

## 22

## Quiver Plots

## 箭头图

有大小、有方向



生存还是死亡，这是一个问题：

要想活的高贵，到底是该忍气吞声

接受厄运的捶打，

还是该拿起武器痛击无尽的烦恼，

打败一切？

*To be, or not to be, that is the question:*

*Whether 'tis nobler in the mind to suffer*

*The slings and arrows of outrageous fortune,*

*Or to take arms against a sea of troubles,*

*And by opposing end them?*

—— 威廉·莎士比亚 (William Shakespeare) | 英国剧作家 | 1564 ~ 1616



- ◀ matplotlib.pyplot.axhline() 绘制水平线
- ◀ matplotlib.pyplot.axvline() 绘制垂直线
- ◀ matplotlib.pyplot.fill\_between() 区域填充颜色
- ◀ matplotlib.pyplot.plot() 绘制线图
- ◀ matplotlib.pyplot.quiver() 绘制箭头图
- ◀ matplotlib.pyplot.scatter() 绘制散点图
- ◀ matplotlib.pyplot.text() 在图片上打印文字
- ◀ numpy.flip() 指定轴翻转数组
- ◀ numpy.fliplr() 左右翻转数组
- ◀ numpy.flipud() 上下翻转数组
- ◀ numpy.meshgrid() 创建网格化数据
- ◀ numpy.prod() 指定轴的元素乘积
- ◀ sympy.diff() 求解符号导数和偏导解析式
- ◀ sympy.lambdify() 将符号表达式转化为函数
- ◀ sympy.symbols() 定义符号变量



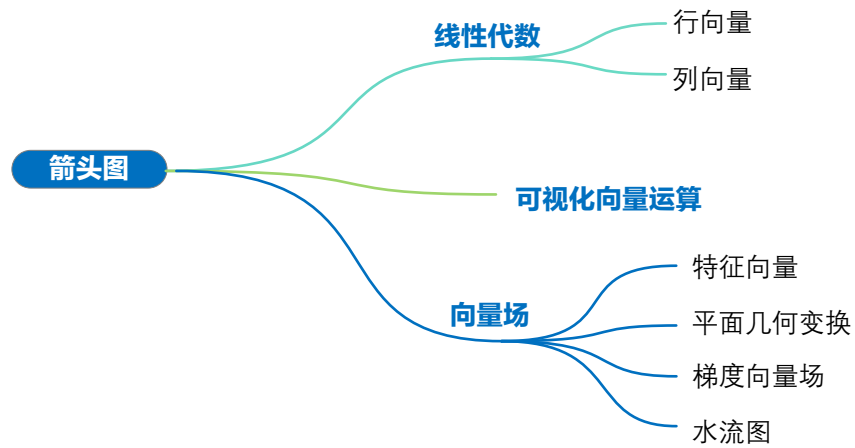
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

## 22.1 向量

《编程不难》专门介绍过向量这个概念。简单来说，**向量** (vector) 可以用有向线段表示，具有方向和大小两个属性。而**标量** (scaler) 只有大小这一个属性。

在二维空间中，一个向量可以表示为一个有序的数对  $(x, y)$ 、 $[x, y]$ 、 $[x, y]^T$ 。如图 1 所示，向量也可以用一个有向线段来表示，线段的起点为原点  $(0, 0)$ ，终点为  $(x, y)$ 。其中， $x$  表示向量在  $x$  轴上的投影， $y$  表示向量在  $y$  轴上的投影，也称为向量的横纵坐标。

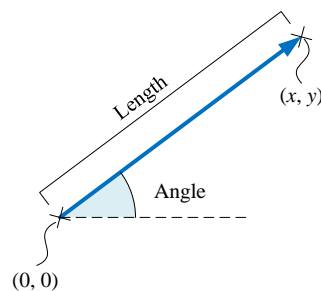


图 1. 向量起点、终点、大小和方向

类似地，在三维空间中，一个向量可以表示为  $(x, y, z)$ 、 $[x, y, z]$ 、 $[x, y, z]^T$ 。三维向量也可以用一个有向线段来表示，线段的起点为原点  $(0, 0, 0)$ ，终点为  $(x, y, z)$ 。其中， $x$ 、 $y$  和  $z$  分别表示向量在  $x$  轴、 $y$  轴和  $z$  轴上的投影长度，也称为向量的三个坐标。

**行向量** (row vector) 是由一系列数字或符号排列成的一行序列。**列向量** (column vector) 是由一系列数字或符号排列成的一列序列。

如图 2 所示，矩阵  $A$  可以视作由一组行向量、列向量构造而成。

而  $A$  的行向量， $\mathbf{a}^{(1)}$ 、 $\mathbf{a}^{(2)}$ 、 $\mathbf{a}^{(3)}$ ，可以看成是平面中的三个箭头，而  $A$  的列向量， $\mathbf{a}_1$ 、 $\mathbf{a}_2$ ，可以看成是三维空间中的两个箭头。

