23 向量

Vectors Meet Coordinate Systems

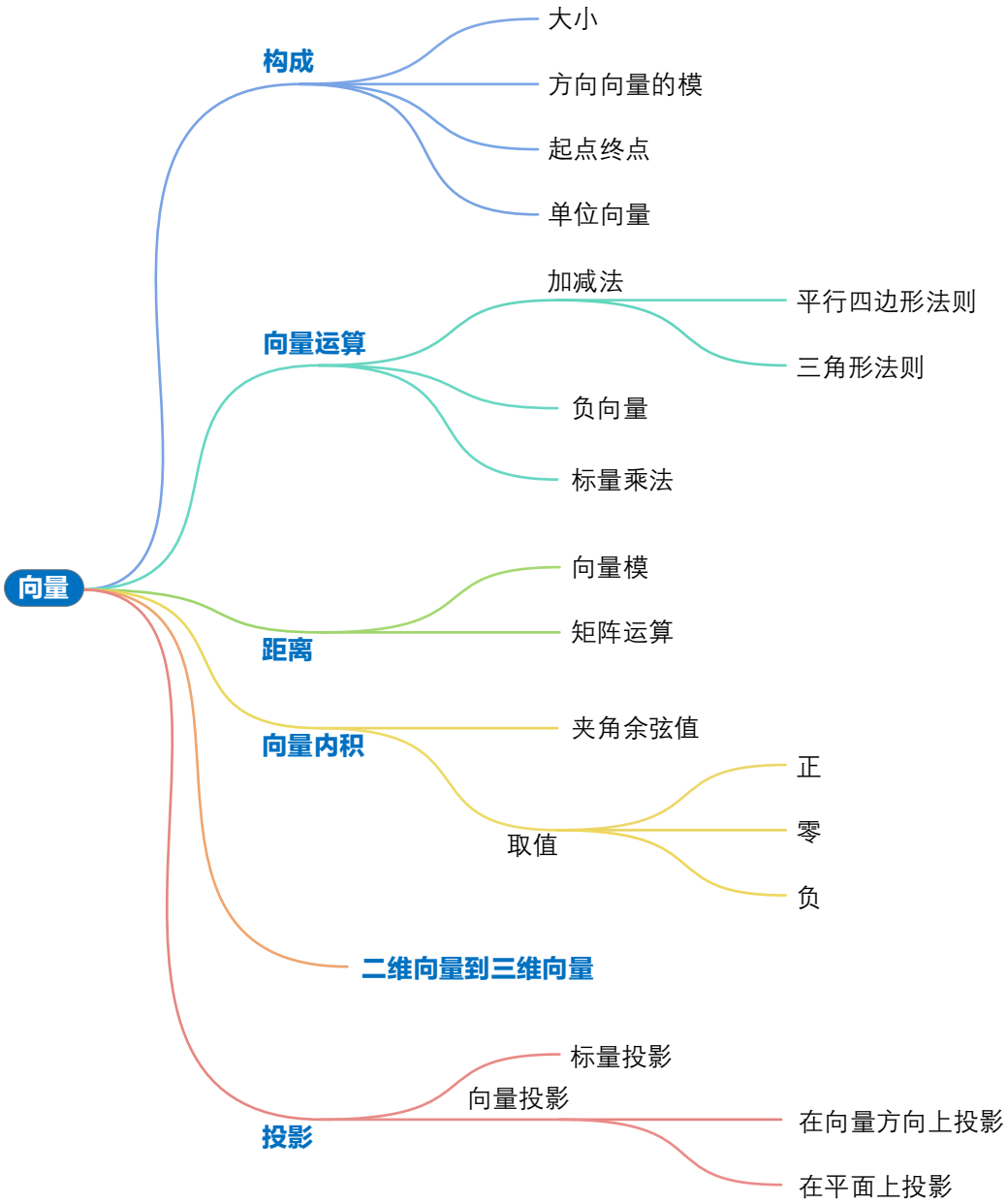
向量遇见坐标系

曾几何时，代数和几何形单影只、踽踽独行；它们各自，蜗步难移、难成大器。然而，代数和几何一见如故，便珠联璧合、琴瑟和鸣；两者取长补短、急流勇进、日臻完美。

As long as algebra and geometry proceeded along separate paths, their advance was slow and their applications limited. But when these sciences joined company, they drew from each other fresh vitality and thenceforward marched on at a rapid pace toward perfection.

—— 约瑟夫·拉格朗日 (Joseph Lagrange) | 法国籍意大利裔数学家和天文学家 | 1736 ~ 1813

- `matplotlib.pyplot.annotate()` 在平面坐标系标注
- `matplotlib.pyplot.quiver()` 绘制箭头图
- `matplotlib.pyplot.quiver()` 绘制箭头图
- `numpy.arccos()` 反余弦
- `numpy.degrees()` 将弧度转化为角度
- `numpy.dot()` 计算向量标量积。值得注意的是，如果输入为一维数组，`numpy.dot()` 输出结果为标量积；如果输入为矩阵，`numpy.dot()` 输出结果为矩阵乘积，相当于矩阵运算符@
- `numpy.linalg.norm()` 计算范数



23.1 向量：有大小、有方向

很多的数学工具的发明的时候，并没有具体的用途。经常发生的情况是，几十年之后、甚至几百年之后，科学家应用某个被尘封的数学工具，完成科学技术的巨大飞跃。本书前文介绍的圆锥曲线就是很好的例子。

但是，也有部分数学工具是为了更好地描述其他学科发现的新理论而创造发展的，比如说向量这个概念。

向量这一定义的发明经过了漫长的 200 年岁月。几乎难以想象，现在大家熟知的向量的记法和运算，竟然是在 19 世纪末才加入数学这个大家庭。

1865 年开始，苏格兰数学物理学家**麦克斯韦** (James Clerk Maxwell) 逐步提出将电、磁场、光统一起来的麦克斯韦方程组。为了更好描述麦克斯韦方程组，美国科学家 Josiah Willard Gibbs 和英国科学家 Oliver Heaviside 分别独立发明了向量的现代记法。

向量是一行或一列数字

本书第一章就介绍了向量。从数据角度，向量无非就是一列或一行数字。如图 1 所示，数据矩阵 $\boldsymbol{X}$ 的每一行是一个行向量，代表一个观察值；$\boldsymbol{X}$ 的每一列为一个列向量，代表某个特征上的所有数据。

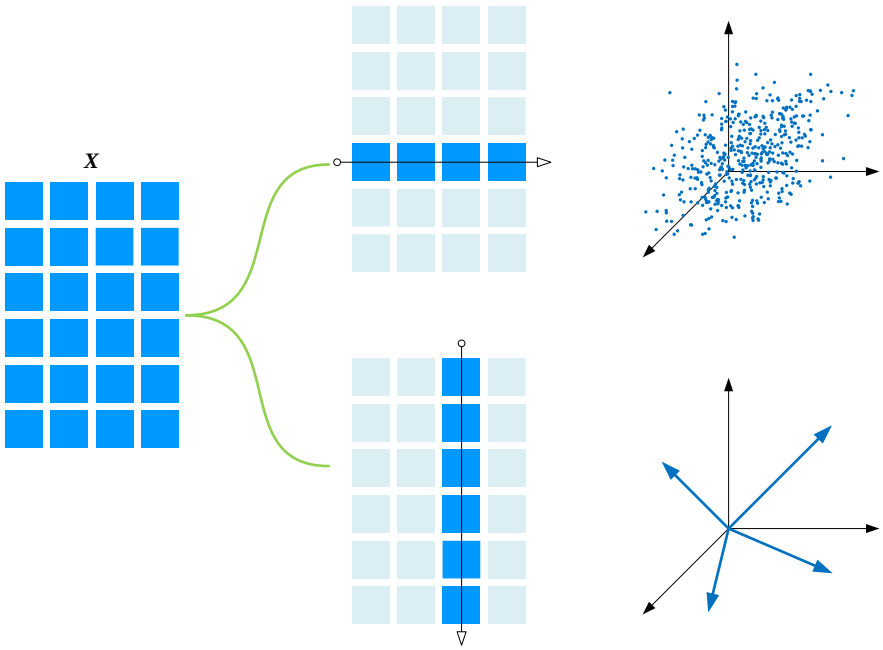


图 1. 观察数据的两个角度

向量的几何意义

但是有了坐标系，向量便不再是无趣的数字，它们变成了一只离弦之箭，可以腾飞。

向量 (vector) 是**既有长度又有方向的量** (a quantity that possesses both magnitude and direction)。物理学中，**位移** (displacement)、**速度** (velocity)、**加速度** (acceleration)、**力** (force) 等物理量都是向量。

