

## 12

## Plane Geometry

## 平面几何图形

用 Matplotlib 绘制各种平面几何形状



艺术就是谎言，但艺术让我们看清真相。

*Art is the lie that enables us to realize the truth.*

—— 毕加索 (Pablo Picasso) | 西班牙艺术家 | 1881 ~ 1973



- ◀ matplotlib.patches.Arc() 绘制弧线
- ◀ matplotlib.patches.Arrow() 绘制箭头
- ◀ matplotlib.patches.Circle() 绘制正圆
- ◀ matplotlib.patches.Ellipse() 绘制椭圆
- ◀ matplotlib.patches.FancyBboxPatch() 绘制 Fancy 矩形框
- ◀ matplotlib.patches.Polygon() 绘制多边形
- ◀ matplotlib.patches.Rectangle() 绘制长方形
- ◀ matplotlib.patches.RegularPolygon() 绘制正多边形
- ◀ matplotlib.pyplot.cm 提供各种预定义色谱方案，比如 matplotlib.pyplot.cm.rainbow
- ◀ matplotlib.pyplot.contour() 绘制平面等高线
- ◀ matplotlib.pyplot.contourf() 绘制填充等高线图
- ◀ numpy.cos() 计算余弦值
- ◀ numpy.diag() 如果 A 为方阵，numpy.diag(A) 函数提取对角线元素，以向量形式输入结果；如果 a 为向量，numpy.diag(a) 函数将向量展开成方阵，方阵对角线元素为 a 向量元素
- ◀ 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  $\odot$
- ◀ numpy.linalg.inv() 矩阵求逆
- ◀ numpy.linalg.norm() 计算范数
- ◀ numpy.meshgrid() 创建网格化数据
- ◀ numpy.sin() 计算正弦值
- ◀ numpy.sqrt() 计算平方根
- ◀ matplotlib.patches.Rectangle() 是 Matplotlib 中的一个图形对象，用于绘制矩形形状
- ◀ matplotlib.pyplot.axhspan() 函数用于在水平方向创建一个跨越指定 y 值范围的色块
- ◀ matplotlib.pyplot.axvspan() 函数用于在垂直方向创建一个跨越指定 x 值范围的色块
- ◀ matplotlib.pyplot.fill() 函数用于绘制多边形，并在其中填充颜色，创建一个封闭区域的图形效果
- ◀ matplotlib.pyplot.fill\_between() 函数用于在两条曲线之间填充颜色，创建一个区域的图形效果
- ◀ matplotlib.pyplot.fill\_betweenx() 函数用于在两条垂直于 x 轴的水平线之间填充颜色，创建一个区域的图形效果
- ◀ numpy.linspace() 在指定的间隔内，返回固定步长的数据

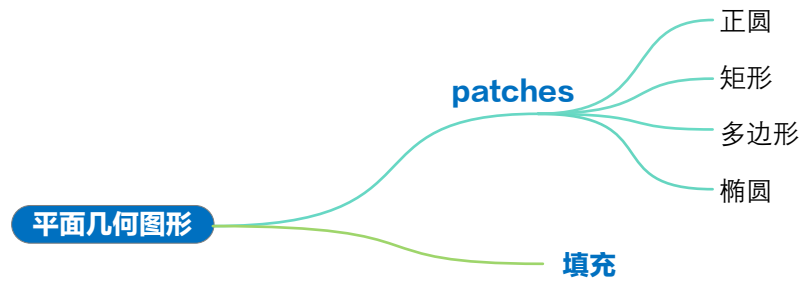
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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)



## 12.1 使用 patches 绘制平面几何形状

相信大家对 `matplotlib.patches` 已经不陌生了。`matplotlib.patches` 是 Matplotlib 库中的一个模块，可以使用它来绘制如图 1 圆形、矩形、多边形、箭头等等。

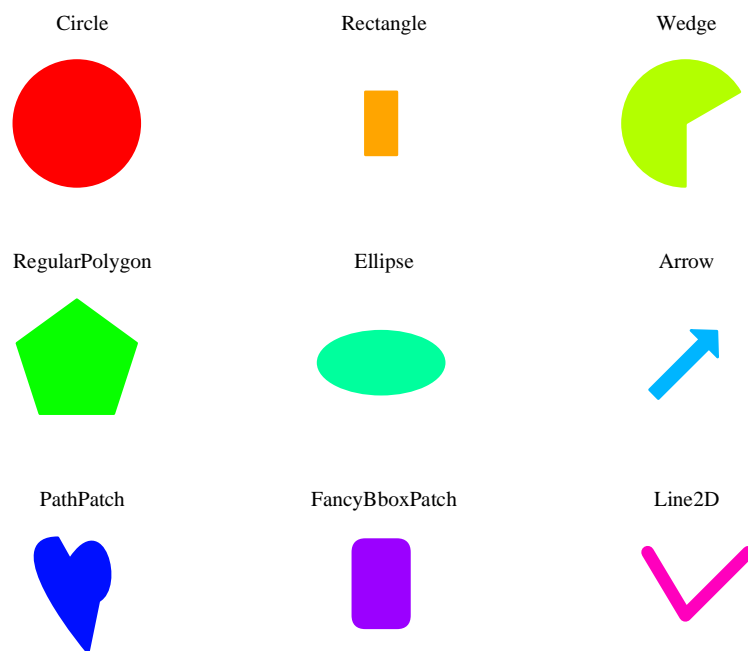


图 1. `matplotlib.patches` 中常见的几何图形 | [Bk\\_2\\_Ch12\\_01.ipynb](#)

图 1 参考 Matplotlib 官方示例，请大家自行学习 [Bk\\_2\\_Ch12\\_01.ipynb](#)。

图 2 所示为利用 `matplotlib.patches` 绘制一组单位圆内接正多边形 (inscribed regular polygon in a unit circle)、单位圆外切正多边形 (circumscribed regular polygons on a unit circle)。

