Chapter 7

Introduction to Matrices

This chapter introduces the theory and application of matrices. It is divided into two main sections.

- Section 7.1 discusses some of the basic properties and operations of matrices strictly from a mathematical perspective. (More matrix operations are discussed in Chapter 9.)
- Section 7.2 explains how to interpret these properties and operations geometrically.

Matrices are of fundamental importance in 3D math, where they are primarily used to describe the relationship between two coordinate spaces. They do this by defining a computation to transform vectors from one coordinate space to another.

7.1 Matrix — A Mathematical Definition

In linear algebra, a matrix is a rectangular grid of numbers arranged into *rows* and *columns*. Recalling our earlier definition of vector as a one-dimensional array of numbers, a matrix may likewise be defined as a *two-dimensional array* of numbers. (The *two* in “two-dimensional array” comes from the fact that there are rows and columns, and it should not be confused with 2D vectors or matrices.) A vector is an array of scalars, and a matrix is an array of vectors.

7.1.1 Matrix Dimensions and Notation

Just as we defined the dimension of a vector by counting how many numbers it contained, we will define the size of a matrix by counting how many rows and columns it contains. An $r \times c$ matrix (read “r by c”) has r rows and c columns. Here is an example of a 4×3 matrix:

$$\begin{bmatrix} 4 & 0 & 12 \\ -5 & \sqrt{4} & 3 \\ 12 & -4/3 & -1 \\ 1/2 & 18 & 0 \end{bmatrix}$$

This 4×3 matrix illustrates the standard notation for writing matrices. We arrange the numbers in a grid, surrounded by square brackets. Note that other authors surround the grid of numbers with parentheses rather than brackets. Other authors use straight vertical lines. We will reserve this notation for an entirely separate concept related to matrices, the *determinant* of a matrix. (We will discuss determinants in Section 9.1.)

As we mentioned in Section 5.2, we will represent a matrix variable with uppercase letters in boldface, for example: **M**, **A**, **R**. When we wish to refer to the individual elements within a matrix, we use subscript notation, usually with the corresponding lowercase letter in italics. This is shown below for a 3×3 matrix:

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

m_{ij} denotes the element in **M** at row i and column j. Matrices use 1-based indices, so the first row and column are numbered one. For example, m_{12} (read "m one two," not "m twelve") is the element in the first row, second column. Notice that this is different from the C programming language, which uses 0-based array indices. A matrix does not have a column 0 or row 0. This difference in indexing can cause some confusion if using actual C arrays to define matrices. (This is one reason we won't use arrays to define matrices in our code.)

7.1.2 Square Matrices

Matrices with the same number of rows as columns are called *square* matrices and are of particular importance. In this book, we will be interested in 2×2, 3×3, and 4×4 matrices.

The *diagonal elements* of a square matrix are those elements where the row and column index are the same. For example, the diagonal elements of the 3×3 matrix **M** are m_{11}, m_{22}, and m_{33}. The other elements are *non-diagonal* elements. The diagonal elements form the *diagonal* of the matrix:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

If all non-diagonal elements in a matrix are zero, then the matrix is a *diagonal matrix*. For example:

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

A special diagonal matrix is the *identity matrix*. The identity matrix of dimension n, denoted $\mathbf{I}_n$, is the $n\times n$ matrix with 1's on the diagonal and 0's elsewhere. For example, the 3×3 identity matrix is:

Equation 7.1: The 3D identity matrix

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Often, the context will make the dimension of the identity matrix used in a particular situation clear. In these cases, we will omit the subscript and simply refer to the identity matrix as $\mathbf{I}$.

The identity matrix is special because it is the *multiplicative identity element* for matrices. (We will learn about matrix multiplication in Section 7.1.6.) The basic idea is that if you multiply a matrix by the identity matrix, you get the original matrix. So, in some ways, the identity matrix is for matrices what the number 1 is for scalars.

7.1.3 Vectors as Matrices

Matrices may have any positive number of rows and columns, including one. We have already encountered matrices with one row or one column: vectors! A vector of dimension n can be viewed either as a $1\times n$ matrix or as an $n\times 1$ matrix. A $1\times n$ matrix is known as a *row vector*, and an $n\times 1$ matrix is known as a *column vector*. Row vectors are written horizontally, and column vectors are written vertically:

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Until now, we have used the two notations interchangeably. Indeed, geometrically they are identical, and in most cases the distinction is not important. However, for reasons that will soon become apparent, when we use vectors with matrices, we must be very clear about whether our vector is a row or column vector.

