

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



Geometric Transformations

8 几何变换

缩放、旋转、剪切、正交投影、镜像、仿射变换（含平移）...

这一章讲解常见几何变换，它们常用在计算机图形学、机器人等等领域。除此以外，本册后续几乎所有章节都依赖本章的这些几何变换。

《线性代数不难》上册，已经在不同场合见缝插针地介绍各种几何操作；本章则系统地讲解常见几何操作，以及它们背后涉及的各种线性代数工具。

这一章实际上是个很好的复习巩固的机会，因为这章用到了前文很多重要线性代数概念，比如矩阵乘法、逆矩阵、行列式、线性变换、基底等等。

本章承上启下，因为本章内容也广泛用到本书后续的各种矩阵分解（比如 QR 分解、特征值分解、Cholesky 分解、LDL 分解、奇异值分解等等）；在数据分析、机器学习算法应用中，我们也会常常用到这些几何变换。

这章的每个几何变换，特别建议大家从这几方面看：

- ▶ **矩阵乘法** 几何视角（特别是矩阵对于零向量 $\mathbf{0}$ ，单位向量 \mathbf{e}_1 、 \mathbf{e}_2 ，全 1 列向量 $\mathbf{1}$ 的变换）。
- ▶ 看**行列式**（绝对值是否大于 1；正负，还是零）；特殊矩阵的**行列式**（对角方阵，上三角矩阵，下三角矩阵 ...）。
- ▶ 从行列式看几何变换导致的**面积**或**体积**变换。
- ▶ 从**逆矩阵**（先看是否存在）看逆变换；特殊矩阵的**逆矩阵**（对角方阵，上三角矩阵，下三角矩阵，正交矩阵 ...）。
- ▶ **线性组合** 角度看平行四边形网格、平行六面体形状变换，是否发生降维。
- ▶ 从**基底**角度看变换矩阵是否构成基底，以及构成怎样基底（标准正交基、规范正交基、正交基、非正交基）。
- ▶ 变换矩阵的特点（是否**对称**，是否**对角**，是否**正交**，是否**可逆**（行列式是否为 0），矩阵乘法是否满足**交换律** ...）。

8.1 几何变换和矩阵乘法



本节你将掌握的核心技能：

- ▶ 线性映射：通过矩阵乘法 $Ax = y$ 理解如何将输入向量 x 映射到输出向量 y 。
- ▶ 在线性变换作用下，原本的单位正方形网格被变换为平行四边形网格；这些平行四边形排列平行且等距，且原点位置保持不变。
- ▶ 特殊列向量的几何意义：零向量、单位向量、全 1 列向量。
- ▶ 回顾矩阵连乘转置： $(AB)^T = B^T A^T$ 。
- ▶ 缩放：对角方阵。
- ▶ 旋转：保持长度和角度的正交线性变换，正交矩阵，行列式为 1。
- ▶ 剪切：上/下三角矩阵，主对角线元素均为 1，行列式为 1。
- ▶ 正交投影：降维，行列式为 0，不可逆矩阵。
- ▶ 镜像：正交矩阵，行列式为 -1。

这一章，缩放、旋转、剪切、正交投影、镜像会一一登场；本节先从矩阵乘法角度聊聊这些几何变换。本节的内容主要是回顾本书前文 7 章的核心知识点，相对来说很简单；本解不会展开讲解这几个几何变换。

回顾矩阵乘法 $Ax = y$

本书前文介绍，如图 1 所示，矩阵乘法相当于用矩阵 A 对列向量 x 进行线性映射 (linear mapping)

$$A_{m \times p} x_{p \times 1} = y_{m \times 1} \quad (1)$$

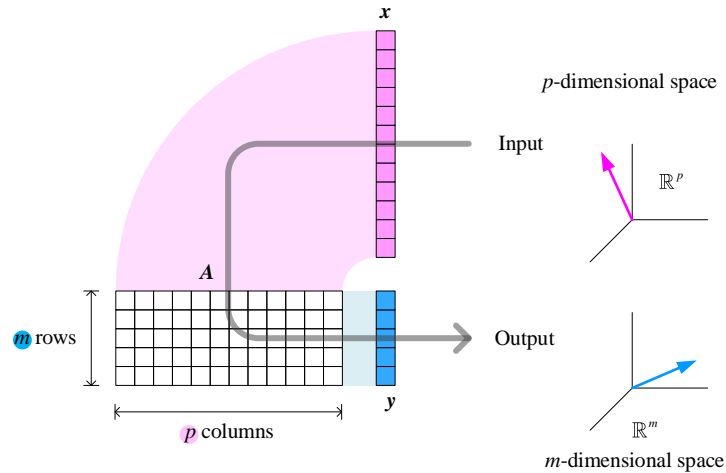
从矩阵形状来看， $(m \times p) @ (p \times 1)$ 夹在中间的 (p) 被“消去”，矩阵乘法结果的形状为 $(m \times 1)$ 。

p 维列向量 x 相当于输入向量，表示某个初始状态； x 是 \mathbb{R}^p 空间中的点。

m 维列向量 y 相当于输出向量，表示经过变换后的结果； y 是 \mathbb{R}^m 空间中的点。

从线性组合角度来看，列向量 y 是矩阵 A 列向量的线性组合。

⚠ 注意，矩阵 A 是个扁平矩阵，显然 A 列向量不构成基底。

图 1. 矩阵乘法 $Ax = y$ 的输入和输出

2 × 2 方阵

给定 2×2 矩阵 A

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \quad (2)$$

用矩阵乘法第三视角展开 $Ax = y$

$$Ax = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} a_{1,1} \\ a_{2,1} \end{bmatrix} + x_2 \begin{bmatrix} a_{1,2} \\ a_{2,2} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = y \quad (3)$$