举个例子，`patches.Circle` 可以创建一个圆形对象。这个对象可以具有不同的参数，如位置、大小、边框颜色、填充颜色、阴影线等。

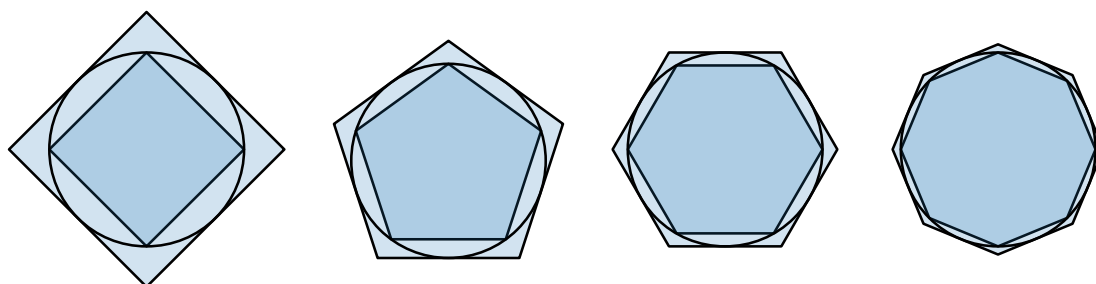


图 2. 利用 `patches` 绘制正圆，以及外切、内接正多边形 | [Bk2\\_Ch24\\_02.ipynb](#)



《数学要素》第3章中，大家会看到我们利用图2介绍如何估算圆周率。

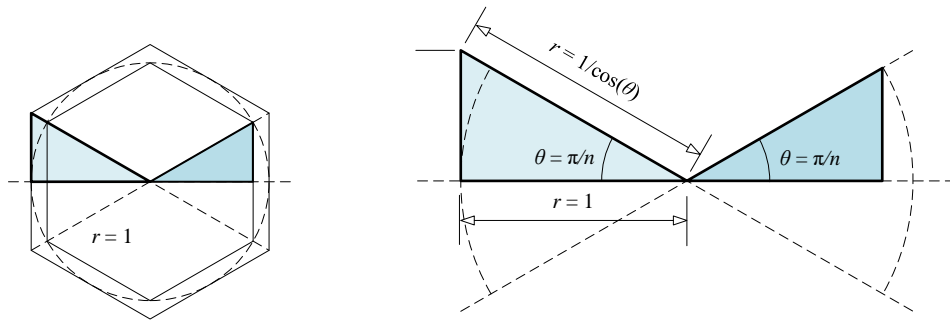


图 3. 圆形内接和外接多边形

Bk\_2\_Ch12\_02.ipynb 绘制图 2，下面聊聊这段代码。

**a** 从 `matplotlib.patches` 导入 `RegularPolygon` 和 `Circle`。`RegularPolygon` 用来绘制正多边形 (regular polygon)，`Circle` 用来绘制正圆形。

**b** 用 `matplotlib.pyplot.subplots()`，简作 `plt.subplots()`，创建一个包含 1 行、4 列的子图布局。返回一个图像对象 `fig` 和一个包含各个子图轴对象的数组 `axs`。

**c** 在 `for` 循环中使用 `zip` 函数，将两个可迭代对象 `[4, 5, 6, 8]` 和 `axs.ravel()` 组合在一起，然后通过 `for` 循环对它们进行迭代。

`[4, 5, 6, 8]` 元素代表正多边形的顶点数 (number of vertices)。

`axs.ravel()` 将之前创建的包含 `Axes` 对象的数组展平成一个一维数组。

在每次 `for` 循环遍历中，`num_vertices` 获取了 `[4, 5, 6, 8]` 中的当前值，而 `ax` 获取了 `axs.ravel()` 中的对应元素，即当前子图轴对象。

**d** 用 `RegularPolygon` 创建正多边形对象实例。

`(0,0)` 指定了多边形的中心坐标。

`numVertices=num_vertices` 指定了多边形的顶点数。

`radius=1` 设置了多边形的外切正多边形的圆半径。图 3 展示其背后的数学原理。

`alpha=0.2` 设置了多边形的透明度。

`edgecolor='k'` 设置多边形的边框颜色为黑色。

**e** 用 `add_patch()` 方法用于将 `hexagon_outer` 图形对象添加到子图中。

`add_patch` 是 `matplotlib` 库中 `Axes` 对象的一个方法，用于向一个子图中添加一个图形元素。这个方法可以添加多种不同类型的图形元素，例如矩形、多边形、圆形、椭圆、箭头等等。在使用 `add_patch` 方法前，需要先创建一个对应的图形元素对象，例如 `Circle`、`Rectangle`、`Polygon`、`Ellipse`、`Arrow` 等。然后，可以使用 `add_patch` 方法将这个对象添加到指定的子图中。在添加完成后，可以使用 `set_*` (比如 `set_edgecolor`、`set_linewidth`) 方法或属性来设置图形元素的属性，例如填充颜色、边框颜色、边框宽度等。