7.1.4 Transposition

Consider a matrix $\mathbf{M}$ with dimensions $r\times c$. The *transpose* of $\mathbf{M}$ (denoted $\mathbf{M}^T$) is the $c\times r$ matrix where the columns are formed from the rows of $\mathbf{M}$. In other words, $\mathbf{M}^T_{ij}=\mathbf{M}_{ji}$. This "flips" the matrix diagonally. Equation 7.2 gives two examples of transposing matrices:

Equation 7.2: Transposing matrices

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \qquad \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

For vectors, transposition turns row vectors into column vectors and vice versa:

Equation 7.3: Transposing converts between row and column vectors

$$\begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix}^T = \begin{bmatrix} x & y & z \end{bmatrix}$$

Transposition notation is often used to write column vectors inline in a paragraph, like this: $[1, 2, 3]^T$.

There are two fairly obvious, but significant, observations concerning matrix transposition:

- $(\mathbf{M}^T)^T = \mathbf{M}$ for a matrix **M** of any dimension. In other words, if we transpose a matrix, and then transpose it again, we get the original matrix. This rule also applies to vectors.
- $\mathbf{D}^T = \mathbf{D}$ for any diagonal matrix **D**, including the identity matrix **I**.

7.1.5 Multiplying a Matrix with a Scalar

A matrix **M** may be multiplied with a scalar *k*, resulting in a matrix of the same dimension as **M**. We denote matrix multiplication with a scalar by placing the scalar and the matrix side by side, usually with the scalar on the left. No multiplication symbol is necessary. The multiplication takes place in the straightforward fashion; each element in the resulting matrix $k\mathbf{M}$ is the product of *k* and the corresponding element in **M**. For example:

Equation 7.4: Multiplying a 4×3 matrix by a scalar

$$k\mathbf{M} = k\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \\ m_{41} & m_{42} & m_{43} \end{bmatrix} = \begin{bmatrix} km_{11} & km_{12} & km_{13} \\ km_{21} & km_{22} & km_{23} \\ km_{31} & km_{32} & km_{33} \\ km_{41} & km_{42} & km_{43} \end{bmatrix}$$

7.1.6 Multiplying Two Matrices

In certain situations, we can take the product of two matrices. The rules that govern when matrix multiplication is allowed, and how the result is computed, may at first seem bizarre. An $r\times n$ matrix **A** may be multiplied by an $n\times c$ matrix **B**. The result, denoted **AB**, is an $r\times c$ matrix.

For example, assume that **A** is a 4×2 matrix, and **B** is a 2×5 matrix. Then **AB** is a 4×5 matrix:

$$\underset{\substack{r\times n \\ 4\times 2}}{\overset{\mathbf{A}}{\begin{bmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{bmatrix}}} \underset{\substack{n\times c \\ 2\times 5}}{\overset{\mathbf{B}}{\begin{bmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{bmatrix}}} = \underset{\substack{r\times c \\ 4\times 5}}{\overset{\mathbf{AB}}{\begin{bmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{bmatrix}}}$$

must match

columns in result

rows in result

If the number of columns in **A** does not match the number of rows in **B**, then the multiplication **AB** is undefined.

Matrix multiplication is computed as follows: let the matrix **C** be the $r \times c$ product **AB** of the $r \times n$ matrix **A** with the $n \times c$ matrix **B**. Then each element c_{ij} is equal to the vector dot product of row i of **A** with column j of **B**. More formally:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

(See Appendix A if you don't know what the symbol that looks like a "Z" means.)

This sounds complicated, but there is a simple pattern. For each element c_{ij} in the result, locate row i in **A** and column j in **B**. Multiply the corresponding elements of the row and column, and sum the products. (This is equivalent to the dot product of row i in **A** with column j in **B**.) c_{ij} is equal to this sum.

Let's look at an example. Below we show how to compute c_{24}:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & \boxed{c_{24}} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix}$$

$$c_{24} = a_{21} b_{14} + a_{22} b_{24}$$

The element in the second row and fourth column of **C** is equal to the dot product of the second row of **A** with the fourth column of **B**.

