The matrix inverse equations $A^{-1}A = I = AA^{-1}$, are sufficient to prove that $A$ is the inverse of $A^{-1}$. *Q.E.D.*

**Example 1.19.** We already learned how to find the inverse of an elementary matrix of type #1; we just negate the one nonzero off-diagonal entry. For example, if

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}, \quad \text{then} \quad E^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}.$$ 

This reflects the fact that the inverse of the elementary row operation that adds twice the first row to the third row is the operation of subtracting twice the first row from the third row.

**Example 1.20.** Let $P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ denote the elementary matrix that has the effect of interchanging rows 1 and 2 of a matrix. Then $P^2 = I$, since doing the same
operation twice in a row has no net effect. This implies that \( P^{-1} = P \) is its own inverse. Indeed, the same result holds for all elementary permutation matrices that correspond to row operations of type \( \#2 \). However, it is not true for more general permutation matrices.

**Lemma 1.21.** If \( A \) and \( B \) are invertible matrices of the same size, then their product, \( AB \), is invertible, and

\[
(AB)^{-1} = B^{-1}A^{-1}.
\]

(1.36)

Note particularly the reversal in order of the factors.

**Proof:** Let \( X = B^{-1}A^{-1} \). Then, by associativity,

\[
X(AB) = B^{-1}A^{-1}AB = B^{-1}B = I, \quad (AB)X = ABB^{-1}A^{-1} = AA^{-1} = I.
\]

Thus \( X \) is both a left and a right inverse for the product matrix \( AB \) and the result follows. Q.E.D.

**Example 1.22.** One verifies, directly, that the inverse of \( A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \) is \( A^{-1} = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \), while the inverse of \( B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \) is \( B^{-1} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \). Therefore, the inverse of their product \( C = AB = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ -1 & 0 \end{pmatrix} \) is given by \( C^{-1} = B^{-1}A^{-1} = \begin{pmatrix} 0 & -1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & -2 \end{pmatrix} \).

We can straightforwardly generalize the preceding result. The inverse of a multiple product of invertible matrices is the product of their inverses, *in the reverse order*:

\[
(A_1A_2 \cdots A_{m-1}A_m)^{-1} = A_m^{-1}A_{m-1}^{-1} \cdots A_2^{-1}A_1^{-1}.
\]

(1.37)

**Warning:** In general, \((A + B)^{-1} \neq A^{-1} + B^{-1}\). This equation is not even true for scalars \((1 \times 1\) matrices)!

**Gauss–Jordan Elimination**

The basic algorithm used to compute the inverse of a square matrix is known as *Gauss–Jordan Elimination*, in honor of Gauss and Wilhelm Jordan, a nineteenth century German engineer. A key fact is that we only need to solve the right inverse equation

\[
AX = I
\]

(1.38)
in order to compute \( X = A^{-1} \). The other equation in (1.33), namely \(XA = I\), will then follow as an automatic consequence. In other words, for square matrices, a right inverse is automatically a left inverse, and conversely! A proof will appear below.

The reader may well ask, then, why use both left and right inverse conditions in the original definition? There are several good reasons. First of all, a rectangular matrix may satisfy one of the two conditions — having either a left inverse or a right inverse — but can never satisfy both. Moreover, even when we restrict our attention to square
matrices, starting with only one of the conditions makes the logical development of the subject considerably more difficult, and not really worth the extra effort. Once we have established the basic properties of the inverse of a square matrix, we can then safely discard the superfluous left inverse condition. Finally, when we generalize the notion of an inverse to a linear operator in Chapter 7, then, unlike square matrices, we cannot dispense with either of the conditions.

Let us write out the individual columns of the right inverse equation (1.38). The $i^{th}$ column of the $n \times n$ identity matrix $I$ is the vector $e_i$ that has a single 1 in the $i^{th}$ slot and 0’s elsewhere, so

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \end{pmatrix}, \quad \ldots \quad e_n = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \end{pmatrix}. \quad (1.39)$$

According to (1.11), the $i^{th}$ column of the matrix product $AX$ is equal to $Ax_i$, where $x_i$ denotes the $i^{th}$ column of $X = (x_1 \ x_2 \ldots \ x_n)$. Therefore, the single matrix equation (1.38) is equivalent to $n$ linear systems

$$Ax_1 = e_1, \quad Ax_2 = e_2, \quad \ldots \quad Ax_n = e_n, \quad (1.40)$$

all having the same coefficient matrix. As such, to solve them we are led to form the $n$ augmented matrices $M_1 = (A \ | \ e_1), \ldots, M_n = (A \ | \ e_n)$, and then perform our Gaussian elimination algorithm on each one. But this would be a waste of effort. Since the coefficient matrix is the same, we will end up performing identical row operations on each augmented matrix. Consequently, it will be more efficient to combine them into one large augmented matrix $M = (A \ | \ e_1 \ldots e_n) = (A \ | \ I)$, of size $n \times (2n)$, in which the right hand sides $e_1, \ldots, e_n$ of our systems are placed into $n$ different columns, which we then recognize as reassembling the columns of an $n \times n$ identity matrix. We may then apply our elementary row operations to reduce, if possible, the large augmented matrix so that its first $n$ columns are in upper triangular form.

**Example 1.23.** For example, to find the inverse of the matrix $A = \begin{pmatrix} 0 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \\ \end{pmatrix}$, we form the large augmented matrix

$$
\begin{pmatrix}
0 & 2 & 1 & 1 & 0 & 0 \\
2 & 6 & 1 & 0 & 1 & 0 \\
1 & 1 & 4 & 0 & 0 & 1
\end{pmatrix}.
$$

Applying the same sequence of elementary row operations as in Section 1.4, we first interchange the rows

$$
\begin{pmatrix}
2 & 6 & 1 & 0 & 1 & 0 \\
0 & 2 & 1 & 1 & 0 & 0 \\
1 & 1 & 4 & 0 & 0 & 1
\end{pmatrix}.
$$
and then eliminate the nonzero entries below the first pivot,
\[
\begin{pmatrix}
  2 & 6 & 1 & | & 0 & 1 & 0 \\
  0 & 2 & 1 & | & 1 & 0 & 0 \\
  0 & -2 & \frac{7}{2} & | & 0 & -\frac{1}{2} & 1
\end{pmatrix}.
\]

Next we eliminate the entry below the second pivot:
\[
\begin{pmatrix}
  2 & 6 & 1 & | & 0 & 1 & 0 \\
  0 & 2 & 1 & | & 1 & 0 & 0 \\
  0 & 0 & \frac{9}{2} & | & 1 & -\frac{1}{2} & 1
\end{pmatrix}.
\]

At this stage, we have reduced our augmented matrix to the upper triangular form \((U | C)\), which is equivalent to reducing the original \(n\) linear systems \(Ax_i = e_i\) to \(n\) upper triangular systems \(Ux_i = c_i\). We could therefore perform \(n\) back substitutions to produce the solutions \(x_i\), which would form the individual columns of the inverse matrix \(X = (x_1 \ldots x_n)\).

In the standard Gauss-Jordan scheme, one instead continues to employ the usual sequence of elementary row operations to fully reduce the augmented matrix to the form \((I | X)\) in which the left hand \(n \times n\) matrix has become the identity, while the right hand matrix is the desired solution \(X = A^{-1}\). Indeed, \((I | X)\) represents the \(n\) trivial, but equivalent, linear systems \(Ix_i = x_i\) with identity coefficient matrix.

Now, the identity matrix has 0’s below the diagonal, just like \(U\). It also has 1’s along the diagonal, whereas \(U\) has the pivots (which are all nonzero) along the diagonal. Thus, the next phase in the procedure is to make all the diagonal entries of \(U\) equal to 1. To do this, we need to introduce the last, and least, of our linear systems operations.

**Linear System Operation #3:** Multiply an equation by a nonzero constant.

This operation does not change the solution, and so yields an equivalent linear system. The corresponding elementary row operation is:

**Elementary Row Operation #3:** Multiply a row of the matrix by a nonzero scalar.

Dividing the rows of the upper triangular augmented matrix \((U | C)\) by the diagonal pivots of \(U\) will produce a matrix of the form \((V | K)\) where \(V\) is *special upper triangular*, meaning it has all 1’s along the diagonal. In the particular example, the result of these three elementary row operations of Type #3 is
\[
\begin{pmatrix}
  1 & 3 & \frac{1}{2} & | & 0 & \frac{1}{2} & 0 \\
  0 & 1 & \frac{1}{2} & | & \frac{1}{2} & 0 & 0 \\
  0 & 0 & 1 & | & \frac{2}{9} & -\frac{1}{9} & \frac{2}{9}
\end{pmatrix},
\]
where we multiplied the first and second rows by \(\frac{1}{2}\) and the third row by \(\frac{2}{5}\).

