

作者	生姜 DrGinger
脚本	生姜 DrGinger
视频	崔崔 CuiCui
开源学习资源	https://github.com/Visualize-ML
平台	https://www.youtube.com/@DrGinger_Jiang https://space.bilibili.com/3546865719052873 https://space.bilibili.com/513194466

1.7 投影



本节你将掌握的核心技能：

- ▶ 标量投影：向量在某方向的“坐标值”。
- ▶ 向量投影：“标量投影 \times 单位向量”，表示投影方向上的正交分量。
- ▶ 正交补概念：正交投影后剩余部分，表示无法被方向向量线性表达的成分。
- ▶ 正交分解：将向量拆分为相互垂直的两个分量，分别表示投影与正交补。
- ▶ 坐标计算：能通过标量投影计算任意向量在旋转直角坐标系中的坐标值。

在上一节讨论向量内积时，我们提到了标量投影这一概念。标量投影可以理解为一个向量在另一个向量方向上的正交投影“长度”，即“坐标值”；标量投影是内积的一种重要应用。本节还将介绍向量投影，相当于“坐标值 \times 方向向量”。此外，我们还将了解正交补，正交补表示向量中不能被投影方向线性表达的部分，与投影方向正交。最后，本节探讨了向量分解的概念，即将一个向量拆分为多个分量，并介绍了标准正交基。

标量投影

如图 1 (a) 所示，给定非零向量 a 、 b ，两者夹角为 θ 。

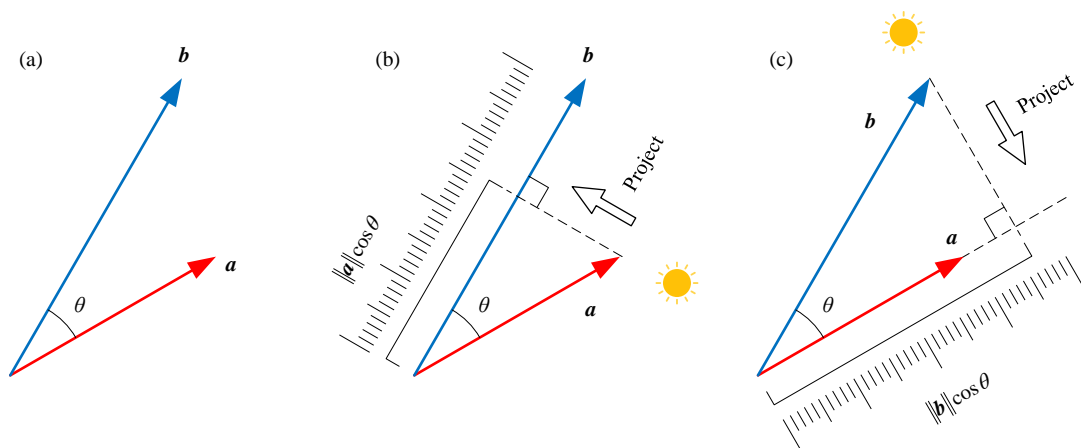


图 1. 标量投影

如图 1 (b) 所示, a 在 b 方向上的**标量投影** (scalar projection) 为

$$\|a\| \cos \theta = \frac{a \cdot b}{\|b\|} = a \cdot \frac{b}{\|b\|} = a \cdot \hat{b} \quad (1)$$

其中, \hat{b} 为非零向量 b 的单位向量 (方向向量)。

顾名思义, 标量投影结果为标量, 因此不包含方向信息。

标量投影可以理解为给定向量在某个特定方向上正交投影的“坐标”。

比如, 图 1 (b) 中, $\|a\| \cos \theta$ 表示向量 a 在向量 b 方向 (\hat{b}) 上的正交投影的“坐标”; 如图 1 (b) 所示, 正交投影意味着图中虚线垂直于向量 b 。

如图 1 (c) 所示, 反过来, b 在 a 方向上的标量投影为

$$\|b\| \cos \theta = \frac{a \cdot b}{\|a\|} = \frac{a}{\|a\|} \cdot b = \hat{a} \cdot b \quad (2)$$

其中, \hat{a} 为非零向量 a 的单位向量 (方向向量)。

也就是说, $\|b\| \cos \theta$ 为向量 b 在向量 a 方向 (\hat{a}) 上的正交投影“坐标”。图 1 (c) 中虚线垂直于向量 a 。

举个例子, 给定向量 $a = [3, 4]^T$, a 在 e_1 方向上的标量投影为

$$a \cdot e_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3 \quad (3)$$

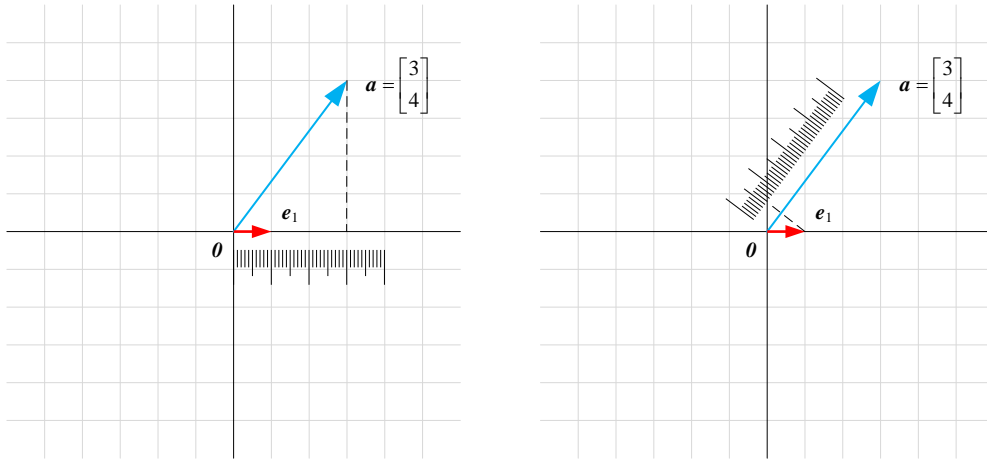
可以直接用向量内积来计算标量投影是因为 e_1 是单位向量。

反过来, 计算 e_1 在 a 方向上的标量投影

$$e_1 \cdot \hat{a} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix} = 3/5 \quad (4)$$