Another way to help remember the pattern is to write **B** above **C**, as shown below. This aligns the proper row from **A** with a column from **B** for each element in the result **C**:

$$\begin{array}{cc} & \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} & \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix} \end{array}$$

$$c_{43} = a_{41} b_{13} + a_{42} b_{23}$$

For geometric applications, we will be particularly interested in multiplying square matrices — the 2×2 and 3×3 cases are especially important to us. Equation 7.5 gives the complete equation for 2×2 matrix multiplication:

Equation 7.5: 2×2 matrix multiplication

$$\begin{aligned}\mathbf{AB} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}\end{aligned}$$

Let's look at a 2×2 example with some real numbers:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} -3 & 0 \\ 5 & 1/2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -7 & 2 \\ 4 & 6 \end{bmatrix} \\ \mathbf{AB} &= \begin{bmatrix} -3 & 0 \\ 5 & 1/2 \end{bmatrix}\begin{bmatrix} -7 & 2 \\ 4 & 6 \end{bmatrix} \\ &= \begin{bmatrix} (-3)(-7)+(0)(4) & (-3)(2)+(0)(6) \\ (5)(-7)+(1/2)(4) & (5)(2)+(1/2)(6) \end{bmatrix} \\ &= \begin{bmatrix} 21 & -6 \\ -33 & 13 \end{bmatrix}\end{aligned}$$

Now for the 3×3 case:

Equation 7.6: 3×3 matrix multiplication

$$\begin{aligned}\mathbf{AB} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} & a_{11}b_{12}+a_{12}b_{22}+a_{13}b_{32} & a_{11}b_{13}+a_{12}b_{23}+a_{13}b_{33} \\ a_{21}b_{11}+a_{22}b_{21}+a_{23}b_{31} & a_{21}b_{12}+a_{22}b_{22}+a_{23}b_{32} & a_{21}b_{13}+a_{22}b_{23}+a_{23}b_{33} \\ a_{31}b_{11}+a_{32}b_{21}+a_{33}b_{31} & a_{31}b_{12}+a_{32}b_{22}+a_{33}b_{32} & a_{31}b_{13}+a_{32}b_{23}+a_{33}b_{33} \end{bmatrix}\end{aligned}$$

And a 3×3 example with some real numbers:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 1 & -5 & 3 \\ 0 & -2 & 6 \\ 7 & 2 & -4 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -8 & 6 & 1 \\ 7 & 0 & -3 \\ 2 & 4 & 5 \end{bmatrix} \\ \mathbf{AB} &= \begin{bmatrix} 1 & -5 & 3 \\ 0 & -2 & 6 \\ 7 & 2 & -4 \end{bmatrix}\begin{bmatrix} -8 & 6 & 1 \\ 7 & 0 & -3 \\ 2 & 4 & 5 \end{bmatrix} \\ &= \begin{bmatrix} (1)(-8)+(-5)(7)+(3)(2) & (1)(6)+(-5)(0)+(3)(4) & (1)(1)+(-5)(-3)+(3)(5) \\ (0)(-8)+(-2)(7)+(6)(2) & (0)(6)+(-2)(0)+(6)(4) & (0)(1)+(-2)(-3)+(6)(5) \\ (7)(-8)+(2)(7)+(-4)(2) & (7)(6)+(2)(0)+(-4)(4) & (7)(1)+(2)(-3)+(-4)(5) \end{bmatrix} \\ &= \begin{bmatrix} -37 & 18 & 31 \\ -2 & 24 & 36 \\ -50 & 26 & -19 \end{bmatrix}\end{aligned}$$

Beginning in Section 9.4, we will also use 4×4 matrices.

A few interesting notes concerning matrix multiplication:

- Multiplying any matrix **M** by a square matrix **S** on either side results in a matrix of the same size as **M**, provided that the sizes of the matrices are such that the multiplication is allowed. If **S** is the identity matrix **I**, then the result is the original matrix **M**:

  $$\mathbf{MI} = \mathbf{IM} = \mathbf{M}$$

(That's the reason it's called the *identity* matrix!)

