Chapter 2

Mathematics: Bare Basics

This chapter delivers an intuitive presentation of the basics of mathematics, which are needed throughout this book. Matrix and vector are indispensable tools in computer graphics and this chapter starts with them.

2.1 Matrix and Vector

Shown below is a matrix with m rows and n columns:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \tag{2.1}$$

Its dimension is denoted as $m \times n$. Two subscripts of an element specify where it is located. For instance, a_{12} is at the first row and second column. Tow matrices A and B can be multiplied if the number of columns in A equals the number of rows in B. If A's dimension is $l \times m$ and B's dimension is $m \times n$, AB is an $l \times n$ matrix. See the example shown below:

$$\begin{aligned} AB &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix} \end{aligned} \tag{2.2}$$

A 2D vector is usually represented as (x, y) and a 3D vector is (x, y, z). Vectors are special instances of matrices. For example, (x, y) is a matrix with a single row and its dimension is 1×2. Similarly, the dimension of (x, y, z) is 1×3. In this sense, (x, y) and (x, y, z) are called *row vectors*. Alternatively, we can use the *column vector* representation, e.g., a 2D vector is written as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} \tag{2.3}$$

It is also a special matrix and its dimension is 2×1.

As a vector is taken as a special matrix, the matrix-matrix multiplication applies to matrix-vector multiplication. If M is a 3×2 matrix and v is a 2D column vector, for example, Mv is computed as follows:

$$\begin{aligned} Mv &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \\ a_{31}x + a_{32}y \end{pmatrix} \end{aligned} \tag{2.4}$$

In this example, the resulting 3×1 matrix is simply a 3D vector.

Given a matrix M, its *transpose* denoted by M^T is created by taking the rows of M as the columns of M^T. For example, the transpose of M given in Equation (2.4) is written as follows:

$$\begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{pmatrix} \tag{2.5}$$

Given a column vector v, its row vector representation is obtained by transposing v. It is v^T.

The matrix-vector multiplication in Equation (2.4), Mv, can be represented in a different way. Instead of the column vector, v, we can use a row vector, v^T, but then it is placed at the left-hand side of M^T:

$$\begin{aligned} v^T M^T &= \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}x + a_{12}y & a_{21}x + a_{22}y & a_{31}x + a_{32}y \end{pmatrix} \end{aligned} \tag{2.6}$$

The result is the same as that in Equation (2.4) but is represented in a row vector.

Whereas OpenGL uses the column vectors and the vector-on-the-right representation for matrix-vector multiplication, Direct3D uses the row vectors and the vector-on-the-left representation.

The *identity matrix* is a square matrix with ones on the main diagonal (from the upper-left element to the lower-right element) and zeros elsewhere. It is denoted by I. For any matrix M, $MI = IM = M$, as shown in the following examples:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{2.7}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{2.8}$$

When two square matrices A and B are multiplied to return an identity matrix, i.e., $AB = I$, B is called the *inverse* of A and is denoted by A^{-1}. Equally, A is the inverse of B. It can be proven that $(AB)^{-1} = B^{-1}A^{-1}$. The similar applies to transpose: $(AB)^T = B^T A^T$.

The coordinates of a 2D vector, v, are represented by (v_x, v_y). Its length denoted by $||v||$ is defined as $\sqrt{v_x^2 + v_y^2}$. If v is a 3D vector, its coordinates are (v_x, v_y, v_z) and $||v||$ is $\sqrt{v_x^2 + v_y^2 + v_z^2}$. Dividing a vector by its length is called normalization, and the resulting vector, $\frac{v}{||v||}$, has the same direction as v. Such a normalized vector is called a *unit vector* in that its length is 1.