We are now over half way towards our goal of an identity matrix on the left. We need only make the entries above the diagonal equal to zero. This can be done by elementary row operations of Type #1, but now we work backwards as in back substitution. First,
eliminate the nonzero entries in the third column lying above the (3, 3) entry; this is done by subtracting one half the third row from the second and also from the first:
\[
\begin{pmatrix}
1 & 3 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
- \frac{1}{2}
\begin{pmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 2 & 9
\end{pmatrix}
= \begin{pmatrix}
1 & 3 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
- \frac{1}{2}
\begin{pmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 2 & 9
\end{pmatrix}.
\]
Finally, subtract \( \frac{1}{3} \) the second from the first to eliminate the remaining nonzero off-diagonal entry:
\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
- \frac{1}{3}
\begin{pmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 2 & 9
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
- \frac{1}{3}
\begin{pmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 2 & 9
\end{pmatrix}.
\]
The final right hand matrix is our desired inverse:
\[
A^{-1} = \begin{pmatrix}
-\frac{23}{18} & \frac{7}{18} & \frac{2}{9} \\
\frac{7}{18} & \frac{1}{18} & -\frac{1}{9} \\
\frac{2}{9} & -\frac{1}{9} & \frac{2}{9}
\end{pmatrix},
\]
thereby completing the Gauss–Jordan procedure. The reader may wish to verify that the final result does satisfy both inverse conditions \( AA^{-1} = I = A^{-1}A \).

We are now able to complete the proofs of the basic results on inverse matrices. First, we need to determine the elementary matrix corresponding to an elementary row operation of type #3. Again, this is obtained by performing the indicated elementary row operation on the identity matrix. Thus, the elementary matrix that multiplies row \( i \) by the nonzero scalar \( c \neq 0 \) is the diagonal matrix having \( c \) in the \( i \)th diagonal position, and 1’s elsewhere along the diagonal. The inverse elementary matrix is the diagonal matrix with \( 1/c \) in the \( i \)th diagonal position and 1’s elsewhere on the main diagonal; it corresponds to the inverse operation that divides row \( i \) by \( c \). For example, the elementary matrix that multiplies the second row of a \( 3 \times n \) matrix by the scalar 5 is
\[
E = \begin{pmatrix}
1 & 0 & 0 \\
0 & 5 & 0 \\
0 & 0 & 1
\end{pmatrix},
\]
and has inverse
\[
E^{-1} = \begin{pmatrix}
1 & 0 & 0 \\
0 & \frac{1}{5} & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

The Gauss–Jordan method tells us how to reduce any nonsingular square matrix \( A \) to the identity matrix by a sequence of elementary row operations. Let \( E_1, E_2, \ldots, E_N \) be the corresponding elementary matrices. Therefore,
\[
E_N E_{N-1} \cdots E_2 E_1 A = I.
\] (1.41)
We claim that the matrix product
\[
X = E_N E_{N-1} \cdots E_2 E_1
\] (1.42)
is the inverse of \( A \). Indeed, formula (1.41) says that \( X A = I \), and so \( X \) is a left inverse. Furthermore, each elementary matrix has an inverse, and so by (1.37), \( X \) itself is invertible, with
\[
X^{-1} = E_1^{-1} E_2^{-1} \cdots E_{N-1}^{-1} E_N^{-1}.
\] (1.43)
Therefore, multiplying the already established formula $XA = I$ on the left by $X^{-1}$, we find $A = X^{-1}$, and so, by Lemma 1.18, $X = A^{-1}$ as claimed. This completes the proof of Theorem 1.16. Finally, equating $A = X^{-1}$ to (1.43), and using the fact that the inverse of an elementary matrix is also an elementary matrix, we have established:

**Proposition 1.24.** Any nonsingular matrix $A$ can be written as the product of elementary matrices.

For example, the $2 \times 2$ matrix $A = \begin{pmatrix} 0 & -1 \\ 1 & 3 \end{pmatrix}$ is converted into the identity matrix by row operations corresponding to the matrices $E_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, corresponding to a row interchange, $E_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, scaling the second row by $-1$, and $E_3 = \begin{pmatrix} 1 & -3 \\ 0 & 1 \end{pmatrix}$ that subtracts 3 times the second row from the first. Therefore,

$$A^{-1} = E_3 E_2 E_1 = \begin{pmatrix} 1 & -3 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ -1 & 0 \end{pmatrix},$$

while

$$A = E_1^{-1} E_2^{-1} E_3^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 3 \end{pmatrix}.$$

As an application, let us prove that the inverse of a nonsingular triangular matrix is also triangular. Specifically:

**Lemma 1.25.** If $L$ is a lower triangular matrix with all nonzero entries on the main diagonal, then $L$ is nonsingular and its inverse $L^{-1}$ is also lower triangular. In particular, if $L$ is special lower triangular, so is $L^{-1}$. A similar result holds for upper triangular matrices.

**Proof:** It suffices to note that if $L$ has all nonzero diagonal entries, one can reduce $L$ to the identity by elementary row operations of Types #1 and #3, whose associated elementary matrices are all lower triangular. Lemma 1.2 implies that the product $\left(1.42\right)$ is then also lower triangular. If $L$ is special, then all the pivots are equal to 1 and so no elementary row operations of Type #3 are required, so the inverse is a product of special lower triangular matrices, and hence is special lower triangular. 

Q.E.D.

**Solving Linear Systems with the Inverse**

An important motivation for the matrix inverse is that it enables one to effect an immediate solution to a nonsingular linear system.

**Theorem 1.26.** If $A$ is invertible, then the unique solution to the linear system $Ax = b$ is given by $x = A^{-1}b$.

**Proof:** We merely multiply the system by $A^{-1}$, which yields $x = A^{-1}Ax = A^{-1}b$, as claimed.

Q.E.D.
Thus, with the inverse in hand, a “more direct” way to solve our example (1.26) is to multiply the right hand side by the inverse matrix:

\[
\begin{pmatrix}
  x \\
  y \\
  z
\end{pmatrix} = \begin{pmatrix}
  -\frac{23}{18} & \frac{7}{18} & \frac{2}{9} \\
  \frac{7}{18} & \frac{1}{18} & -\frac{1}{9} \\
  \frac{2}{9} & -\frac{1}{9} & \frac{2}{9}
\end{pmatrix}\begin{pmatrix}
  2 \\
  7 \\
  3
\end{pmatrix} = \begin{pmatrix}
  \frac{5}{6} \\
  \frac{5}{6} \\
  \frac{1}{3}
\end{pmatrix},
\]

reproducing our earlier solution.

However, while aesthetically appealing, the solution method based on the inverse matrix is hopelessly inefficient as compared to forward and back substitution based on a (permuted) LU factorization, and should not be used. A complete justification of this dictum will be provided in Section 1.7. In contrast to what you might have learned in an introductory linear algebra course, you should never use the matrix inverse for practical computations! This is not to say that the inverse is completely without merit. Far from it! The inverse continues to play a fundamental role in the theoretical side of linear algebra, as well as providing important insight into the algorithms that are used in practice. But the basic message of practical, applied linear algebra is that LU decomposition and Gaussian Elimination are fundamental; inverses are only used for theoretical purposes, and are to be avoided in all but the most elementary practical computations.

**Remark:** The reader may have learned a version of the Gauss–Jordan algorithm for solving a single linear system that replaces the back substitution step by a further application of all three types of elementary row operations in order to reduce the coefficient matrix to the identity. In other words, to solve \( Ax = b \), we start with the augmented matrix \( M = ( A \mid b ) \) and use all three types of elementary row operations to produce (assuming nonsingularity) the fully reduced form \( ( I \mid x ) \), representing the trivial, equivalent system \( I x = x \), with the solution \( x \) to the original system in its final column. However, as we shall see, back substitution is much more efficient, and is the method of choice in all practical situations.

**The LDV Factorization**

The Gauss–Jordan construction leads to a slightly more detailed version of the LU factorization, which is useful in certain situations. Let \( D \) denote the diagonal matrix having the same diagonal entries as \( U \); in other words, \( D \) has the pivots on its diagonal and zeros everywhere else. Let \( V \) be the special upper triangular matrix obtained from \( U \) by dividing each row by its pivot, so that \( V \) has all 1’s on the diagonal. We already encountered \( V \) during the course of the Gauss–Jordan method. It is easily seen that \( U = DV \), which implies the following result.

**Theorem 1.27.** A matrix \( A \) is regular if and only if it admits a factorization

\[
A = LDV,
\]

where \( L \) is special lower triangular matrix, \( D \) is a diagonal matrix having the nonzero pivots on the diagonal, and \( V \) is special upper triangular.
For the matrix appearing in Example 1.5, we have \( U = DV \), where

\[
U = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad V = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},
\]

producing the \( A = LDV \) factorization

\[
\begin{pmatrix} 2 & 1 & 1 \\ 4 & 5 & 2 \\ 2 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\]

**Proposition 1.28.** If \( A = LU \) is regular, then the factors \( L \) and \( U \) are each uniquely determined. The same holds for its \( A = LDV \) factorization.

**Proof:** Suppose \( LU = \widetilde{L} \widetilde{U} \). Since the diagonal entries of all four matrices are non-zero, Lemma 1.25 implies that they are invertible. Therefore,

\[
\widetilde{L}^{-1}L = \widetilde{L}^{-1}LUU^{-1} = \widetilde{L}^{-1}\widetilde{L}\widetilde{U}U^{-1} = \widetilde{U}U^{-1}.
\] (1.45)

The left hand side of the matrix equation (1.45) is the product of two special lower triangular matrices, and so, according to Lemma 1.2, is itself special lower triangular — with 1’s on the diagonal. The right hand side is the product of two upper triangular matrices, and hence is itself upper triangular. Comparing the individual entries, the only way such a special lower triangular matrix could equal an upper triangular matrix is if they both equal the diagonal identity matrix. Therefore, \( \widetilde{L}^{-1}L = 1 = \widetilde{U}U^{-1} \), which implies that \( \widetilde{L} = L \) and \( \widetilde{U} = U \), and proves the result. The \( LDV \) version is an immediate consequence. Q.E.D.