请大家计算向量 a 、 e_2 相互标量投影的结果。

图 2. 向量 a 、 e_1 相互标量投影

代码 1 完成上述标量投影计算。下面聊聊其中关键词句。

- a** 创建了两个一维数组，代表 a 、 e_1 两个向量。
- b** 分别计算 a 和 e_1 、 e_1 和 a 点积，请大家判断结果是否相同。
- c** 用 `numpy.linalg.norm()` 计算 a 、 e_1 两个向量的长度。
- d** 计算 a 在 e_1 方向上的标量投影，即 a 、 e_1 两者的向量内积除以 e_1 的向量长度。
- e** 计算 e_1 在 a 方向上的标量投影，即 a 、 e_1 两者的向量内积除以 a 的向量长度。

代码 1. 计算标量投影 |  LA_01_07_01.ipynb

```
## 初始化
import numpy as np

## 定义向量
a a = np.array([3, 4])
    e1 = np.array([1, 0])

## 计算点积
b dot_a_e1 = np.dot(a, e1)
    dot_e1_a = np.dot(e1, a)

## 计算向量模长
c norm_a = np.linalg.norm(a)
    norm_e1 = np.linalg.norm(e1)

## 计算标量投影
d scalar_proj_a_on_e1 = dot_a_e1 / norm_e1 # a 在 e1 上的投影
e scalar_proj_e1_on_a = dot_e1_a / norm_a # e1 在 a 上的投影
```

RGB 颜色标量投影

想象一束黄光穿过一片红色滤光片后，只有红光的强度被保留下来。这个强度值可以类比为黄光向红色轴的标量投影。

如图 3 (a) 所示，在 RGB 空间中，黄光向量为 $[1, 1, 0]^T$ ，红色滤光片对应的红色向量为 $[1, 0, 0]^T$ ，黄光在红光向量上的标量投影为

$$\frac{[1 \ 1 \ 0]^T \cdot [1 \ 0 \ 0]^T}{\|[1 \ 0 \ 0]^T\|} = 1 \quad (5)$$

这个标量投影可以理解为黄光在红色方向上的“坐标”，即红光的强度。这说明，黄光中有 100% 强度的红光。

反过来，如图 3 (b) 所示，红光在黄光上的标量投影为

$$\frac{[1 \ 1 \ 0]^T \cdot [1 \ 0 \ 0]^T}{\|[1 \ 1 \ 0]^T\|} = \frac{\sqrt{2}}{2} \quad (6)$$

黄光向量长度为 $\sqrt{2}$ ，红光在其中占了 $1/2$ 。

? 类似地，请大家计算品红、蓝光的相互标量投影。

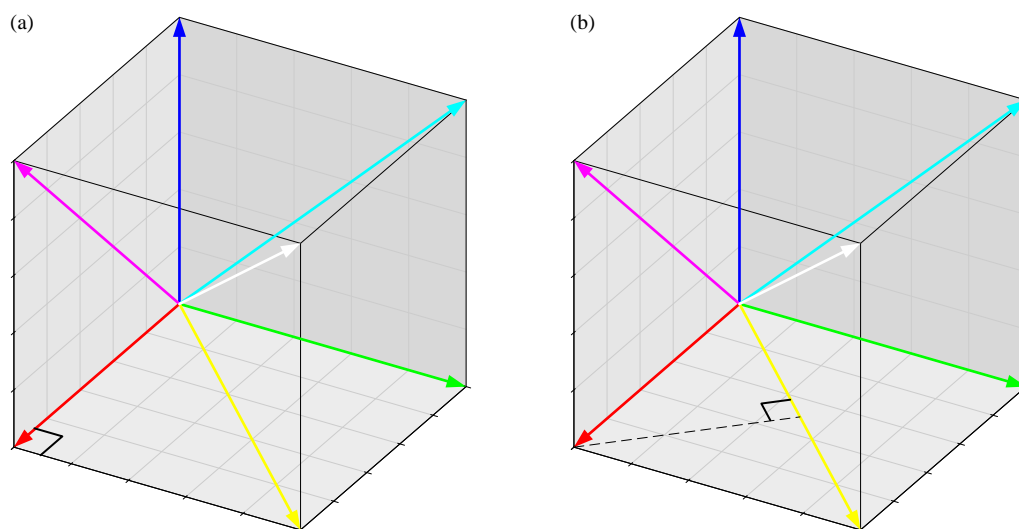


图 3. 红光、白光的相互标量投影

而青色 $[0, 1, 1]^T$ 通过红色滤光片时，则不会有任何光通过；原因很简单，青色光中没有任何红色成分。标量投影可以帮我们验证这一点

$$\frac{[0 \ 1 \ 1]^T \cdot [1 \ 0 \ 0]^T}{\|[1 \ 0 \ 0]^T\|} = 0 \quad (7)$$

换个角度，蓝绿平面上任意向量在红色方向标量投影都 0。

? 请大家计算蓝光、绿光在红光方向的标量投影。

再试想一束白光穿过一片红色滤光片后，只有红光通过；这相当于白光向红光方向上的标量投影。

在 RGB 空间中，白光向量为 $[1, 1, 1]^T$ ，红色向量为 $[1, 0, 0]^T$ ，白光在红光向量上的标量投影为

$$\frac{[1 \ 1 \ 1]^T \cdot [1 \ 0 \ 0]^T}{\|[1 \ 0 \ 0]^T\|} = 1 \quad (8)$$

这个值可以理解为白光在红色轴上的“坐标”，即红光的强度。这说明，白光中有 100% 强度的红光。

反过来，红光在白光上的标量投影为

$$\frac{[1 \ 1 \ 1]^T \cdot [1 \ 0 \ 0]^T}{\|[1 \ 1 \ 1]^T\|} = \frac{\sqrt{3}}{3} \quad (9)$$