**f** 也用 `RegularPolygon` 创建正多边形实例。

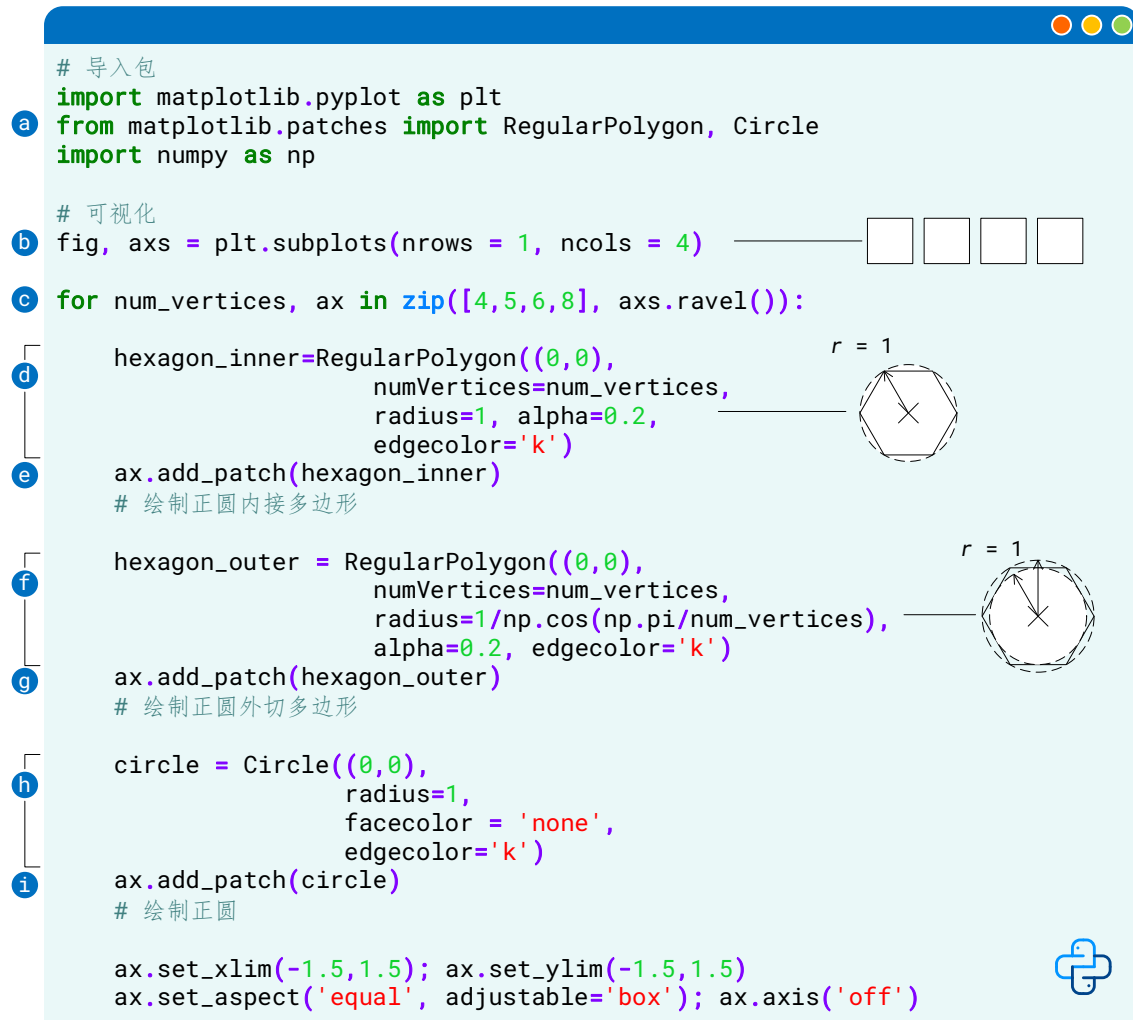
`radius=1/np.cos(np.pi/num_vertices)` 设置了多边形的外切正多边形的圆半径。图 3 展示其背后的数学原理。

**g** 也用 `add_patch()` 方法添加图形对象。

**h** 用 `Circle` 创建圆对象实例。

`(0,0)` 指定了圆的中心坐标。`radius=1` 设置了圆的半径为 1。`facecolor='none'` 设置了圆的填充颜色为透明。`edgecolor='k'` 设置了圆的边框颜色为黑色。

**i** 用 `add_patch()` 方法添加圆形对象。



代码 1. 利用 patches 绘制正圆，以及外切、内接正多边形 | Bk\_2\_Ch12\_02.ipynb

图 11 ~ 图 14 所示为利用 patches 中各种图形创作的生成艺术，相关代码留给大家自行学习。

## 12.2 填充

### 沿横轴填充

`fill_between()` 是 `matplotlib.pyplot` 库中的一个函数，用于绘制两个曲线之间的填充区域。

`fill_between(x, y1, y2, ...)` 函数可以接受两个数组 `x` 和 `y1`，以及另外一个数组 `y2`，它们都是相同长度的。这个函数会将 `y1` 和 `y2` 之间的区域填充，并在 `x` 上绘制。

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 4 所示三个例子展示曲线和水平线之间沿横轴填充。`fill_between()` 可以使用参数 `where` 来指定  $y_1$  和  $y_2$  之间的填充区域，可以使用参数 `facecolor` 来指定填充颜色，使用参数 `alpha` 来指定填充区域的透明度。