如图 2 所示，位移向量的大小代表前进的距离大小，而向量的方向代表位移方向。和向量相对的是标量；**标量** (scalar, scalar quantity) 是有大小没有方向的量，用实数表示。

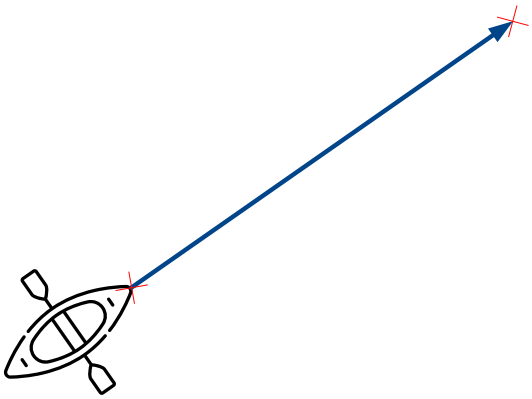


图 2. 位移向量

向量定义

给定一个列向向量 $\boldsymbol{a}$:

$$\boldsymbol{a}=\begin{bmatrix}4\\3\end{bmatrix} \tag{1}$$

如图 3 所示，向量 $\boldsymbol{a}$ 可以表达为一个带箭头的线段，它以原点 O (0, 0)为起点，指向终点 A (4, 3)；因此，向量 $\boldsymbol{a}$ 也可以写作 $\overrightarrow{OA}$:

$$\overrightarrow{OA}=\begin{bmatrix}4\\3\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}4\\3\end{bmatrix} \tag{2}$$

注意，丛书很少使用 $\overrightarrow{OA}$ 这种向量记法。

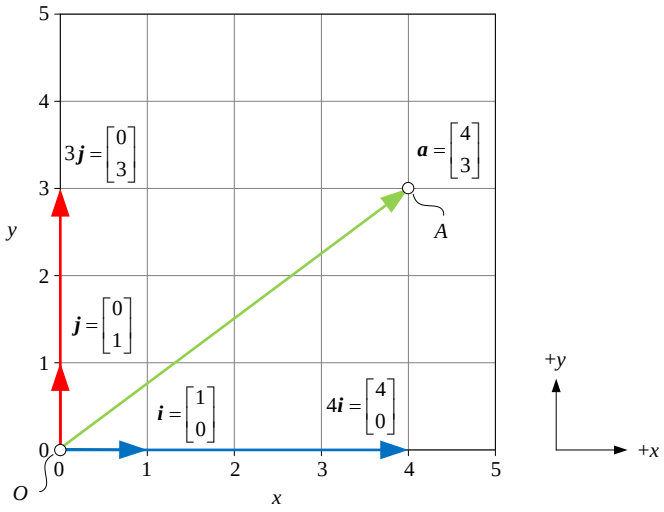


图 3. 平面直角坐标系，向量 $\boldsymbol{a}$ 的定义

向量模

如图 3 所示，向量 $\boldsymbol{a}$ 的长度就是线段 OA 的长度：

$$\|\boldsymbol{a}\| = \sqrt{4^2 + 3^2} = 5 \tag{3}$$

$\|\boldsymbol{a}\|$ 计算向量 $\boldsymbol{a}$ 的长度，也叫模，也叫 L^2 范数。L^2 范数是 L^p 范数的一种。

向量分解

类似物理学中力的分解，向量 $\boldsymbol{a}$ 还可以写成如下两个向量的和：

$$\boldsymbol{a} = 4\boldsymbol{i} + 3\boldsymbol{j} = 4\times\begin{bmatrix}1\\0\end{bmatrix} + 3\times\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}4\\3\end{bmatrix} \tag{4}$$

$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 常被称作横轴和纵轴上的单位向量，具体定义为：

$$\boldsymbol{i} = \begin{bmatrix}1\\0\end{bmatrix}, \quad \boldsymbol{j} = \begin{bmatrix}0\\1\end{bmatrix} \tag{5}$$

单位向量 (unit vector) 是指模 (长度) 等于 1 的向量，即：

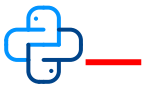
$$\|\boldsymbol{i}\| = 1, \quad \|\boldsymbol{j}\| = 1 \tag{6}$$

本系列丛书后续也会使用 $\boldsymbol{e}_1$ 和 $\boldsymbol{e}_2$ 代表横轴纵轴的单位向量。

任何非零向量除以自身的模，得到向量方向上的单位向量。(1) 中向量 $\boldsymbol{a}$ 的单位向量为：

$$\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} = \frac{1}{5}\times\begin{bmatrix}4\\3\end{bmatrix} = \begin{bmatrix}0.8\\0.6\end{bmatrix} \tag{7}$$

以下代码绘制图 3。



```
# Bk3_Ch23_1

import numpy as np
import matplotlib.pyplot as plt

# draw vectors starting from origin

def draw_vector(vector,RBG,label):
    array = np.array([[0, 0, vector[0], vector[1]]])
    X, Y, U, V = zip(*array)
    plt.quiver(X, Y, U, V,angles='xy', scale_units='xy',scale=1,color = RBG)

    # add labels to the sample data

    label = label + f" ({vector[0]},{vector[1]})"

    plt.annotate(label, # text
                 (vector[0],vector[1]), # point to label
                 textcoords="offset points",
                 xytext=(0,10),
                 # distance from text to points (x,y)
                 ha='center')
                 # horizontal alignment center

# define one vector
a = np.array([4,3])
i = np.array([1,0])
j = np.array([0,1])

fig, ax = plt.subplots()

draw_vector(4*i, np.array([0,112,192])/255, '4i')
draw_vector(3*j, np.array([255,0,0])/255, '3j')

draw_vector(i, np.array([0,112,192])/255, 'i')
draw_vector(j, np.array([255,0,0])/255, 'j')

draw_vector(a,np.array([146,208,80])/255, 'a')

plt.xlabel('$x$')
plt.ylabel('$y$')

plt.axis('scaled')
ax.set_xlim([0, 5])
ax.set_ylim([0, 5])
ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
plt.show()
```

23.2 几何视角看向量运算

有了平面直角坐标系，向量的加法、减法和标量乘法更容易理解。

向量加法

图 4 所示 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相加对应的等式为。

$$\boldsymbol{a}+\boldsymbol{b}=\begin{bmatrix}4\\1\end{bmatrix}+\begin{bmatrix}1\\3\end{bmatrix}=\begin{bmatrix}5\\4\end{bmatrix} \tag{8}$$

如图 4 所示，从几何角度 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相加相当于物理学中两个力的合成。