白色向量长度为 $\sqrt{3}$ ，红光在其中占了 $1/3$ 。

为了更好呈现投影效果，图 4 采用了两个不同视角。

? 请大家计算白光、蓝光两个向量之间相互标量投影，白光、绿光两个向量之间相互标量投影。

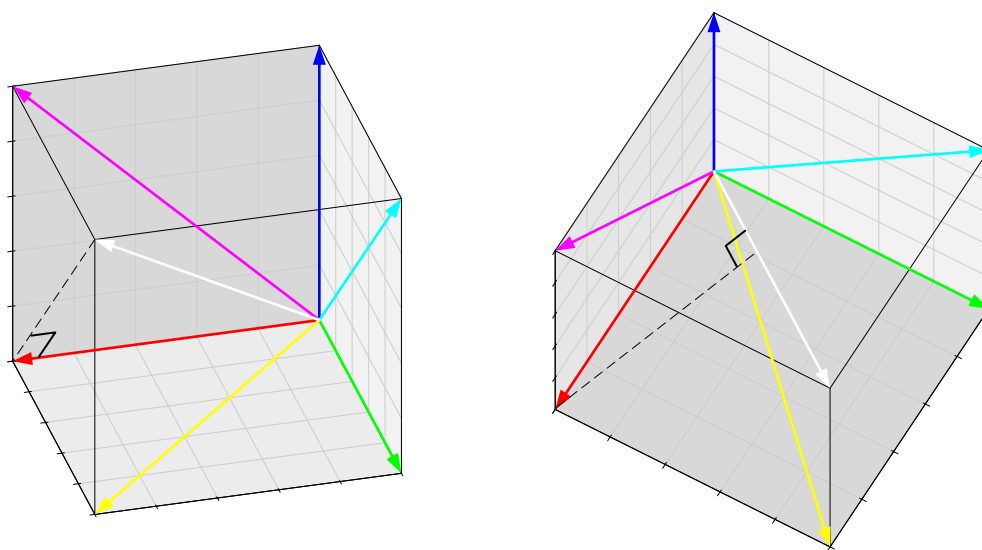


图 4. 红光、白光的相互标量投影

向量投影

通过本节前文的学习，大家知道标量投影告诉我们某个向量在指定方向上有多大，即“坐标”，就像测量一个物体在某条直线上的影子的长度。

而**向量投影** (vector projection) 相当于“**坐标值** × **方向向量**”；这个方向向量就是投影方向的单位向量。

如图 5 (a) 所示，给定非零向量 a 、 b ， a 在 b 方向上的**向量投影**为

$$\text{proj}_b a = \underbrace{\|a\| \cos \theta}_{\text{Scalar projection}} \times \underbrace{\hat{b}}_{\text{Direction vector}} = \frac{a \cdot b}{\|b\|} \times \frac{b}{\|b\|} = \frac{a \cdot b}{\|b\|^2} b = \frac{a \cdot b}{b \cdot b} b \quad (10)$$

$\frac{a \cdot b}{\|b\|}$ 为 a 在 b 方向上**标量投影**，即**坐标值**； $\frac{b}{\|b\|}$ 为 b 的**方向向量**，即单位向量、归一化向量。

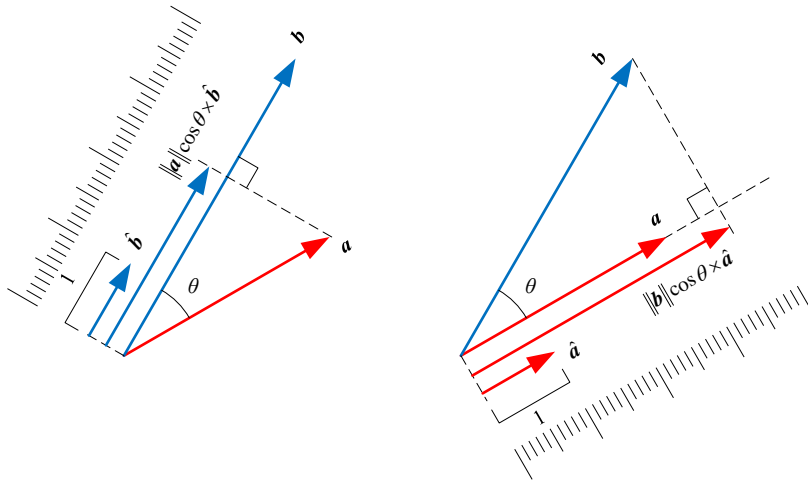


图 5. 向量投影

反过来，如图 5 (b) 所示， b 在 a 方向上的**向量投影**为

$$\text{proj}_a b = \underbrace{\|b\| \cos \theta}_{\text{Scalar projection}} \times \underbrace{\hat{a}}_{\text{Direction vector}} = \frac{a \cdot b}{\|a\|} \times \frac{a}{\|a\|} = \frac{a \cdot b}{\|a\|^2} a = \frac{a \cdot b}{a \cdot a} a \quad (11)$$

同理， $\frac{a \cdot b}{\|a\|}$ 为 b 在 a 方向上**标量投影** (坐标值)； $\frac{a}{\|a\|}$ 为 a 的**方向向量** (单位向量)。