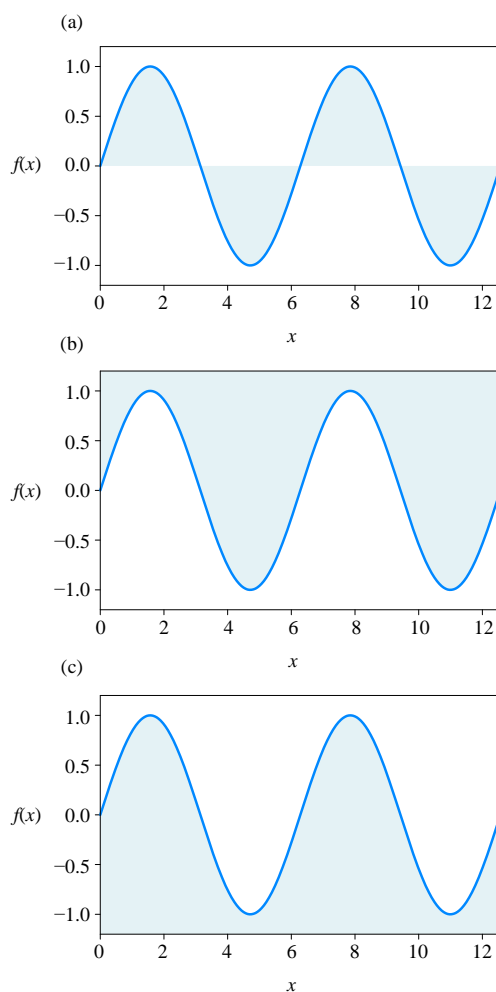



图 4. 曲线和水平线之间沿横轴填充 |  Bk\_2\_Ch12\_07.ipynb

如图 5 所示，通过设置条件，我们还可以给满足不同条件的区域填充不同颜色。图 6 还给出两个例子，展示两条曲线之间沿横轴填充。

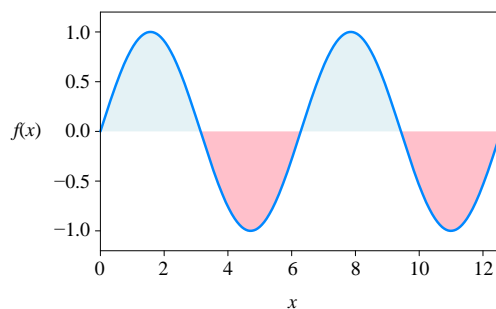


图 5. 曲线和水平线之间沿横轴填充，不同颜色 |  Bk\_2\_Ch12\_07.ipynb

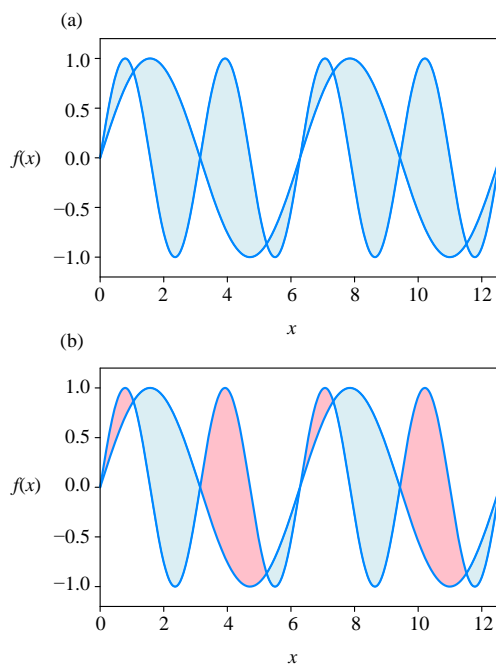

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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. 两条曲线之间沿横轴填充 |  Bk\_2\_Ch12\_07.ipynb

Jupyter 笔记 Bk\_2\_Ch12\_07.ipynb 绘制图 4、图 5、图 6。

### 沿纵轴填充

`fill_betweenx` 则可以用来绘制两个曲线在  $y$  轴方向之间的填充区域。`fill_betweenx(y, x1, x2, ...)` 函数接受两个数组  $y$  和  $x1$ , 以及另外一个数组  $x2$ , 它们都是相同长度的。这个函数会将  $x1$  和  $x2$  之间的区域填充, 并在  $y$  上绘制。图 7 所示为沿纵轴方向填充的 6 个例子。

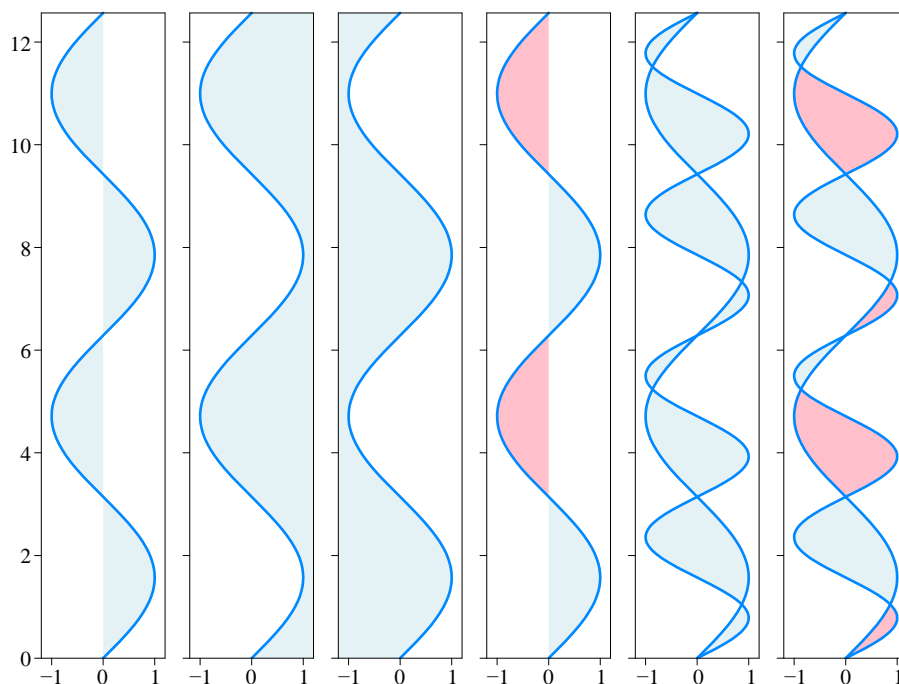


图 7. 沿纵轴方向填充 | Bk\_2\_Ch12\_08.ipynb

### 填充阴影线

Matplotlib 中我们还可以用 `hatch` 给各种填充增加阴影线。`hatch` 是 `matplotlib` 库中的一个属性，用于给某些图形元素添加填充图案。要使用 `hatch` 属性，需要将它设置为一个字符串，该字符串描述了所需的填充图案类型。`matplotlib` 库中提供了多种不同的填充图案类型，下面给出几个例子：

```
'/'：斜杠填充
'\'：反斜杠填充
'.'：点状填充
'o'：圆形填充
'-'：横向线性填充
'+'：十字线填充
'x'：斜十字线填充
'|'：纵向线性填充
```



BK\_2\_Ch12\_09.ipynb 给出几种常见的阴影线，请大家自行学习。也请大家在前两个 Jupyter 笔记使用不同的填充阴影线。

### 参考填充色块

本书前文介绍过如何绘制水平、竖直参考线，类似地，我们也可以绘制参考填充色块。`axhspan` 是 `matplotlib` 库中的一个函数，用于在一个子图中绘制一个水平的矩形。



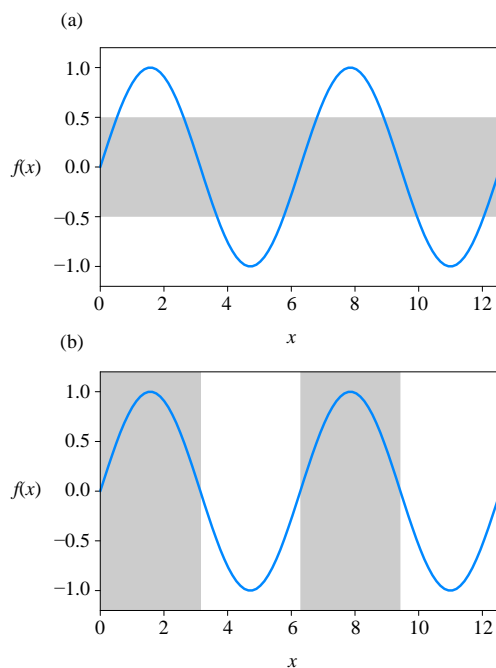


图 8. 参考填充色块 | [Bk\\_2\\_Ch12\\_10.ipynb](#)

