

第03讲 向量的四则运算 课后习题答案

作者：欧新宇 (Xinyu OU)

本文档所展示的测试结果，均运行于：Intel Core i7-7700K CPU 4.2GHz

• 作业要求及提交

1. 将所有运行结果保存为一个 word 文档（特别推荐保存为 pdf 文档进行提交）
2. 要求：使用编程环境完成下列习题，并按题目顺序进行排版，每个题目要求按如下顺序组织（若存在）：
 - 0). 题目(将题目完整Copy到作业文档中，可以通过公式编辑器编辑或截图方式)；
 - 1). 需要手工书写的部分，请尽量在word文档中进行编辑（迫不得已时，可书写在纸上并拍照）；
 - 2). 代码（尽量通过从编程环境截图粘贴）；
 - 3). 运行结果贴于文档中。（复制运行结果到文档或通过截图粘贴）
 - x). **如果熟悉本编程环境'Jupyter Notebook'也可以直接在本环境中编写所有文稿及代码，并打印成pdf文档进行提交。**
3. 将文档上传至 课堂派 平台

注意：截图只需要截取必要部分。此外，请确保截图清晰可见。

1. 设 $v_1 = (1, 1, 0)^T$, $v_2 = (0, 1, 1)^T$, $v_3 = (3, 4, 0)^T$, 求 $a = v_1 - v_2$ 及 $b = 3v_1 + 2v_2 - v_3$ 。

• 手工推导

$$a = v_1 - v_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-0 \\ 1-1 \\ 0-1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$b = 3v_1 + 2v_2 - v_3 = 3 * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + 2 * \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 3*1+2*0-3 \\ 3*1+2*1-4 \\ 3*0+2*1-0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

• Python描述

```
# 0. 载入运算库
import numpy as np

# 1. 定义已知量
v1 = np.array([[1,1,0]]).T
v2 = np.array([[0,1,1]]).T
v3 = np.array([[3,4,0]]).T

# 2. 计算和数据处理
print('a = {}'.format(v1-v2))
print('b = {}'.format(3*v1+2*v2-v3))
```

```
a = [[ 1]
      [ 0]
      [-1]]
b = [[0]
      [1]
      [2]]
```

2. 设 $3(a_1 - a) + 2(a_2 + a) = 5(a_3 + a)$ 。其中,
 $a_1 = (2, 5, 1, 3)^T, a_2 = (10, 1, 5, 10)^T, a_3 = (4, 1, -1, 1)^T$, 求 a 。

• 基础推导

$$3(a_1 - a) + 2(a_2 + a) = 5(a_3 + a)$$

$$\Rightarrow 3a_1 - 3 * a + 2a_2 + 2 * a = 5a_3 + 5 * a$$

$$\Rightarrow 3a_1 + 2a_2 - 5a_3 = 6 * a$$

$$\Rightarrow a = \frac{3}{6}a_1 + \frac{2}{6}a_2 - \frac{5}{6}a_3$$

• 手工推导

$$a = \frac{3}{6} * \begin{bmatrix} 2 \\ 5 \\ 1 \\ 3 \end{bmatrix} + \frac{2}{6} * \begin{bmatrix} 10 \\ 1 \\ 5 \\ 10 \end{bmatrix} - \frac{5}{6} \begin{bmatrix} 4 \\ 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{6+20-20}{6} \\ \frac{15+2-10}{6} \\ \frac{3+10+5}{6} \\ \frac{9+20-5}{6} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

• Python描述

```
import numpy as np
a1=np.array([[2,5,1,3]]).T
a2=np.array([[10,1,5,10]]).T
a3=np.array([[4,1,(-1),1]]).T

print(3/6*a1 + 2/6*a2 - 5/6*a3)
```

```
[[1.]
 [2.]
 [3.]
 [4.]]
```

【结果分析】

1. 乘法的符号，注意区分： `*` , `.dot` , `.cross` , `.*`。
2. “1.”? 表示1是一个浮点数，表示带小数。此时小数部分=0，因为1.0 = 1，所以省略了小数部分，变成“1.”。值得注意的是Python是一种弱变量语法，它不需要显示定义数据类型，因此数据类型之间根据需要可以在一定范围内自动进行转换。例如在计算中产生分数的时候，整型就自动转换为了浮点型。本题中，由于后面的乘法运算，将分数部分约分了，所以就以浮点型的形式显示整数，如：1., 2., 3.。

3. $u^T = (3, 2, 2), v^T = (5, 3, 1)$, 求 u 和 v 的内积 (点乘) 和外积 (叉乘)。

• 手工推导

$$u \cdot v = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = 3 * 5 + 2 * 3 + 2 * 1 = 15 + 6 + 2 = 23$$

$$u \times v = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} \times \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} = \begin{bmatrix} 2 * 1 - 2 * 3 \\ 2 * 5 - 3 * 1 \\ 3 * 3 - 2 * 5 \end{bmatrix} = \begin{bmatrix} -4 \\ 7 \\ -1 \end{bmatrix}$$

- Python描述

```
import numpy as np
u=np.array([[3,2,2]])
v=np.array([[5,3,1]])

dot = np.dot(u, v.T) # 不是数学规则，而是程序的限制，大家只需要记住
cross = np.cross(u, v)
print('点乘: {}, 叉乘: {}'.format(dot, cross))
```

点乘: [[23]], 叉乘: [[-4 7 -1]]