总结来说，标量投影只给出投影方向上的“坐标”。向量投影通过引入方向向量指定了投影的“坐标 + 方向”。



LA_01_06_02.ipynb 介绍如何计算向量投影，请大家自行学习。

RGB 颜色向量投影

在 RGB 空间中，红光 $[1, 0, 0]^T$ 本身就是单位向量；前文已经算出来黄光在红光上的标量投影。

因此，黄光向量为 $[1, 1, 0]^T$ 在红光向量上的向量投影为


$$1 \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (12)$$

反过来，黄光的单位向量为

$$\frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad (13)$$

前文计算得到红光在黄光上的标量投影为 $\frac{\sqrt{2}}{2}$ ；这样容易计算红光在黄光上的向量投影

$$\frac{\sqrt{2}}{2} \times \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2} \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad (14)$$

 请大家计算白光在红光方向上的向量投影；也请计算红光在白光方向上的向量投影。

正交补

图 3 (a) 中，黄色向量朝红色向量正交投影之后，还有一部分“剩余”，这部分剩余就是绿色向量。

剩余的绿色向量与红色向量垂直。换句话说，绿色向量表示的是黄色向量中无法用红色向量表示的部分。

在向量投影中，这个绿色向量也有自己的名字——**正交补** (orthogonal rejection, orthogonal complement)；我们也管它叫它也被称为**正交余量** (orthogonal remainder)、**垂直分量** (perpendicular component)、**法向分量** (normal component)，强调它与投影方向的正交关系。

简单来说，如果一个向量正交投影到某个方向后，无法被该方向表示的剩余部分就是这个方向的正交补。

给定非零向量 a 、 b ， a 在 b 方向上的正交补为

$$\text{oproj}_b a = a - \text{proj}_b a \quad (15)$$

$\text{oproj}_b a$ 为向量 a 中不能被 b 线性表达的成分。显然， $\text{oproj}_b a$ 垂直于 b 。

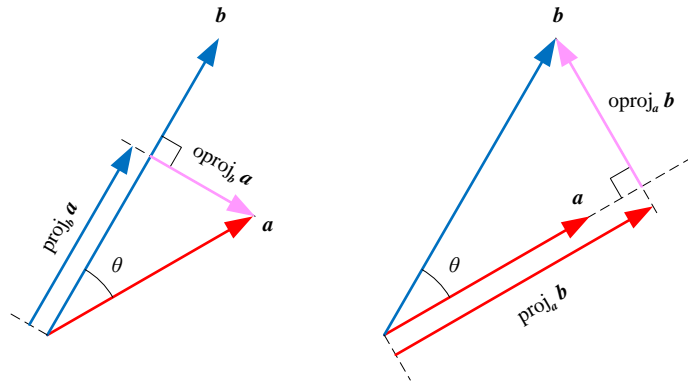


图 6. 正交补

反过来, b 在 a 方向上的正交补为

$$\text{oproj}_a b = b - \text{proj}_a b \quad (16)$$

$\text{oproj}_a b$ 为向量 b 中不能被 a 线性表达的成分。显然, $\text{oproj}_a b$ 垂直于 a 。

向量正交分解

向量分解的核心思想是：把一个向量拆分为两个或多个分量，使它们相加后仍等于原向量。

实际上, 大家对向量分解并不陌生。前文提过, 在平面直角坐标系中, 每个向量都可以表示为坐标轴上的两个分量之和。

我们可以把 $x = [x_1, x_2]^T$ 写成

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = x_1 e_1 + x_2 e_2 \quad (17)$$

举个例子, 向量 $a = [3, 4]^T$ 可以写成

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} = 3 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 3e_1 + 4e_2 \quad (18)$$

如图 7 所示, 水平分量, 即沿 x_1 轴的部分, 为 $[3, 0]^T$, 可以写成 $3e_1$ 。

竖直分量, 即沿 x_2 轴的部分, 为 $[0, 4]^T$, 可以写成 $4e_2$ 。这显然是个向量正交分解。

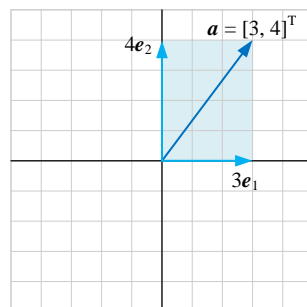


图 7. 向量 $\mathbf{a} = [3, 4]^T$ 在 $[\mathbf{e}_1, \mathbf{e}_2]$ 中的向量分解

图 7 所示的实际上是向量的正交分解 (orthogonal decomposition of a vector)。向量正交分解将一个向量拆分为两个或多个相互正交的分量。这种分解方式确保了分量之间没有相互影响。在欧几里得空间中，正交分解的存在性和唯一性依赖于内积的定义，使得它在数学、物理和工程领域中具有广泛的应用。

\mathbf{e}_1 、 \mathbf{e}_2 构成了 **标准正交基** (standard basis, natural basis)，记作 $[\mathbf{e}_1, \mathbf{e}_2]$ 。

标准正交基 是向量空间中的一组基向量，通常用于表示欧几里得空间中的坐标轴正方向。

下面，让我们逐字解释 **标准正交基**。

先从 **基底** (basis) 开始。**基底** 是一个向量空间中的一组 **线性无关** 向量，它们的 **线性组合** 可以表示该空间中的任意向量。

基底 中每个向量叫 **基向量** (basis vector)。**基向量** 就好比支撑空间网格结构的骨架。

本书前文提过，**线性无关** 是指一组向量中没有任何一个向量可以表示为其他向量的线性组合，即这些向量在方向上彼此独立，无法被替代。

本书前文也讲过，**线性组合** 是指将一组向量分别乘以标量系数后相加，生成一个新的向量的操作。比如， \mathbf{e}_1 和 \mathbf{e}_2 通过线性组合 $(x_1\mathbf{e}_1 + x_2\mathbf{e}_2)$ 能够表达平面上任意向量 \mathbf{x} 。