这个函数通常用于强调某个区域的范围或表示一个特定的数据区间，如图 8 (a) 所示。

`axhspan` 函数接受四个参数：`ymin`、`ymax`、`xmin` 和 `xmax`。其中，`ymin` 和 `ymax` 表示矩形的纵向范围，`xmin` 和 `xmax` 表示矩形的横向范围。类似地，`axvspan` 可以绘制竖直方向参考填充色块，如图 8 (b) 所示。

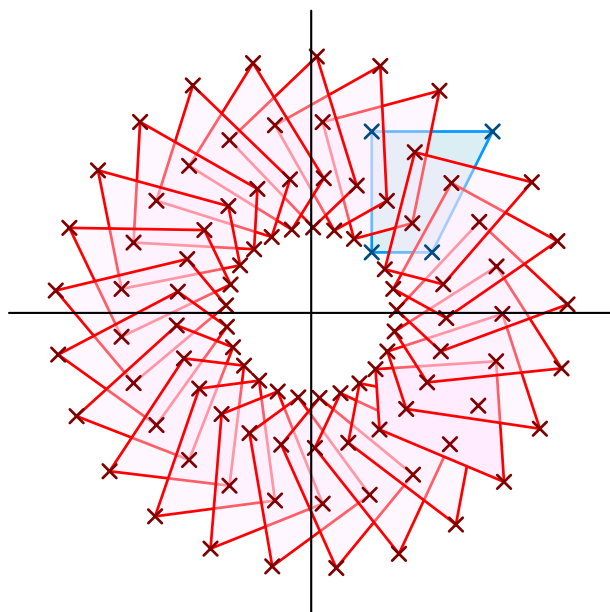

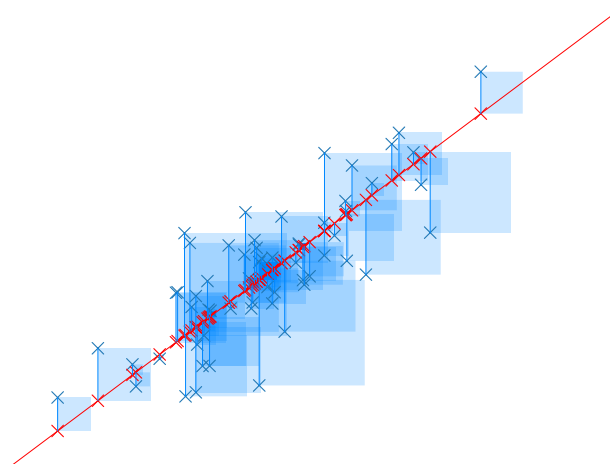
## 用 fill 填充

`fill` 是 `matplotlib` 库中常用的一个函数，用于绘制填充区域。使用 `fill` 函数可以将一个多边形区域填充成指定的颜色。

`fill` 函数接受两个参数：`x` 坐标数组和 `y` 坐标数组，用于指定要填充的多边形区域的顶点坐标。`x` 和 `y` 的长度必须相同，且每个元素都对应一个多边形的顶点。图 9 给出的例子还用到了旋转，《矩阵力量》会介绍如何利用线性代数工具完成旋转操作。

图 10 所示为利用正方形可视化最小二乘回归原理。

图 15 所示利用网格 + 填充平面几何形状展示平面仿射变换。这背后的数学工具会在鸢尾花书《矩阵力量》揭示。

图 9. 用 fill 首尾连接封闭填充 |  Bk\_2\_Ch12\_11.ipynb图 10. 添加几何元素 |  Bk\_2\_Ch12\_12.ipynb

想要理解如何用 `patches` 绘制各种几何图形，大家可以参考如下链接。

[https://matplotlib.org/stable/api/patches\\_api.html](https://matplotlib.org/stable/api/patches_api.html)