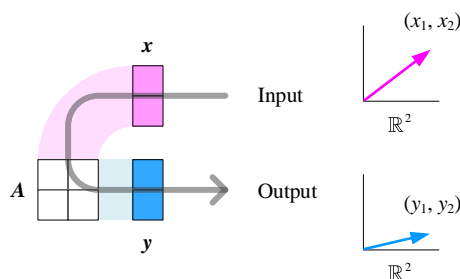
$a_1 \qquad a_2$

从矩阵形状来看， $(2 \times 2) @ (2 \times 1)$ 夹在中间的 (2) 被“消去”，矩阵乘法结果的形状为 (2×1) 。

如图 2 所示，对于矩阵乘法 $Ax = y$ ，矩阵 A 的形状为 2×2 。

输入向量 x 是 2×1 向量，相当于平面直角坐标系 \mathbb{R}^2 中的一个点 (x_1, x_2) ；也可以看作是平面上一个起点位于原点 $(0,0)$ ，终点位于 (x_1, x_2) 的向量。

输出向量 y 也是 2×1 向量，也是 \mathbb{R}^2 中的一个点 (y_1, y_2) ；也是平面直角坐标系 \mathbb{R}^2 中一个起点位于原点 $(0,0)$ ，终点位于 (y_1, y_2) 的向量。列向量 y 是 a_1 、 a_2 的线性组合。

图 2. $Ax = y$ ，矩阵 A 为 2×2

平行四边形

下面让我们看几个特殊列向量的矩阵乘法。

如图 3 所示，我们关注左图单位正方形（淡黄色）的顶点对应四个列向量：零向量 $\mathbf{0}$ ，单位向量 \mathbf{e}_1 、 \mathbf{e}_2 ，全 1 列向量 \mathbf{I} 。

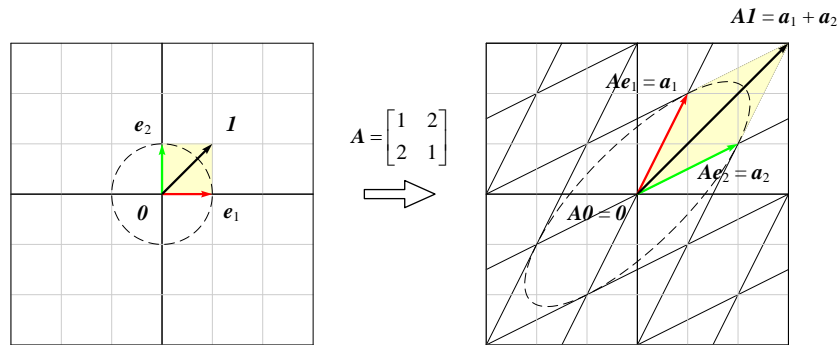


图 3. 平面上四个特殊列向量

如图 3 右图所示，经过 $Ax=y$ 线性映射后，单位正方形变成了平行四边形。

平行四边形四个顶点分别为：

- ▶ 零向量 $\mathbf{0}$ ，对应矩阵乘法 $A\mathbf{0}$
- ▶ A 的第一列向量 \mathbf{a}_1 ，对应矩阵乘法 $A\mathbf{e}_1$
- ▶ A 的第二列向量 \mathbf{a}_2 ，对应矩阵乘法 $A\mathbf{e}_2$
- ▶ A 的前两向量之和 $\mathbf{a}_1 + \mathbf{a}_2$ ，对应矩阵乘法 $A\mathbf{I}$

单位正方形对应的面积为 1，对应单位矩阵 \mathbf{I} 的行列式。180 度逆时针来看，单位矩阵 \mathbf{I} 第一列向量 \mathbf{e}_1 领先第二列向量 \mathbf{e}_2 。

图 3 中，方阵 A 的行列式为 -3，这意味着平行四边形的面积为 3；负号的原因是，180 度逆时针来看， A 的第一列向量 \mathbf{a}_1 落后第二列向量 \mathbf{a}_2 。

四个特殊列向量的矩阵乘法

下面让我们看看这四个矩阵乘法。

零向量 $\mathbf{0}$ 经过线性变换，对应矩阵乘法 $A\mathbf{0}$

$$A\mathbf{0} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0} \quad (4)$$

我们发现结果还是零向量 $\mathbf{0}$ ，即**原点位置不变**！这一点对于线性变换特别重要！

单位向量 \mathbf{e}_1 经过线性变换，对应矩阵乘法 $A\mathbf{e}_1$

$$Ae_1 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a_{1,1} \\ a_{2,1} \end{bmatrix} = \mathbf{a}_1 \quad (5)$$

以上矩阵乘法相当于提取矩阵 A 的第一列向量 \mathbf{a}_1 。 \mathbf{a}_1 对应图 3 右侧平行四边形中的红色向量。

单位向量 \mathbf{e}_2 经过线性变换，对应矩阵乘法 Ae_2

$$Ae_2 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,2} \\ a_{2,2} \end{bmatrix} = \mathbf{a}_2 \quad (6)$$