图 3 所示为  $B = A^T$  的行列向量。而  $B$  的行向量是三维空间的两个箭头， $B$  的列向量是平面中的三个箭头。

本章将介绍如何用箭头图可视化向量、向量运算。

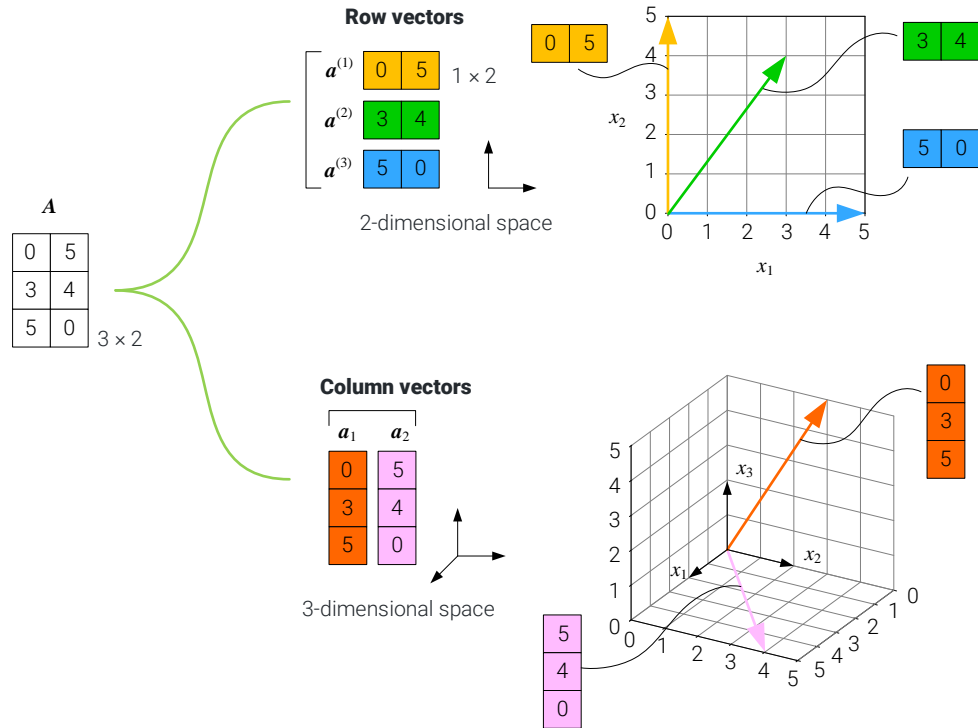


图 2. 行向量和列向量, 图片来自《编程不难》

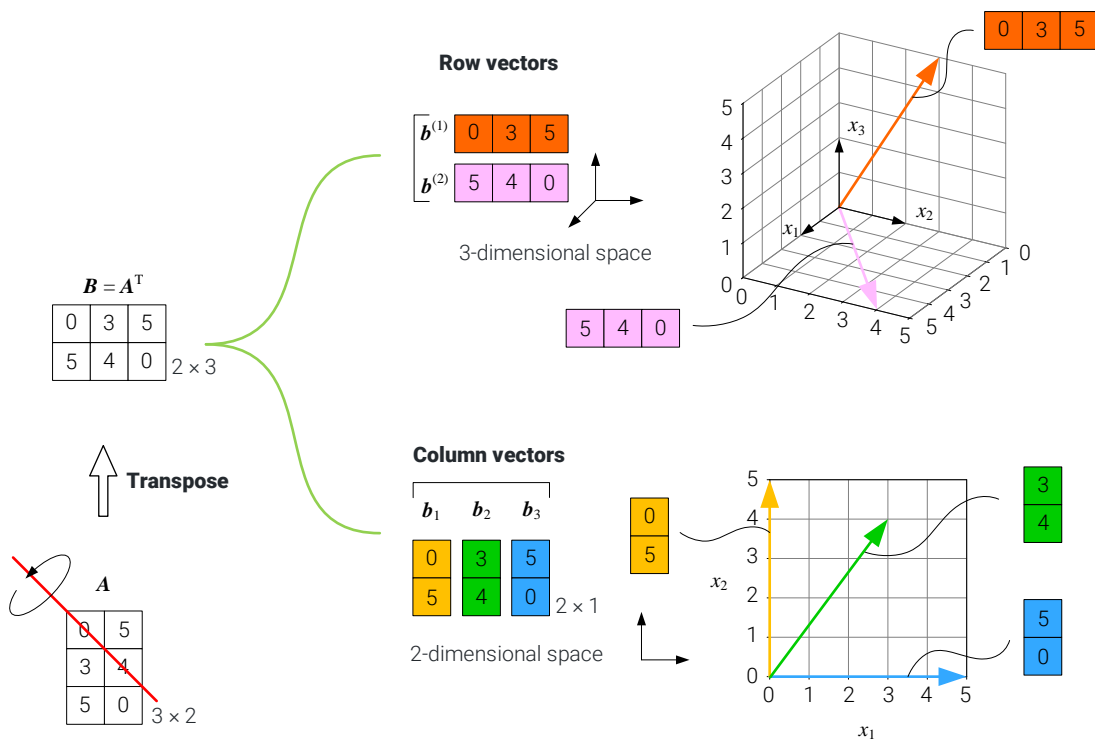


图 3. 转置之后矩阵的行向量和列向量, 图片来自《编程不难》

## 22.2 箭头

`quiver` 是 `matplotlib.pyplot` 模块中的一个函数，用于绘制二维、三维箭头图。二维箭头图的函数和基本参数为 `matplotlib.pyplot.quiver(x, y, u, v, scale=1)`。

其中， $x$  和  $y$  是箭头起始点的坐标， $u$  和  $v$  是箭头在两个方向的投影量。默认情况下，箭头的长度是按照输入数据的比例来绘制的，可以通过 `scale` 参数进行调整。图 4 所示为 `quiver` 箭头的常用参数。

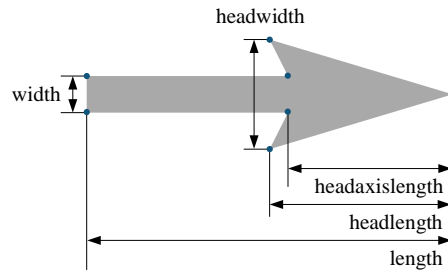


图 4. 箭头的参数

### 可视化向量运算

图 5 所示为利用箭头图可视化向量加法。图 6 所示为利用箭头图展示向量长度，即模 (norm)。图 7 可视化向量减法。

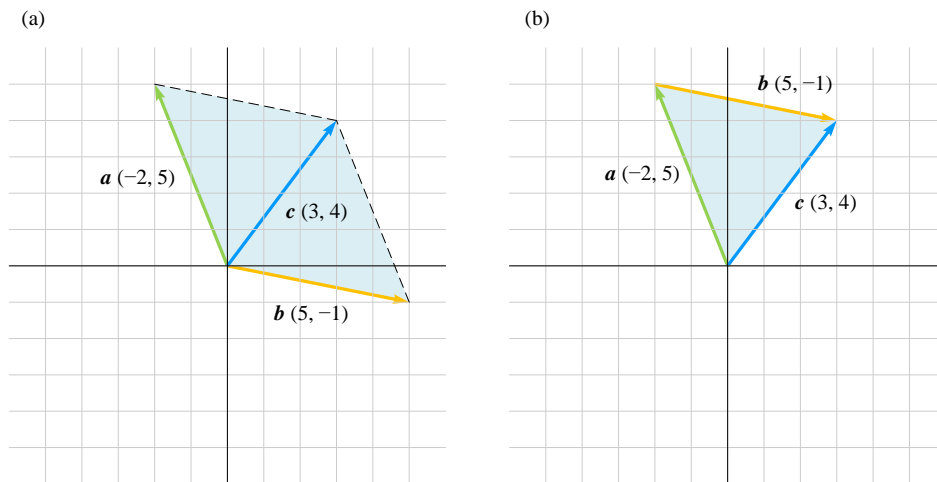
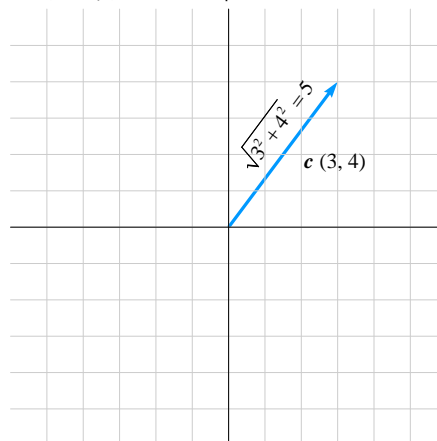