本章介绍了绘制平面几何图形的几种方法。请大家格外注意几种常见仿射变换。此外，本书后续还会用到平面等高线方法呈现平面几何图形，请大家注意对比学习。

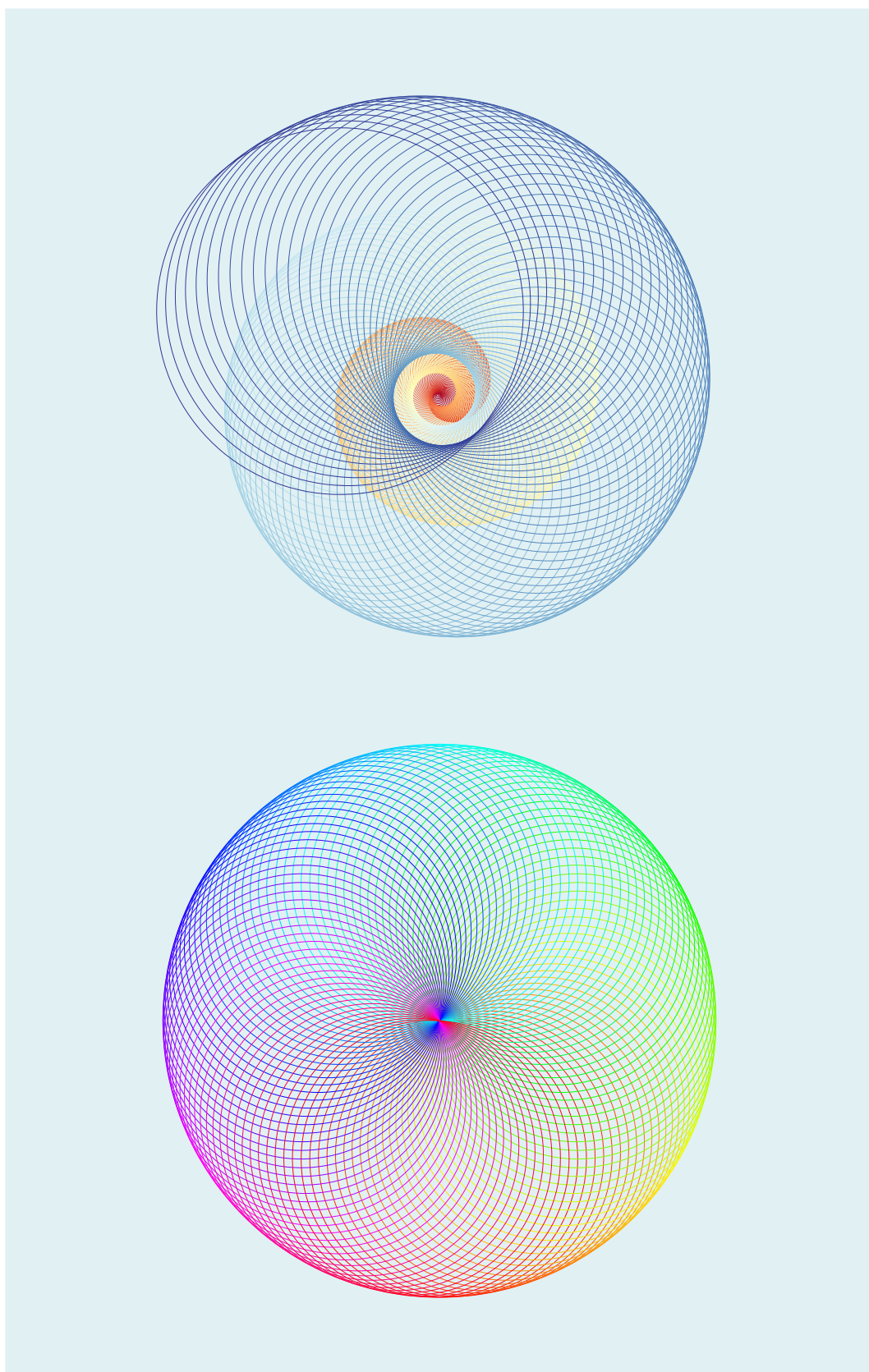


图 11. 两组旋转正圆 |  Bk\_2\_Ch12\_03.ipynb

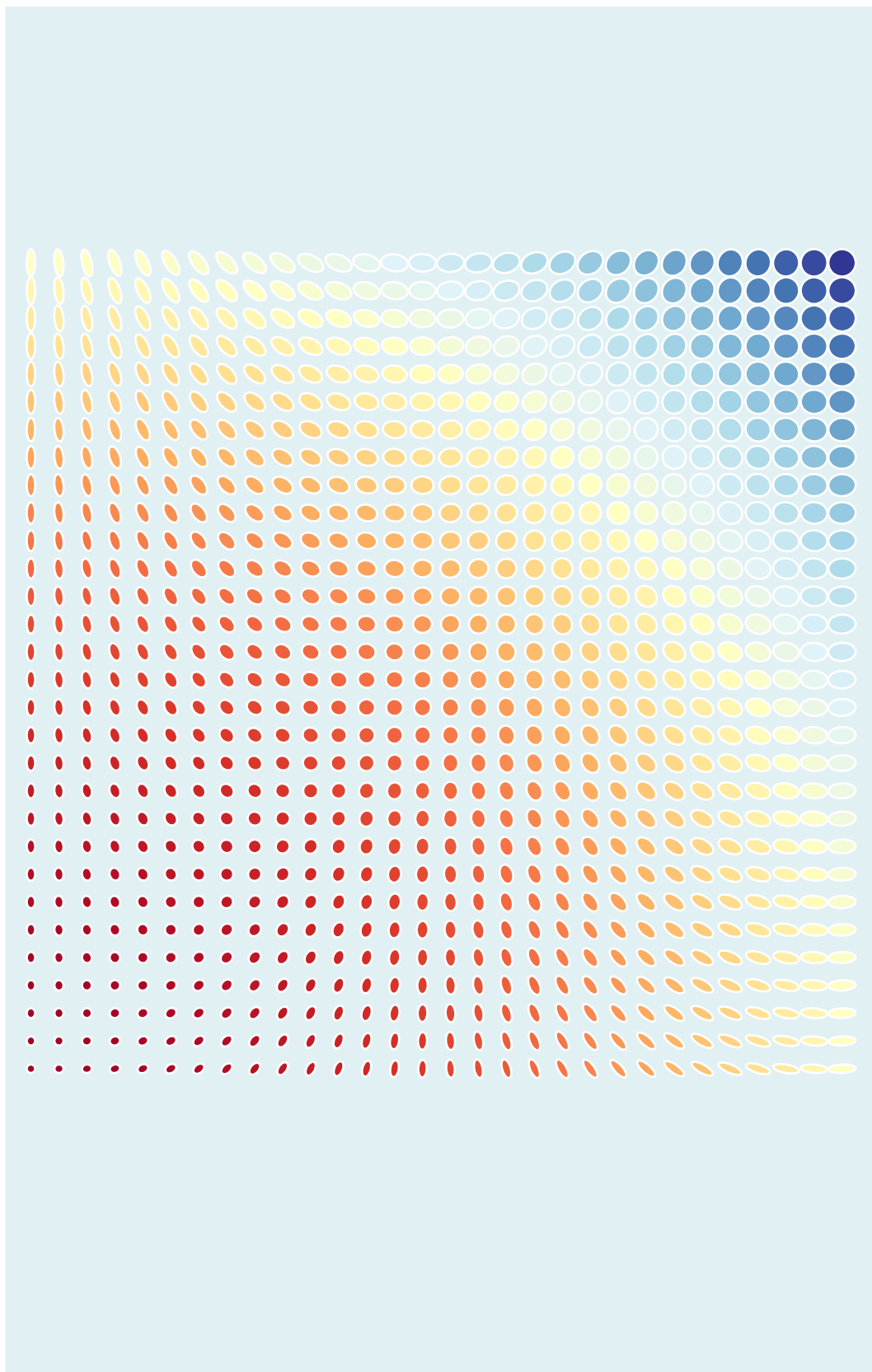


图 12. 一组旋转椭圆 |  Bk\_2\_Ch12\_04.ipynb

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)