2.2 Coordinate System and Basis

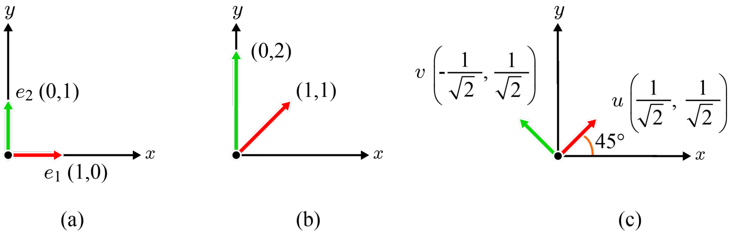

y
e2 (0,1)
e1 (1,0)
x
(a)
y
(0,2)
(1,1)
x
(b)
y
$v\left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$
$u\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$
45°
x
(c)

Fig. 2.1: Basis examples. (a) Standard basis. (b) A valid basis that is neither standard nor orthonormal. (c) Another valid basis that is not standard but orthonormal.

In the 2D coordinate system shown in Fig. 2.1-(a), consider two vectors, e_1 and e_2. Every 2D vector can be defined as a *linear combination*[1] of e_1 and e_2, e.g., $(3,5) = 3e_1 + 5e_2$. In this sense, $\{e_1, e_2\}$ is called a *basis*. As e_1 is along the x-axis and e_2 is along the y-axis, $\{e_1, e_2\}$ is named the *standard* basis. Observe that e_1 and e_2 are unit vectors that are orthogonal to each other, i.e., e_1 and e_2 are *orthonormal*.

A different set of vectors may work as a basis. Consider $\{(1,1),(0,2)\}$ shown in Fig. 2.1-(b). The example vector, $(3,5)$, can be defined by linearly combining $(1,1)$ and $(0,2)$, i.e., $(3,5) = 3(1,1) + 1(0,2)$. Observe that

[1] Given n vectors, $v_1, v_2, \ldots, v_n$, and n scalars, $c_1, c_2, \ldots, c_n$, the linear combination of the vectors with the scalars is $c_1v_1 + c_2v_2 + \cdots + c_nv_n$. It is also a vector.

$\{(1,1),(0,2)\}$ is not an orthonormal basis. It is generally easier to work with an orthonormal basis rather than an arbitrary basis. Fig. 2.1-(c) shows an orthonormal basis, which is not standard. Note that $(3,5) = 4\sqrt{2}u + \sqrt{2}v$.

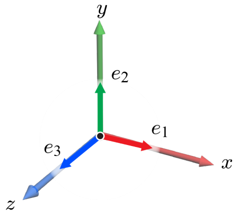


Fig. 2.2: Standard basis for the 3D coordinate system.

As shown in Fig. 2.2, the standard basis in the 3D coordinate system is $\{e_1, e_2, e_3\}$, where $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, and $e_3 = (0,0,1)$. Obviously it is an orthonormal basis.

2.3 Dot Product

Consider two n-dimensional vectors, a and b. When their coordinates are represented by $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$, respectively, their dot product, denoted by $a \cdot b$, is defined as follows:

$$a \cdot b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n \tag{2.9}$$

Dot product is also defined as follows:

$$a \cdot b = ||a|| ||b|| cos\theta \tag{2.10}$$

where θ is the angle between a and b. Note that $a \cdot b = 0$ if a and b are perpendicular to each other. See Fig. 2.3 for 2D cases. If θ is an acute angle, $a \cdot b$ is positive; if θ is an obtuse angle, $a \cdot b$ is negative. The same observation can be made when a and b are 3D vectors.

Suppose that a is a unit vector, i.e., $\|a\| = 1$. Then, $a \cdot b$ becomes $\|b\| cos\theta$. It is the length of b projected onto a. See Fig. 2.4-(a). The projected length