As you may have guessed, the more general cases requiring one or more row interchanges lead to a permuted \( LDV \) factorization in the following form.

**Theorem 1.29.** A matrix \( A \) is nonsingular if and only if there is a permutation matrix \( P \) such that

\[
P A = LDV,
\] (1.46)

where \( L, D, V \) are as before.

Uniqueness does not hold for the more general permuted factorizations (1.29), (1.46) since there may be various permutation matrices that place a matrix \( A \) in regular form \( PA \); see Exercise \( \text{II} \) for an explicit example. Moreover, unlike the regular case, the pivots, i.e., the diagonal entries of \( U \), are no longer uniquely defined, but depend on the particular combination of row interchanges employed during the course of the computation.

### 1.6. Transposes and Symmetric Matrices.

Another basic operation on a matrix is to interchange its rows and columns. If \( A \) is an \( m \times n \) matrix, then its transpose, denoted \( A^T \), is the \( n \times m \) matrix whose \((i,j)\) entry equals the \((j,i)\) entry of \( A \); thus

\[
B = A^T \quad \text{means that} \quad b_{ij} = a_{ji}.
\]
For example, if

\[ A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad \text{then} \quad A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}. \]

Note that the rows of \( A \) are the columns of \( A^T \) and vice versa. In particular, the transpose of a row vector is a column vector, while the transpose of a column vector is a row vector.

For example, if \( \mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \), then \( \mathbf{v}^T = (1 \ 2 \ 3) \).

The transpose of a scalar, considered as a \( 1 \times 1 \) matrix, is itself: \( c^T = c \) for \( c \in \mathbb{R} \).

Remark: Most vectors appearing in applied mathematics are column vectors. To conserve vertical space in this text, we will often use the transpose notation, e.g., \( \mathbf{v} = (v_1 \ v_2 \ v_3)^T \), as a compact way of writing column vectors.

In the square case, transpose can be viewed as “reflecting” the matrix entries across the main diagonal. For example,

\[
\begin{pmatrix}
1 & 2 & -1 \\
3 & 0 & 5 \\
-2 & -4 & 8
\end{pmatrix}^T = \begin{pmatrix} 1 & 3 & -2 \\
2 & 0 & -4 \\
-1 & 5 & 8
\end{pmatrix}.
\]

In particular, the transpose of a lower triangular matrix is upper triangular and vice-versa.

Performing the transpose twice gets you back to where you started:

\[(A^T)^T = A. \quad (1.47)\]

Unlike the inverse, the transpose is compatible with matrix addition and scalar multiplication:

\[(A + B)^T = A^T + B^T, \quad (cA)^T = cA^T. \quad (1.48)\]

The transpose is also compatible with matrix multiplication, but with a twist. Like the inverse, the transpose reverses the order of multiplication:

\[(AB)^T = B^T A^T. \quad (1.49)\]

The proof of (1.49) is a straightforward consequence of the basic laws of matrix multiplication. An important special case is the product between a row vector \( \mathbf{v}^T \) and a column vector \( \mathbf{w} \). In this case,

\[\mathbf{v}^T \mathbf{w} = (\mathbf{v}^T \mathbf{w})^T = \mathbf{w}^T \mathbf{v}, \quad (1.50)\]

because the product is a scalar and so equals its own transpose.

**Lemma 1.30.** The operations of transpose and inverse commute. In other words, if \( A \) is invertible, so is \( A^T \), and its inverse is

\[A^{-T} \equiv (A^T)^{-1} = (A^{-1})^T. \quad (1.51)\]
Proof: Let \( Y = (A^{-1})^T \). Then, according to (1.49),
\[
Y A^T = (A^{-1})^T A^T = (A A^{-1})^T = I^T = I.
\]
The proof that \( A^T Y = I \) is similar, and so we conclude that \( Y = (A^T)^{-1} \). Q.E.D.

**Factorization of Symmetric Matrices**

The most important class of square matrices are those that are unchanged by the transpose operation.

**Definition 1.31.** A square matrix is *symmetric* if it equals its own transpose: \( A = A^T \).

Thus, \( A \) is symmetric if and only if its entries satisfy \( a_{ji} = a_{ij} \) for all \( i, j \). In other words, entries lying in “mirror image” positions relative to the main diagonal must be equal. For example, the most general symmetric \( 3 \times 3 \) matrix has the form

\[
A = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}.
\]

Note that any diagonal matrix, including the identity, is symmetric. A lower or upper triangular matrix is symmetric if and only if it is, in fact, a diagonal matrix.

The \( LDV \) factorization of a nonsingular matrix takes a particularly simple form if the matrix also happens to be symmetric. This result will form the foundation of some significant later developments.

**Theorem 1.32.** A symmetric matrix \( A \) is regular if and only if it can be factored as
\[
A = LDL^T, \tag{1.52}
\]
where \( L \) is a special lower triangular matrix and \( D \) is a diagonal matrix with nonzero diagonal entries.

Proof: We already know, according to Theorem 1.27, that we can factorize
\[
A = LDV, \tag{1.53}
\]
We take the transpose of both sides of this equation and use the fact that the transpose of a matrix product is the product of the transposes in the reverse order, whence
\[
A^T = (LDV)^T = V^T D^T L^T = V^T D L^T, \tag{1.54}
\]
where we used the fact that a diagonal matrix is automatically symmetric, \( D^T = D \). Note that \( V^T \) is special lower triangular, and \( L^T \) is special upper triangular. Therefore (1.54) gives the \( LDV \) factorization of \( A^T \).

In particular, if \( A = A^T \), then we can invoke the uniqueness of the \( LDV \) factorization, cf. Proposition 1.28, to conclude that \( L = V^T \), and \( V = L^T \), (which are two versions of the same equation). Replacing \( V \) by \( L^T \) in (1.53) proves the factorization (1.52). Q.E.D.
Example 1.33. Let us find the $LDL^T$ factorization of the particular symmetric matrix $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix}$. This is done by performing the usual Gaussian elimination algorithm. Subtracting twice the first row from the second and also the first row from the third produces the matrix \( \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix} \). We then add one half of the second row of the latter matrix to its third row, resulting in the upper triangular form
\[
U = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & \frac{5}{2} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & \frac{5}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & 1 \end{pmatrix} = DV,
\]
which we further factorize by dividing each row of $U$ by its pivot. On the other hand, the special lower triangular matrix associated with the row operations is $L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -\frac{1}{2} & 1 \end{pmatrix}$, which, as guaranteed by Theorem 1.32, is the transpose of $V = L^T$. Therefore, the desired $A = LU = LDL^T$ factorizations of this particular symmetric matrix are
\[
\begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & \frac{5}{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & 1 \end{pmatrix} = DL^T.
\]

Example 1.34. Let us look at a general $2 \times 2$ symmetric matrix
\[
A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}.
\]
Regularity requires that the first pivot be $a \neq 0$. A single row operation will place $A$ in upper triangular form $U = \begin{pmatrix} a & c \\ 0 & \frac{ac-b^2}{a} \end{pmatrix}$. The associated lower triangular matrix is $L = \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix}$. Thus, $A = LU$. Finally, $D = \begin{pmatrix} a & 0 \\ 0 & \frac{ac-b^2}{a} \end{pmatrix}$ is just the diagonal part of $U$, and we find $U = DL^T$, so that the $LDL^T$ factorization is explicitly given by
\[
\begin{pmatrix} a & b \\ b & c \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} a & 0 \\ 0 & \frac{ac-b^2}{a} \end{pmatrix} \begin{pmatrix} 1 & b \\ 0 & a \end{pmatrix}.
\]

Remark: If $A = LDL^T$, then $A$ is necessarily symmetric. Indeed,
\[
A^T = (LDL^T)^T = (L^T)^T D^T L^T = LD L^T = A.
\]
However, not every symmetric matrix has an $LDL^T$ factorization. A simple example is the irregular but invertible $2 \times 2$ matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.
1.7. Practical Linear Algebra.

For pedagogical reasons, the examples and exercises that have been used to illustrate the algorithms are all based on rather small $(2 \times 2$ or $3 \times 3)$ matrices. In such cases, or even for matrices of moderate size, the differences between the various approaches to solving linear systems (Gauss, Gauss-Jordan, matrix inverse, etc.) are relatively unimportant, particularly if one has a decent computer or even hand calculator to perform the tedious parts. However, real-world applied mathematics deals with much larger linear systems, and the design of efficient algorithms is critical. For example, numerical solutions of ordinary differential equations will typically lead to matrices with hundreds or thousands of entries, while numerical solution of partial differential equations arising in fluid and solid mechanics, weather prediction, image and video processing, chemical reactions, quantum mechanics, molecular dynamics, and many other areas will often lead to matrices with millions of entries. It is not hard for such systems to tax even the most sophisticated supercomputer. Thus, it is essential that we look into the computational details of competing algorithms in order to compare their efficiency, and thereby gain some experience with the issues underlying the design of high performance numerical algorithms.