- Matrix multiplication is *not* commutative:
  $\mathbf{AB} \neq \mathbf{BA}$
- Matrix multiplication is associative:
  $$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$
  (Assuming that the sizes of **A**, **B**, and **C** are such that multiplication is allowed, note that if (**AB**)**C** is defined, then **A**(**BC**) is always defined as well.) The associativity of matrix multiplication extends to multiple matrices. For example:
  $$\mathbf{ABCDEF} = ((((\mathbf{AB})\mathbf{C})\mathbf{D})\mathbf{E})\mathbf{F} = \mathbf{A}((((\mathbf{BC})\mathbf{D})\mathbf{E})\mathbf{F}) = (\mathbf{AB})(\mathbf{CD})(\mathbf{EF})$$
  It is interesting to note that although all parenthesizations compute the correct result, some groupings require fewer scalar multiplications than others. The problem of finding the parenthesization that minimizes the number of scalar multiplications is known as the *matrix chain* problem.
- Matrix multiplication also associates with multiplication by a scalar or a vector:
  $$(k\mathbf{A})\mathbf{B} = k(\mathbf{AB}) = \mathbf{A}(k\mathbf{B}) \quad (\mathbf{vA})\mathbf{B} = \mathbf{v}(\mathbf{AB})$$
- Transposing the product of two matrices is the same as taking the product of their transposes in reverse order:
  $$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$
  This can be extended to more than two matrices:
  $$(\mathbf{M}_1\mathbf{M}_2\cdots\mathbf{M}_{n-1}\mathbf{M}_n)^T = {\mathbf{M}_n}^T{\mathbf{M}_{n-1}}^T\cdots{\mathbf{M}_2}^T{\mathbf{M}_1}^T$$

7.1.7 Multiplying a Vector and a Matrix

Since a vector can be considered a matrix with one row or one column, we can multiply a vector and a matrix using the rules discussed in the previous section. It becomes very important whether we are using row or column vectors. Below we show how 3D row and column vectors may be pre- or post-multiplied by a 3×3 matrix:

Equation 7.7: Multiplying 3D row and column vectors with a 3×3 matrix

$$\begin{bmatrix} x & y & z \end{bmatrix}\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} xm_{11}+ym_{21}+zm_{31} & xm_{12}+ym_{22}+zm_{32} & xm_{13}+ym_{23}+zm_{33} \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} xm_{11}+ym_{12}+zm_{13} \\ xm_{21}+ym_{22}+zm_{23} \\ xm_{31}+ym_{32}+zm_{33} \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} x & y & z \end{bmatrix} = (undefined)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = (undefined)$$

As you can see, when we multiply a row vector on the left by a matrix on the right, the result is a row vector. When we multiply a matrix on the left by a column vector on the right, the result is a column vector. The other two combinations are not allowed; you cannot multiply a matrix on the left by a row vector on the right, nor can you multiply a column vector on the left by a matrix on the right.

There are three interesting observations concerning vector-times-matrix multiplication:

- Each element in the resulting vector is the dot product of the original vector with a single row or column from the matrix.
- Each element in the matrix determines how much "weight" a particular element in the input vector contributes to an element in the output vector. For example, m_{11} controls how much of the input x value goes toward the output x value.
- Vector-times-matrix multiplication distributes over vector addition. That is, for vectors **v** and **w** and matrices **M**:

  $$(\mathbf{v} + \mathbf{w})\mathbf{M} = \mathbf{vM} + \mathbf{wM}$$

7.1.8 Row vs. Column Vectors

In this section, we will explain why the distinction between row and column vectors is significant and give our rationale for preferring row vectors. In Equation 7.7, when we multiply a row vector on the left with a matrix on the right, we get the row vector:

$$\begin{bmatrix} xm_{11} + ym_{21} + zm_{31} & xm_{12} + ym_{22} + zm_{32} & xm_{13} + ym_{23} + zm_{33} \end{bmatrix}$$

Compare that with the result when a column vector on the right is multiplied by a matrix on the left:

$$\begin{bmatrix} xm_{11} + ym_{12} + zm_{13} \\ xm_{21} + ym_{22} + zm_{23} \\ xm_{31} + ym_{32} + zm_{33} \end{bmatrix}$$

Disregarding the fact that one is a row vector and the other is a column vector, the values for the components of the vector are *not* the same! This is why the distinction between row and column vectors is so important.

In this book, we will use column vectors *only* when the distinction between row and column vectors is not important. If the distinction is at all relevant (for example, if vectors are used in conjunction with matrices), then we will use row vectors.

There are several reasons for using row vectors instead of column vectors:

- Row vectors format nicely when they are used inline in a paragraph. For example, the row vector [1, 2, 3] fits nicely in this sentence. But notice how the column vector

  $$\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

causes formatting problems. The same sorts of problems occur in source code as well. Some authors use transposed row vectors to write column vectors inline in their text, like $[4, 5, 6]^T$. Using row vectors from the beginning avoids all this weirdness.