图 5. 可视化二维向量加法

Bk\_2\_Ch22\_1.ipynb



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

图 6. 可视化二维向量长度 | Bk\_2\_Ch22\_1.ipynb

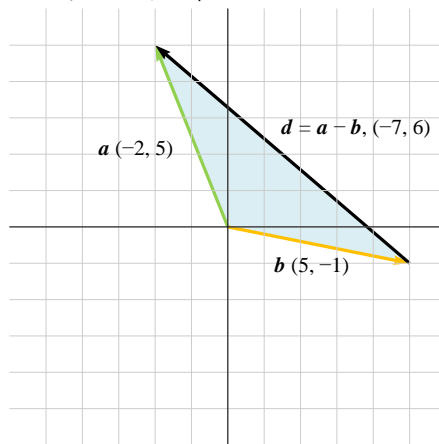


图 7. 可视化二维向量减法 | Bk\_2\_Ch22\_1.ipynb

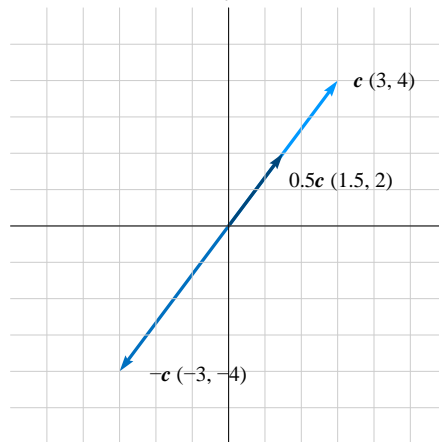


图 8. 可视化标量乘向量 | Bk\_2\_Ch22\_1.ipynb

Bk\_2\_Ch22\_1.ipynb 绘制上述代码，其中代码 1 绘制图 5 (a)，下面聊聊其中核心语句。

**a** 在轴对象 `ax` 上用 `quiver()` 方法绘制平面箭头图，表达向量  $a$ 。  
 $(0, 0)$  代表向量起点坐标。`a[0]` 代表向量在  $x$  轴上的投影，`a[1]` 代表向量在  $y$  轴上的投影。参数 `angles='xy'` 指定箭头应该以  $x$  和  $y$  轴的角度来表示。  
 参数 `scale_units='xy'` 指定箭头的比例应该根据  $x$  和  $y$  轴的单位来缩放。  
 参数 `scale=1` 指定箭头的长度应该乘以的比例因子。在这里，箭头的长度将乘以 1，保持原始长度。本书在绘制平面箭头图时，一般都会用 `angles='xy'`，`scale_units='xy'`，`scale=1` 这组设置。

**?** 请大家自己想办法将 `angles='xy'`，`scale_units='xy'`，`scale=1` 保存在一个变量里，然后在绘制箭头图时调用这个变量。

**b** 在轴对象 `ax` 上用 `text()` 方法创建文本注释。

$(-3, 2.5)$  是文本注释的具体坐标。

' $\$a (-2, 5)\$$ ' 是要显示的文本内容。'\$' 符号会让文本被解释为数学表达。

`fontsize=10` 设置文本字体大小。

**c** 和 **d** 用箭头图表达向量  $b$ ，并注释。**e** 和 **f** 用箭头图表达向量  $c$ ，并注释。

**g** 用 `plot()` 绘制向量  $a$ 、向量  $c$  终点连线，颜色 `c` 为黑色，线型 `ls` 为划线。

**h** 用 `plot()` 绘制向量  $b$ 、向量  $c$  终点连线。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

**i** 是首尾相连封闭坐标点，每一行代表一个平面坐标点。

**j** 用 `matplotlib.pyplot.fill()`，简作 `plt.fill()`，来填充多边形。

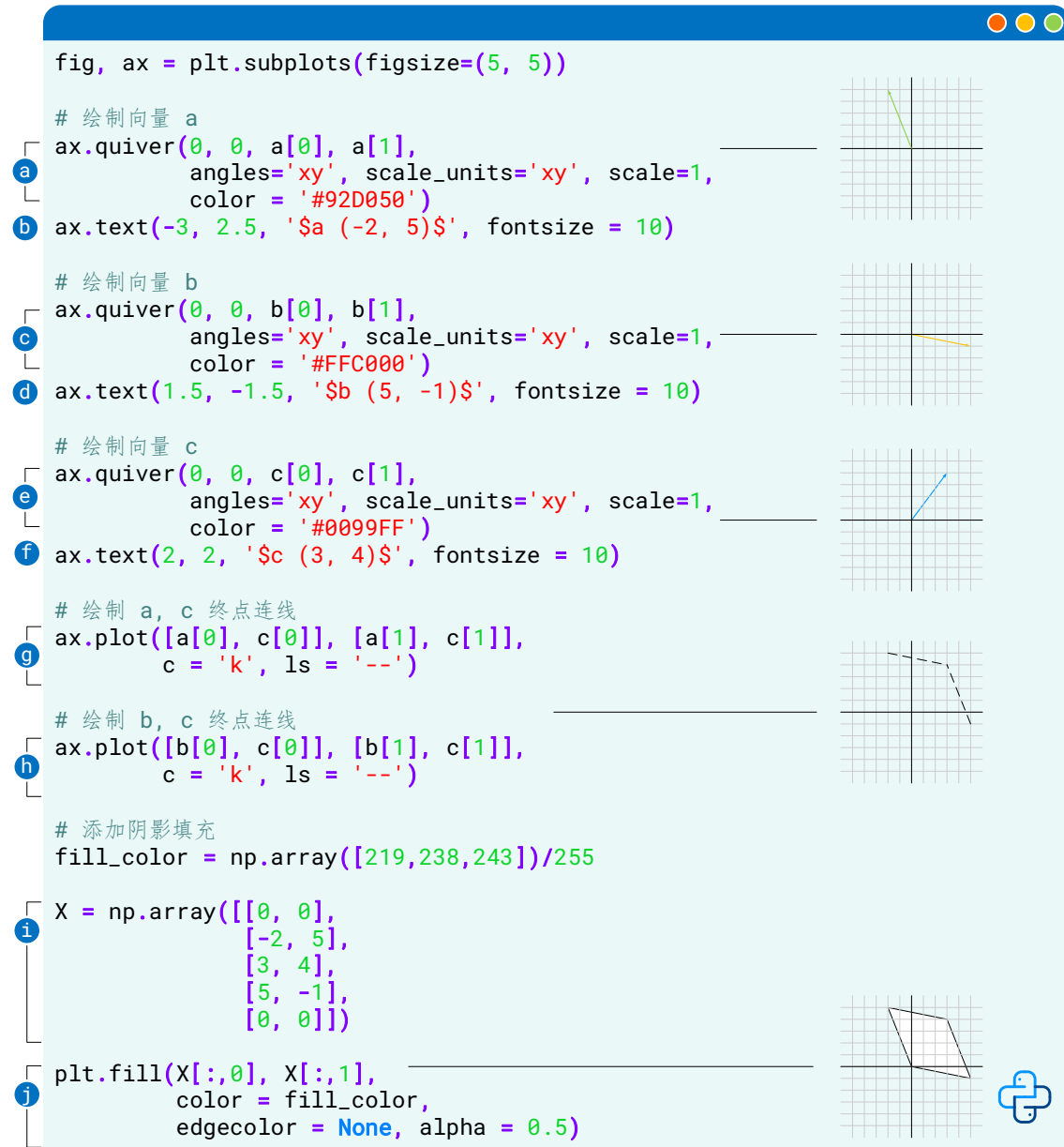
`X[:,0]` 和 `X[:,1]` 分别是要填充的多边形顶点横、纵坐标。注意，`X` 中最后一行和第一行重复；实际上，`X` 最后一行数据可以删除，也能绘制封闭填充形状。