The most basic question is: how many arithmetic operations are required for each of our algorithms? We shall keep track of additions and multiplications separately, since the latter typically take slightly longer to perform in a computer processor. However, we shall not distinguish between addition and subtraction, nor between multiplication and division, as these typically rely on the same floating point algorithm. We shall also assume that the matrices and vectors are generic, with few, if any, zero entries. Modifications of the basic algorithms for sparse matrices, meaning those that have lots of zero entries, are an important topic of research, since these include many of the large matrices that appear in applications to differential equations. We refer the interested reader to more advanced numerical linear algebra texts, e.g., [121, 119], for further developments.

First, for ordinary multiplication of an $n \times n$ matrix $A$ and a vector $b$, each entry of the product $Ab$ requires $n$ multiplications of the form $a_{ij}b_j$ and $n - 1$ additions to sum the resulting products. Since there are $n$ entries, this means a total of $n^2$ multiplications and $n(n - 1) = n^2 - n$ additions. Thus, for a matrix of size $n = 100$, one needs about 10,000 distinct multiplications and a similar (but slightly fewer) number of additions. If $n = 1,000,000 = 10^6$ then $n^2 = 10^{12}$, which is phenomenally large, and the total time required to perform the computation becomes a significant issue.

Let us look at the regular Gaussian Elimination algorithm, referring back to our program. First, we count how many arithmetic operations are based on the $j^{th}$ pivot $m_{jj}$. For each of the $n - j$ rows lying below it, we must perform one division to compute the factor $l_{ij} = m_{ij}/m_{jj}$ used in the elementary row operation. The entries in the column below the pivot will be made equal to zero automatically, and so we need only compute the updated entries lying below and to the right of the pivot. There are $(n - j)^2$ such entries in the coefficient matrix and an additional $n - j$ entries in the last column of the augmented matrix. Let us concentrate on the former for the moment. For each of these, we replace

\[ \text{See [31] for more sophisticated computational methods to speed up matrix multiplication.} \]
$m_{ik}$ by $m_{ik} - l_{ij} m_{jk}$, and so must perform one multiplication and one addition. Therefore, for the $j$th pivot there are a total of $(n - j)(n - j + 1)$ multiplications — including the initial $n - j$ divisions needed to produce the $l_{ij}$ — and $(n - j)^2$ additions needed to update the coefficient matrix. Therefore, to reduce a regular $n \times n$ matrix to upper triangular form requires a total of

$$
\sum_{j=1}^{n} (n - j)(n - j + 1) = \frac{n^3 - n}{3} \text{ multiplications, and}
$$

$$
\sum_{j=1}^{n} (n - j)^2 = \frac{2n^3 - 3n^2 + n}{6} \text{ additions.}
$$

Thus, when $n$ is large, both require approximately $\frac{1}{3} n^3$ operations.

We should also be keeping track of the number of operations on the right hand side of the system. No pivots appear there, and so there are

$$
\sum_{j=1}^{n} (n - j) = \frac{n^2 - n}{2}
$$

multiplications and the same number of additions required to produce the right hand side in the resulting triangular system $U \mathbf{x} = \mathbf{c}$. For large $n$, this number is considerably smaller than the coefficient matrix totals (1.57), (1.58).

The next phase of the algorithm can be similarly analyzed. To find the value of

$$
x_j = \frac{1}{u_{jj}} \left( c_j - \sum_{i=j+1}^{n} u_{ji} x_i \right)
$$

once we have computed $x_{j+1}, \ldots, x_n$, requires $n - j + 1$ multiplications/divisions and $n - j$ additions. Therefore, the Back Substitution phase of the algorithm requires

$$
\sum_{j=1}^{n} (n - j + 1) = \frac{n^2 + n}{2} \text{ multiplications, and}
\sum_{j=1}^{n} (n - j) = \frac{n^2 - n}{2} \text{ additions.}
$$

For $n$ large, both of these are approximately equal to $\frac{1}{2} n^2$. Comparing these results, we conclude that the bulk of the computational effort goes into the reduction of the coefficient matrix to upper triangular form.

Forward substitution, to solve $L \mathbf{c} = \mathbf{b}$, has the same operations count, except that since the diagonal entries of $L$ are all equal to 1, no divisions are required, and so we use a total of $\frac{1}{2}(n^2 - n)$ multiplications and the same number of additions. Thus, once we have computed the $LU$ decomposition of the matrix $A$, the Forward and Back Substitution process requires about $n^2$ arithmetic operations of the two types, which is the same as the

---

\* In Exercise the reader is asked to prove these summation formulae by induction.
number of operations needed to perform the matrix multiplication $A^{-1}b$. Thus, even if we know the inverse of the coefficient matrix $A$, it is still just as efficient to use Forward and Back Substitution to compute the solution!

As noted above, the computation of $L$ and $U$ requires about $\frac{1}{3} n^3$ arithmetic operations of each type. On the other hand, to complete the full-blown Gauss–Jordan elimination scheme, we must perform all the elementary row operations on the large augmented matrix, which has size $n \times 2n$. Therefore, during the reduction to upper triangular form, there are an additional $\frac{1}{3} n^3$ operations of each type required. Moreover, we then need to perform an additional $\frac{1}{3} n^3$ operations to reduce $U$ to the identity matrix, and a corresponding $\frac{1}{2} n^3$ operations on the right hand matrix, too. (All these are approximate totals, based on the leading term in the actual count.) Therefore, Gauss–Jordan requires a grand total of $\frac{5}{3} n^3$ operations to complete, just to find $A^{-1}$; multiplying the right hand side to obtain the solution $\mathbf{x} = A^{-1}\mathbf{b}$ involves another $n^2$ operations. Thus, the Gauss–Jordan method requires approximately five times as many arithmetic operations, and so would take five times as long to complete, as compared to the more elementary Gaussian Elimination and Back Substitution algorithm. These observations serve to justify our earlier contention that matrix inversion is inefficient, and should never be used to solve linear systems in practice.

**Tridiagonal Matrices**

Of course, in special cases, the arithmetic operation count might be considerably reduced, particularly if $A$ is a sparse matrix with many zero entries. A number of specialized techniques have been designed to handle such sparse linear systems. A particularly important class are the tridiagonal matrices

$$ A = \begin{pmatrix}
q_1 & r_1 \\
p_1 & q_2 & r_2 \\
p_2 & q_3 & r_3 \\
\vdots & \ddots & \ddots \\
p_{n-2} & q_{n-1} & r_{n-1} \\
p_{n-1} & q_n
\end{pmatrix} \quad (1.61) $$

with all entries zero except for those on the main diagonal, $a_{i,i} = q_i$, on the subdiagonal, meaning the $n - 1$ entries $a_{i+1,i} = p_i$ immediately below the main diagonal, and the superdiagonal, meaning the entries $a_{i,i+1} = r_i$ immediately above the main diagonal. (Zero entries are left blank.) Such matrices arise in the numerical solution of ordinary differential equations and the spline fitting of curves for interpolation and computer graphics. If
We then solve the tridiagonal linear system of the diagonal entries.

Multiplying out \( LU \), and equating the result to \( A \) leads to the equations

\[
\begin{align*}
    d_1 &= q_1, \\
    l_1 u_1 + d_2 &= q_2, \\
    &\vdots \\
    l_{j-1} u_{j-1} + d_{j} &= q_{j}, \\
    &\vdots \\
    l_{n-2} u_{n-2} + d_{n-1} &= q_{n-1}, \\
    l_{n-1} u_{n-1} + d_{n} &= q_{n}.
\end{align*}
\]

These elementary algebraic equations can be successively solved for the entries of \( L \) and \( U \) in the order \( d_1, u_1, l_1, d_2, u_2, l_2, d_3, u_3 \ldots \). The original matrix \( A \) is regular provided none of the diagonal entries \( d_1, d_2, \ldots \) are zero, which allows the recursive procedure to proceed.

Once the \( LU \) factors are in place, we can apply Forward and Back Substitution to solve the tridiagonal linear system \( A \mathbf{x} = \mathbf{b} \). We first solve \( L \mathbf{c} = \mathbf{b} \) by Forward Substitution, which leads to the recursive equations

\[
\begin{align*}
    c_1 &= b_1, \\
    c_2 &= b_2 - l_1 c_1, \\
    &\vdots \\
    c_n &= b_n - l_{n-1} c_{n-1}.
\end{align*}
\]

We then solve \( U \mathbf{x} = \mathbf{c} \) by Back Substitution, again recursively:

\[
\begin{align*}
    x_n &= \frac{c_n}{d_n}, \\
    x_{n-1} &= \frac{c_{n-1} - u_{n-1} x_n}{d_{n-1}}, \\
    &\vdots \\
    x_1 &= \frac{c_1 - u_1 x_2}{d_1}.
\end{align*}
\]

As you can check, there are a total of \( 5n - 4 \) multiplications/divisions and \( 3n - 3 \) additions/subtractions required to solve a general tridiagonal system of \( n \) linear equations — a striking improvement over the general case.