- More importantly, when we discuss how matrix multiplication can be used to perform coordinate space transformations, it will be convenient for the vector to be on the left and the matrix on the right. In this way, the transformation will read like a sentence. This is especially important when more than one transformation takes place. For example, if we wish to transform a vector **v** by the matrices **A**, **B**, and **C**, in that order, we write **vABC**. Notice that the matrices are listed in order of transformation from left to right. If column vectors are used, then the matrix is on the left, and the transformations will occur in order from right to left. In this case, we would write **CBAv**. We will discuss concatenation of multiple transformation matrices in detail in Section 8.7.
- DirectX uses row vectors.

The arguments in favor of column vectors are:

- Column vectors usually format nicer in equations. (Examine Equation 7.7 on page 89.)
- Linear algebra textbooks typically use column vectors.
- Several famous computer graphics "bibles" use column vectors. (For example, [8], [17].)
- OpenGL uses column vectors.

Different authors use different conventions. When you use someone else's equation or source code, be very careful that you know whether they are using row or column vectors. If a book uses column vectors, its equations for matrices will be transposed compared to the equations we present in this book. In addition, when column vectors are used, vectors are pre-multiplied by a matrix, as opposed to the convention chosen in this book, to multiply row vectors by a matrix on the right. This causes the order of multiplication to be reversed between the two styles when multiple matrices and vectors are multiplied together. For example, the multiplication **vABC** is valid only with row vectors. The corresponding multiplication would be written **CBAv** if column vectors were used.

Transposition-type mistakes like this can be a common source of frustration when programming 3D math. Luckily, the C++ matrix classes we will present in Chapter 11 are designed so that direct access to the individual matrix elements is seldom needed. Thus, the frequency of these types of errors is minimized.

7.2 Matrix — A Geometric Interpretation

In general, a square matrix can describe any *linear transformation*. In Section 8.8.1, we will provide a complete definition of linear transformation. For now, it will suffice to say that a linear transformation preserves straight and parallel lines, and there is no translation — the origin does not move. While a linear transformation preserves straight lines, other properties of the geometry, such as lengths, angles, areas, and volumes, are possibly altered by the transformation. In a non-technical sense, a linear transformation may "stretch" the coordinate space, but it doesn't "curve" or "warp" it. This is a very useful set of transformations:

- Rotation
- Scale
- Orthographic projection
- Reflection
- Shearing

Chapter 8 discusses each of these transformations in detail. For now, we will attempt to gain some understanding of the relationship between a particular matrix and the transform it represents.

7.2.1 How Does a Matrix Transform Vectors?

In Section 4.2.4, we discussed how a vector may be interpreted geometrically as a sequence of axially-aligned displacements. For example, the vector [1, –3, 4] can be interpreted as a displacement of [1, 0, 0], followed by a displacement of [0, –3, 0], followed by a displacement of [0, 0, 4]. Section 5.8.2 described how this sequence of displacements can be interpreted as a sum of vectors according to the triangle rule:

$$\begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

In general, for any vector **v**, we can write **v** in "expanded" form:

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix}$$

Let's rewrite this expression in a slightly different form:

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + z\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Notice that the unit vectors on the right-hand side are *x*-, *y*-, and *z*-axes. We have just expressed mathematically a concept we established in Section 4.2.3: each coordinate of a vector specifies the signed displacement parallel to corresponding axes.

Let's rewrite the sum one more time. This time, we will define the vectors **p**, **q**, and **r** to be unit vectors pointing in the +*x*, +*y*, and +*z*, directions, respectively:

Equation 7.8: Expressing a vector as a linear combination of basis vectors

$$\mathbf{v} = x\mathbf{p} + y\mathbf{q} + z\mathbf{r}$$

Now we have expressed the vector **v** as a linear combination of the vectors **p**, **q**, and **r**. The vectors **p**, **q**, and **r** are known as *basis vectors*. We are accustomed to using the cardinal axes as basis vectors, but, in fact, a coordinate space may be defined using *any* three vectors, provided the three vectors are *linearly independent* (which basically means that they don't lie in a plane). If we construct a 3×3 matrix **M** using **p**, **q**, and **r** as the rows of the matrix, we get:

Equation 7.9: Interpreting a matrix as a set of basis vectors

$$\mathbf{M} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x & \mathbf{p}_y & \mathbf{p}_z \\ \mathbf{q}_x & \mathbf{q}_y & \mathbf{q}_z \\ \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z \end{bmatrix}$$