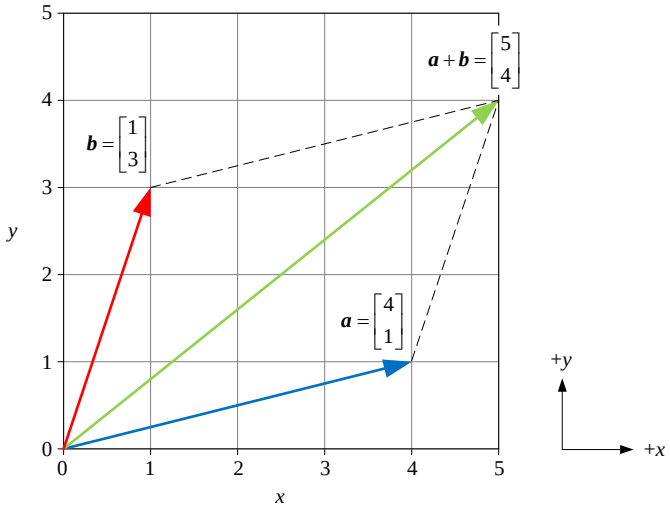


图 4. $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相加

正四边形和三角形法则

平面直角坐标系上，两个向量相加有两种方法——正四边形法则 (parallelogram law)、三角形法则 (triangle law)。

图 5 (a) 所示为平行四边形法则。将向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 平移至公共起点，以 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的两条边作平行四边形，$\boldsymbol{a}+\boldsymbol{b}$ 为公共起点所在平行四边形对角线。

三角形法则则更为常用。如图 5 (b) 所示，在平面内，将向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 首尾相连，$\boldsymbol{a}+\boldsymbol{b}$ 的结果对应为向量 $\boldsymbol{a}$ 的起点与向量 $\boldsymbol{b}$ 的终点相连构成的向量；也就是说 $\boldsymbol{a}+\boldsymbol{b}$ 始于 $\boldsymbol{a}$ 的起点，指向 $\boldsymbol{b}$ 终点。

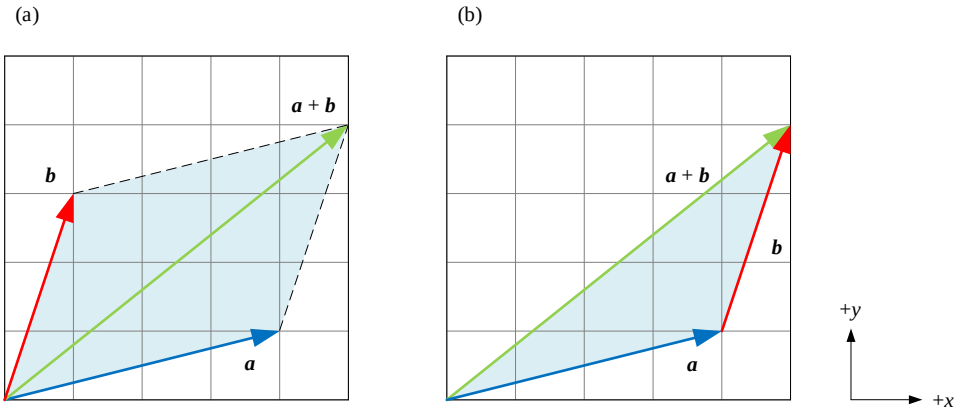


图 5. 平行四边形法则和三角形法则计算 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相加

n 个向量相加，$\boldsymbol{a}_1 + \boldsymbol{a}_2 + \ldots + \boldsymbol{a}_n$，将它们首尾相连，第一个向量 $\boldsymbol{a}_1$ 的起点与最后一个向量 $\boldsymbol{a}_n$ 的终点相连构成的向量。图 6 所示为采用三角形法则计算 5 个向量相加。

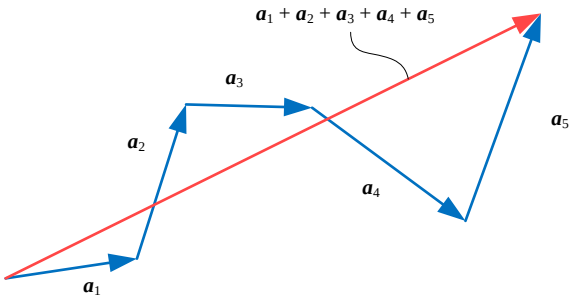


图 6. 三角形法则计算 5 个向量相加

向量减法

$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相减。

$$\boldsymbol{a}-\boldsymbol{b}=\begin{bmatrix}4\\1\end{bmatrix}-\begin{bmatrix}1\\3\end{bmatrix}=\begin{bmatrix}3\\-2\end{bmatrix} \tag{9}$$

从几何角度，也可以用三角形法则来求解。

如图 7 所示，将 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量平移至公共起点，以 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 作为两边构造三角形，向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的终点连线为第三条边；这个三角形的第三边就是 $\boldsymbol{a}-\boldsymbol{b}$ 的结果，$\boldsymbol{a}-\boldsymbol{b}$ 的方向为减向量 $\boldsymbol{b}$ 终点指向被减向量 $\boldsymbol{a}$ 终点。

此外，$\boldsymbol{a}-\boldsymbol{b}$ 可以 $\boldsymbol{a}$ 视作和–$\boldsymbol{b}$ 相加。–$\boldsymbol{b}$ 叫做 $\boldsymbol{b}$ 的负向量。从数字角度，$\boldsymbol{b}$ 和–$\boldsymbol{b}$ 对应元素为相反数；从几何角度角度，$\boldsymbol{b}$ 和–$\boldsymbol{b}$ 大小相同、方向相反；也就是说，$\boldsymbol{b}$ 和–$\boldsymbol{b}$ 起点终点相互调换。

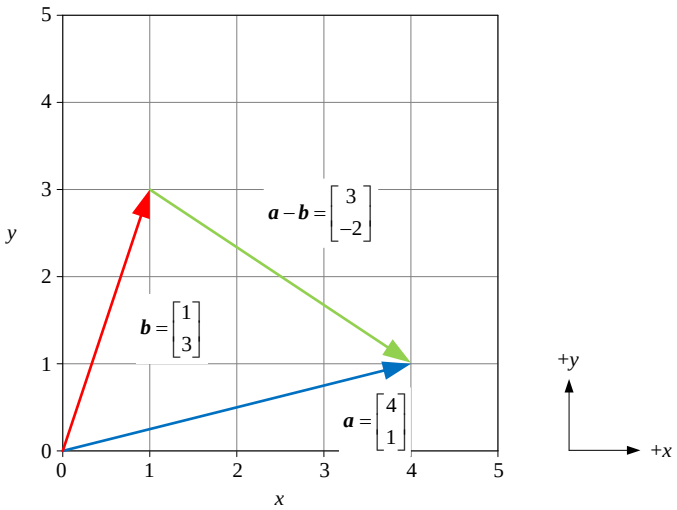


图 7. $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量相减，三角形法则

标量乘法

图 8 所示 $\boldsymbol{a}$ 的两个标量乘法。

$$0.5\boldsymbol{a} = 0.5 \times \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad 2\boldsymbol{a} = 2 \times \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \tag{10}$$