`color=fill_color` 指定填充颜色。

`edgecolor=None` 设定多边形边缘的颜色，设置为 `None` 表示不绘制边缘。

`alpha=0.5` 是填充颜色的透明度，取值范围为 0（完全透明）到 1（完全不透明）。

请大家自行分析 `Bk_2_Ch22_1.ipynb` 剩余代码。



代码 1. 可视化向量加法 | `Bk_2_Ch22_1.ipynb`

### 三维向量

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

图 9 所示为利用三维箭头图可视化三维向量加法。图 10 所示为向量投影到  $xy$  平面、 $xz$  平面、 $yz$  平面。图 11 所示为向量投影到  $x$  轴、 $y$  轴、 $z$  轴。

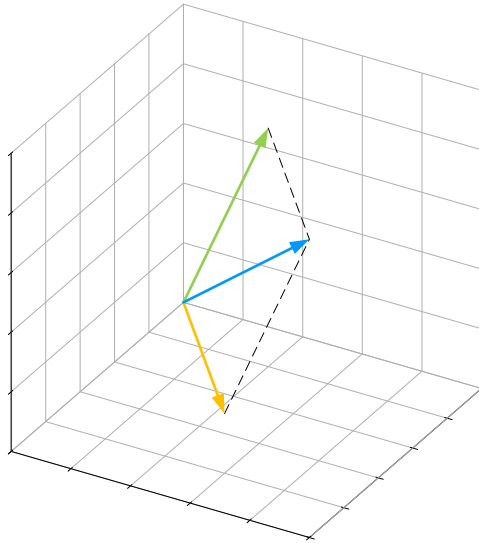


图 9. 可视化三维向量加法

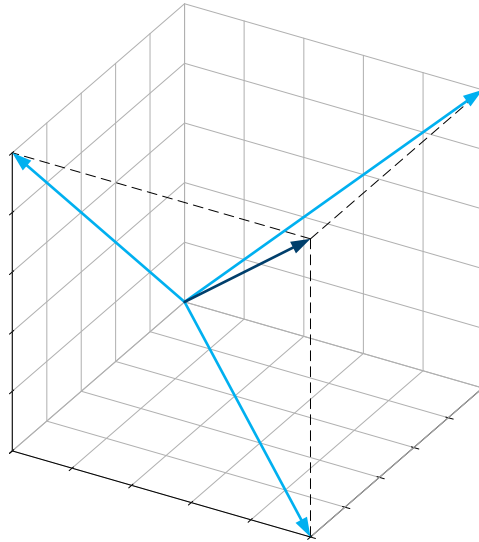
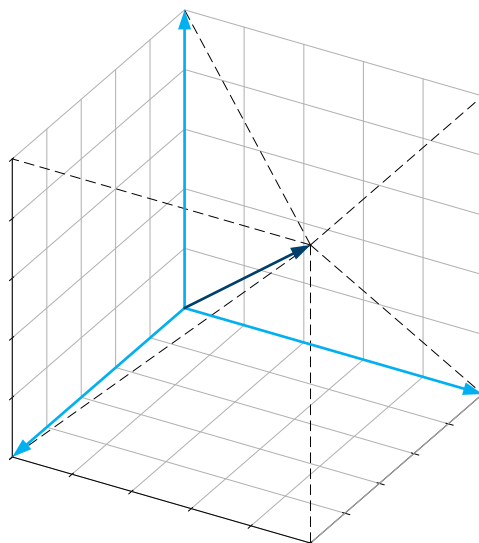


图 10. 三维向量投影到平面



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

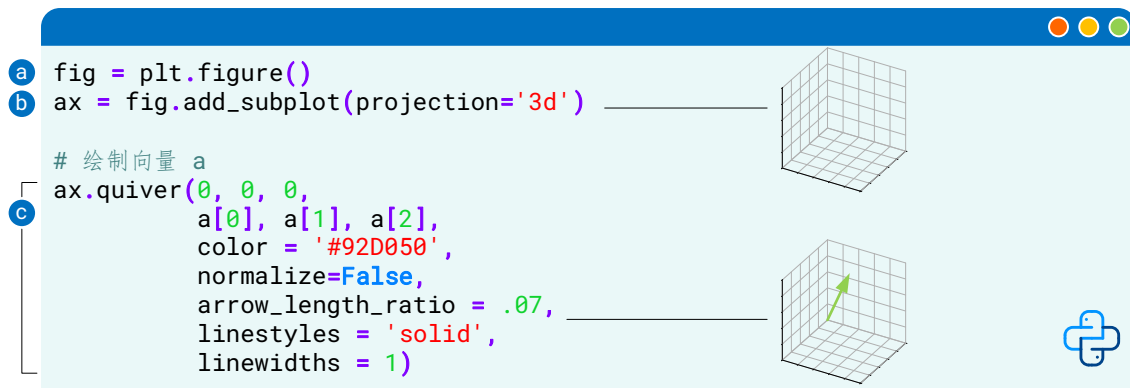
欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)



图 11. 三维向量投影到轴

Bk\_2\_Ch22\_2.ipynb 绘制图 9 ~ 图 11，代码相对比较简单，我们仅仅介绍代码 2 中这几句。

- a 用 `matplotlib.pyplot.figure()`，简作 `plt.figure()`，创建图形对象 `fig`。
- b 用 `add_subplot()` 在 `fig` 上添加三维轴对象 `ax`。
- c 用 `quiver()` 方法在三维轴对象 `ax` 上绘制箭头图，代表三维向量。  
 $(0, 0, 0)$  代表向量起点坐标。  
`a[0]`、`a[1]`、`a[2]` 分别为向量在  $x$ 、 $y$ 、 $z$  轴上的分量。  
`color='#92D050'` 表示箭头的颜色。  
`normalize=False` 表示不对箭头进行归一化；如果设置为 `True`，箭头长度将被归一化为 1。  
`arrow_length_ratio=.07` 表示箭头头部的长度与整个箭头长度的比例。  
`linestyle='solid'` 设置箭头的线条样式，这里是实线。  
`linewidths=1` 设置箭头的线条宽度。



代码 2. 可视化三维向量 | Bk\_2\_Ch22\_2.ipynb

Bk\_2\_Ch22\_2.ipynb 还绘制图 14。图 14 中红色箭头代表起点为原点的  $x$  轴正方向单位向量，绿色箭头为  $y$  轴正方向单位向量，蓝色箭头为  $z$  轴正方向单位向量。本书前文用过类似代码生成图 14 子图布局，请大家回顾。

## 22.3 向量场

除了单独绘制箭头图，我们还可以绘制向量场。向量场是指在空间中的每一个点都存在一个向量的集合。在数学中，向量场通常用函数来描述，这个函数将每个点映射到该点处的向量。这个函数被称为向量场的场函数或者向量场的定义式。向量场可以用来描述许多物理现象，例如流体力学中的速度场，电场、磁场、水流、风向等等。

### 可视化特征向量

图 15 所示为利用向量场可视化特征向量。

图 15 (a) 中每一个蓝色箭头代表一个特定方向的单位向量  $\mathbf{v}$ 。给定矩阵  $\mathbf{A}$ ，图中的红色箭头代表  $\mathbf{A}\mathbf{v}$  的计算结果。特别地，如果  $\mathbf{v}$  和  $\mathbf{A}\mathbf{v}$  在同一方向上，即满足  $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ 。 $\mathbf{v}$  叫做  $\mathbf{A}$  的特征向量 (eigen vector)， $\lambda$  叫做  $\mathbf{A}$  的特征值 (eigen value)。