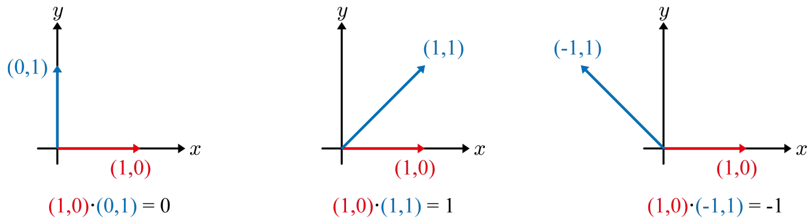
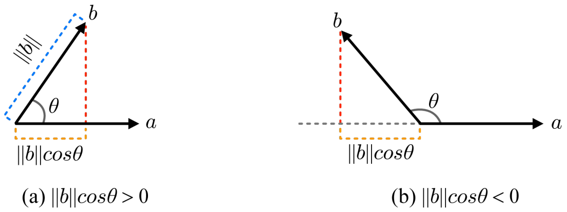


Fig. 2.3: Dot product of two vectors reveals their relative orientation.



(a) $||b||cos\theta > 0$

(b) $||b||cos\theta < 0$

Fig. 2.4: Signed length. (a) The projected length $||b||cos\theta$ is positive. (b) The projected length becomes negative because $cos\theta$ is negative.

is negative if θ is an obtuse angle, as shown in Fig. 2.4-(b). In this sense, $||b||cos\theta$ is called the *signed length*.

Note that, given a vector v, $||v||^2 = v \cdot v$. If v is a unit vector, $v \cdot v = 1$. An orthonormal basis has an interesting and useful feature. Consider the standard basis as an example. In the 2D standard basis, $\{e_1, e_2\}$, the following holds: $e_1 \cdot e_1 = e_2 \cdot e_2 = 1$ and $e_1 \cdot e_2 = e_2 \cdot e_1 = 0$. In the 3D standard basis, $\{e_1, e_2, e_3\}$, a similar observation is made: $e_i \cdot e_j = 1$ if $i = j$, and $e_i \cdot e_j = 0$ otherwise. This applies to any orthonormal basis.

2.4 Cross Product

The cross product takes as input two 3D vectors, a and b, and outputs another 3D vector which is perpendicular to both a and b, as shown in Fig. 2.5-(a). The cross product is denoted by $a \times b$ and its direction is defined by the *right-hand rule*: The direction of $a \times b$ is indicated by the thumb of the

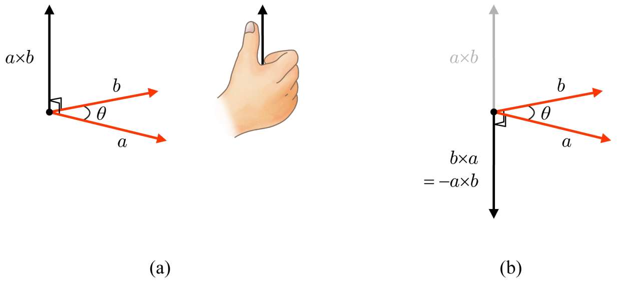


Fig. 2.5: Cross product and right-hand rule. (a) $a \times b$. (b) $b \times a$.

right hand when the other fingers curl from a to b. The length of $a \times b$ equals the area of the parallelogram that a and b span:

$$||a \times b|| = ||a||||b|| \sin\theta \tag{2.11}$$

The right-hand rule implies that the direction of $b \times a$ is opposite to that of $a \times b$, i.e., $b \times a = -a \times b$, but their lengths are the same, as illustrated in Fig. 2.5-(b). In this sense, the cross product operation is called *anti-commutative*.

When the coordinates of a and b are represented by (a_x, a_y, a_z) and (b_x, b_y, b_z), respectively, those of $a \times b$ are $(a_yb_z - a_zb_y, a_zb_x - a_xb_z, a_xb_y - a_yb_x)$. See [Note: Derivation of cross product][2].

[Note: Derivation of cross product]

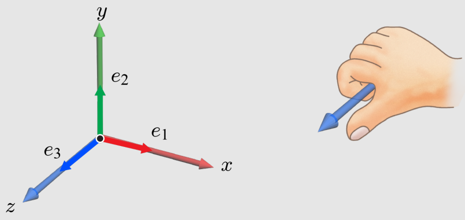


Fig. 2.6: Standard basis and right-hand rule. The thumb of the right hand points toward the positive end of e_3 when the other four fingers curl from e_1 to e_2, i.e., $e_1 \times e_2 = e_3$. We also obtain that $e_2 \times e_3 = e_1$ and $e_3 \times e_1 = e_2$.