上式相当于提取 A 的第二列向量 \mathbf{a}_2 。 \mathbf{a}_2 对应图 3 右侧平行四边形中的绿色向量。

全 1 列向量 \mathbf{I} 经过线性变换，对应矩阵乘法 $A\mathbf{I}$

$$Ae_2 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} + a_{1,2} \\ a_{2,1} + a_{2,2} \end{bmatrix} = \mathbf{a}_1 + \mathbf{a}_2 \quad (7)$$

上式相当于计算 A 的列向量之和， $\mathbf{a}_1 + \mathbf{a}_2$ 。

$\mathbf{a}_1 + \mathbf{a}_2$ 对应图 3 右侧平行四边形中的黑色向量。

此外，我们还需要注意到图 3 另外一个细节——网格变化。

图 3 左图的网格是单位正方形，这代表的是标准正交基 $[\mathbf{e}_1, \mathbf{e}_2]$ 张成的空间，对应线性组合

$$k_1\mathbf{e}_1 + k_2\mathbf{e}_2 \quad (8)$$

其中， k_1, k_2 可以是任意实数。

如图 3 右图所示，经过矩阵 A 的线性变换之后，如果 $\mathbf{a}_1, \mathbf{a}_2$ 线性无关，平行四边形网格对应基底 $[\mathbf{a}_1, \mathbf{a}_2]$ 张成的空间，对应线性组合

$$k_1\mathbf{a}_1 + k_2\mathbf{a}_2 \quad (9)$$

其中， k_1, k_2 同样可以取得任意实数。

在**线性变换**作用下，原本的**单位正方形网格**被变换为**平行四边形网格**；这些平行四边形排列**平行且等距**，且**原点位置保持不变**。

仅仅“平行”还不够，我们还特别强调“等距”！如图 4 所示，单位正方形网格变换的结果还是平行的网格，但是不等距。因此图 4 这两个变换并不是线性变换。

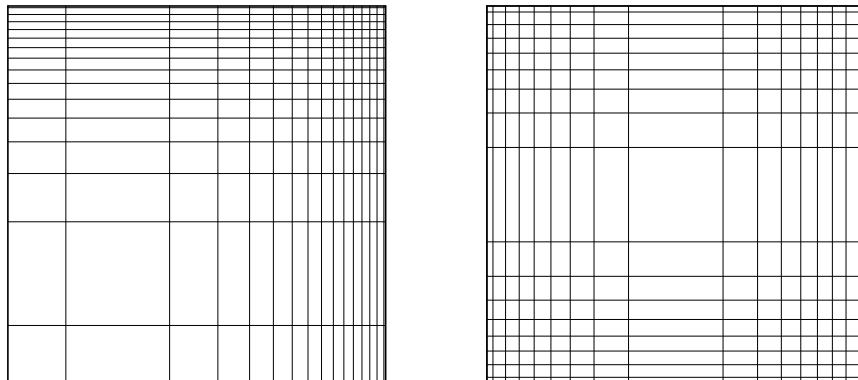


图 4. 非线性变换



大家可能也注意到，图 3 左图中的单位圆变成了椭圆。这是本书后续要着重介绍的内容，本章先按下不表，大家有个印象就好。

矩阵乘法转置

对 (3) 矩阵乘法转置得到

$$\begin{aligned} (\mathbf{Ax})^T &= \mathbf{x}^T \mathbf{A}^T = [x_1 \quad x_2] \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \\ &= x_1 \mathbf{a}_1^T + x_2 \mathbf{a}_2^T \\ &= x_1 [a_{1,1} \quad a_{2,1}] + x_2 [a_{1,2} \quad a_{2,2}] \\ &= [y_1 \quad y_2] = \mathbf{y}^T \end{aligned} \quad (10)$$

上式用二维行向量描述平面上一个点。

(10) 这个式子帮我们回忆一个矩阵乘法性质

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \quad (11)$$

这告诉我们，如下矩阵乘法相当于一系列连续几何变换

$$\mathbf{DCBAx} \quad (12)$$

对列向量 \mathbf{x} 作用的次序从右向左，即 $\mathbf{A} \rightarrow \mathbf{B} \rightarrow \mathbf{C} \rightarrow \mathbf{D}$ 。