“**标准**”在“标准正交基”中的含义与 natural basis 或 standard basis 中的“自然 (natural)”或“标准 (standard)”有关。“标准”指的是这组基向量是向量空间中最自然、最常用的基，通常选择坐标轴正方向的单位向量作为基向量。比如，单位向量 \mathbf{e}_1 指向 x_1 轴正方向，单位向量 \mathbf{e}_2 指向 x_2 轴正方向。

大家对上述这些向量空间概念有些印象就好，本书后续将专门系统讲解它们。

整理 (16)，我们可以把向量 \mathbf{a} 分解为两部分

$$\mathbf{a} = \text{proj}_{\mathbf{b}} \mathbf{a} + \text{oproj}_{\mathbf{b}} \mathbf{a} \quad (19)$$

如图 8 所示， $\text{proj}_{\mathbf{b}} \mathbf{a}$ 、 $\text{oproj}_{\mathbf{b}} \mathbf{a}$ 这两个向量分量相互垂直；其中，向量投影 $\text{proj}_{\mathbf{b}} \mathbf{a}$ 平行于向量 \mathbf{b} ，正交补 $\text{oproj}_{\mathbf{b}} \mathbf{a}$ 垂直于向量 \mathbf{b} 。

同样，向量 \mathbf{b} 可以分解为两部分

$$\mathbf{b} = \text{proj}_{\mathbf{a}} \mathbf{b} + \text{oproj}_{\mathbf{a}} \mathbf{b} \quad (20)$$

$\text{proj}_{\mathbf{a}} \mathbf{b}$ 、 $\text{oproj}_{\mathbf{a}} \mathbf{b}$ 这两个向量分量相互垂直；其中，向量投影 $\text{proj}_{\mathbf{a}} \mathbf{b}$ 平行于向量 \mathbf{a} ，正交补 $\text{oproj}_{\mathbf{a}} \mathbf{b}$ 垂直于向量 \mathbf{a} 。

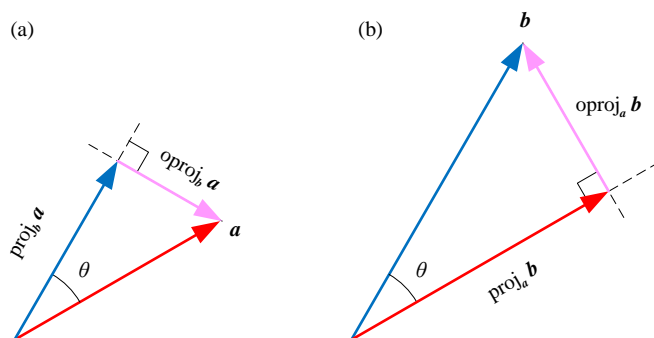


图 8. 正交向量分解

高中物理课本中，力、速度的分解可以采用的这种向量正交分解方式。

如图 9 (a) 所示，斜面上的物体受到重力作用；重力可以分解为两个分量，一个沿坡度向下，一个垂直坡面。

如图 9 (b) 所示，斜面上的物体沿破面向下运动；速度可以分解为两个分量，一个水平，一个竖直。

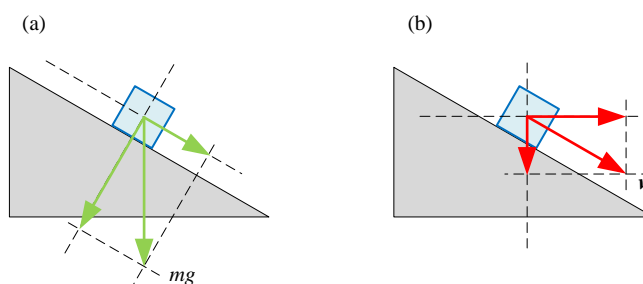


图 9. 力、速度的正交分解

计算坐标

在图 10 (a) 这个大家熟悉的平面直角坐标系中，向量 $\mathbf{a} = [4, 3]^T$ 的坐标为 (4, 3)。

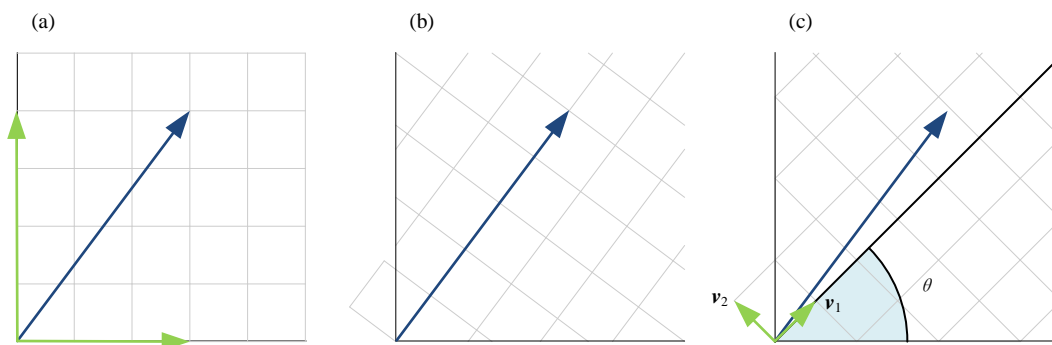


图 10. 向量的正交分解

如图 10 (b) 所示，向量 $\mathbf{a} = [4, 3]^T$ 可以正交分解为

$$\begin{bmatrix} 4 \\ 3 \end{bmatrix} = 5 \times \begin{bmatrix} 4/5 \\ 3/5 \end{bmatrix} + 0 \times \begin{bmatrix} -3/5 \\ 4/5 \end{bmatrix} \quad (21)$$