图 15 (a) 中每一对向量 ( $\mathbf{v}$  和  $\mathbf{A}\mathbf{v}$ ) 都有自己特定的起点。

图 15 (b) 也用来展示特征值分解，不同的是这幅图中，每一对向量 ( $\mathbf{v}$  和  $\mathbf{A}\mathbf{v}$ ) 的起点完全相同，都是原点  $(0,0)$ 。而且，图 15 (b) 的每一对向量用特定的颜色渲染。这幅图中，我们还看到单位圆 (unit circle) 转化为旋转椭圆。



鸢尾花书《矩阵力量》将专门介绍特征值分解、特征向量、特征值这些概念。

Bk\_2\_Ch22\_3.ipynb 绘制图 15 两幅图，代码比较简单，我们只分析代码 3 这一段。

**a** 绘制用向量场展示线性映射之前的所有向量，即图 15 (a) 中所有蓝色箭头。大家查看代码时会发现，`xx1`、`xx2`、`uu`、`vv` 均为二维数组，而且形状相同。`xx1`、`xx2` 分别代表所有蓝色箭头的起点横、纵坐标；`uu`、`vv` 分别代表所有蓝色箭头在横、纵轴方向投影。

**b** 也用向量场展示线性映射之后所有向量，即图 15 (a) 中所有红箭头。

请大家自行分析 Bk\_2\_Ch22\_3.ipynb 剩余代码。

```
fig, ax = plt.subplots(figsize = (6,6))

# 绘制线性映射之前的向量
ax.quiver(xx1,xx2, # 向量始点位置坐标，网格化数据
          uu,vv,   # 两个方向的投影量
          angles='xy', scale_units='xy',
          scale=0.8, # 稍微放大
          width = 0.0025, # 宽度，默认0.005
          edgecolor='none', facecolor= 'b')

# 绘制线性映射之后的向量
ax.quiver(xx1,xx2,
          uu_new,vv_new,
          angles='xy', scale_units='xy',
          scale=0.8,
          width = 0.0025,
          edgecolor='none', facecolor= 'r')
```

代码 3. 可视化特征向量 | Bk\_2\_Ch22\_3.ipynb

## 平面几何变换

图 15 所示为利用平面箭头图可视化平面几何变换。

这幅图的特殊之处是，我们用色谱渲染不同箭头。颜色映射的依据是图 15 (a) 箭头的角度值。

Bk\_2\_Ch22\_4.ipynb 绘制图 15，下面聊聊代码 4 中语句。

**a** 构造复数， $1j$  为虚数单位。本书第 32 章会专门介绍如何可视化复数和复数函数。

**b** 用 `numpy.angle()` 计算复数辐角，对应向量和  $x$  轴正方向夹角。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

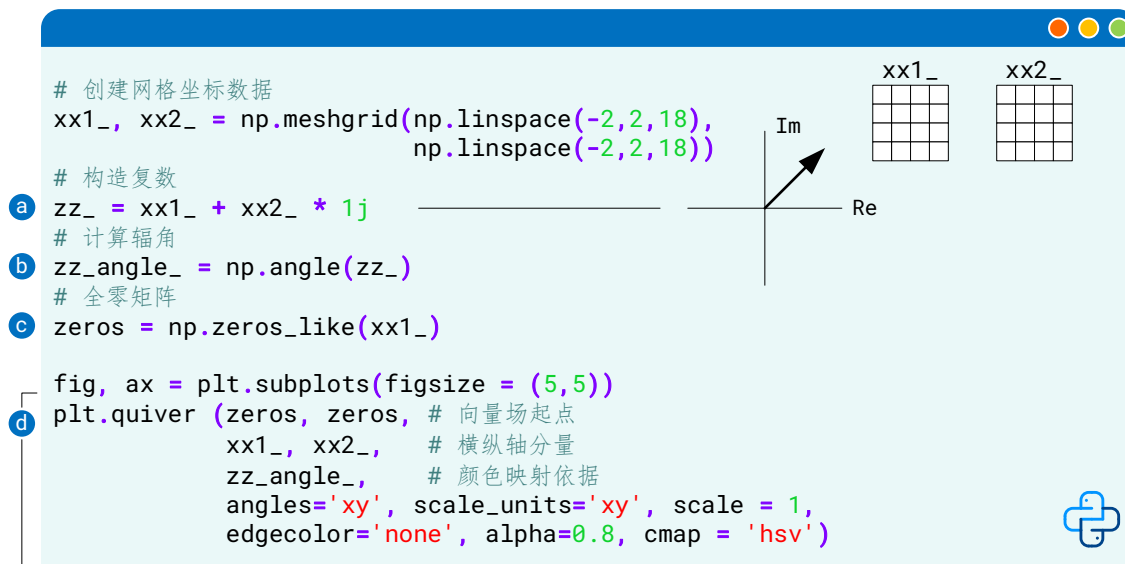
版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

- c 用 `numpy.zeros_like()` 产生全 0 数组，用作向量场中所有向量起点坐标。
  - d 用 `matplotlib.pyplot.quiver()`，简作 `plt.quiver()`，绘制向量场。
- 值得注意的是，第 5 个函数输入 `zz_angle_` 为角度值，用来作为颜色映射的依据。参数 `cmap` 指定颜色映射为 'hsv'。



代码 4. 用色谱渲染向量场 | Bk\_2\_Ch22\_4.ipynb

## 梯度向量场

图 17 所示为利用向量场二元、三元函数的梯度向量场。

简单来说，对于多元函数，梯度（gradient）表示函数在特定点变化最快的方向。

图 17（a）上来看，对于二元函数，如果梯度向量长度越大，说明该处越陡峭。如果梯度向量长度很小，说明该处越平缓。特别地如果梯度向量长度为 0，说明该处切面平行于水平面。


Bk\_2\_Ch22\_5.ipynb 绘制图 17（a）；此外，这段代码还绘制图 12。

- a 将 `plotly.figure_factory` 导入，简作 `ff`；这个模块提供了很多特殊的可视化方案。
- b 调用 `plotly.figure_factory.create_quiver()`，简作 `ff.create_quiver()`，绘制向量场，图像具有交互属性。
- c 设置图像宽度和高度。

```

a import plotly.figure_factory as ff
b fig = ff.create_quiver(xx1_, xx2_,
                        V[0], V[1],
                        scale=0.38,
                        arrow_scale=.28,
                        line_width=1)
c fig.update_layout(autosize=False,
                    width=500, height=500)
fig.show()