从几何角度，向量的标量乘法就是向量的缩放——长度缩放、方向不变或反向。

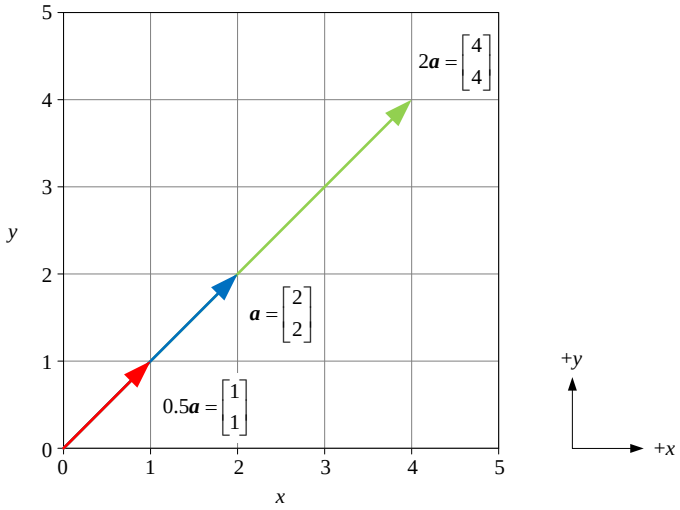


图 8. 向量 $\boldsymbol{a}$ 的标量乘法

以下代码绘制图 4、图 7、图 8。

```
# Bk3_Ch23_2

import numpy as np
import matplotlib.pyplot as plt

# draw vectors starting from origin

def draw_vector(vector,RBG,label):
    array = np.array([[0, 0, vector[0], vector[1]]])
    X, Y, U, V = zip(*array)
    plt.quiver(X, Y, U, V,angles='xy', scale_units='xy',scale=1,color = RBG)

    # add labels to the sample data

    label = label + f" ({vector[0]},{vector[1]})"

    plt.annotate(label, # text
                 (vector[0],vector[1]), # point to label
                 textcoords="offset points",
                 xytext=(0,10),
                 # distance from text to points (x,y)
                 ha='center')
```

```
                # horizontal alignment center

# define two vectors

a = np.array([4,1])
b = np.array([1,3])

# addition of a and b

fig, ax = plt.subplots()

draw_vector(a, np.array([0,112,192])/255, 'a')
draw_vector(b, np.array([255,0,0])/255, 'b')
draw_vector(a + b,np.array([146,208,80])/255, 'a + b')

plt.xlabel('$x$')
plt.ylabel('$y$')

plt.axis('scaled')
ax.set_xlim([0, 5])
ax.set_ylim([0, 5])
ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
plt.show()

# subtraction, a - b
fig, ax = plt.subplots()

draw_vector(a, np.array([0,112,192])/255, 'a')
draw_vector(b, np.array([255,0,0])/255, 'b')
draw_vector(a - b,np.array([146,208,80])/255, 'a - b')

plt.xlabel('$x$')
plt.ylabel('$y$')

plt.axis('scaled')
ax.set_xlim([0, 5])
ax.set_ylim([-2, 3])
ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
plt.show()

a = np.array([2,2])

# scalar multiplication
fig, ax = plt.subplots()

draw_vector(2*a, np.array([146,208,80])/255, '2*a')
draw_vector(a, np.array([0,112,192])/255, 'a')
draw_vector(0.5*a, np.array([255,0,0])/255, '0.5*a')


plt.xlabel('$x$')
plt.ylabel('$y$')

plt.axis('scaled')
ax.set_xlim([0, 5])
ax.set_ylim([0, 5])
ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
plt.show()
```

23.3 向量简化距离运算

本书前文介绍计算两点之间的距离。本节引入向量计算，距让离计算变得更直观。

先看第一种解释。图 9 所示，给定 $A\,(x_A, y_A)$ 和 $B\,(x_B, y_B)$ 两点，B 为起点 A 点为终点的向量 $\overrightarrow{BA}$ 可以写作：

$$\overrightarrow{BA} = \begin{bmatrix} x_A - x_B \\ y_A - y_B \end{bmatrix} \tag{11}$$

而 $\overrightarrow{BA}$ 的向量模便对应 AB 线段的长度：

$$\left\|\overrightarrow{BA}\right\| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} = \sqrt{(4-1)^2 + (1-3)^2} = \sqrt{13} \tag{12}$$

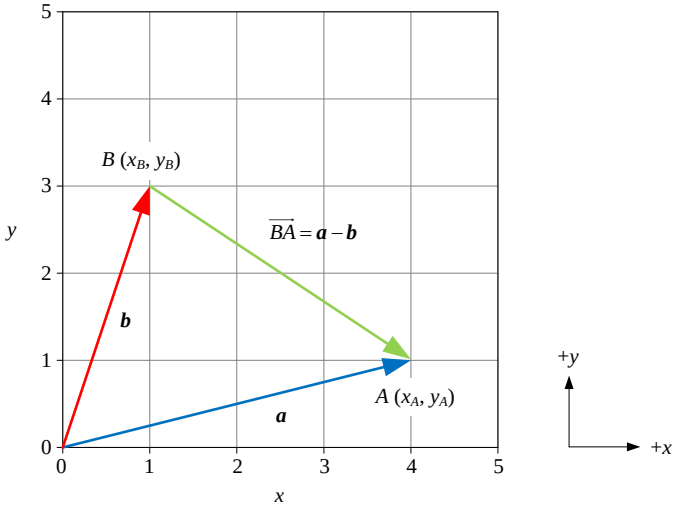


图 9. 计算 A 和 B 距离，即 AB 长度

向量模

另外一个角度，将 A 和 B 的坐标写成向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，AB 距离可以用 $\boldsymbol{a} - \boldsymbol{b}$ 模来计算。

$$d = \|\boldsymbol{a} - \boldsymbol{b}\| = \sqrt{(\boldsymbol{a} - \boldsymbol{b}) \cdot (\boldsymbol{a} - \boldsymbol{b})} \tag{13}$$

$\boldsymbol{a} - \boldsymbol{b}$ 为。

$$\boldsymbol{a} - \boldsymbol{b} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix} \tag{14}$$

将 (14) 代入 (13)，得到。

$$d = \|\boldsymbol{a} - \boldsymbol{b}\| = \sqrt{3^2 + (-2)^2} = \sqrt{13} \tag{15}$$

如果 $\boldsymbol{a}$、$\boldsymbol{b}$ 均为列向量，用用矩阵乘法来写 (13)：

$$d = \sqrt{(\boldsymbol{a} - \boldsymbol{b})^{\mathrm{T}} (\boldsymbol{a} - \boldsymbol{b})} = \sqrt{\begin{bmatrix} 3 \\ -2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 3 \\ -2 \end{bmatrix}} = \sqrt{13} \tag{16}$$

23.4 向量内积与向量夹角

向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的内积的定义为：

$$\boldsymbol{a}\cdot\boldsymbol{b}=\|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos\theta \tag{17}$$

在坐标系的助力下，向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的内积有了几何含义。

图 10 所示，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的内积为 $\boldsymbol{a}$ 的模乘向量 $\boldsymbol{b}$ 在向量 $\boldsymbol{a}$ 方向上的分量值；$\boldsymbol{b}$ 在 $\boldsymbol{a}$ 方向上的分量值为 $\|\boldsymbol{b}\|\cos\theta$，这个值也叫 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 方向上的标量投影。

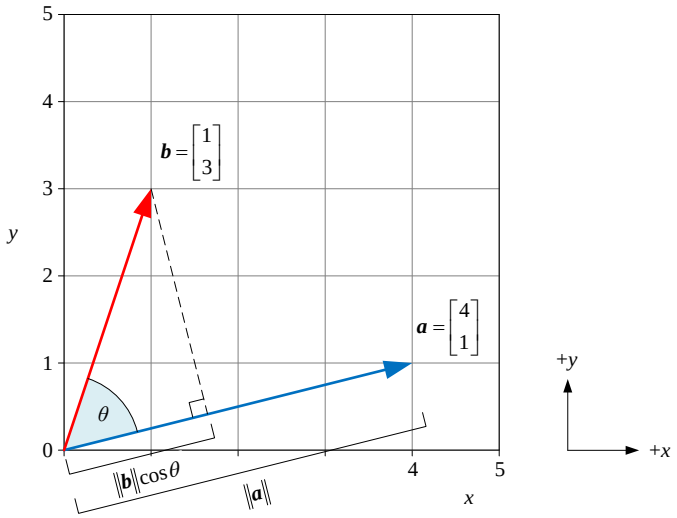


图 10. 内积的定义

向量夹角

这样，向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角 θ 的余弦值可以通过下式求得。

$$\cos\theta=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \tag{18}$$

举个例子，图 4 中 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角余弦值。

$$\cos\theta=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}=\frac{1\times4+3\times1}{\sqrt{10}\sqrt{17}}\approx0.537 \tag{19}$$

通过反余弦求得角度约为 $\theta=57.53°$。以下代码计算 θ。

```
# Bk3_Ch23_3

import numpy as np
a = [4,1]
b = [1,3]

# L2 norms
a_norm = np.linalg.norm(a)
b_norm = np.linalg.norm(b)

# dot product
a_dot_b = np.dot(a, b)

# cosine result
cos_result = a_dot_b/a_norm/b_norm

# radian to degree
# np.arccos(cos_result)*180/np.pi
angle = np.degrees(np.arccos(cos_result))

print(angle)
```