**Example 1.35.** Consider the \( n \times n \) tridiagonal matrix

\[
A = \begin{pmatrix}
4 & 1 & & \\
1 & 4 & 1 & \\
& 1 & 4 & 1 \\
& & \ddots & \ddots \\
& & & 1 & 4 \\
& & & & 1 & 4
\end{pmatrix}
\]
in which the diagonal entries are all \( q_i = 4 \), while the entries immediately above and below the main diagonal are all \( p_i = r_i = 1 \). According to (1.63), the tridiagonal factorization (1.62) has \( u_1 = u_2 = \ldots = u_{n-1} = 1 \), while

\[
\begin{align*}
    d_1 &= 4, & l_j &= 1/d_j, & d_{j+1} &= 4 - l_j, & j &= 1, 2, \ldots, n-1.
\end{align*}
\]

The computed values are

\[
\begin{array}{c|ccccccc}
    j & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
    \hline
    d_j & 4 & 3.75 & 3.733333 & 3.732143 & 3.732057 & 3.732051 & 3.732051 \\
    l_j & .25 & .266666 & .267857 & .267942 & .267948 & .267949 & .267949 \\
\end{array}
\]

These converge rapidly to

\[
\begin{align*}
    d_j &\rightarrow 2 + \sqrt{3} = 3.732051 \ldots, \\
l_j &\rightarrow 2 - \sqrt{3} = .2679492 \ldots,
\end{align*}
\]

which makes the factorization for large \( n \) almost trivial. The numbers \( 2 \pm \sqrt{3} \) are the roots of the quadratic equation \( x^2 - 4x + 1 = 0 \); an explanation of this observation will be revealed in Chapter 19.

**Pivoting Strategies**

Let us now consider the practical side of pivoting. As we know, in the irregular situations when a zero shows up in a diagonal pivot position, a row interchange is required to proceed with the elimination algorithm. But even when a nonzero element appear in the current pivot position, there may be good numerical reasons for exchanging rows in order to install a more desirable element in the pivot position. Here is a simple example:

\[
\begin{align*}
0.01 x + 1.6 y &= 32.1, \\
x + 0.6 y &= 22. \\
\end{align*}
\]  
\hspace{1cm} (1.66)

The exact solution to the system is \( x = 10, \ y = 20 \). Suppose we are working with a very primitive calculator that only retains 3 digits of accuracy. (Of course, this is not a very realistic situation, but the example could be suitably modified to produce similar difficulties no matter how many digits of accuracy our computer retains.) The augmented matrix is

\[
\begin{pmatrix}
0.01 & 1.6 & 32.1 \\
1 & 0.6 & 22
\end{pmatrix}
\]

Choosing the \((1,1)\) entry as our pivot, and subtracting 100 times the first row from the second produces the upper triangular form

\[
\begin{pmatrix}
0.01 & 1.6 & 32.1 \\
0 & -159.4 & -3188
\end{pmatrix}
\]

Since our calculator has only three–place accuracy, it will round the entries in the second row, producing the augmented coefficient matrix

\[
\begin{pmatrix}
0.01 & 1.6 & 32.1 \\
0 & -159.0 & -3190
\end{pmatrix}
\]
Gaussian Elimination With Partial Pivoting

start
for $i = 1$ to $n$
    set $\sigma(i) = i$
next $i$
for $j = 1$ to $n$
    if $m_{\sigma(i),j} = 0$ for all $i \geq j$, stop; print “$A$ is singular”
    choose $i > j$ such that $m_{\sigma(i),j}$ is maximal
    interchange $\sigma(i) \leftrightarrow \sigma(j)$
    for $i = j + 1$ to $n$
        set $l_{\sigma(i)j} = m_{\sigma(i)j}/m_{\sigma(j)j}$
        for $k = j + 1$ to $n + 1$
            set $m_{\sigma(i)k} = m_{\sigma(i)k} - l_{\sigma(i)j} m_{\sigma(j)k}$
        next $k$
    next $i$
next $j$
end

The solution by back substitution gives $y = \frac{3190}{159} = 20.0628\ldots \approx 20.1$, and then $x = 100 (32.1 - 1.6 y) = 100 (32.1 - 32.16) \approx 100 (32.1 - 32.2) = -10$. The relatively small error in $y$ has produced a very large error in $x$ — not even its sign is correct!

The problem is that the first pivot, .01, is much smaller than the other element, 1, that appears in the column below it. Interchanging the two rows before performing the row operation would resolve the difficulty — even with such an inaccurate calculator! After the interchange, we have

\[
\begin{pmatrix}
1 & .6 & 22 \\
.01 & 1.6 & 32.1
\end{pmatrix},
\]

which results in the rounded-off upper triangular form

\[
\begin{pmatrix}
1 & .6 & 22 \\
0 & 1.594 & 31.88
\end{pmatrix} \approx \begin{pmatrix}
1 & .6 & 22 \\
0 & 1.59 & 31.9
\end{pmatrix}.
\]

The solution by back substitution now gives a respectable answer:

\[
y = 31.9/1.59 = 20.0628\ldots \approx 20.1, \quad x = 22 - .6 y = 22 - 12.06 \approx 22 - 12.1 = 9.9.
\]

The general strategy, known as Partial Pivoting, says that at each stage, we should use the largest legitimate (i.e., lying on or below the diagonal) element as the pivot, even
if the diagonal element is nonzero. In a computer implementation of pivoting, there is no need to waste processor time physically exchanging the row entries in memory. Rather, one introduces a separate array of pointers that serve to indicate which original row is currently in which permuted position. More specifically, one initializes $\sigma(1) = 1, \ldots, \sigma(n) = n$. Interchanging row $i$ and row $j$ of the coefficient or augmented matrix is then accomplished by merely interchanging $\sigma(i)$ and $\sigma(j)$. Thus, to access a matrix element that is currently in row $i$ of the augmented matrix, one merely retrieves the element that is in row $\sigma(i)$ in the computer’s memory. An explicit implementation of this strategy is provided below. A program for partial pivoting that includes row pointers appears above.

Partial pivoting will solve most problems, although there can still be difficulties. For instance, it will not handle the system

$$10x + 1600y = 3210, \quad x + .6y = 22,$$

obtained by multiplying the first equation in (1.66) by 1000. The tip-off is that, while the entries in the column containing the pivot are smaller, those in its row are much larger. The solution to this difficulty is Full Pivoting, in which one also performs column interchanges — preferably with a column pointer — to move the largest legitimate element into the pivot position. In practice, a column interchange is just a reordering of the variables in the system, which, as long as one keeps proper track of the order, also doesn’t change the solutions.

Finally, there are some matrices that are hard to handle even with pivoting tricks. Such *ill-conditioned* matrices are typically characterized by being “almost” singular†. A famous example of an ill-conditioned matrix is the $n \times n$ *Hilbert matrix*

$$H_n = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n} \\ 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{2}{3} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2} \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \frac{1}{n+3} & \cdots & \frac{1}{2n-1} \end{pmatrix}.$$  \hspace{1cm} (1.67)

In Proposition 3.36 we will prove that $H_n$ is nonsingular for all $n$. However, the solution of a linear system whose coefficient matrix is a Hilbert matrix $H_n$, even for moderately sized $n$, is a very challenging problem, even if one uses high precision computer arithmetic‡.

† This can be quantified by saying that their determinant is very small, but non-zero; see also Sections 8.5 and 10.3.

‡ In computer algebra systems such as MAPLE or MATHEMATICA, one can use exact rational arithmetic to perform the computations. Then the important issues are time and computational efficiency.
This is because the larger \( n \) is, the closer \( H_n \) is, in a sense, to being singular.

The reader is urged to try the following computer experiment. Fix a moderately large value of \( n \), say 20. Choose a column vector \( x \) with \( n \) entries chosen at random. Compute \( b = H_n x \) directly. Then try to solve the system \( H_n x = b \) by Gaussian Elimination. If it works for \( n = 20 \), try \( n = 50 \) or 100. This will give you a good indicator of the degree of precision used by your computer program, and the accuracy of the numerical solution algorithm.

1.8. General Linear Systems.

So far, we have only treated linear systems involving the same number of equations as unknowns, and then only those with nonsingular coefficient matrices. These are precisely the systems that always have a unique solution. We now turn to the problem of solving a general linear system of \( m \) equations in \( n \) unknowns. The cases not covered as yet are rectangular systems, with \( m \neq n \), as well as square systems with singular coefficient matrices. The basic idea underlying the Gaussian Elimination Algorithm for nonsingular systems can be straightforwardly adapted to these cases, too. One systematically utilizes the same elementary row operations so as to manipulate the coefficient matrix into a particular reduced form generalizing the upper triangular form we aimed for in the earlier square, nonsingular cases.

Definition 1.36. An \( m \times n \) matrix is said to be in row echelon form if it has the following “staircase” structure:

\[
U = \begin{pmatrix}
\oplus & * & \ldots & * & \ldots & \ldots & \ldots & \ldots & * & \ldots & * & \ldots & * & \ldots & * \\
0 & 0 & \ldots & 0 & \oplus & \ldots & * & \ldots & \ldots & \ldots & * & \ldots & * & \ldots & * \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & \oplus & \ldots & \ldots & \ldots & * & \ldots & * & \ldots & * \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & \ldots & \ldots & 0 & \oplus & \ldots & * \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & \ldots & \ldots & 0 & 0 & 0 \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & \ldots & \ldots & 0 & 0 & 0 \ldots & 0
\end{pmatrix}
\]

The entries indicated by \( \oplus \) are the pivots, and must be nonzero. The first \( r \) rows of \( U \) each contain one pivot, but not all columns need to contain a pivot. The entries below the “staircase”, indicated by the solid line, are all zero, while the non-pivot entries above the staircase, indicated by stars, can be anything. The last \( m - r \) rows are identically zero, and do not contain any pivots. Here is an explicit example of a matrix in row echelon form:

\[
\begin{pmatrix}
3 & 1 & 0 & 2 & 5 & -1 \\
0 & -1 & -2 & 1 & 8 & 0 \\
0 & 0 & 0 & 0 & 2 & -4 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
The three pivots, which are the first three nonzero entries in the nonsero rows, are, respectively, 3, −1, 2. There may, in exceptional situations, be one or more initial all zero columns.