对 (12) 转置得到

$$\mathbf{x}^T \mathbf{A}^T \mathbf{B}^T \mathbf{C}^T \mathbf{D}^T \quad (13)$$

这时候，对 \mathbf{x}^T 作用的次序从左向右。

下面让我们简单聊聊不同几何变换对应的矩阵乘法运算。

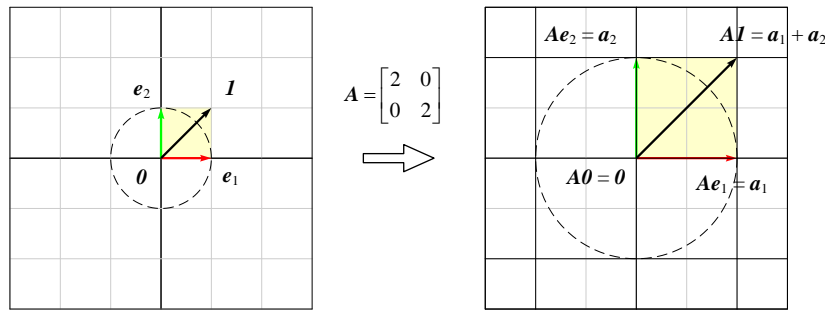
缩放

如图 5 所示，平面**缩放** (scaling) 是一种按比例拉伸、压缩平面几何形状的线性变换；变换矩阵对应的 \mathbf{A} 为对角方阵。

单位正方形网格变为正方形网格 (如图 5)、矩形网格。



缩放变换很重要，本书后文要介绍的特征值分解、奇异值分解、主成分分析等话题都会用到它。

图 5. 矩阵 A 为 2×2 , 缩放

把图 5 中方阵 A 写成列向量 $[a_1, a_2]$ 形式, 具体如下

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (14)$$

如果 a_1, a_2 线性无关, 基底 $[a_1, a_2]$ 为正交基, 网格为矩形 (正方形是特殊的矩形)。

旋转

如图 6 所示, 平面绕原点**旋转** (rotate) 是一种保持长度和角度不变的线性变换, 其对应的方阵是正交矩阵, 且行列式为 1。

回顾前文介绍的正交矩阵的性质。如果矩阵 A 为正交矩阵, A 满足 $AA^T = A^T A = I$ 。

观察图 6 的网格变换, 我们容易发现, 横平竖直的单位正方形网格 (标准正交基) 变成了旋转的单位正方形网格 (规范正交基)。



旋转变换特别重要, 我们会在谱分解、奇异值分解、主成分分析等话题中广泛用到。

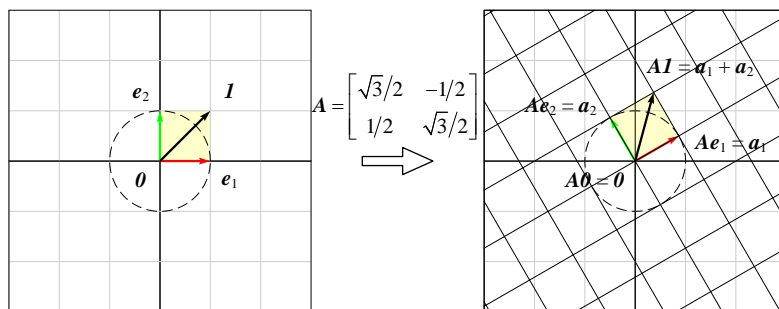
图 6. 矩阵 A 为 2×2 , (绕原点) 旋转

图 6 中矩阵 A 的列向量 a_1, a_2 均为单位向量, 且正交。

剪切

如图 7 所示，平面**剪切** (shear) 是一种使图形沿某一方向倾斜变形的线性变换；线性变换对应的矩阵为三角矩阵 (上三角或下三角，主对角线元素为 1)。

剪切变换对应方阵的行列式为 1，即保持面积不变。



有趣的是，图 7 中单位圆也变成了椭圆，这是本书后文要在 LDL 分解讨论的话题。

矩阵 A 的列向量 a_1 为单位向量，指向水平轴正方向。准确来说，图 7 展示的水平**剪切** (horizontal shear)。

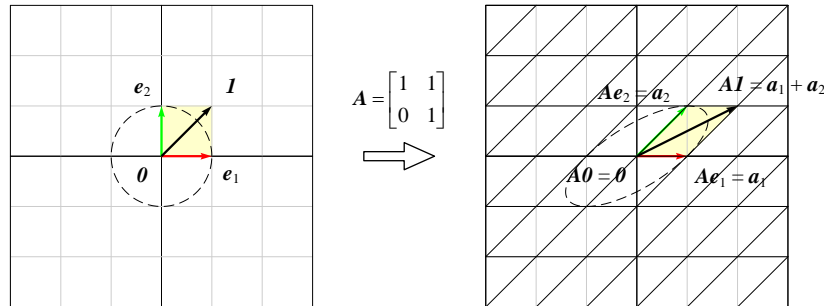


图 7. 矩阵 A 为 2×2 ，水平剪切

正交投影

如图 8 所示，平面正交投影是将点垂直投影到某一子空间的线性变换，图形发生降维。



注意，正交投影对应的方阵行列式为 0，这意味着正交投影矩阵不可逆！

这也意味着方阵的列向量线性相关，不能构成基底！比如，图 8 中矩阵 A 的列向量 a_1 、 a_2 满足 $a_1 + a_2 = 0$ 。



正交投影和本书前文的标量投影、向量投影密切相关；正交投影和施密特正交化、QR 分解、最小二乘法等密切相关，这些是下一章要讨论的核心话题。

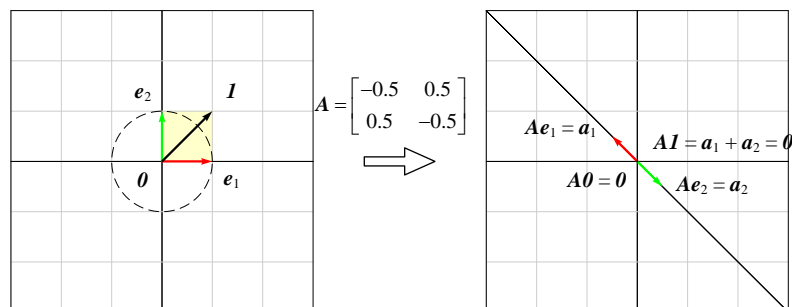


图 8. 矩阵 A 为 2×2 ，正交投影

镜像

如图 9 所示，平面**镜像** (reflection) 是一种关于某条 (过原点的) 直线对称翻转的线性变换，其线性变换对应的方阵也是正交矩阵，且行列式为-1。