内积正负

如图 11 所示，两个向量内积结果为正，说明向量夹角在 0°到 90° (包括 0°，不包括 90°)；内积结果为 0，说明两个向量垂直；内积结果为负，向量夹角在 90°到 180° (包括 180°，不包括 90°)。

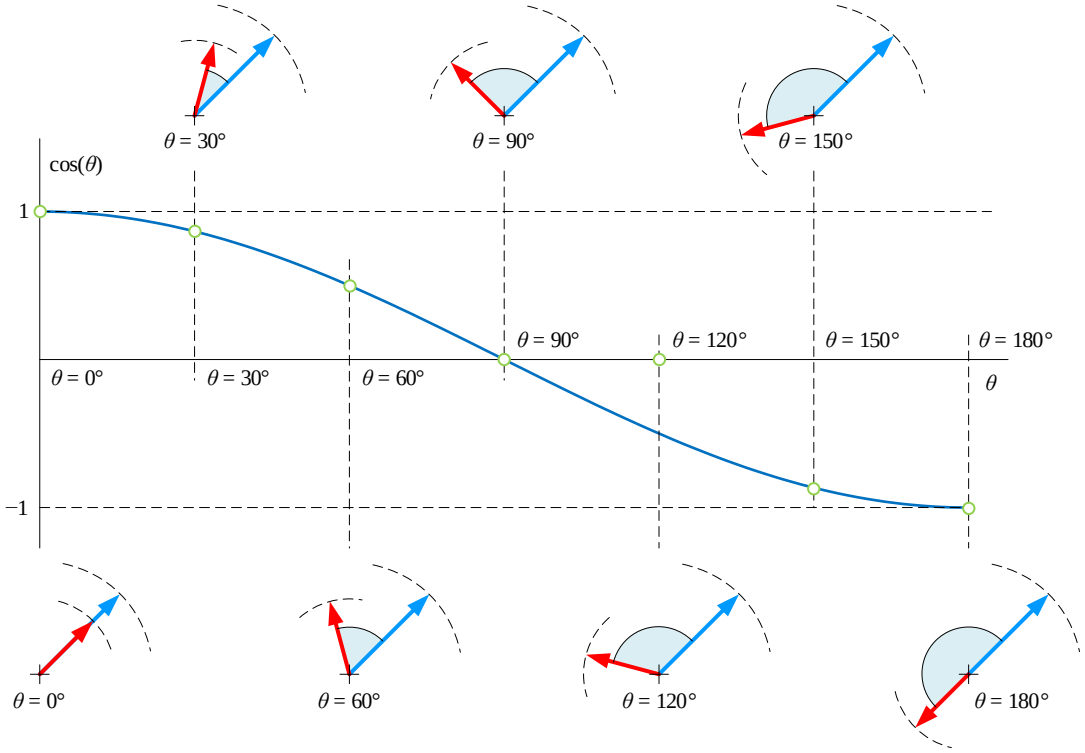


图 11. 向量夹角和余弦值关系

在平面直角坐标系中，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 相互垂直，因此两者内积为 0。

$$\boldsymbol{i}\cdot\boldsymbol{j}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}0\\1\end{bmatrix}=\boldsymbol{i}^{\mathrm{T}}\boldsymbol{j}=\begin{bmatrix}1 & 0\end{bmatrix}@\begin{bmatrix}0\\1\end{bmatrix}=0 \tag{20}$$

23.5 二维到三维

本章前文定义 $\boldsymbol{i}$ 和 $\boldsymbol{j}$ 为二维平面直角坐标系横轴和纵轴上的单位向量，在它们的基础上各加一行 0 就可以得到三维坐标系的横轴和纵轴单位向量。

$$\boldsymbol{i}=\begin{bmatrix}1\\0\\0\end{bmatrix},\quad \boldsymbol{j}=\begin{bmatrix}0\\1\\0\end{bmatrix} \tag{21}$$

另外，再加一个表达 z 轴正方向的单位向量 $\boldsymbol{k}$，我们就构造三维正交单位向量 $\boldsymbol{i}$、$\boldsymbol{j}$、$\boldsymbol{k}$。

$$\boldsymbol{k}=\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{22}$$

利用 $\boldsymbol{i}$、$\boldsymbol{j}$、$\boldsymbol{k}$，也可以在三维直角坐标系中构造图 3 向量 $\boldsymbol{a}$，具体如图 12。

$$\boldsymbol{a}=4\boldsymbol{i}+3\boldsymbol{j}=4\times\begin{bmatrix}1\\0\\0\end{bmatrix}+3\times\begin{bmatrix}0\\1\\0\end{bmatrix}=\begin{bmatrix}4\\3\\0\end{bmatrix} \tag{23}$$

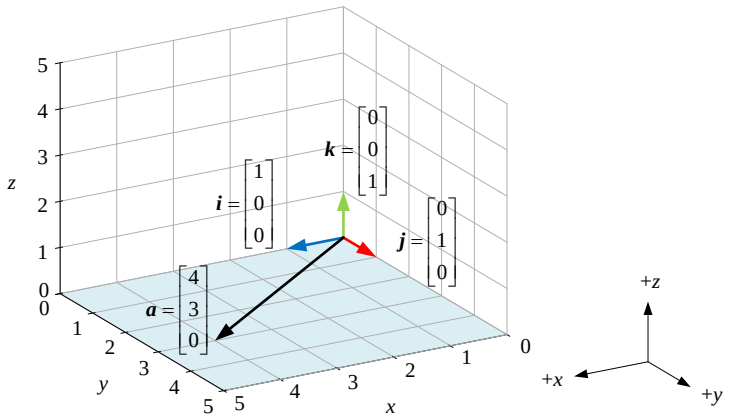


图 12. 三维直角坐标系，向量 $\boldsymbol{a}$ 的定义

投影

图 13 所示为三维直角坐标系中向量 $\boldsymbol{c}$ 和 $\boldsymbol{a}$ 的关系为。

$$\boldsymbol{c} = \boldsymbol{a} + 5\boldsymbol{k} = \begin{bmatrix} 4 \\ 3 \\ 0 \end{bmatrix} + 5 \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} \tag{24}$$

观察图 13，发现向量 $\boldsymbol{a}$ 相当于 $\boldsymbol{c}$ 在 xy 平面的投影。白话说，在向量 $\boldsymbol{c}$ 正上方点一盏灯，在水平面内的影子就是 $\boldsymbol{a}$。这就是我们在本书前文几何部分介绍的投影 (projection)。

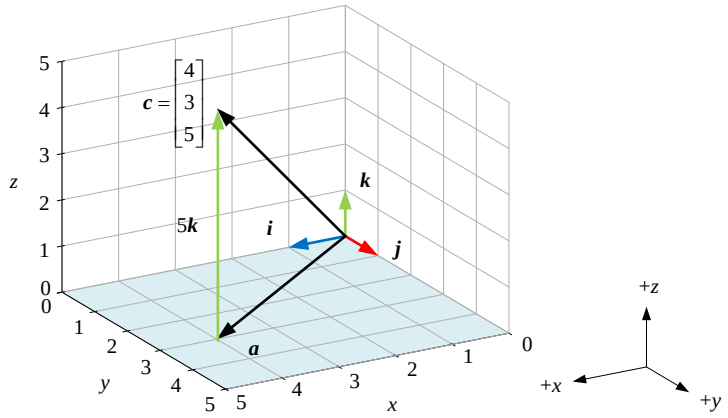


图 13. 三维直角坐标系，向量 $\boldsymbol{a}$ 和向量 c 的关系

值得注意的，向量 $\boldsymbol{c}$ 和 $\boldsymbol{a}$ 之差为 $5\boldsymbol{k}$，而 $5\boldsymbol{k}$ 垂直于 xy 平面。也就是说，$5\boldsymbol{k}$ 垂直 $\boldsymbol{i}$，同时垂直于 $\boldsymbol{j}$；即，$5\boldsymbol{k}$ 和 $\boldsymbol{i}$ 内积为 0，$5\boldsymbol{k}$ 和 $\boldsymbol{j}$ 内积为 0：

$$5\boldsymbol{k} \cdot \boldsymbol{i} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0, \quad 5\boldsymbol{k} \cdot \boldsymbol{j} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0 \tag{25}$$