```

代码 5. 用 `plotly.figure_factory.create_quiver()` 绘制平面向量场 |  Bk\_2\_Ch22\_5.ipynb

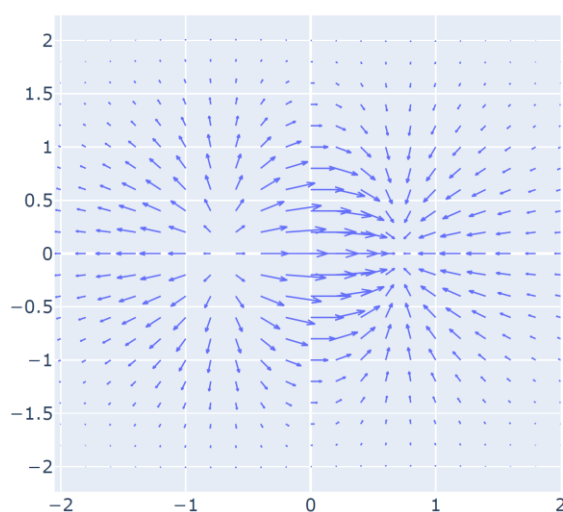


图 12. 用 `plotly.figure_factory.create_quiver()` 绘制平面向量场

图 17 (b) 所示为三元函数  $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$  的三维空间梯度向量场。

Bk\_2\_Ch22\_6.ipynb 绘制图 17 (b); 此外, Bk\_2\_Ch22\_6.ipynb 还绘制了图 13 这个向量场图, 图中用圆锥代表向量。

```

a import plotly.graph_objects as go
b fig = go.Figure(data = go.Cone(
    x=xxx1.ravel(),
    y=xxx2.ravel(),
    z=xxx3.ravel(),
    u=V[0].ravel(),
    v=V[1].ravel(),
    w=V[2].ravel(),
    colorscale='RdYlBu',
    sizemode="absolute",
    sizeref=18))
c fig.update_layout(autosize=False,
                    width=600, height=600)
fig.show()

```

代码 6. 用 `plotly.graph_objects.Cone()` 绘制三维向量场 | Bk\_2\_Ch22\_5.ipynb

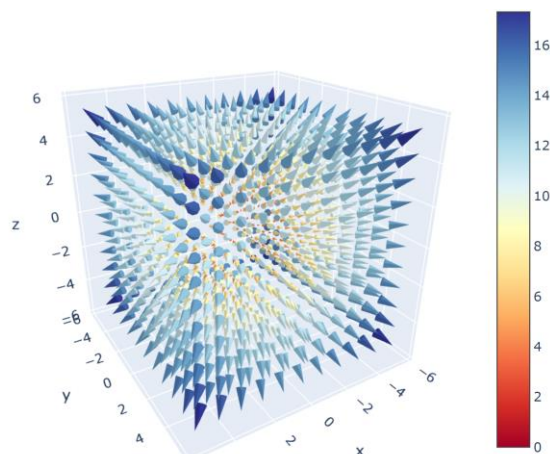


图 13. 用 `plotly.figure_factory.create_quiver()` 绘制平面向量场

## 水流图

图 18 所示为用 `matplotlib.pyplot.streamplot()` 绘制的水流图。这个函数的用法和 `quiver()` 颇为相似。Bk\_2\_Ch22\_7.ipynb 绘制图 18。

这段代码还介绍用 `plotly.figure_factory.create_streamline()` 绘制具有交互属性的水流图，请大家自行学习这段代码。

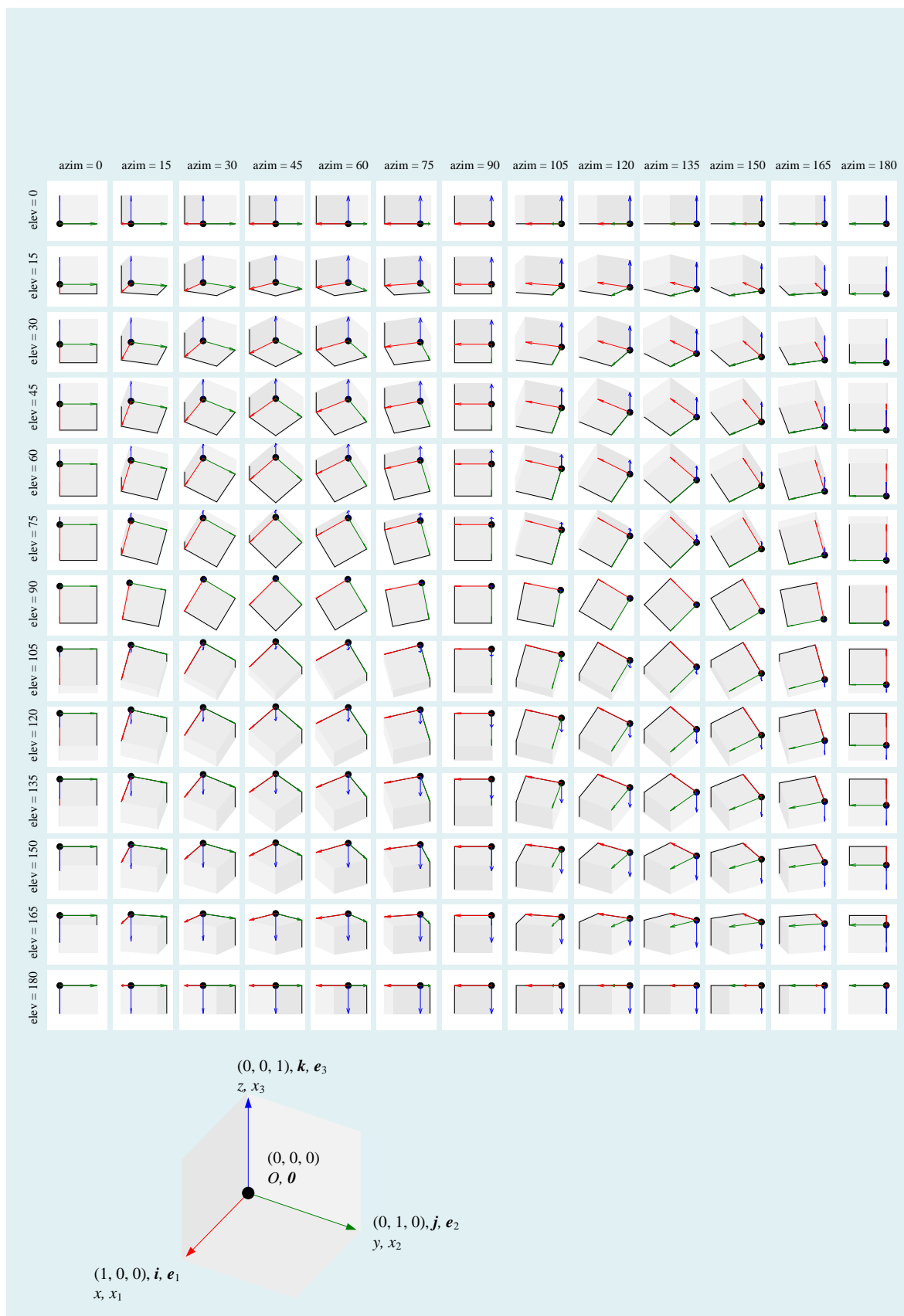


图 14. 不同视角下观察“红绿蓝”单位向量

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课程视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