可以这样理解，在这个“旋转正方网格”描述的坐标系中，向量 $\mathbf{a} = [4, 3]^T$ 的坐标为 (5, 0)。

实际上，平面有无数个这种“旋转正方网格”；我们能找到无数组正交归一化向量，它们满足

$$\mathbf{v}_1 = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} \quad (22)$$

请大家计算 \mathbf{v}_1 、 \mathbf{v}_2 的长度，验证两者长度是否为 1；也请验算 \mathbf{v}_1 、 \mathbf{v}_2 的内积为 0。

也就是说，当 θ 取不同值时，我们可以得到不同的“旋转正方网格”；而这些网格都可以成为描述 $\mathbf{a} = [4, 3]^T$ 的“直角”坐标系。

问题来了，怎么获得在不同“旋转正方网格”的坐标呢？

这使用到了上一节介绍的向量标量投影。

比如， $\mathbf{a} = [4, 3]^T$ 在 \mathbf{v}_1 上的标量投影为

$$\frac{\mathbf{a} \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|} = \mathbf{a} \cdot \mathbf{v}_1 = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} = 4 \cos \theta + 3 \sin \theta \quad (23)$$

这个值就是 \mathbf{a} 在 \mathbf{v}_1 上的坐标值。

类似地， $\mathbf{a} = [4, 3]^T$ 在 \mathbf{v}_2 上的标量投影为

$$\frac{\mathbf{a} \cdot \mathbf{v}_2}{\|\mathbf{v}_2\|} = \mathbf{a} \cdot \mathbf{v}_2 = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} \sin \theta \\ -\cos \theta \end{bmatrix} = 4 \sin \theta - 3 \cos \theta \quad (24)$$

$\mathbf{a} = [4, 3]^T$ 在 \mathbf{v}_1 上的向量分量 (向量投影) 为

$$(4 \cos \theta + 3 \sin \theta) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (25)$$


$\mathbf{a} = [4, 3]^T$ 在 \mathbf{v}_2 上的向量分量 (向量投影) 为

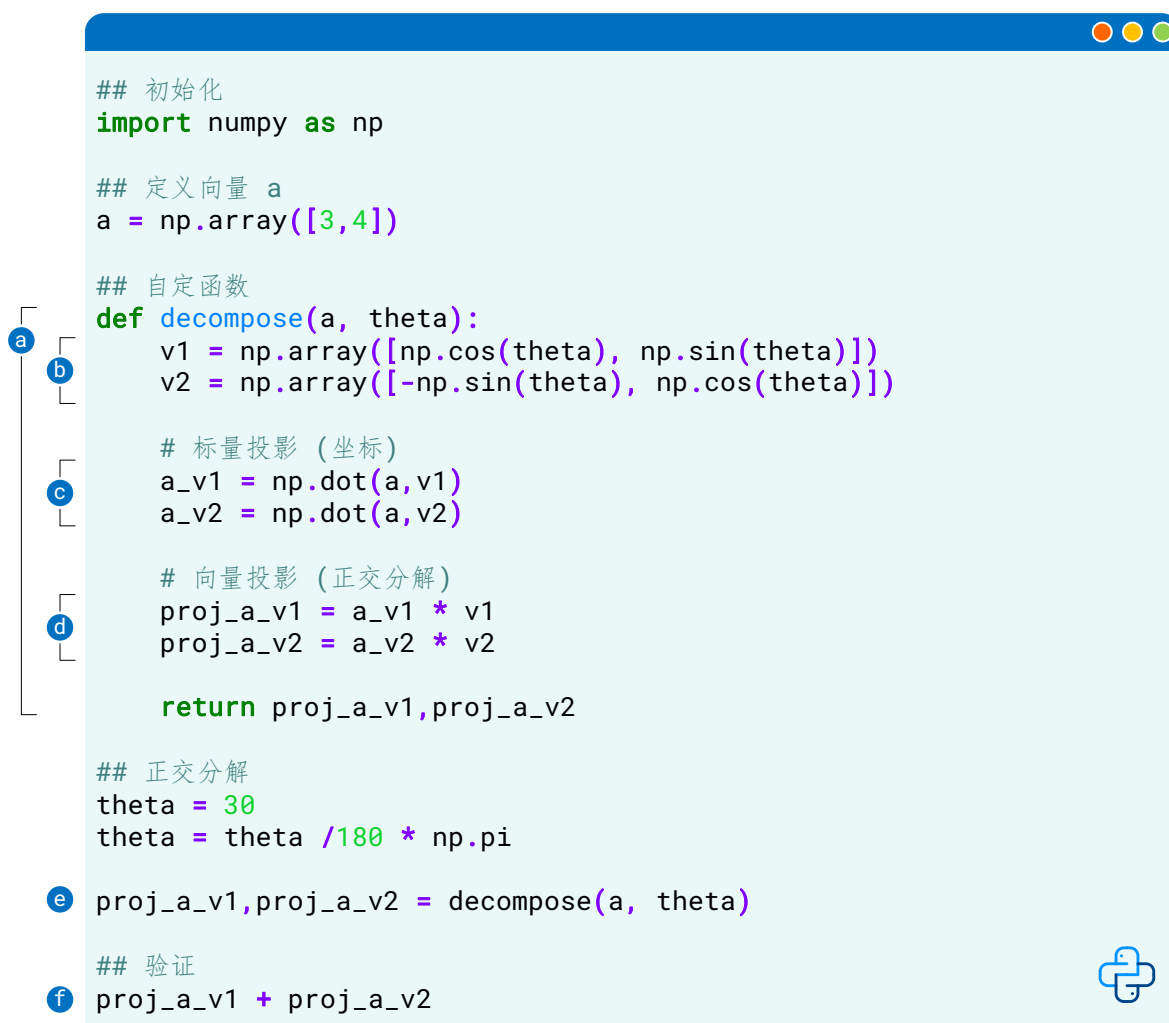
$$(4 \sin \theta - 3 \cos \theta) \cdot \begin{bmatrix} \sin \theta \\ -\cos \theta \end{bmatrix} \quad (26)$$