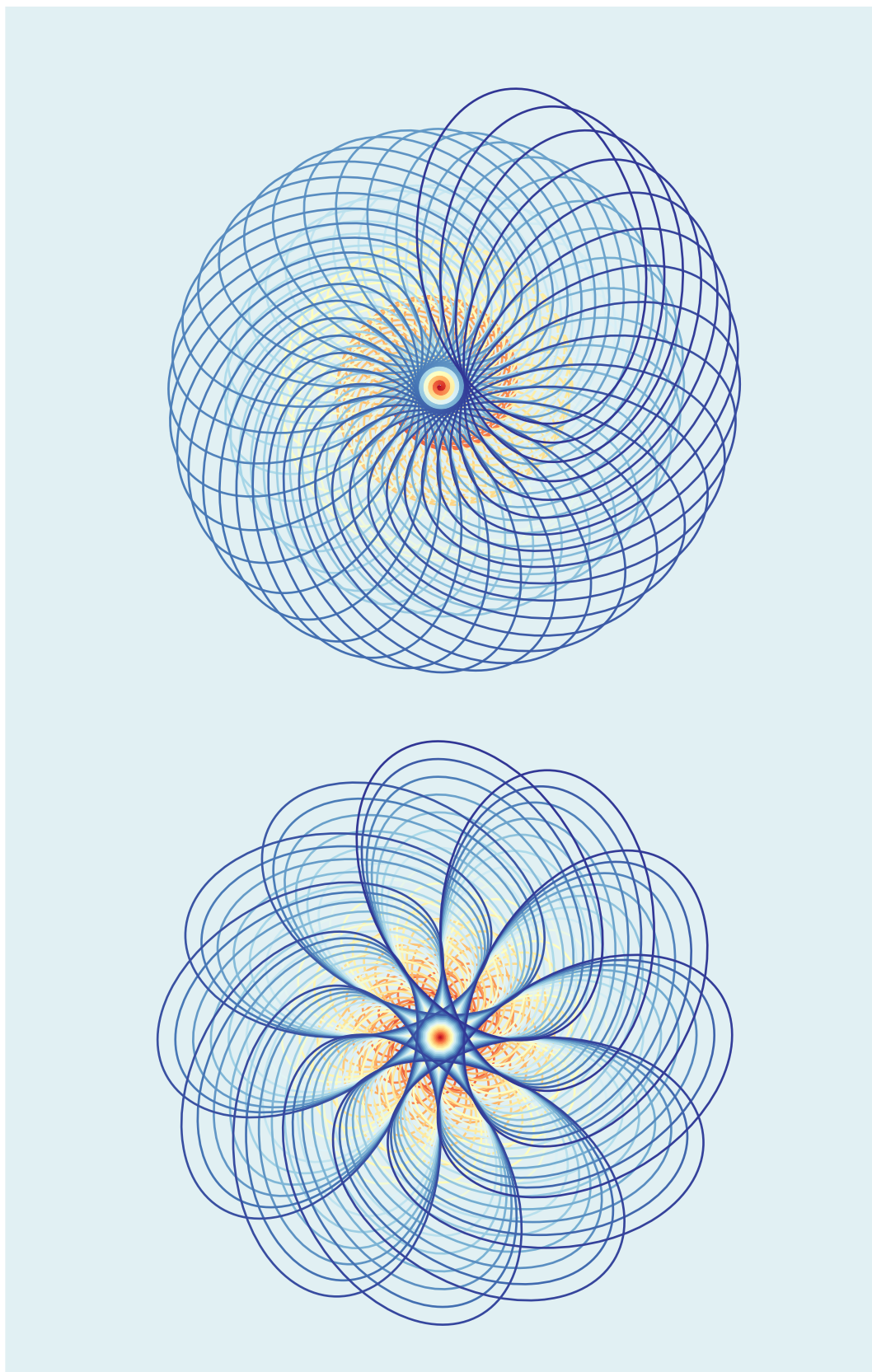


图 13. 两组旋转椭圆 |  Bk\_2\_Ch12\_05.ipynb



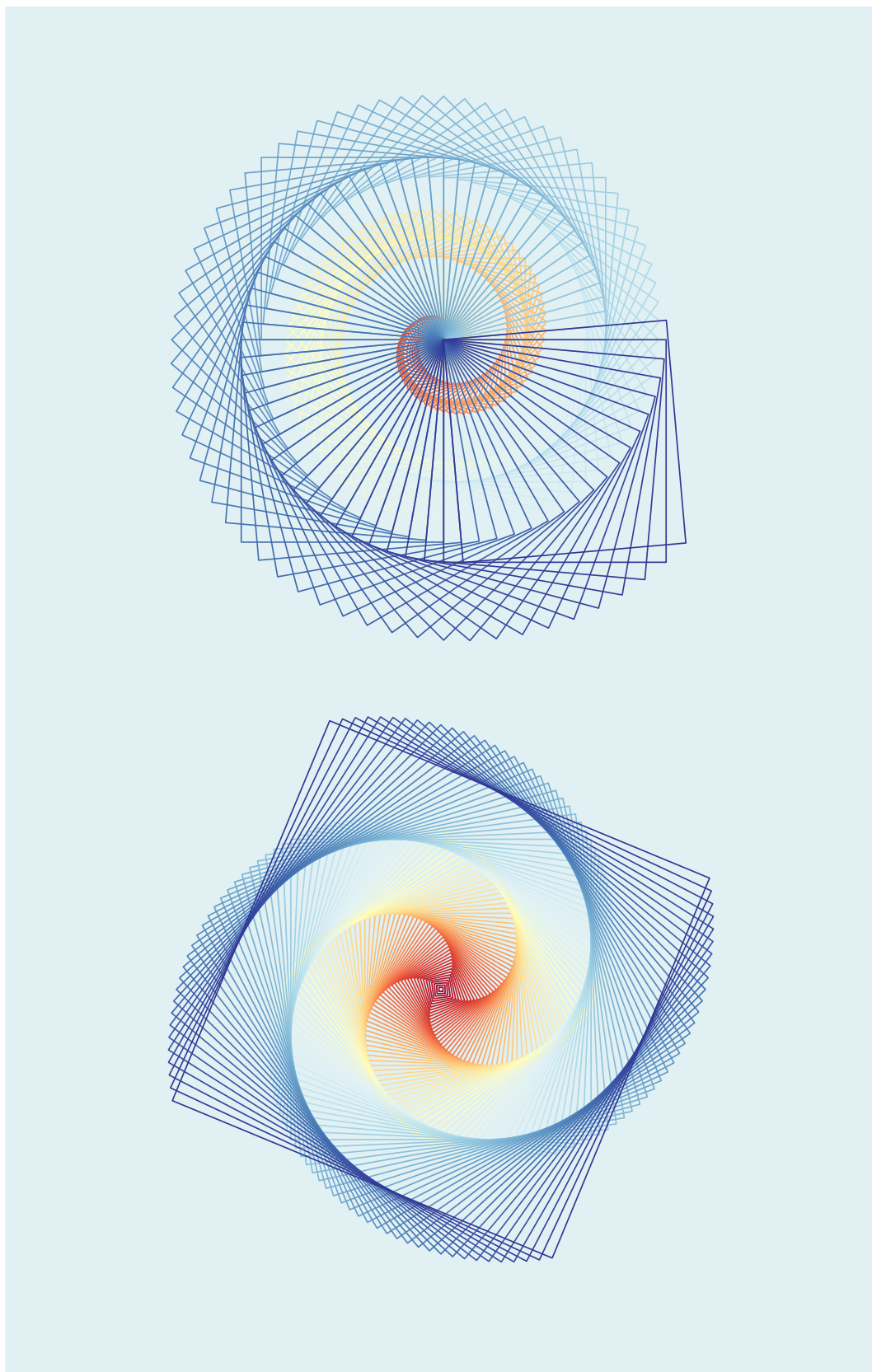