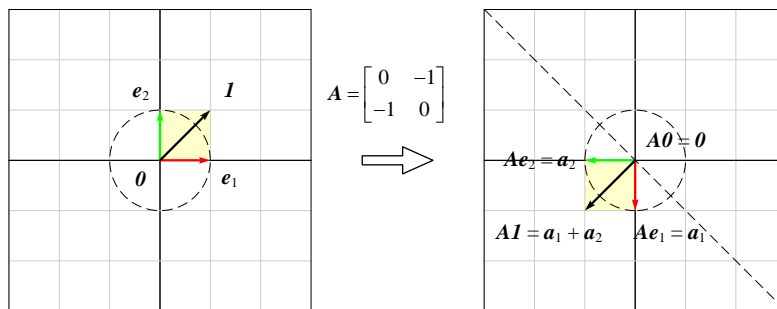


图 9. 矩阵 A 为 2×2 ，镜像

图 9 中矩阵 A 的列向量 a_1 、 a_2 也都是单位向量，且正交。但是，这个方阵不同于图 6 中旋转矩阵。


可视化

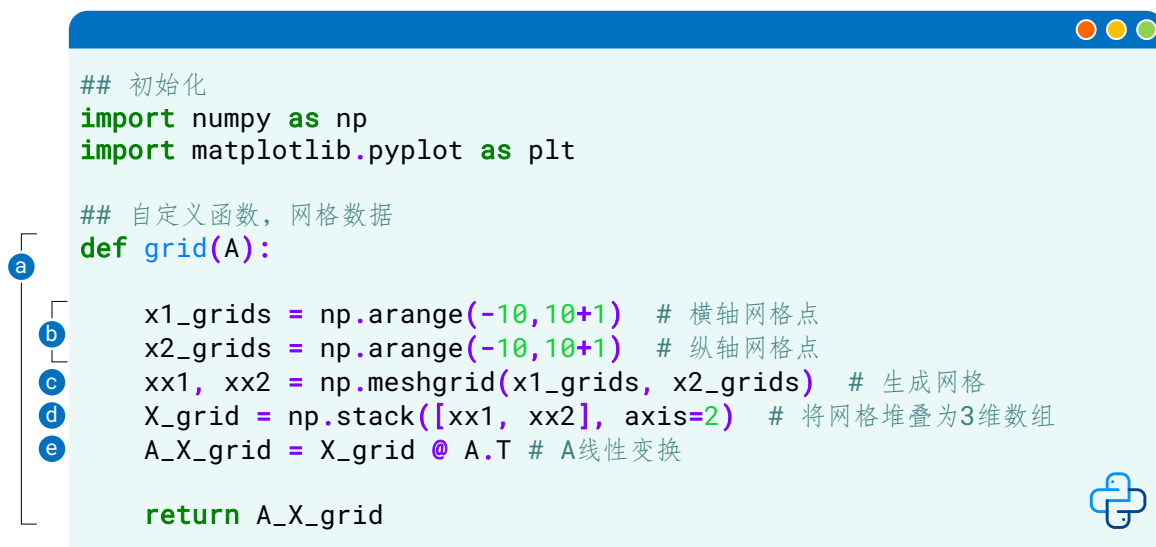
代码文件 LA_08_01_01.ipynb 绘制图 3 右图。

下面先聊聊其中的自定义函数。

代码 1 定义一个名叫 grid 的函数，它接收一个参数 A ，这个参数是一个矩阵，用来描述一个二维线性变换。

- a** 是自定义函数主体。
- b** 用 `numpy.arange()` 创建横轴、纵轴的整数网格点。
- c** 用 `numpy.meshgrid()` 函数，把上面两个一维数组扩展成一个二维的网格，`xx1` 表示所有横坐标，`xx2` 表示所有纵坐标。
- d** 用 `numpy.stack()` 把两个网格叠加在一起，形成一个三维数组，方便后续做矩阵操作。
- e** 对这个网格中的每个点进行线性变换，`@` 是矩阵乘法符号，`A.T` 表示 A 的转置，是为了匹配维度。变换之后，网格上的每个点位置都会变。