这两个分量之和为

$$(4 \cos \theta + 3 \sin \theta) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + (4 \sin \theta - 3 \cos \theta) \cdot \begin{bmatrix} \sin \theta \\ -\cos \theta \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \quad (27)$$

代码 2 展示如何对向量完成正交分解。

代码 2. 向量正交分解 |  LA_01_07_03.ipynb



下面聊聊代码 2 中关键语句。

a 定义了一个函数，名字叫 `decompose`，意思是“分解”，它接收两个输入参数：一个向量 a 和一个角度 θ （单位是弧度）。函数的作用是把向量 a 分解成两个方向上的分量。

b 用角度 θ 来构造一个单位向量 v_1 ，方向是顺时针旋转 θ 角度得到的方向。`np.cos` 和 `np.sin` 分别表示余弦和正弦，用于计算水平、竖直的分量。这个向量的长度为 1，表示的是我们想要的第一个方向向量。

接着构造第二个方向的单位向量 v_2 ，它和 v_1 垂直，也就是说两个方向彼此正交。

c 计算向量 a 在单位向量（方向向量） v_1 方向上的投影大小，叫做标量投影，使用的是 `np.dot` 函数，它表示“点积”或者“内积”。点积的结果是一个数值，代表 a 沿着 v_1 方向“有多少”。

接着计算向量 a 在单位向量（方向向量） v_2 方向上的投影大小，叫做标量投影。

d 标量投影的大小乘上单位方向 v_1 ，得到的是一个新的向量 `proj_a_v1`，它表示向量 a 在 v_1 方向上的那部分分量。这一步把“大小”和“方向”结合起来，变成一个向量。

接着计算向量 a 在 v_2 方向上的那部分分量。

e 调用前面定义的 `decompose` 函数，把向量 a 和角度 θ 作为参数传进去。函数运行后会返回两个向量： a 在第一个方向上的投影，以及在垂直方向上的投影。

f 最后我们把这两个投影向量相加，得到一个新的向量。如果前面的分解正确，这两个投影相加应该等于原来的向量 a 。

分解白光

回到 RGB 空间，图 11 所示为白光的三种向量正交分解方式。

如图 11 (a) 所示，白光向量可以正交分解为黄光、蓝光；显然，蓝光垂直于黄光，蓝光也和 x_1x_2 平面正交。

类似地，白光向量也可以正交分解为品红色光、绿光，如图 11 (b) 所示；而绿光垂直于品红色光，绿光和 x_1x_3 平面正交。白光向量还可以分解为品青色光、红光，如图 11 (c) 所示；而红光垂直于品青色光，红光和 x_2x_3 平面正交。

实际上，白光分解成“红光 + 绿光 + 蓝光”也是向量正交分解；红光、绿光、蓝光，这三个向量分量相互正交。

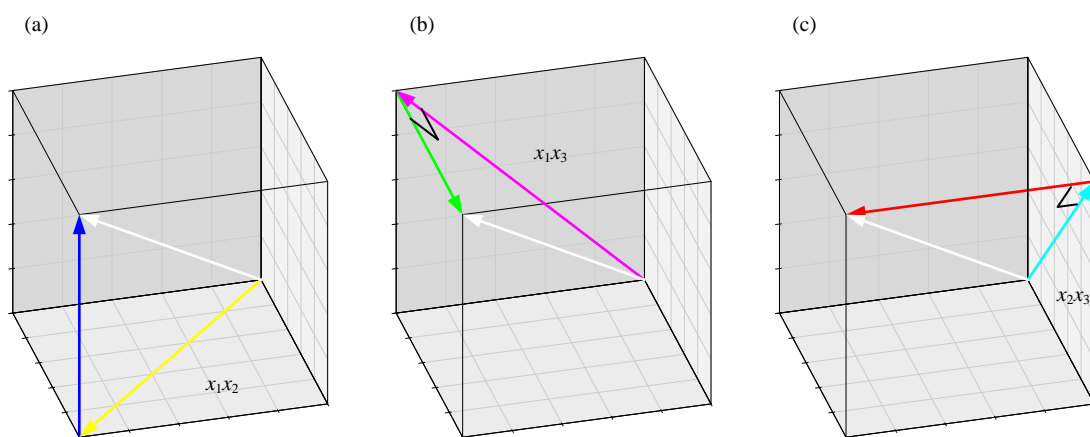
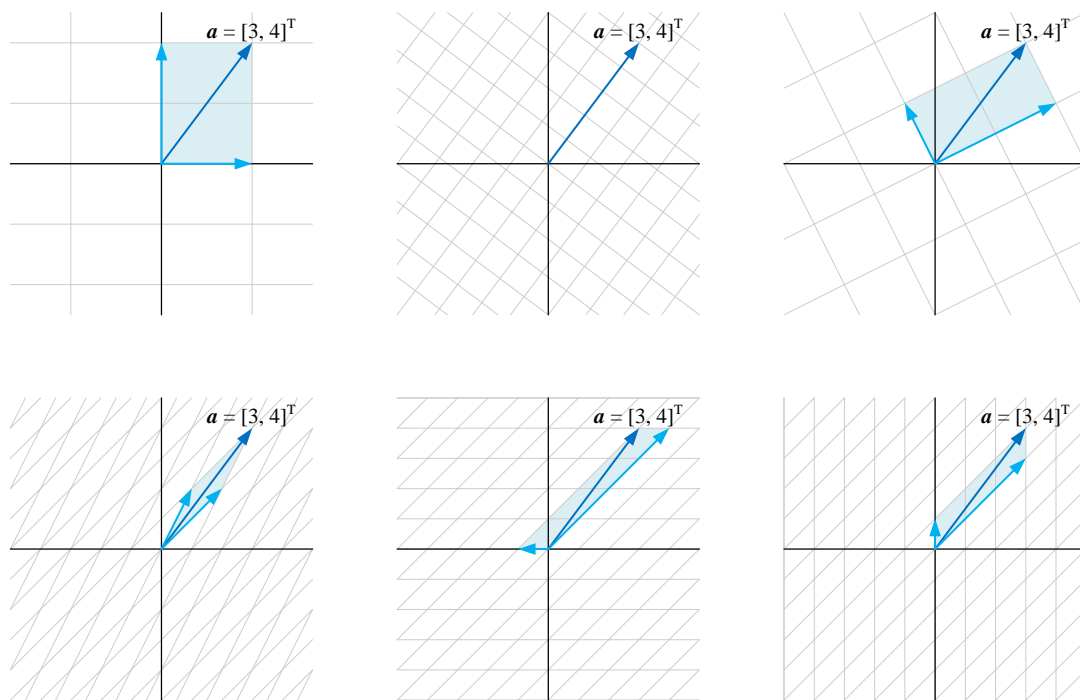