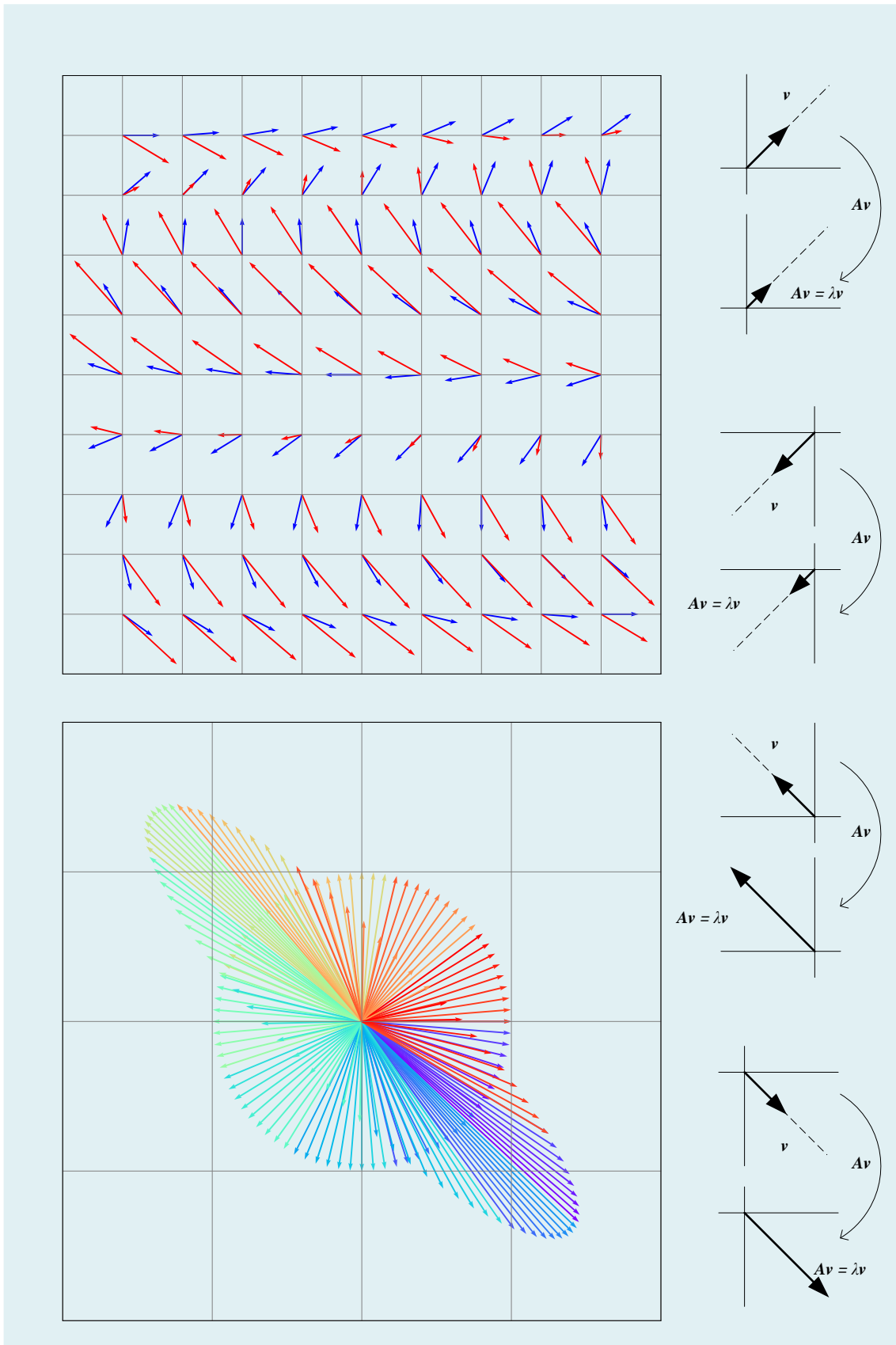


图 15. 可视化特征向量

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)



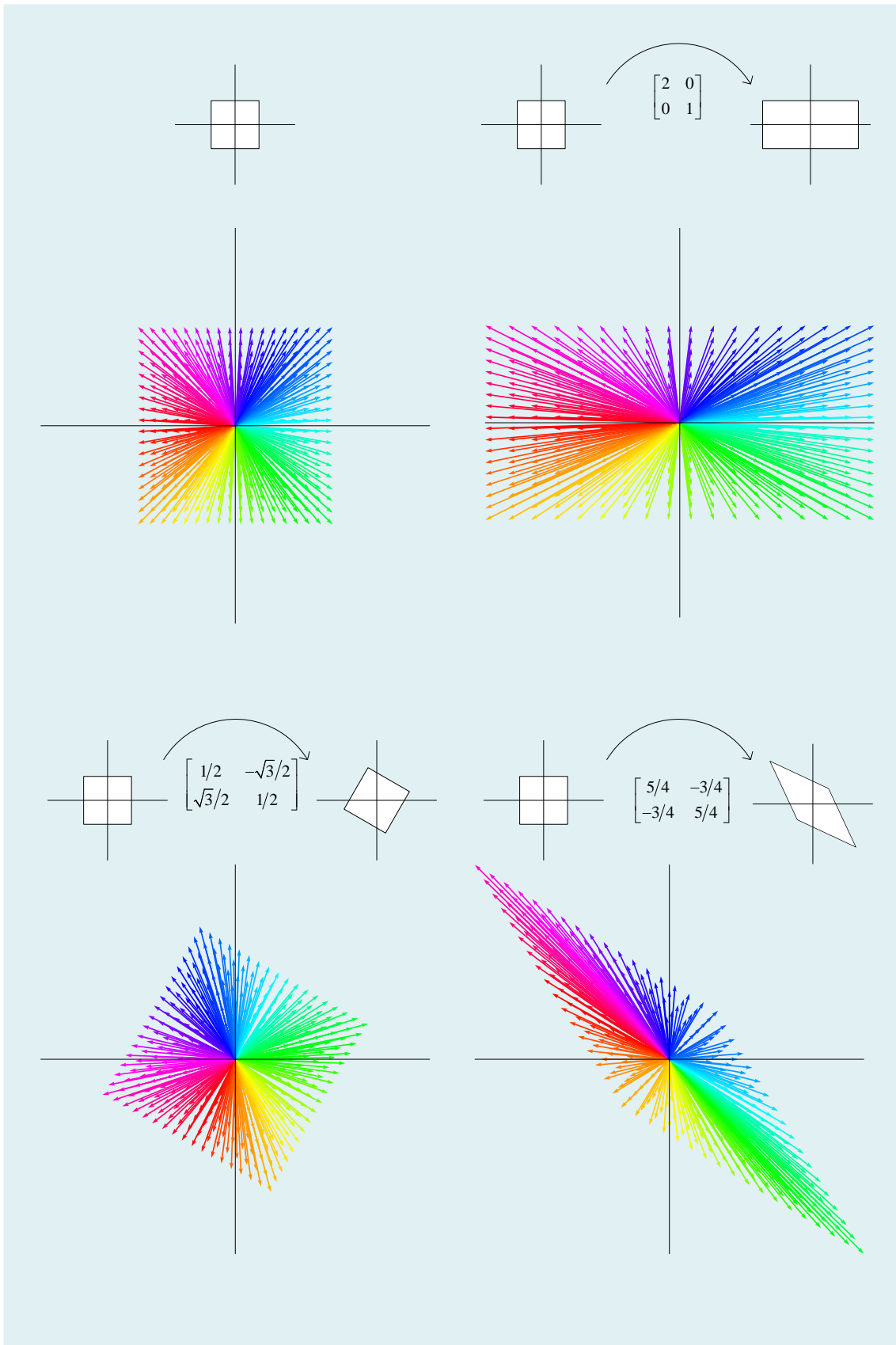


图 16. 可视化几何变换

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)