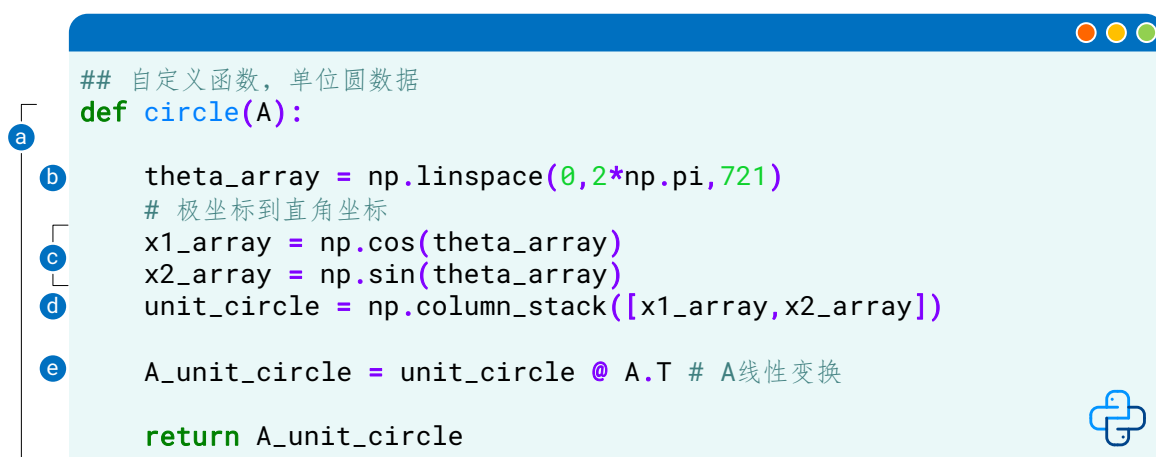
代码 1. 自定义函数，网格数据 |  LA_08_01_01.ipynb




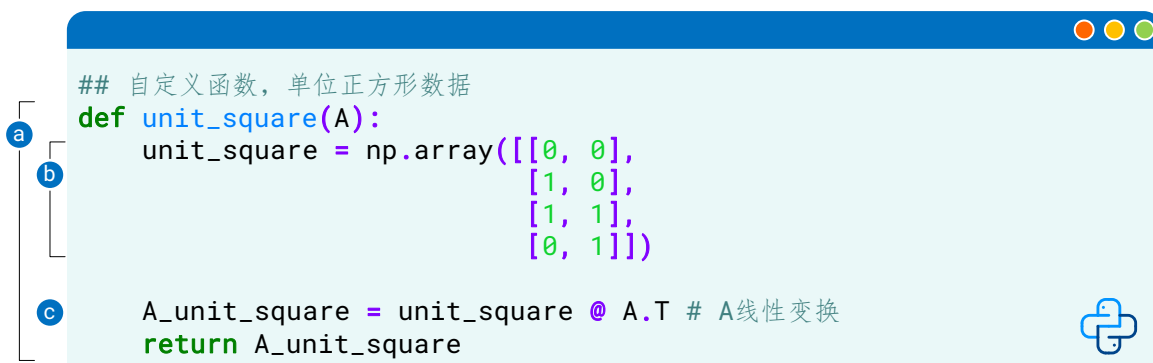
代码 2 为自定义函数，创建单位圆数据。

- a 定义一个名叫 circle 的函数，同样接受一个变换矩阵 A，它的作用是生成单位圆数据并进行线性变换。最后，用 return 返回变换后数据，给别的函数使用。
- b 用 `numpy.linspace()` 创建一个数组，表示从 0 到一圈 360 度 (用弧度表示，即 2π) 之间的角度，总共 721 个点。
- c 默认半径为 1 (单位圆)，计算每个角度对应的余弦 (横坐标)、正弦 (纵坐标)。
- d 用 `numpy.column_stack()` 把横纵坐标合并成一个二维数组，每一行是单位圆上的一个点。
- e 对单位圆上的每个点进行线性变换，得到变换后的形状轮廓的数据；本例中得到的是旋转椭圆。

代码 2. 自定义函数，单位圆数据 | LA_08_01_01.ipynb



代码 3 自定义函数创建单位正方形数据，并线性变换。代码比较简单请大家自行分析。值得注意的是 b 中四个行向量代表单位正方形的四个顶点。

代码 3. 自定义函数，单位正方形数据 |  LA_08_01_01.ipynb


```


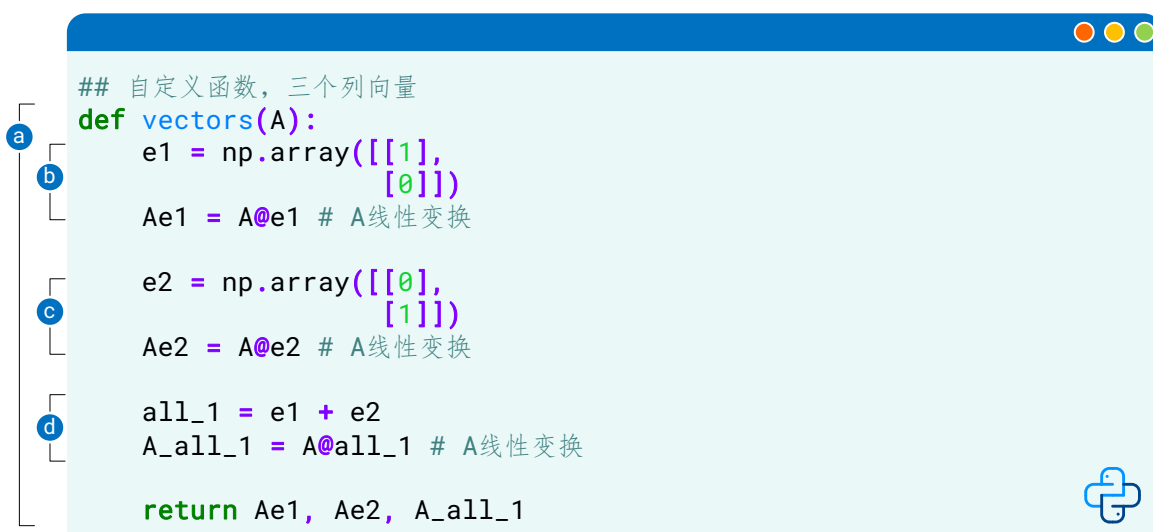
## 自定义函数，单位正方形数据
def unit_square(A):
    unit_square = np.array([[0, 0],
                             [1, 0],
                             [1, 1],
                             [0, 1]])

    A_unit_square = unit_square @ A.T # A线性变换
    return A_unit_square