Multiplying a vector by this matrix, we get:

$$\begin{aligned} \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} \mathbf{p}_x & \mathbf{p}_y & \mathbf{p}_z \\ \mathbf{q}_x & \mathbf{q}_y & \mathbf{q}_z \\ \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z \end{bmatrix} &= \begin{bmatrix} x\mathbf{p}_x + y\mathbf{q}_x + z\mathbf{r}_x & x\mathbf{p}_y + y\mathbf{q}_y + z\mathbf{r}_y & x\mathbf{p}_z + y\mathbf{q}_z + z\mathbf{r}_z \end{bmatrix} \\ &= x\mathbf{p} + y\mathbf{q} + z\mathbf{r} \end{aligned}$$

This is the same as our original equation for computing **v** after transformation. We have discovered the key idea that:

> If we interpret the rows of a matrix as the basis vectors of a coordinate space, then multiplication by the matrix performs a coordinate space transformation. If **aM**=**b**, we say that **M** transformed **a** to **b**.

From this point forward, the terms *transformation* and *multiplication* will be largely synonymous.

The bottom line is that there's nothing especially magical about matrices. They simply provide a compact way to represent the mathematical operations required to perform a coordinate space transformation. Furthermore, using linear algebra to manipulate matrices is a convenient way to take simple transformations and derive more complicated transformations. We will investigate this idea in Section 8.7.

7.2.2 What Does a Matrix Look Like?

"Unfortunately, no one can be told what the matrix is — you have to see it for yourself." This is not only a line from a great movie, it's true for linear algebra matrices as well. Until you develop an ability to visualize a matrix, it is just nine numbers in a box. We have stated that a matrix represents a coordinate space transformation. So when we visualize the matrix, we are visualizing the transformation, the new coordinate system. But what does this transformation look like? What is the relationship between a particular 3D transformation (i.e., rotation, shearing, etc.) and those nine numbers inside a 3×3 matrix? How can we construct a matrix to perform a given transform (other than by copying the equations blindly out of a book)?

To begin to answer these questions, let's examine what happens when the basis vectors [1, 0, 0], [0, 1, 0], and [0, 0, 1] are multiplied by an arbitrary matrix **M**:

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix}$$

As you can see, when we multiply the basis vector [1, 0, 0] by **M**, the resulting vector is the first row of **M**. Similar statements can be made regarding the other two rows. This is a critical observation:

> Each row of a matrix can be interpreted as a basis vector after transformation.

This is the same basic idea that we discovered in the previous section, only we have come at it from a slightly different angle. This very powerful concept has two important implications:

- First, we have a simple way to take any matrix and visualize what sort of transformation the matrix represents. Later in this section we will give examples of how to do this in 2D and 3D.
- Second, we have the ability to make the reverse construction — given a desired transformation (i.e. rotation, scale, etc.). We can derive a matrix which represents that transformation. All we have to do is figure out what the transformation does to basis vectors and fill in those transformed basis vectors into the rows of a matrix. This trick is used extensively in Chapter 8, where we will discuss the fundamental transformations and show how to construct matrices to perform those transformations.

Let's look at a couple of examples. First we will examine a 2D example to get ourselves warmed up and then a full-fledged 3D example. Examine the following 2×2 matrix:

$$\mathbf{M} = \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$$

What sort of transformation does this matrix represent? First, let's extract the basis vectors **p** and **q** from the rows of the matrix:

$$\begin{aligned} \mathbf{p} &= \begin{bmatrix} 2 & 1 \end{bmatrix} \\ \mathbf{q} &= \begin{bmatrix} -1 & 2 \end{bmatrix} \end{aligned}$$

Figure 7.1 shows these vectors in the Cartesian plane, along with the "original" basis vectors (the x-axis and y-axis), for reference:

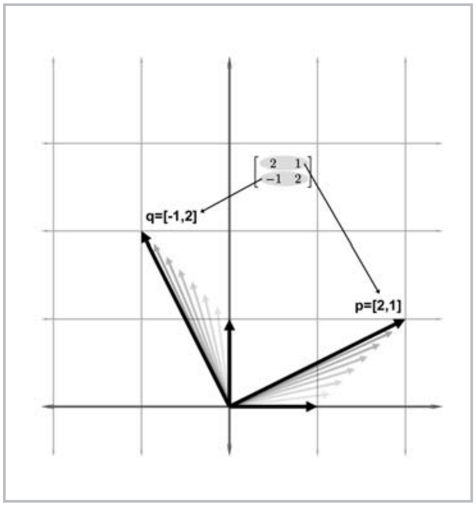


Figure 7.1: Visualizing the row vectors of a 2D transform matrix

As Figure 7.1 illustrates, the $+x$ basis vector is transformed into the vector labeled **p** above, and the y basis vector is transformed into the vector labeled **q**. So one way to visualize a matrix in 2D is to visualize the "L" formed by the row vectors. In this example, we can easily see that part of the transformation represented by **M** is a counterclockwise rotation of about 26°.