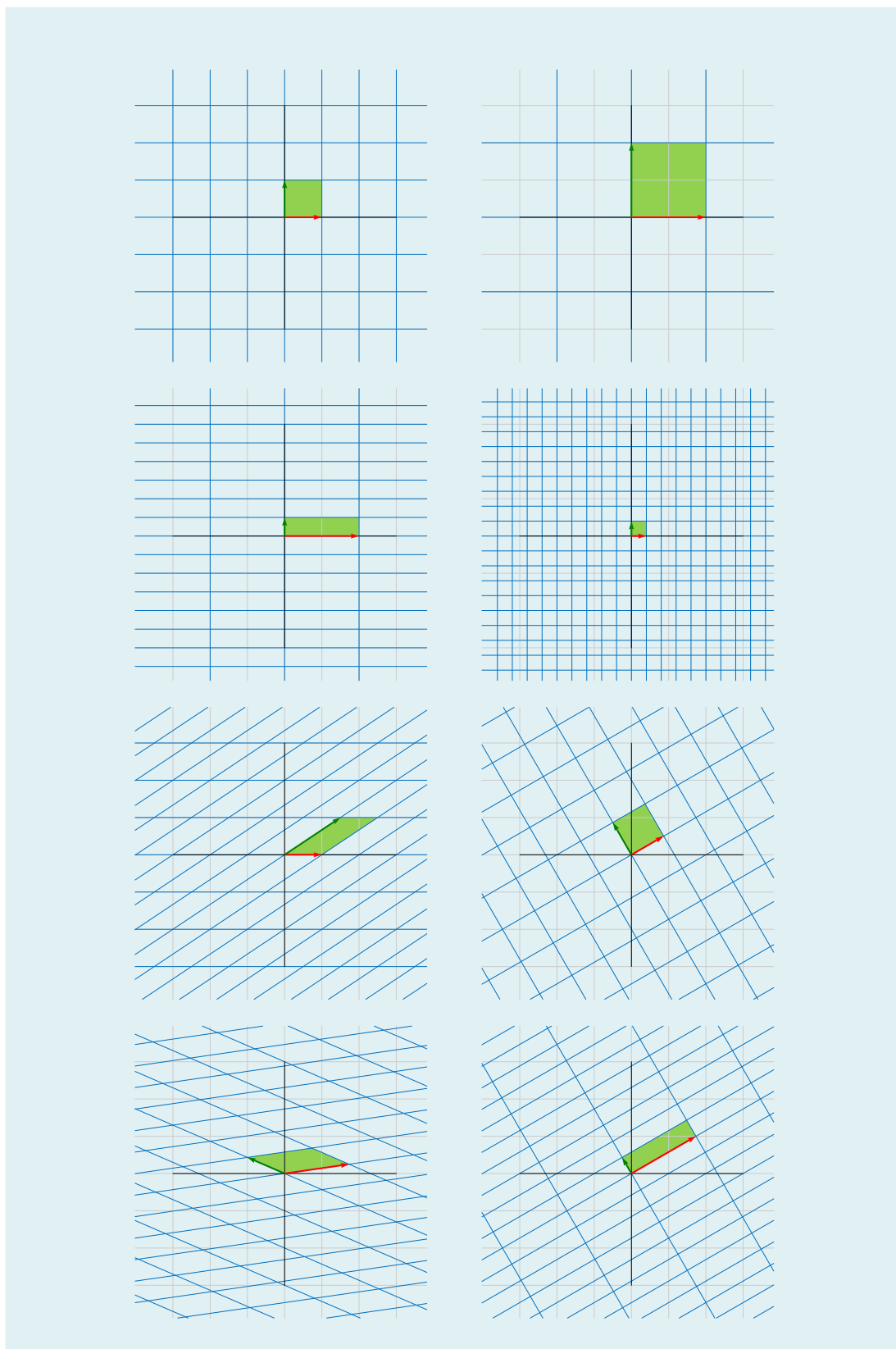


图 14. 两组旋转正方形 |  Bk\_2\_Ch12\_06.ipynb

图 15. 几何变换 |  Bk\_2\_Ch12\_13.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)