```

代码 4 自定义函数，创建单位正方形数据。

- a 定义一个函数 `vectors`，作用是生成单位向量 e_1 、 e_2 ，全 1 列向量 I ，并查看它们线性变换后的效果。
- b 定义横轴正方向单位向量 e_1 。然后，对这个向量进行线性变换。
- c 定义纵轴正方向单位向量 e_2 。然后，对这个向量进行相同线性变换。
- d 将 e_1 、 e_2 相加得到全 1 列向量，再对其进行线性变换。

代码 4. 自定义函数，单位正方形数据 |  LA_08_01_01.ipynb


```

## 自定义函数，三个列向量
def vectors(A):
    e1 = np.array([[1],
                   [0]])
    Ae1 = A@e1 # A线性变换

    e2 = np.array([[0],
                   [1]])
    Ae2 = A@e2 # A线性变换

    all_1 = e1 + e2
    A_all_1 = A@all_1 # A线性变换

    return Ae1, Ae2, A_all_1

```

代码 5 为自定义可视化函数。

- a 定义一个叫 `visualize` 的函数，用来把上面所有变换后的图形画出来。
- b 调用前面的自定义函数，分别生成线性变换后网格、单位正方形、单位圆、三个向量的数据。
- c 先一行一行地画出变换后的网格，表示横向网格线；然后再一列一列地画出变换后的网格，表示纵向网格线。
- d 用 `quiver()` 绘制三个箭头。这个函数之前已经讲过很多次，请大家自行回顾参数。
- e 用 `plot()` 绘制单位圆线性变换后的结果。

f 用 fill() 绘制单位正方形变换后的结果，背景填充颜色为淡黄色。

代码 5. 自定义可视化函数 |  LA_08_01_01.ipynb

```
## 定义可视化函数
def visualize(A):
    fig, ax = plt.subplots()

    # 调用自定义函数创建数据
    A_X_grid = grid(A)
    A_unit_square = unit_square(A)
    A_unit_circle = circle(A)
    Ae1, Ae2, A_all_1 = vectors(A)

    # 绘制变换后的网格
    for i in range(A_X_grid.shape[0]):
        plt.plot(A_X_grid[i, :, 0], A_X_grid[i, :, 1],
                 'k-', linewidth=0.5) # 绘制网格行
    for j in range(A_X_grid.shape[1]):
        plt.plot(A_X_grid[:, j, 0], A_X_grid[:, j, 1],
                 'k-', linewidth=0.5) # 绘制网格列

    # 绘制变换后的列向量
    plt.quiver(0, 0, Ae1[0], Ae1[1],
               angles='xy', scale_units='xy',
               scale=1, color=[1, 0, 0], zorder = 1e5)
    plt.quiver(0, 0, Ae2[0], Ae2[1],
               angles='xy', scale_units='xy',
               scale=1, color=[0, 1, 0], zorder = 1e5)
    plt.quiver(0, 0, A_all_1[0], A_all_1[1],
               angles='xy', scale_units='xy',
               scale=1, color=[0, 0, 0], zorder = 1e5)

    # 绘制变换后单位圆
    plt.plot(A_unit_circle[:,0], A_unit_circle[:,1],
             c = 'k', ls = '--')

    # 绘制变换后单位正方形
    plt.fill(A_unit_square[:,0], A_unit_square[:,1],
            c = '#FFFFCC', ls = '--', zorder = 1)

    # 装饰
    plt.axvline(x=0, color='k', zorder=0)
    plt.axhline(y=0, color='k', zorder=0)
    ax.tick_params(axis='both', which='both', length=0,
                  labelbottom=False, labelleft=False)
    ax.set_aspect(1)
    lim = 3; ax.set_xlim([-lim, lim]); ax.set_ylim([-lim, lim])
    plt.xticks(np.arange(-lim, lim))
    plt.yticks(np.arange(-lim, lim))
    ax.grid(linestyle='--', linewidth=0.15, color=[0.8, 0.8, 0.8])
```

3 × 3 方阵

最后，让我们把线性变换扩展到三维空间。给定 3×3 矩阵 A

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \quad (15)$$

3×3 矩阵 A 完成了 3 维向量 x 到 3 维向量 y 的映射。

如图 10 所示，对于矩阵乘法 $Ax = y$ ，当矩阵 A 的形状为 3×3 (3 行、3 列)，输入向量 x 是 3×1 向量， x 代表三维空间 \mathbb{R}^3 的坐标点。