以下代码绘制图 13。

```
# Bk3_Ch23_4

import numpy as np
import matplotlib.pyplot as plt

def draw_vector(vector,RBG,label,zdir):
    array = np.array([[0, 0, 0, vector[0], vector[1], vector[2]]])
    X, Y, Z, U, V, W = zip(*array)
    plt.quiver(X, Y, Z, U, V, W, normalize = False, color = RBG,
               arrow_length_ratio=0.1)

    label = label + ' (%d, %d, %d)' %(vector[0], vector[1], vector[2])

    ax.text(vector[0], vector[1], vector[2], label, zdir,
            verticalalignment='center')

# define one vector
c = np.array([4, 3, 5])
a = np.array([4, 3, 0])
```

```
i = np.array([1, 0, 0])
j = np.array([0, 1, 0])
k = np.array([0, 0, 1])

fig = plt.figure()
ax = fig.gca(projection='3d')

draw_vector(a,np.array([0,0,0])/255, 'a',a)
draw_vector(c,np.array([0,0,0]), 'c',c)

draw_vector(i, np.array([0,112,192])/255,  'i',(1,0,0))
draw_vector(j, np.array([255,0,0])/255,    'j',(0,1,0))
draw_vector(k, np.array([146,208,80])/255, 'k',(0,0,1))

plt.show()
ax.set_proj_type('ortho')

ax.set_xlim(0,5)
ax.set_ylim(0,5)
ax.set_zlim(0,5)
ax.spines['left'].set_position('zero')

plt.tight_layout()
ax.set_xlabel('$\it{x}$')
ax.set_ylabel('$\it{y}$')
ax.set_zlabel('$\it{z}$')

ax.view_init(azim=60, elev=20)
# ax.view_init(azim=30, elev=20)
ax.xaxis._axinfo["grid"].update({"linewidth":0.25, "linestyle" : ":"})
ax.yaxis._axinfo["grid"].update({"linewidth":0.25, "linestyle" : ":"})
ax.zaxis._axinfo["grid"].update({"linewidth":0.25, "linestyle" : ":"})
```

23.6 投影：影子的长度

投影分为两种——标量投影 (scalar projection) 和向量投影 (vector projection)。

标量投影

投影得到的结果为标量，就是标量投影。利用向量内积的计算原理，向量 $\boldsymbol{b}$ 在向量 $\boldsymbol{a}$ 方向上投影得到的线段长度就是标量投影。

$$\|\boldsymbol{b}\|\cos\theta = \|\boldsymbol{b}\|\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|} \tag{26}$$

如图 14 所示，$\boldsymbol{b}$ 在 $\boldsymbol{a}$ 方向上的标量投影为。

$$\|\boldsymbol{b}\|\cos\theta = \frac{7}{\sqrt{17}} \tag{27}$$

$\boldsymbol{a}$ 在 $\boldsymbol{b}$ 方向上的标量投影为。

$$\|\boldsymbol{a}\|\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{b}\|} = \frac{7}{\sqrt{10}} \tag{28}$$

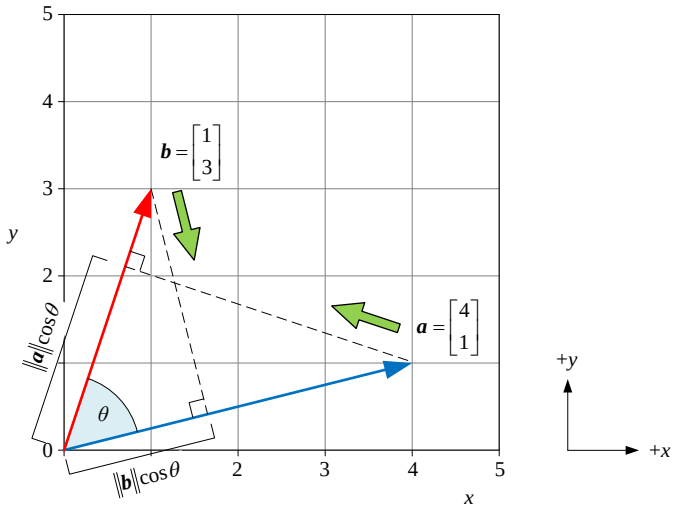


图 14. 标量投影

向量投影

向量投影则是在标量基础上，加上方向；也就是说 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 方向上标量投影，再乘 $\boldsymbol{a}$ 的单位向量：

$$\text{proj}_{\boldsymbol{a}}\,\boldsymbol{b} = \|\boldsymbol{b}\|\cos\theta\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|}\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \tag{29}$$

特别地，如果 $\boldsymbol{a}$ 在单位向量 $\boldsymbol{v}$ 方向上标量投影为：

$$\|\boldsymbol{a}\|\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{v}}{\|\boldsymbol{v}\|} = \boldsymbol{a}\cdot\boldsymbol{v} \tag{30}$$

如图 15 所示，$\boldsymbol{a}$ 在单位向量 $\boldsymbol{v}$ 方向上向量投影：

$$\text{proj}_{\boldsymbol{v}}\,\boldsymbol{a} = \left(\|\boldsymbol{a}\|\cos\theta\right)\boldsymbol{v} = \left(\boldsymbol{a}\cdot\boldsymbol{v}\right)\boldsymbol{v} \tag{31}$$

本系列丛书里面最常见的就是上述投影规则。若 $\boldsymbol{a}$ 和 $\boldsymbol{v}$ 均为列向量，可以用矩阵乘法规则重新写 (31)：

$$\text{proj}_{\boldsymbol{v}}\,\boldsymbol{a} = \left(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{v}\right)\boldsymbol{v} = \left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{a}\right)\boldsymbol{v} \tag{32}$$

投影是线性代数中有关向量最重要的几何操作，没有之一。投影会经常出现在本系列丛书不同板块中。本节会多花一些笔墨和大家聊聊向量投影，给大家建立更加直观的印象。

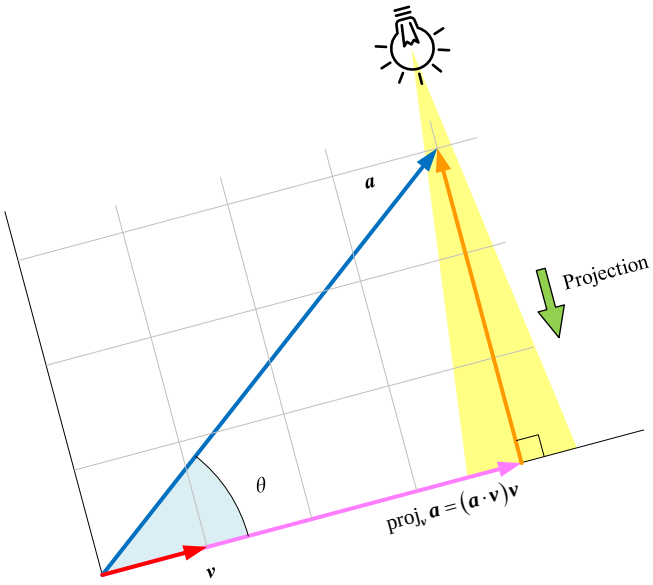


图 15. 正交投影的意义

如图 16 所示，向量 $\boldsymbol{a}$ 在 $\boldsymbol{i}$ 方向上向量投影为：

$$\text{proj}_{\boldsymbol{i}}\,\boldsymbol{a} = \left(\begin{bmatrix}4\\3\end{bmatrix} \cdot \begin{bmatrix}1\\0\end{bmatrix}\right)\boldsymbol{i} = 4\boldsymbol{i} = \begin{bmatrix}4\\0\end{bmatrix} \tag{33}$$