**Proposition 1.37.** Any matrix can be reduced to row echelon form by a sequence of elementary row operations of Types #1 and #2.

In matrix language, Proposition 1.37 implies that if \( A \) is any \( m \times n \) matrix, then there exists an \( m \times m \) permutation matrix \( P \) and an \( m \times m \) special lower triangular matrix \( L \) such that

\[
PA = LU,
\]

where \( U \) is in row echelon form. The factorization is not unique.

A constructive proof of this result is based on the general Gaussian elimination algorithm, which proceeds as follows. Starting at the top left of the matrix, one searches for the first column which is not identically zero. Any of the nonzero entries in that column may serve as the pivot. Partial pivoting indicates that it is probably best to choose the largest one, although this is not essential for the algorithm to proceed. One places the chosen pivot in the first row of the matrix via a row interchange, if necessary. The entries below the pivot are made equal to zero by the appropriate elementary row operations of Type #1. One then proceeds iteratively, performing the same reduction algorithm on the submatrix consisting of all entries strictly to the right and below the pivot. The algorithm terminates when either there is a pivot in the last row, or all of the rows lying below the last pivot are identically zero, and so no more pivots can be found.

**Example 1.38.** Let us illustrate the general Gaussian Elimination algorithm with a particular example. Consider the linear system

\[
\begin{align*}
x + 3y + 2z - u &= a, \\
2x + 6y + z + 4u + 3v &= b, \\
-x - 3y - 3z + 3u + v &= c, \\
3x + 9y + 8z - 7u + 2v &= d,
\end{align*}
\]

of 4 equations in 5 unknowns, where \( a, b, c, d \) are specified numbers\(^\dagger\). The coefficient matrix is

\[
A = \begin{pmatrix}
1 & 3 & 2 & -1 & 0 \\
2 & 6 & 1 & 4 & 3 \\
-1 & -3 & -3 & 3 & 1 \\
3 & 9 & 8 & -7 & 2
\end{pmatrix}.
\]

To solve the system, we introduce the augmented matrix

\[
\begin{pmatrix}
1 & 3 & 2 & -1 & 0 & | & a \\
2 & 6 & 1 & 4 & 3 & | & b \\
-1 & -3 & -3 & 3 & 1 & | & c \\
3 & 9 & 8 & -7 & 2 & | & d
\end{pmatrix}
\]

\(^\dagger\) It will be convenient to work with the right hand side in general form, although the reader may prefer, at least initially, to assign specific values to \( a, b, c, d \).
obtained by appending the right hand side of the system. The upper left entry is nonzero, and so can serve as the first pivot; we eliminate the entries below it by elementary row operations, resulting in
\[
\begin{pmatrix}
1 & 3 & 2 & -1 & 0 & a \\ 0 & 0 & -3 & 6 & 3 & b - 2a \\ 0 & 0 & -1 & 2 & 1 & c + a \\ 0 & 0 & 2 & -4 & 2 & d - 3a
\end{pmatrix}.
\]
Now, the second column contains no suitable nonzero entry to serve as the second pivot. (The top entry already lies in a row with a pivot in it, and so cannot be used.) Therefore, we move on to the third column, choosing the \((2, 3)\) entry, \(-3\), as our second pivot. Again, we eliminate the entries below it, leading to
\[
\begin{pmatrix}
1 & 3 & 2 & -1 & 0 & a \\ 0 & 0 & -3 & 6 & 3 & b - 2a \\ 0 & 0 & 0 & 0 & 4 & d + \frac{2}{3}b - \frac{13}{3}a
\end{pmatrix}.
\]
The final pivot is in the last column, and we interchange the last two rows in order to place the coefficient matrix in row echelon form:
\[
\begin{pmatrix}
1 & 3 & 2 & -1 & 0 & a \\ 0 & 0 & -3 & 6 & 3 & b - 2a \\ 0 & 0 & 0 & 0 & 4 & d + \frac{2}{3}b - \frac{13}{3}a
\end{pmatrix}.
\]
There are three pivots, \(-1, -3, 4\), sitting in positions \((1, 1)\), \((2, 3)\) and \((3, 5)\). Note the staircase form, with the pivots on the steps and everything below the staircase being zero. Recalling the row operations used to construct the solution (and keeping in mind that the row interchange that appears at the end also affects the entries of \(L\)), we find the factorization \((1.68)\) has the explicit form
\[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0
\end{pmatrix}.
\]

We shall return to find the solution to our system after a brief theoretical interlude.

**Warning:** In the augmented matrix, pivots can never appear in the last column, representing the right hand side of the system. Thus, even if \(c - \frac{1}{3}b + \frac{5}{3}a \neq 0\), that entry does not qualify as a pivot.

We now introduce the most important numerical quantity associated with a matrix.

**Definition 1.39.** The rank of a matrix \(A\) is the number of pivots.
For instance, the rank of the matrix (1.70) equals 3, since its reduced row echelon form, i.e., the first five columns of (1.71), has three pivots. Since there is at most one pivot per row and one pivot per column, the rank of an \( m \times n \) matrix is bounded by both \( m \) and \( n \), and so \( 0 \leq r \leq \min\{m,n\} \). The only matrix of rank 0 is the zero matrix, which has no pivots.

**Proposition 1.40.** A square matrix of size \( n \times n \) is nonsingular if and only if its rank is equal to \( n \).

Indeed, the only way an \( n \times n \) matrix can end up having \( n \) pivots is if its reduced row echelon form is upper triangular with nonzero diagonal entries. But a matrix that reduces to such triangular form is, by definition, nonsingular.

Interestingly, the rank of a matrix does not depend on which elementary row operations are performed along the way to row echelon form. Indeed, performing a different sequence of row operations — say using partial pivoting versus no pivoting — can produce a completely different row echelon form. The remarkable fact, though, is that all such row echelon forms end up having exactly the same number of pivots, and this number is the rank of the matrix. A formal proof of this fact will appear in Chapter 2.

Once the coefficient matrix has been reduced to row echelon form, the solution proceeds as follows. The first step is to see if there are any incompatibilities. Suppose one of the rows in the row echelon form of the coefficient matrix is identically zero, but the corresponding entry in the last column of the augmented matrix is nonzero. What linear equation would this represent? Well, the coefficients of all the variables are zero, and so the equation is of the form \( 0 = c \), where \( c \), the number on the right hand side of the equation, is the entry in the last column. If \( c \neq 0 \), then the equation cannot be satisfied. Consequently, the entire system has no solutions, and is an incompatible linear system. On the other hand, if \( c = 0 \), then the equation is merely \( 0 = 0 \), and so is trivially satisfied. For example, the last row in the echelon form (1.71) is all zero, and hence the last entry in the final column must also vanish in order that the system be compatible. Therefore, the linear system (1.69) will have a solution if and only if the right hand sides \( a, b, c, d \) satisfy the linear constraint

\[
\frac{5}{3} a - \frac{1}{3} b + c = 0. \tag{1.72}
\]

In general, if the system is incompatible, there is nothing else to do. Otherwise, every zero row in the echelon form of the augmented matrix also has a zero entry in the last column, and the system is compatible, so one or more solutions exist. To find the solution(s), we work backwards, starting with the last row that contains a pivot. The variables in the system naturally split into two classes.

**Definition 1.41.** In a linear system \( Ux = c \) in row echelon form, the variables corresponding to columns containing a pivot are called basic variables, while the variables corresponding to the columns without a pivot are called free variables.

The solution to the system proceeds by a version of the Back Substitution procedure. The nonzero equations are solved, in reverse order, for the basic variable corresponding to its pivot. Each result is substituted into the preceding equations before they in turn are
solved. The remaining free variables, if any, are allowed to take on any values whatsoever, and the solution then specifies all the basic variables in terms of the free variables, which serve to parametrize the general solution.

**Example 1.42.** Let us illustrate this construction with our particular example. Assuming the compatibility condition (1.72), the reduced augmented matrix (1.71) is

\[
\begin{pmatrix}
1 & 3 & 2 & -1 & 0 & a \\
0 & 0 & -3 & 6 & 3 & b - 2a \\
0 & 0 & 0 & 0 & 4 & d + \frac{2}{3}b - \frac{13}{3}a \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

The pivots are found in columns 1, 3, 5, and so the corresponding variables, \(x, z, v\), are basic; the other variables, \(y, u\), are free. We will solve the reduced system for the basic variables in terms of the free variables.