[2] In this book, the note in a shaded box can be skipped over, and no difficulty will be encountered for further reading.

In Fig. 2.6, the relative orientation among the basis vectors, e_1, e_2, and e_3, is described using the right-hand rule:

$$\begin{aligned} e_1 \times e_2 &= e_3 \\ e_2 \times e_3 &= e_1 \\ e_3 \times e_1 &= e_2 \end{aligned} \tag{2.12}$$

The anti-commutativity of the cross product leads to the following:

$$\begin{aligned} e_2 \times e_1 &= -e_3 \\ e_3 \times e_2 &= -e_1 \\ e_1 \times e_3 &= -e_2 \end{aligned} \tag{2.13}$$

Equation (2.11) also asserts that

$$e_1 \times e_1 = e_2 \times e_2 = e_3 \times e_3 = \mathbf{0} \tag{2.14}$$

where $\mathbf{0}$ is the *zero vector*, $(0, 0, 0)$.

When $a = (a_x, a_y, a_z)$ and $b = (b_x, b_y, b_z)$, a is rewritten in terms of the standard basis as $a_x e_1 + a_y e_2 + a_z e_3$. Similarly, b is rewritten as $b_x e_1 + b_y e_2 + b_z e_3$. Then, $a \times b$ is derived as follows:

$$\begin{aligned} a \times b &= (a_x e_1 + a_y e_2 + a_z e_3) \times (b_x e_1 + b_y e_2 + b_z e_3) \\ &= a_x b_x (e_1 \times e_1) + a_x b_y (e_1 \times e_2) + a_x b_z (e_1 \times e_3) + \\ &\quad\ a_y b_x (e_2 \times e_1) + a_y b_y (e_2 \times e_2) + a_y b_z (e_2 \times e_3) + \\ &\quad\ a_z b_x (e_3 \times e_1) + a_z b_y (e_3 \times e_2) + a_z b_z (e_3 \times e_3) \\ &= a_x b_x \mathbf{0} + a_x b_y e_3 - a_x b_z e_2 \\ &\quad\ - a_y b_x e_3 + a_y b_y \mathbf{0} + a_y b_z e_1 \\ &\quad\ + a_z b_x e_2 - a_z b_y e_1 + a_z b_z \mathbf{0} \\ &= (a_y b_z - a_z b_y) e_1 + (a_z b_x - a_x b_z) e_2 + (a_x b_y - a_y b_x) e_3 \end{aligned} \tag{2.15}$$

i.e., the coordinates of $a \times b$ are $(a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x)$.

2.5 Line, Ray, and linear Interpolation

A line is defined by two points. Fig. 2.7-(a) shows a line defined by p_0 and p_1. The line is equivalently defined by p_0 and the vector, $p_1 - p_0$, which connects p_0 and p_1. Consider the equation in parameter t:

$$p(t) = p_0 + t(p_1 - p_0) \tag{2.16}$$

The function $p(t)$ maps a scalar value of t to a specific point in the line. When $t = 1$, for example, $p(t) = p_1$. When $t = 0.5$, $p(t)$ represents the midpoint between p_0 and p_1. Fig. 2.7-(a) shows a few other instances of $p(t)$.

In Equation (2.16), t is in the range of $[-\infty, \infty]$ and $p(t)$ represents an infinite line. If t is limited to the range of $[0, \infty]$, $p(t)$ represents a *ray*. It starts from p_0 and is infinitely extended along the direction vector, $p_1 - p_0$. In contrast, when t is limited to a finite range, $p(t)$ represents a *line segment*. If the range is $[0, 1]$, for example, $p(t)$ is a line segment connecting p_0 and p_1.