注意，$\boldsymbol{i}$ 就是单位向量。

用矩阵乘法计算 (33)：

$$\text{proj}_{\boldsymbol{i}}\,\boldsymbol{a} = \left(\begin{bmatrix}4\\3\end{bmatrix}^{\mathrm{T}} @ \begin{bmatrix}1\\0\end{bmatrix}\right)\boldsymbol{i} = \left(\begin{bmatrix}1\\0\end{bmatrix}^{\mathrm{T}} @ \begin{bmatrix}4\\3\end{bmatrix}\right)\boldsymbol{i} = 4\boldsymbol{i} = \begin{bmatrix}4\\0\end{bmatrix} \tag{34}$$

$\boldsymbol{a}$ 在 $\boldsymbol{j}$ 方向上向量投影：

$$\text{proj}_{\boldsymbol{j}}\,\boldsymbol{a} = \left(\begin{bmatrix}4\\3\end{bmatrix} \cdot \begin{bmatrix}0\\1\end{bmatrix}\right)\boldsymbol{j} = 3\boldsymbol{j} = \begin{bmatrix}0\\3\end{bmatrix} \tag{35}$$

用矩阵乘法计算 (35)：

$$\text{proj}_{\boldsymbol{j}}\,\boldsymbol{a} = \left(\begin{bmatrix}4\\3\end{bmatrix}^{\mathrm{T}} @ \begin{bmatrix}0\\1\end{bmatrix}\right)\boldsymbol{j} = \left(\begin{bmatrix}0\\1\end{bmatrix}^{\mathrm{T}} @ \begin{bmatrix}4\\3\end{bmatrix}\right)\boldsymbol{j} = 3\boldsymbol{j} = \begin{bmatrix}0\\3\end{bmatrix} \tag{36}$$

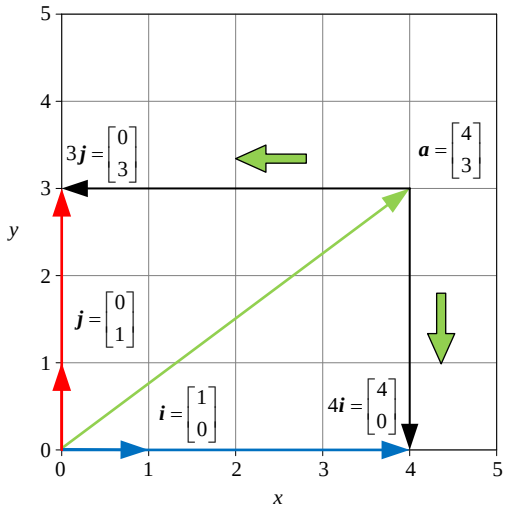


图 16. $\boldsymbol{a}$ 分别在 $\boldsymbol{i}$ 和 $\boldsymbol{j}$ 方向上向量投影

三维向量投影

我们再看三维向量投影。如图 17 所示，$\boldsymbol{c}$ 在 $\boldsymbol{i}$ 方向上标量投影。

$$\text{proj}_{\boldsymbol{i}}\,\boldsymbol{c} = \left(\begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) \boldsymbol{i} = 4\boldsymbol{i} = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} \tag{37}$$

用矩阵乘法写 (37)。

$$\text{proj}_{\boldsymbol{i}}\,\boldsymbol{c} = \left(\begin{bmatrix} 4 & 3 & 5 \end{bmatrix} @ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) \boldsymbol{i} = \left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} @ \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} \right) \boldsymbol{i} = 4\boldsymbol{i} = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} \tag{38}$$

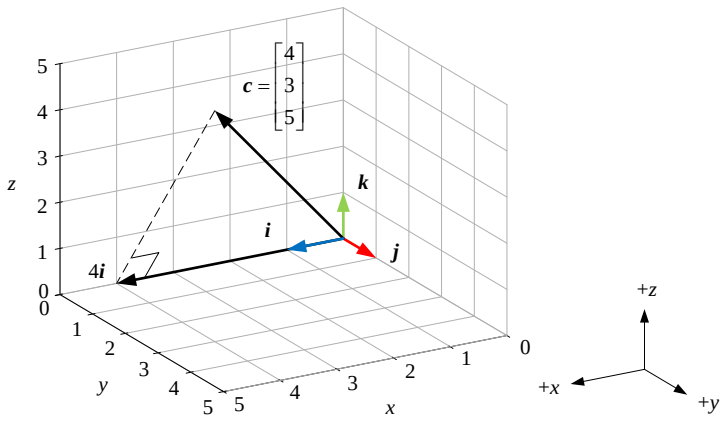


图 17. 向量 $\boldsymbol{c}$ 在向量 $\boldsymbol{i}$ 方向上投影

如图 18 所示，$\boldsymbol{c}$ 在 $\boldsymbol{j}$ 方向上标量投影为：

$$\text{proj}_{\boldsymbol{j}}\boldsymbol{c} = \left(\begin{bmatrix}4\\3\\5\end{bmatrix}\cdot\begin{bmatrix}0\\1\\0\end{bmatrix}\right)\boldsymbol{j} = 3\boldsymbol{j} = \begin{bmatrix}0\\3\\0\end{bmatrix} \tag{39}$$

同样，用矩阵乘法写 (39)：

$$\text{proj}_{\boldsymbol{j}}\boldsymbol{c} = \left(\begin{bmatrix}4 & 3 & 5\end{bmatrix}@\begin{bmatrix}0\\1\\0\end{bmatrix}\right)\boldsymbol{j} = \left(\begin{bmatrix}0 & 1 & 0\end{bmatrix}@\begin{bmatrix}4\\3\\5\end{bmatrix}\right)\boldsymbol{j} = 3\boldsymbol{j} = \begin{bmatrix}0\\3\\0\end{bmatrix} \tag{40}$$

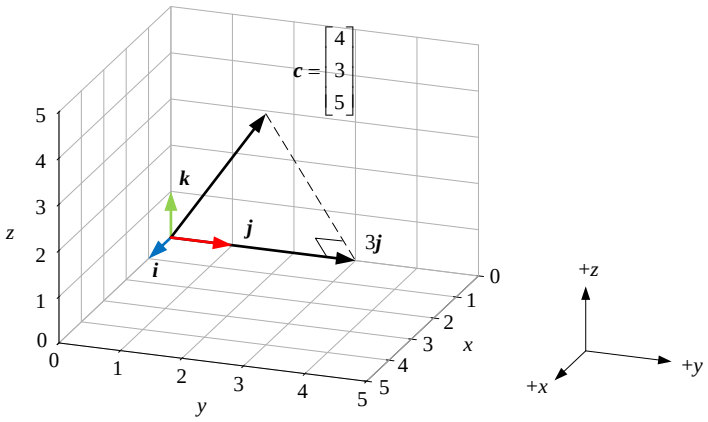


图 18. 向量 $\boldsymbol{c}$ 在向量 $\boldsymbol{j}$ 方向上投影

如图 19 所示，$\boldsymbol{c}$ 在 $\boldsymbol{k}$ 方向上标量投影：

$$\text{proj}_{\boldsymbol{k}}\boldsymbol{c} = \left(\begin{bmatrix}4\\3\\5\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\boldsymbol{k} = 5\boldsymbol{k} = \begin{bmatrix}0\\0\\5\end{bmatrix} \tag{41}$$