As a specific example, the values \(a = 0, b = 3, c = 1, d = 1\), satisfy the compatibility constraint (1.72). The resulting augmented echelon matrix (1.71) corresponds to the system

\[
\begin{align*}
x + 3y + 2z - u &= 0, \\
-3z + 6u + 3v &= 3, \\
4v &= 3, \\
0 &= 0.
\end{align*}
\]

We now solve the equations, in reverse order, for the basic variables, and then substitute the resulting values in the preceding equations. The result is the general solution

\[
v = \frac{3}{4}, \quad z = -1 + 2u + v = -\frac{1}{4} + 2u, \quad x = -3y - 2z + u = \frac{1}{2} - 3y - 3u.
\]

The free variables \(y, u\) are completely arbitrary; any value they assume will produce a solution to the original system. For instance, if \(y = -2, u = 1 - \pi\), then \(x = 3\pi + \frac{7}{2}, z = \frac{7}{4} - 2\pi, v = \frac{3}{4}\). But keep in mind that this is merely one of an infinite number of different solutions.

In general, if the \(m \times n\) coefficient matrix of a system of \(m\) linear equations in \(n\) unknowns has rank \(r\), there are \(m - r\) all zero rows in the row echelon form, and these \(m - r\) equations must have zero right hand side in order that the system be compatible and have a solution. Moreover, there are a total of \(r\) basic variables and \(n - r\) free variables, and so the general solution depends upon \(n - r\) parameters.

Summarizing the preceding discussion, we have learned that there are only three possible outcomes for the solution to a general linear system.

**Theorem 1.43.** A system \(Ax = b\) of \(m\) linear equations in \(n\) unknowns has either (i) exactly one solution, (ii) no solutions, or (iii) infinitely many solutions.

Case (ii) occurs if the system is incompatible, producing a zero row in the echelon form that has a nonzero right hand side. Case (iii) occurs if the system is compatible and there are one or more free variables. This happens when the system is compatible and the rank of the coefficient matrix is strictly less than the number of columns: \(r < n\). Case
(i) occurs for nonsingular square coefficient matrices, and, more generally, for compatible systems for which \( r = n \), implying there are no free variables. Since \( r \leq m \), this case can only arise if the coefficient matrix has at least as many rows as columns, i.e., the linear system has at least as many equations as unknowns.

A linear system can never have a finite number — other than 0 or 1 — of solutions. Thus, any linear system that has more than one solution automatically has infinitely many. This result does not apply to nonlinear systems. As you know, a real quadratic equation \( ax^2 + bx + c = 0 \) can have either 2, 1, or 0 real solutions.

**Example 1.44.** Consider the linear system

\[
\begin{align*}
y + 4z &= a, \\
3x - y + 2z &= b, \\
x + y + 6z &= c,
\end{align*}
\]

consisting of three equations in three unknowns. The augmented coefficient matrix is

\[
\begin{pmatrix}
0 & 1 & 4 & | & a \\
3 & -1 & 2 & | & b \\
1 & 1 & 6 & | & c
\end{pmatrix}.
\]

Interchanging the first two rows, and then eliminating the elements below the first pivot leads to

\[
\begin{pmatrix}
3 & -1 & 2 & | & b \\
0 & 1 & 4 & | & a \\
0 & \frac{4}{3} & \frac{16}{3} & | & c - \frac{1}{3}b
\end{pmatrix}.
\]

The second pivot is in the \((2,2)\) position, but after eliminating the entry below it, we find the row echelon form to be

\[
\begin{pmatrix}
3 & -1 & 2 & | & b \\
0 & 1 & 4 & | & a \\
0 & 0 & 0 & | & c - \frac{1}{3}b - \frac{4}{3}a
\end{pmatrix}.
\]

Since we have a row of all zeros, the original coefficient matrix is singular, and its rank is only 2.

The compatibility condition for the system follows from this last row in the reduced echelon form, and so requires

\[
\frac{4}{3}a + \frac{1}{3}b - c = 0.
\]

If this is not satisfied, the system has no solutions; otherwise it has infinitely many. The free variable is \( z \), since there is no pivot in the third column. The general solution is

\[
y = a - 4z, \quad x = \frac{1}{3}b + \frac{1}{3}y - \frac{2}{3}z = \frac{1}{3}a + \frac{1}{3}b - 2z,
\]

where \( z \) is arbitrary.

Geometrically, Theorem 1.43 is indicating something about the possible configurations of linear subsets (lines, planes, etc.) of an \( n \)-dimensional space. For example, a single linear equation \( ax + by + cz = d \) defines a plane \( P \) in three-dimensional space. The solutions to a system of three linear equations in three unknowns is the intersection \( P_1 \cap P_2 \cap P_3 \) of three planes. Generically, three planes intersect in a single common point; this is case (i).
of the theorem, and occurs if and only if the coefficient matrix is nonsingular. The case of infinitely many solutions occurs when the three planes intersect on a common line, or, even more degenerately, when they all coincide. On the other hand, parallel planes, or planes intersecting in parallel lines, have no common point of intersection, and this corresponds to the third case of a system with no solutions. Again, no other possibilities occur; clearly one cannot have three planes having exactly 2 points in their common intersection — it is either 0, 1 or ∞. Some possible geometric configurations are illustrated in Figure 1.1.

**Homogeneous Systems**

A linear system with all 0’s on the right hand side is called a *homogeneous system*. In matrix notation, a homogeneous system takes the form

$$A \mathbf{x} = 0. \quad (1.73)$$

Homogeneous systems are always compatible, since $\mathbf{x} = \mathbf{0}$ is a solution, known as the *trivial solution*. If the homogeneous system has a nontrivial solution $\mathbf{x} \neq \mathbf{0}$, then Theorem 1.43 assures that it must have infinitely many solutions. This will occur if and only if the reduced system has one or more free variables. Thus, we find:

**Theorem 1.45.** A homogeneous linear system $A \mathbf{x} = 0$ of $m$ equations in $n$ unknowns has a nontrivial solution $\mathbf{x} \neq \mathbf{0}$ if and only if the rank of $A$ is $r < n$. If $m < n$, the system always has a nontrivial solution. If $m = n$, the system has a nontrivial solution if and only if $A$ is singular.

**Example 1.46.** Consider the homogeneous linear system

$$2x_1 + x_2 + 5x_4 = 0, \quad 4x_1 + 2x_2 - x_3 + 8x_4 = 0, \quad -2x_1 - x_2 + 3x_3 - 4x_4 = 0,$$

with coefficient matrix

$$A = \begin{pmatrix} 2 & 1 & 0 & 5 \\ 4 & 2 & -1 & 8 \\ -2 & -1 & 3 & -4 \end{pmatrix}.$$  

Since the system is homogeneous and has fewer equations than unknowns, Theorem 1.45 assures us that it has infinitely many solutions, including the trivial solution $x_1 = x_2 =
When solving a homogeneous system, the final column of the augmented matrix consists of all zeros. As it will never be altered by row operations, it is a waste of effort to carry it along during the process. We therefore perform the Gaussian Elimination algorithm directly on the coefficient matrix $A$. Working with the $(1,1)$ entry as the first pivot, we first obtain
\[
\begin{pmatrix}
2 & 1 & 0 & 5 \\
0 & 0 & -1 & -2 \\
0 & 0 & 3 & 1
\end{pmatrix}.
\]
The $(2,3)$ entry is the second pivot, and we apply one final row operation to place the matrix in row echelon form
\[
\begin{pmatrix}
2 & 1 & 0 & 5 \\
0 & 0 & -1 & -2 \\
0 & 0 & 0 & -5
\end{pmatrix}.
\]
This corresponds to the reduced homogeneous system
\[
2x_1 + x_2 + 5x_4 = 0, \quad -x_3 - 2x_4 = 0, \quad -5x_4 = 0.
\]
Since there are three pivots in the final row echelon form, the rank of the matrix $A$ is 3. There is one free variable, namely $x_2$. Using Back Substitution, we easily obtain the general solution
\[
x_1 = -\frac{1}{2}t, \quad x_2 = t, \quad x_3 = x_4 = 0,
\]
which depends upon a single free parameter $t = x_2$.

**Example 1.47.** Consider the homogeneous linear system
\[
2x - y + 3z = 0, \quad -4x + 2y - 6z = 0, \quad 2x - y + z = 0, \quad 6x - 3y + 3z = 0,
\]
with coefficient matrix $A = \begin{pmatrix}
2 & -1 & 3 \\
-4 & 2 & -6 \\
2 & -1 & 1 \\
6 & -3 & 3
\end{pmatrix}$. The system admits the trivial solution $x = y = z = 0$, but in this case we need to complete the elimination algorithm before we can state whether or not there are other solutions. After the first stage, the coefficient matrix has the form
\[
\begin{pmatrix}
2 & -1 & 3 \\
0 & 0 & 0 \\
0 & 0 & -2 \\
0 & 0 & -6
\end{pmatrix}.
\]
To continue, we need to interchange the second and third rows to place a nonzero entry in the final pivot position; after that the reduction to row echelon form is immediate:
\[
\begin{pmatrix}
2 & -1 & 3 \\
0 & 0 & -2 \\
0 & 0 & 0 \\
0 & 0 & -6
\end{pmatrix} \rightarrow \begin{pmatrix}
2 & -1 & 3 \\
0 & 0 & -2 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}.
\]
Thus, the system reduces to the equations
\[
2x - y + 3z = 0, \quad -2z = 0, \quad 0 = 0, \quad 0 = 0,
\]
where the third and fourth equations are trivially compatible, as they must be in the homogeneous case. The rank is equal to two, which is less than the number of columns, and so, even though the system has more equations than unknowns, it has infinitely many solutions. These can be written in terms of the free variable \(y\), and so the general solution is \(x = \frac{1}{2} y, \ z = 0\), where \(y\) is arbitrary.