Of course, *all* vectors are affected by a linear transformation, not just the basis vectors. While we can get a very good idea what this transformation looks like from the "L," we can gain further insight on the effect the transformation has on the rest of the vectors by completing the 2D parallelogram formed by the basis vectors:

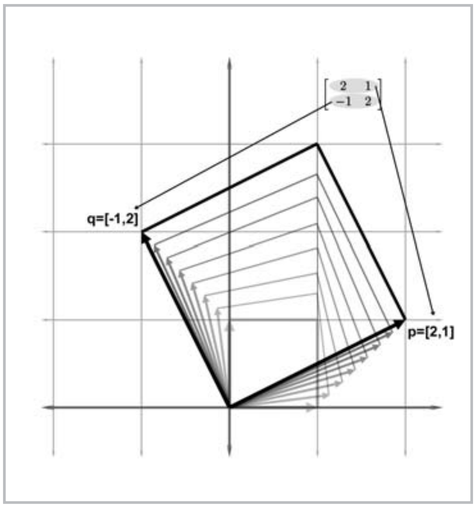
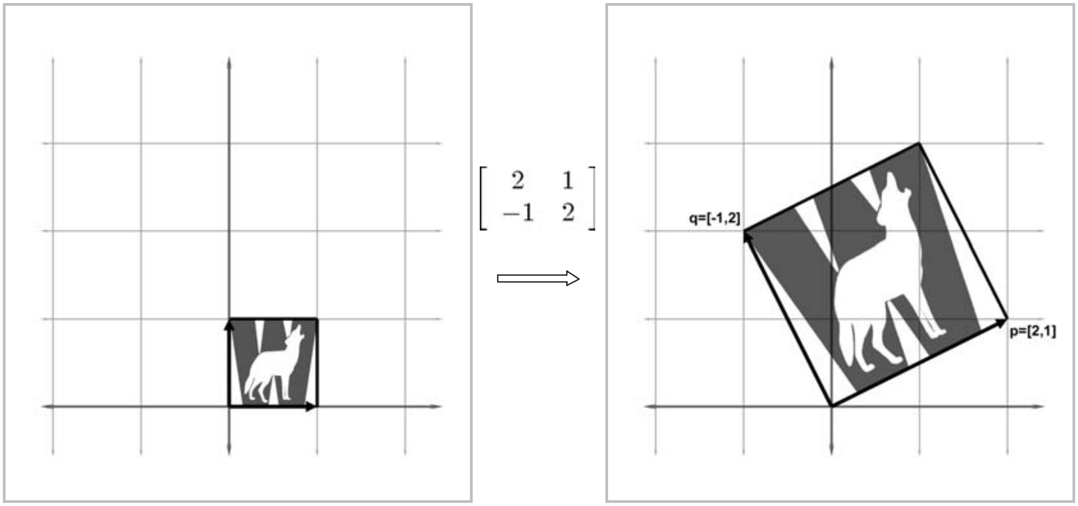


Figure 7.2: The 2D parallelogram formed by the rows of a matrix

This parallelogram is also known as a “skew box.” Drawing an object inside the box can also help:



$$\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$$

Figure 7.3: Drawing an object inside the box helps visualize the transformation

It is clear that our example matrix **M** not only rotates the coordinate space, it also scales it.

We can extend the techniques we used to visualize 2D transformations into 3D. In 2D, we had two basis vectors that formed an “L.” In 3D, we have three basis vectors, and they form a “tripod.” First, let’s show an object before transformation. Figure 7.4 shows a teapot, a unit cube, and the basis vectors in the “identity” position:

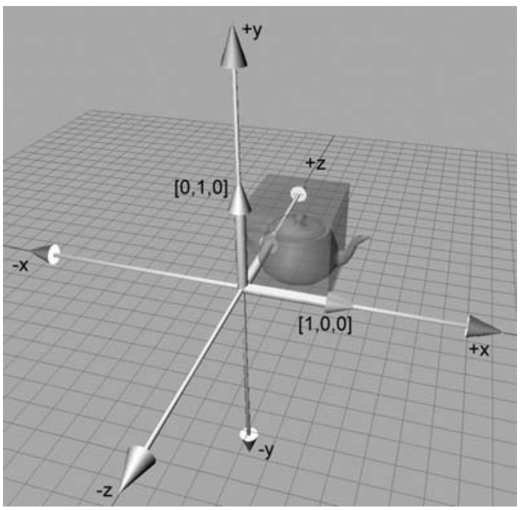


Figure 7.4: Teapot, unit cube, and basis vectors before transformation

(In order to avoid cluttering up the diagram, we have not labeled the +*z* basis vector [0,0,1], which is partially obscured by the teapot and cube.)