用矩阵乘法写 (41)：

$$\text{proj}_{\boldsymbol{k}}\boldsymbol{c} = \left(\begin{bmatrix}4 & 3 & 5\end{bmatrix}@\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\boldsymbol{k} = \left(\begin{bmatrix}0 & 0 & 1\end{bmatrix}@\begin{bmatrix}4\\3\\5\end{bmatrix}\right)\boldsymbol{k} = 5\boldsymbol{k} = \begin{bmatrix}0\\0\\5\end{bmatrix} \tag{42}$$

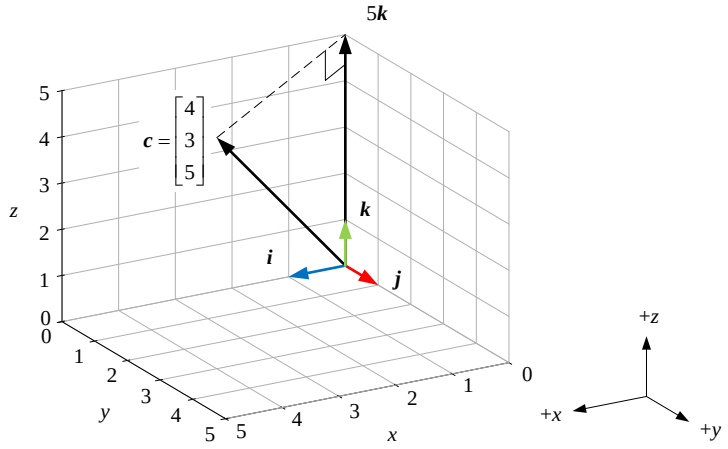


图 19. 向量 $\boldsymbol{c}$ 在向量 $\boldsymbol{k}$ 方向上投影

平面投影

本节最后聊一下三维向量在各个平面投影。

以图 20 为例，因为 $\boldsymbol{i}$ 和 $\boldsymbol{j}$ 张起了 xy 平面，向量 $\boldsymbol{c}$ 在 xy 平面投影，相当于向量 $\boldsymbol{c}$ 分别在 $\boldsymbol{i}$ 和 $\boldsymbol{j}$ 向量投影，再合成。

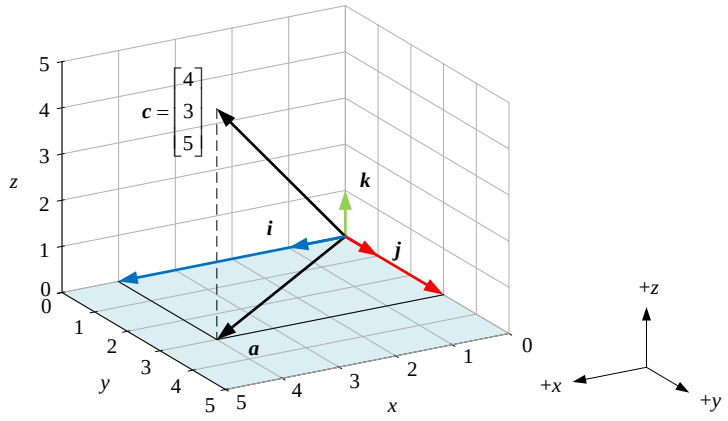


图 20. 向量 $\boldsymbol{c}$ 在 xy 平面投影

换个角度求解这个问题。$\boldsymbol{c}$ 在 xy 平面上投影为 $\boldsymbol{a}$:

$$\boldsymbol{a} = x\boldsymbol{i} + y\boldsymbol{j} \tag{43}$$

其中，x 和 y 为未知量；为了求解 x 和 y，我们需要构造两个等式。

首先计算 $\boldsymbol{c} - \boldsymbol{a}$,

$$\boldsymbol{c} - \boldsymbol{a} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} - (x\boldsymbol{i} + y\boldsymbol{j}) = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} - x\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - y\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3-x \\ 4-y \\ 5 \end{bmatrix} \tag{44}$$

根据前文分析，我们知道 $\boldsymbol{c}-\boldsymbol{a}$ 分别垂直 $\boldsymbol{i}$ 和 $\boldsymbol{j}$，这样我们可以构造两个等式。

$$\begin{aligned}(\boldsymbol{c}-\boldsymbol{a})\cdot\boldsymbol{i}&=0\\(\boldsymbol{c}-\boldsymbol{a})\cdot\boldsymbol{j}&=0\end{aligned}\tag{45}$$

将 (44) 代入 (45)。

$$\begin{bmatrix}3-x\\4-y\\5\end{bmatrix}\cdot\begin{bmatrix}1\\0\\0\end{bmatrix}=0,\quad\begin{bmatrix}3-x\\4-y\\5\end{bmatrix}\cdot\begin{bmatrix}0\\1\\0\end{bmatrix}=0\tag{46}$$

整理得到方程组，并求解 x 和 y。

$$\begin{cases}3-x=0\\4-y=0\end{cases}\Rightarrow\begin{cases}x=3\\y=4\end{cases}\tag{47}$$

这样计算得到，$\boldsymbol{c}$ 在 xy 平面上投影为 $\boldsymbol{a}=3\boldsymbol{i}+4\boldsymbol{j}$。

图 21 和图 22 分别所示为向量 $\boldsymbol{c}$ 在 yz 和 xz 平面投影投影，请读者自行计算投影结果。

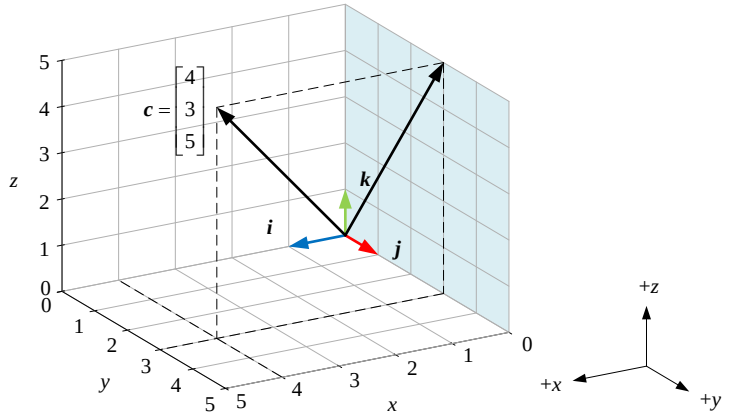


图 21. 向量 $\boldsymbol{c}$ 在 yz 平面投影

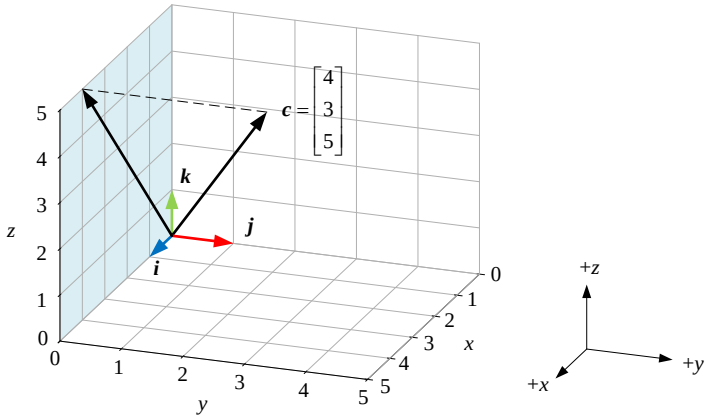
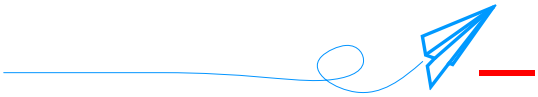


图 22. 向量 $\boldsymbol{c}$ 在 xz 平面投影



向量是线性代数中的多面手，它可以是一行或一列数，也可以是矩阵的一行或一列；有了坐标系，向量和空间中的点、线等元素建立的连线，这时向量摇身一变，变成了有方向的线段。

正是因为向量有几何内涵，线性代数的知识都可以用几何视角来理解。本系列丛书介绍在线性代数知识时都会给出几何视角，请大家格外留意。

下面，请大家准备开始本书最后“鸡兔同笼三部曲”的学习之旅。