1.9. Determinants.

You may be surprised that, so far, we have left undeveloped a topic that often assumes a central role in basic linear algebra: determinants. As with matrix inverses, while determinants can be useful in low dimensions and for theoretical purposes, they are mostly irrelevant when it comes to large scale applications and practical computations. Indeed, the best way to compute a determinant is (surprise) Gaussian Elimination! However, you should be familiar with the basics of determinants, and so for completeness, we shall provide a very brief introduction.

The determinant of a square matrix \(A\), written \(\det A\), is a number that immediately tells whether the matrix is singular or not. (Rectangular matrices do not have determinants.) We already encountered, (1.34), the determinant of a \(2 \times 2\) matrix, which is equal to the product of the diagonal entries minus the product of the off-diagonal entries:

\[
\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc.
\]

The determinant is nonzero if and only if the matrix has an inverse. Our goal is to generalize this construction to general square matrices.

There are many different ways to define determinants. The difficulty is that the actual formula is very unwieldy — see (1.81) below — and not well motivated. We prefer an axiomatic approach that explains how our elementary row operations affect the determinant. In this manner, one can compute the determinant by Gaussian elimination, which is, in fact, the fastest and most practical computational method in all but the simplest situations. In effect, this remark obviates the need to ever compute a determinant.

**Theorem 1.48.** The determinant of a square matrix \(A\) is the uniquely defined scalar quantity \(\det A\) that satisfies the following axioms:

1. Adding a multiple of one row to another does not change the determinant.
2. Interchanging two rows changes the sign of the determinant.
3. Multiplying a row by any scalar (including zero) multiplies the determinant by the same scalar.
4. Finally, the determinant function is fixed by setting

\[
\det I = 1.
\]  

(1.74)

Checking that all four of these axioms hold in the \(2 \times 2\) case (1.34) is left as an elementary exercise for the reader. A particular consequence of axiom 3 is that when we multiply a row of any matrix \(A\) by the zero scalar, the resulting matrix, which has a row of all zeros, necessarily has zero determinant.

**Lemma 1.49.** Any matrix with one or more all zero rows has zero determinant.
Since the determinantal axioms tell how determinants behave under all three of our elementary row operations, we can use Gaussian elimination to compute a general determinant, recovering $\det A$ from its permuted $LU$ factorization.

**Theorem 1.50.** If $A$ is a regular matrix, with $A = LU$ factorization as in (1.21), then

$$\det A = \det U = \prod_{i=1}^{n} u_{ii},$$

(1.75)

equals the product of the pivots. More generally, if $A$ is nonsingular, and requires $k$ row interchanges to arrive at its permuted $LU$ factorization $PA = LU$, then

$$\det A = \det P \det U = (-1)^k \prod_{i=1}^{n} u_{ii},$$

(1.76)

Finally, $A$ is singular if and only if $\det A = 0$.

**Proof:** In the regular case, one only needs elementary row operations of type #1 to reduce $A$ to upper triangular form $U$, and axiom 1 says these do not change the determinant. Therefore $\det A = \det U$. Proceeding with the full Gauss–Jordan scheme, the next phase is to divide each row in $U$ by its pivot, leading to the special upper triangular matrix $V$ with all 1’s on the diagonal. Axiom 3 implies

$$\det A = \det U = \left( \prod_{i=1}^{n} u_{ii} \right) \det V,$$

(1.77)

Finally, we can reduce $V$ to the identity by further row operations of Type #1, and so by (1.74),

$$\det V = \det I = 1.$$  
(1.78)

Combining equations (1.77), (1.78) proves the theorem for the regular case. The nonsingular case follows without difficulty — each row interchange changes the sign of the determinant, and so $\det A$ equals $\det U$ if there have been an even number of interchanges, but equals $-\det U$ if there have been an odd number.

Finally, if $A$ is singular, then we can reduce it to a matrix with at least one row of zeros by elementary row operations of types #1 and #2. Lemma 1.49 implies that the resulting matrix has zero determinant, and so $\det A = 0$, also. **Q.E.D.**

**Corollary 1.51.** The determinant of a diagonal matrix is the product of the diagonal entries. The same result holds for both lower triangular and upper triangular matrices.

**Example 1.52.** Let us compute the determinant of the $4 \times 4$ matrix

$$A = \begin{pmatrix} 1 & 0 & -1 & 2 \\ 2 & 1 & -3 & 4 \\ 0 & 2 & -2 & 3 \\ 1 & 1 & -4 & -2 \end{pmatrix}. $$
We perform our usual Gaussian Elimination algorithm, successively leading to the matrices

\[
A \rightarrow \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & -1 & 0 \\ 0 & 2 & -2 & 3 \\ 0 & 1 & -3 & -4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & -2 & -4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 3 \end{pmatrix},
\]

where we used a single row interchange to obtain the final upper triangular form. Owing to the row interchange, the determinant of the original matrix is \(-1\) times the product of the pivots:

\[
\det A = -1 \cdot 1 \cdot 1 \cdot (-2) \cdot 3 = 6.
\]

In particular, this tells us that \(A\) is nonsingular. But, of course, this was already implied by the elimination, since the matrix reduced to upper triangular form with 4 pivots.

Let us now present some of the basic properties of determinants.

**Lemma 1.53.** The determinant of the product of two square matrices of the same size is the product of the determinants:

\[
\det(AB) = \det A \det B. \quad (1.79)
\]

**Proof:** The product formula holds if \(A\) is an elementary matrix; this is a consequence of the determinantal axioms, combined with Corollary 1.51. By induction, if \(A = E_1 E_2 \cdots E_N\) is a product of elementary matrices, then (1.79) also holds. Therefore, the result holds whenever \(A\) is nonsingular. On the other hand, if \(A\) is singular, then according to Exercise 1.51, \(A = E_1 E_2 \cdots E_N Z\), where the \(E_i\) are elementary matrices, and \(Z\), the row echelon form, is a matrix with a row of zeros. But then \(Z B = W\) also has a row of zeros, and so \(AB = E_1 E_2 \cdots E_N W\) is also singular. Thus, both sides of (1.79) are zero in this case. \(Q.E.D.\)

It is a remarkable fact that, even though matrix multiplication is not commutative, and so \(AB \neq BA\) in general, it is nevertheless always true that both products have the same determinant: \(\det(AB) = \det(BA)\). Indeed, both are equal to the product \(\det A \det B\) of the individual determinants because ordinary (scalar) multiplication is commutative.

**Lemma 1.54.** Transposing a matrix does not change its determinant:

\[
\det A^T = \det A. \quad (1.80)
\]

**Proof:** By inspection, this formula holds if \(A\) is an elementary matrix. If \(A = E_1 E_2 \cdots E_N\) is a product of elementary matrices, then using (1.49), (1.79) and induction

\[
\det A^T = \det(E_1 E_2 \cdots E_N)^T = \det(E_N^T E_{N-1}^T \cdots E_1^T) = \det E_N^T \det E_{N-1}^T \cdots \det E_1^T \\
= \det E_N \det E_{N-1} \cdots \det E_1 = \det E_1 \det E_2 \cdots \det E_N \\
= \det(E_1 E_2 \cdots E_N) = \det A.
\]

The middle equality follows from the commutativity of ordinary multiplication. This proves the nonsingular case; the singular case follows from Lemma 1.30, which implies that \(A^T\) is singular if and only if \(A\) is. \(Q.E.D.\)
Remark: Lemma 1.54 has the interesting consequence that one can equally well use “elementary column operations” to compute determinants. We will not develop this approach in any detail here, since it does not help us to solve linear equations.

Finally, we state the general formula for a determinant; a proof can be found in [135].

**Theorem 1.55.** If $A$ is an $n \times n$ matrix with entries $a_{ij}$, then

$$
\det A = \sum_{\pi} \pm a_{1,\pi(1)} a_{2,\pi(2)} \cdots a_{n,\pi(n)}. \tag{1.81}
$$

The sum in (1.81) is over all possible permutations $\pi$ of the columns of $A$. The summands consist of all possible ways of choosing $n$ entries of $A$ with one entry in each column and 1 entry in each row of $A$. The sign in front of the indicated term depends on the permutation $\pi$; it is $+$ if $\pi$ is an even permutation, meaning that its matrix can be reduced to the identity by an even number of row interchanges, and $-$ is $\pi$ is odd. For example, the six terms in the well-known formula

$$
\det \begin{pmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{pmatrix}
= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}
- a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31} \tag{1.82}
$$

for a $3 \times 3$ determinant correspond to the six possible $3 \times 3$ permutation matrices (1.27).

The proof that (1.81) obeys the basic determinantal axioms is straightforward, but, will not be done here. The reader might wish to try the $3 \times 3$ case to be convinced that it works. This explicit formula proves that the determinant function is well-defined, and formally completes the proof of Theorem 1.48.

Unfortunately, the explicit determinant formula (1.81) contains $n!$ terms, and so, as soon as $n$ is even moderately large, is completely impractical for computation. The most efficient way is still our mainstay — Gaussian Elimination coupled the fact that the determinant is $\pm$ the product of the pivots!

Determinants have many fascinating and theoretically important properties. However, in our applications, these will not be required, and so we conclude this very brief introduction to the subject.