Now consider the 3D transformation matrix below:

$$\begin{bmatrix} 0.707 & -0.707 & 0 \\ 1.250 & 1.250 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Extracting the basis vectors from the rows of the matrix, we can visualize the transformation represented by this matrix. The transformed basis vectors, cube, and teapot are shown below:

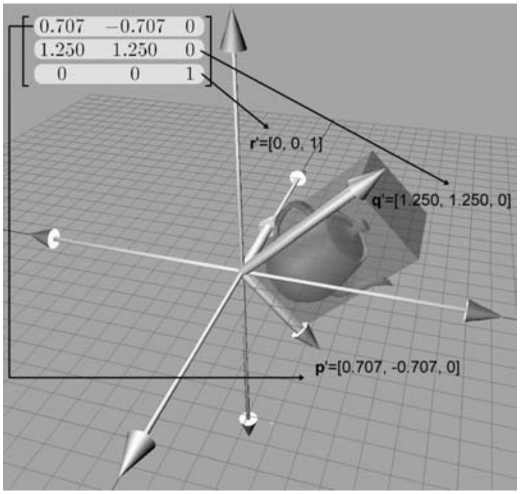


Figure 7.5: Teapot, unit cube, and basis vectors after transformation

As you can see, the transformation consists of a clockwise rotation about the *z*-axis by about 45° and a non-uniform scale that makes the teapot "taller" than it was originally. Notice that the +*z* basis vector was unaltered by the transformation because the third row of the matrix is [0,0,1].

7.2.3 Summary

Before we move on, let's review the key concepts of Section 7.2:

- The rows of a square matrix can be interpreted as the basis vectors of a coordinate space.
- To transform a vector from the original coordinate space to the new coordinate space, we multiply the vector by the matrix.
- The transformation from the original coordinate space to the coordinate space defined by these basis vectors is a linear transformation. A linear transformation preserves straight lines, and parallel lines remain parallel. However, angles, lengths, areas, and volumes may be altered after transformation.
- Multiplying the zero vector by any square matrix results in the zero vector. Therefore, the linear transformation represented by a square matrix has the same origin as the original coordinate space. The transformation does not contain translation.

- We can visualize a matrix by visualizing the basis vectors of the coordinate space after transformation. These basis vectors form an "L" in 2D and a tripod in 3D. Using a box or auxiliary object also helps in visualization.

7.3 Exercises

1. Use the following matrices:

$$\mathbf{A} = \begin{bmatrix} 13 & 4 & -8 \\ 12 & 0 & 6 \\ -3 & -1 & 5 \\ 10 & -2 & 5 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 15 & 8 \\ -7 & 3 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 3 \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} a & g \\ b & h \\ c & i \\ d & j \\ f & k \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

   a. For each matrix **A** through **F** above, give the dimensions of the matrix and identify the matrix as square and/or diagonal.
   b. Determine if the following matrix multiplications are allowed, and if so, give the dimensions of the resulting matrix.
      - DA
      - AD
      - BC
      - AF
      - E^TB
      - DFA
   c. Compute the following transpositions:
      - $\mathbf{A}^T$
      - $\mathbf{E}^T$
      - $\mathbf{B}^T$

2. Compute the following products:

   a. $\begin{bmatrix} 1 & -2 \\ 5 & 0 \end{bmatrix} \begin{bmatrix} -3 & 7 \\ 4 & 1/3 \end{bmatrix}$

   b. $\begin{bmatrix} 3 & -1 & 4 \end{bmatrix} \begin{bmatrix} -2 & 0 & 3 \\ 5 & 7 & -6 \\ 1 & -4 & 2 \end{bmatrix}$

3. Manipulate the following matrix product to remove the parentheses:

$$\left((\mathbf{AB})^T (\mathbf{CDE})^T\right)^T$$

4. What type of transformation is represented by the following 2D matrix?

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$