Equation (2.16) can be rephrased as follows:

$$p(t) = (1-t)p_0 + tp_1 \tag{2.17}$$

If t is in the range $[0, 1]$, $p(t)$ corresponds to a *linear interpolation* of p_0 and p_1. As illustrated in Fig. 2.7-(b), it is described as a *weighted sum* of two points: the weights for p_0 and p_1 are $(1-t)$ and t, respectively.

The function $p(t)$ is vector-valued, e.g., $p(t) = (x(t), y(t), z(t))$ for 3D space. When $p_0 = (x_0, y_0, z_0)$ and $p_1 = (x_1, y_1, z_1)$, linear interpolation is applied to each of the x-, y-, and z-coordinates:

$$p(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} (1-t)x_0 + tx_1 \\ (1-t)y_0 + ty_1 \\ (1-t)z_0 + tz_1 \end{pmatrix} \tag{2.18}$$

Whatever attributes are associated with the end points, they can be linearly interpolated. Suppose that the endpoints are associated with colors c_0 and c_1, respectively, where $c_0 = (R_0, G_0, B_0)$ and $c_0 = (R_1, G_1, B_1)$. Then, the color $c(t)$ is defined as follows:

$$c(t) = (1-t)c_0 + tc_1 = \begin{pmatrix} (1-t)R_0 + tR_1 \\ (1-t)G_0 + tG_1 \\ (1-t)B_0 + tB_1 \end{pmatrix} \tag{2.19}$$

Fig. 2.7-(c) shows some examples of color interpolation. When $t = 0$, $c(t) = c_0$. As t increases, $c(t)$ becomes closer to c_1. When $t = 1$, $c(t) = c_1$.

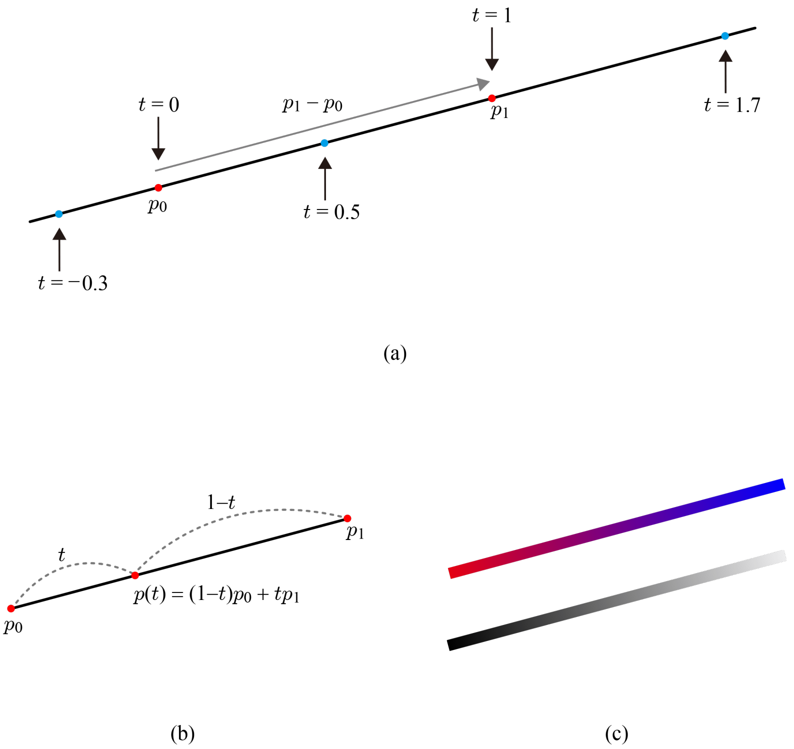


Fig. 2.7: Line and linear interpolation. (a) The line connecting p_0 and p_1 is described by adding $t(p_1 - p_0)$ to p_0. When t is restricted to $[0, 1]$, the line segment between p_0 and p_1 is obtained. (b) The line segment between p_0 and p_1 is represented as a linear interpolation of p_0 and p_1. (c) Color interpolation examples.