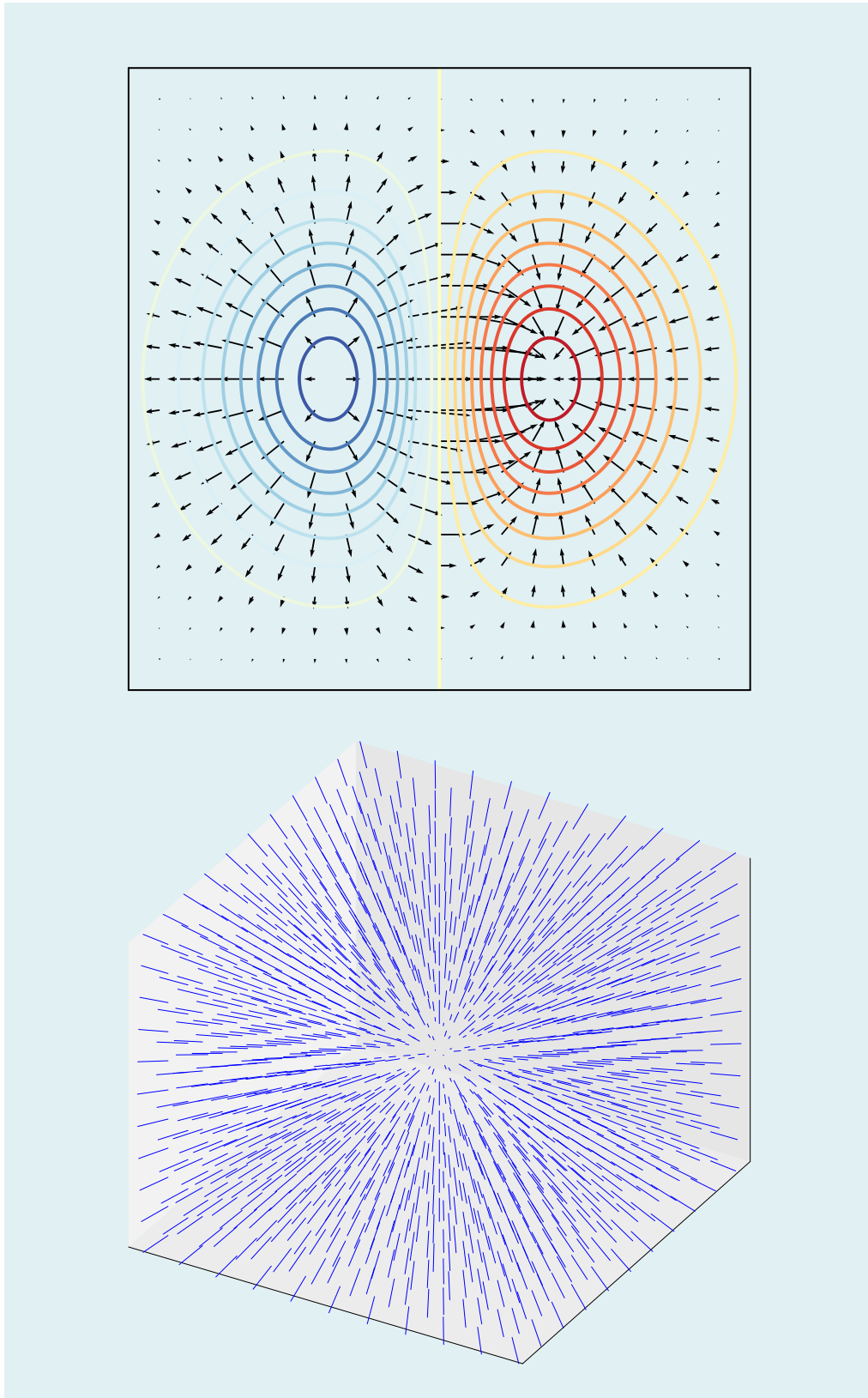


图 17. 可视化梯度向量场

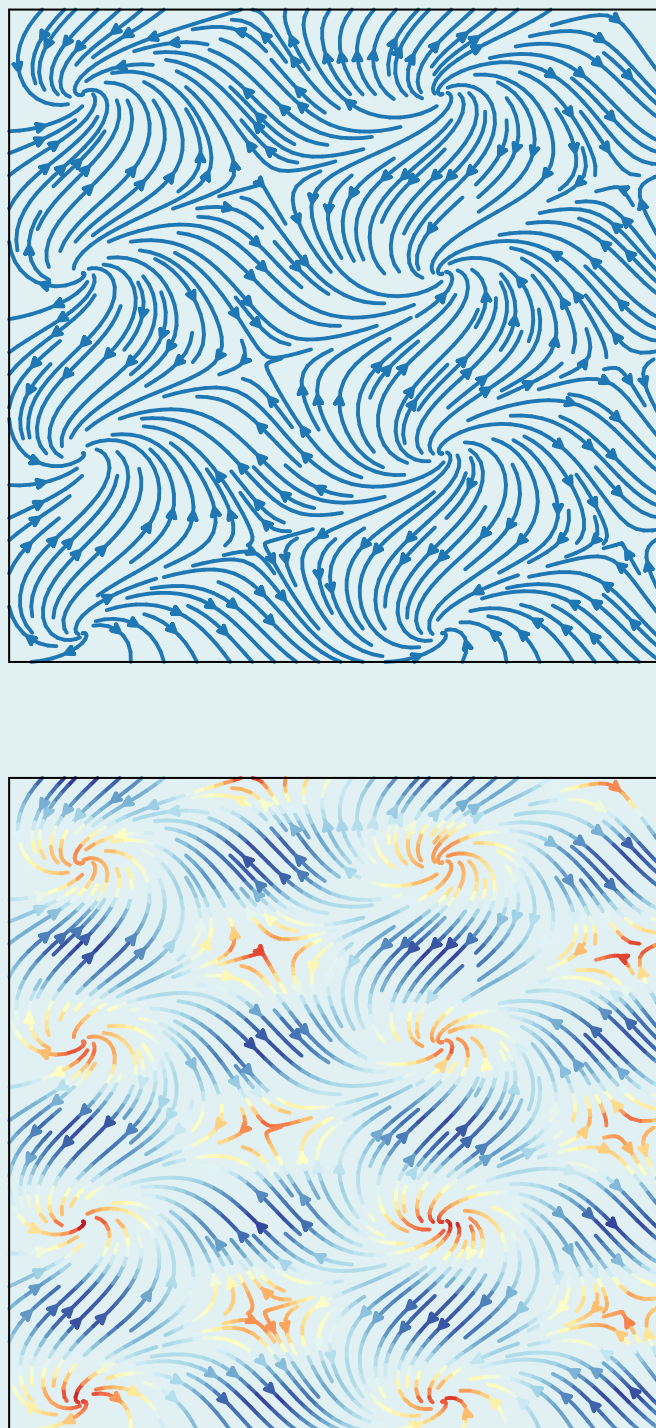


图 18. 水流图