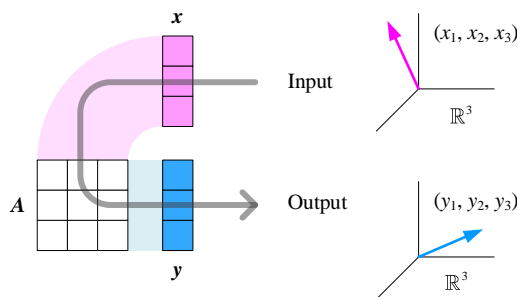


图 10. 矩阵 A 为 3×3

输出向量 y 也是 3×1 向量， y 也是三维空间 \mathbb{R}^3 的坐标点。

从矩阵形状来看， $(3 \times 3) @ (3 \times 1)$ 夹在中间的 (3) 被“消去”，矩阵乘法结果的形状为 (3×1) 。

同样，让我们看几个特殊矩阵乘法，具体如图 11。

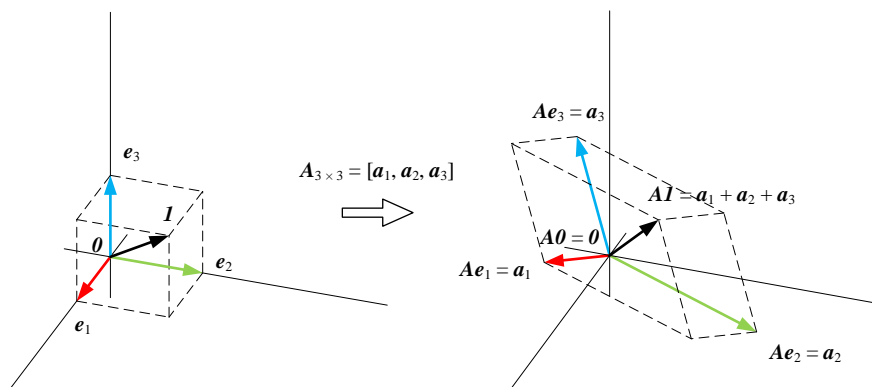


图 11. 三维空间中几个特殊列向量

零向量 0 经过线性变换，对应矩阵乘法 $A0$

$$A\mathbf{0} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0} \quad (16)$$

同样地，三维空间中的线性变换中原点位置没有变化。

单位向量 \mathbf{e}_1 经过线性变换，对应矩阵乘法 $A\mathbf{e}_1$

$$A\mathbf{e}_1 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \end{bmatrix} = \mathbf{a}_1 \quad (17)$$

上式矩阵乘法提取的是方阵 A 第一列。

单位向量 \mathbf{e}_2 经过线性变换，对应矩阵乘法 $A\mathbf{e}_2$

$$A\mathbf{e}_2 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \end{bmatrix} = \mathbf{a}_2 \quad (18)$$

上式提取的 A 第二列。

单位向量 \mathbf{e}_3 经过线性变换，对应矩阵乘法 $A\mathbf{e}_3$

$$A\mathbf{e}_3 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \end{bmatrix} = \mathbf{a}_3 \quad (19)$$

上式提取的 A 第三列。

全 1 列向量 \mathbf{I} 经过线性变换，对应矩阵乘法 $A\mathbf{I}$

$$A\mathbf{I} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} + a_{1,2} + a_{1,3} \\ a_{2,1} + a_{2,2} + a_{2,3} \\ a_{3,1} + a_{3,2} + a_{3,3} \end{bmatrix} = \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3 \quad (20)$$

上式为 A 的三个列向量求和。

一个矩阵 A 乘以 (形状匹配) 的全 1 列向量，即 $A\mathbf{I}$ ，得到的是行向量求和。请大家思考，全 1 列项的转置 \mathbf{I}^T (形状匹配) 乘上一个矩阵 A ，即 $\mathbf{I}^T A \mathbf{I}$ ，得到结果是什么？矩阵乘法 $\mathbf{I}^T A \mathbf{I}$ 得到的又是什么？ $\mathbf{I}^T A \mathbf{I}$ 两个全 1 列项量形状是否相同？什么时候相同？

如图 11 所示，我们发现单位立方体，变成了平行六面体。

图 11 左图代表的是标准正交基 $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ 张成的空间，对应线性组合

$$k_1 \mathbf{e}_1 + k_2 \mathbf{e}_2 + k_3 \mathbf{e}_3 \quad (21)$$

其中, k_1 、 k_2 、 k_3 可以是任意实数。

图 11 右图所示为经过矩阵 A 的线性变换之后, 如果 \mathbf{a}_1 、 \mathbf{a}_2 、 \mathbf{a}_3 线性无关, 平行六面体对应基底 $[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ 张成的空间, 对应线性组合

$$k_1\mathbf{a}_1 + k_2\mathbf{a}_2 + k_3\mathbf{a}_3 \quad (22)$$

其中, k_1 、 k_2 、 k_3 同样可以取得任意实数。

套用本节前文的话, 在**线性变换**作用下, 原本的**单位正方体网格**被变换为**平行六面体网格**; 这些平行六面体排列**平行且等距**, 且**原点位置保持不变**。



请大家用 DeepSeek/ChatGPT 等工具完成本节如下习题。

Q1. 请用矩阵乘法计算如下方阵分别对零向量 $\mathbf{0}$, 单位向量 \mathbf{e}_1 、 \mathbf{e}_2 , 全 1 列向量 $\mathbf{1}$ 的变换。

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

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

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

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

▶ $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

▶ $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

▶ $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

▶ $\begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}$

▶ $\begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$

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

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

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

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

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

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

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

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

▶ $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

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

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

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

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

Q2. 请用语言大致描述 **Q1** 方阵完成怎样的几何变换。建议先用 **Q1** 结果判断，然后用本节配套代码，逐个将方阵带入计算并可视化检验自己的判断。

Q3. 请分析 **Q1** 中哪些矩阵列向量线性相关？

Q4. 请分析 **Q1** 中哪些矩阵列向量可以构成基底，分析基底类型，并解释为什么？

Q5. 请计算 **Q1** 方阵每个列向量的长度 (大小、模、 L^2 范数)，列向量之间的夹角 (用内积)。

Q6. 请在方格纸上绘制 **Q1** 每个方阵列向量构成的平行四边形。

Q7. 请计算 **Q1** 每个方阵的行列式；请思考方阵有怎样特点，怎么算最便捷；并解释行列式正、负或为 0 的原因。

Q8. 请解释 **Q1** 每个方阵是否存在逆矩阵；如果存在的话，请计算逆矩阵。