图 11. 白光的三个分解

不仅仅是横平竖直的单位正方网格

图 12 展示了几种不同的方式来分解向量 \mathbf{a} 。有些网格是矩形的，有些是旋转后的正方形，还有些是一般的平行四边形。尽管形状不同，这些网格都具有相同的特点——平行、等距，并且经过原点。

显然，向量 \mathbf{a} 的分解方式有无数种；反向来看，存在无数种不同类型的网格（基）可以铺满整个平面。

图 12. 向量 $a = [3, 4]^T$ 的不同分解方式

如图 13 所示，在 RGB 空间，我们可以找到近乎无数个白光的正交、非正交分解。我们在 RGB 空间先随机生成一个颜色向量，然后计算白色向量和这个颜色向量之差；再用三角形法则首尾相连绘制这两个颜色向量，就可以还原白光向量。

向量分解可以理解为向量加法的逆运算，即将一个向量拆分成两个或多个分量，使它们的和等于原向量。

换句话说，若向量加法是将多个向量合成为一个整体，那么向量分解就是在已知结果的情况下，找出满足加法关系的组成部分。

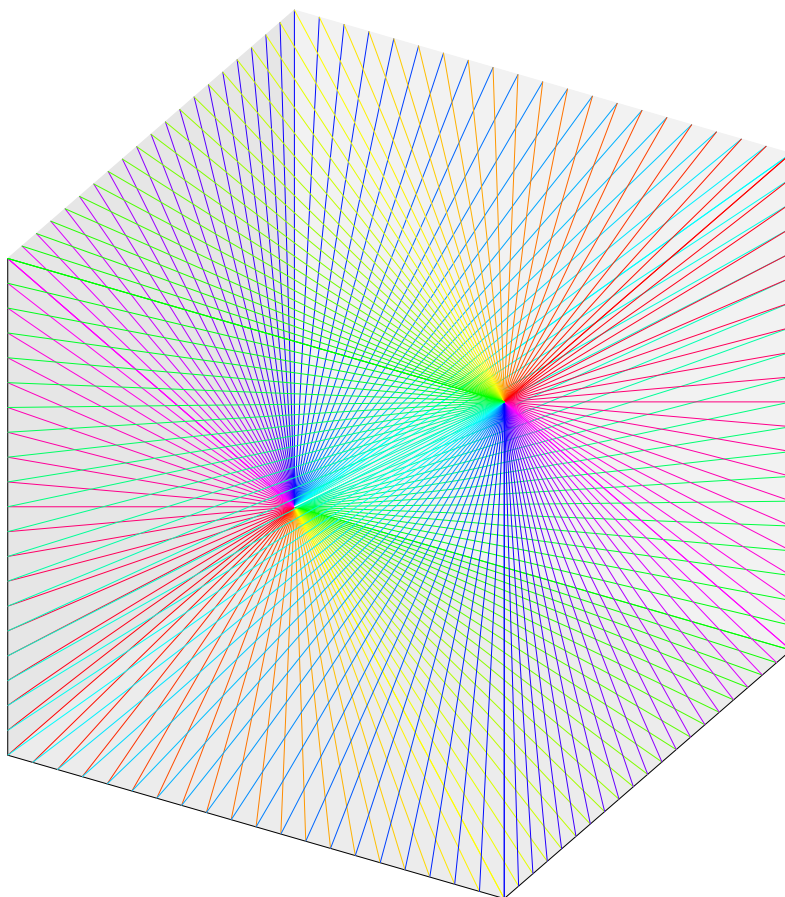


图 13. 白光更多向量正交、非正交分解



请大家用 DeepSeek/ChatGPT 等工具完成本节如下习题。

- Q1.** 请写代码计算向量 $\mathbf{a} = [3, 4]^T$ 在 $[1, 1]$ 上的标量投影，并手算验证。
- Q2.** 请写代码计算向量 $\mathbf{a} = [3, 4]^T$ 在 $[-1, 1]$ 上的向量投影，并手算验证。
- Q3.** 给定向量 $\mathbf{a} = [3, 4]^T$ 、 $\mathbf{b} = [-1, 0]$ ，将向量 \mathbf{a} 分解为
- ▶ 平行 \mathbf{b} 的分量
 - ▶ 垂直 \mathbf{b} 的分量
- Q4.** 请试着写出 10 个类似图 12 的向量分解。