

4

Decorations

装饰

图脊、图轴、网格、线条、标注…个个都是艺术家



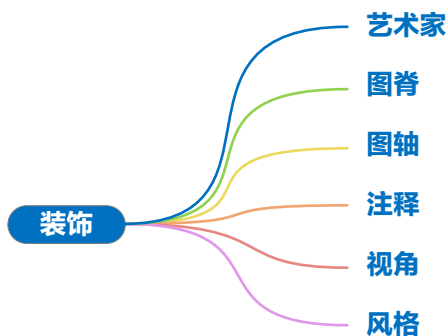
这个世界就是我们想象力的画布。

This world is but a canvas to our imagination.

—— 亨利·戴维·梭罗 (Henry David Thoreau) | 作家、诗人 | 1817 ~ 1862



- ◀ matplotlib.gridspec.GridSpec() 创建一个规则的子图网格布局
- ◀ matplotlib.pyplot.grid() 在当前图表中添加网格线
- ◀ matplotlib.pyplot.plot() 绘制折线图
- ◀ matplotlib.pyplot.subplot() 用于在一个图表中创建一个子图，并指定子图的位置或排列方式
- ◀ matplotlib.pyplot.subplots() 创建一个包含多个子图的图表，返回一个包含图表对象和子图对象的元组
- ◀ matplotlib.pyplot.title() 设置当前图表的标题，等价于 ax.set_title()
- ◀ matplotlib.pyplot.xlabel() 设置当前图表 x 轴的标签，等价于 ax.set_xlabel()
- ◀ matplotlib.pyplot.xlim() 设置当前图表 x 轴显示范围，等价于 ax.set_xlim()
- ◀ matplotlib.pyplot.xticks() 设置当前图表 x 轴刻度位置，等价于 ax.set_xticks()
- ◀ matplotlib.pyplot.ylabel() 设置当前图表 y 轴的标签，等价于 ax.set_ylabel()
- ◀ matplotlib.pyplot.ylim() 设置当前图表 y 轴显示范围，等价于 ax.set_ylim()
- ◀ matplotlib.pyplot.yticks() 设置当前图表 y 轴刻度位置，等价于 ax.set_yticks()
- ◀ numpy.arange() 创建一个具有指定范围、间隔和数据类型的等间隔数组
- ◀ numpy.exp() 计算给定数组中每个元素的 e 的指数值
- ◀ numpy.linspace() 用于在指定的范围内创建等间隔的一维数组，可以指定数组的长度
- ◀ numpy.sin() 用于计算给定弧度数组中每个元素的正弦值



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

4.1 艺术家

在 Matplotlib 中，**艺术家** (artist) 是指图形的每个可见（甚至没那么明显的）元素，如图脊、图轴、坐标轴、标题、标签、图例、线条、网格、色块等等。每个艺术家对象都有自己的默认属性和方法，用于控制其外观和行为。

如图 1 所示，图形艺术家构成了一个层级结构。艺术家是**图形对象** (figure object)，它包含了所有其他艺术家。图形外框是**图脊** (frame)，图形对象之下是**图轴** (axis)，用于绘制数据和刻度线。图轴各种其他艺术家，如刻度线、刻度标签和轴标题。图 4 所示为一幅平面等高线中艺术家的层级结构。

绘图时，可以用 `zorder` 控制艺术家在层级结构中的放置顺序。

本章首先介绍图脊、图轴、注释这三个艺术家。

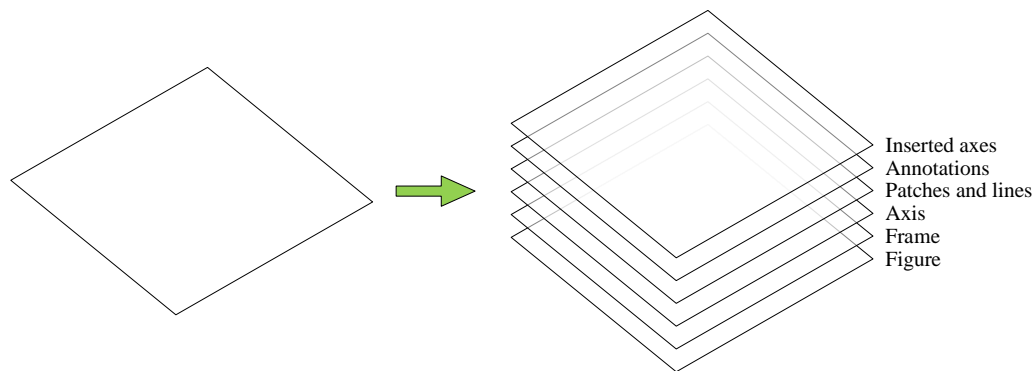


图 1. 一张图片的层级结构

4.2 图脊

在 Matplotlib 中，**图脊** (spine) 是指图形中的边框线，用于界定图形的边界。图脊由四条边框线组成：**上脊** (top spine)、**下脊** (bottom spine)、**左脊** (left spine) 和**右脊** (right spine)。这些脊线可以通过 Matplotlib 的 `Axes.spines` 属性进行访问和定制。

表 1 所示为一些修改图脊设计方案，本章不展开讲解表中这些设置，也不要求大家死记硬背。用到的时候，再来自行学习 `Bk_2_Ch04_01.ipynb`。

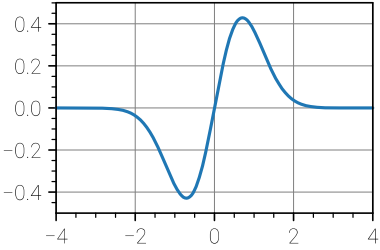
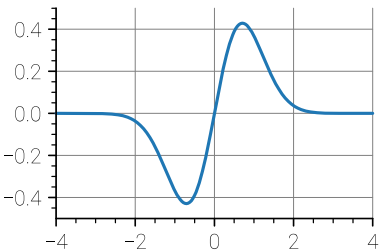
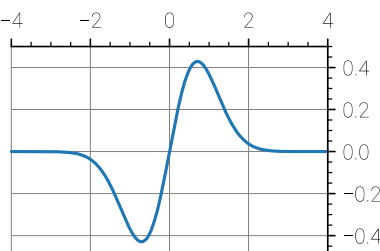
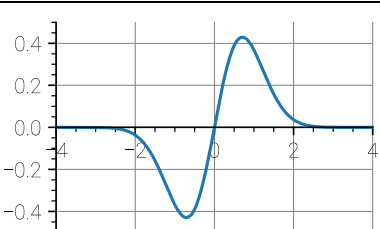
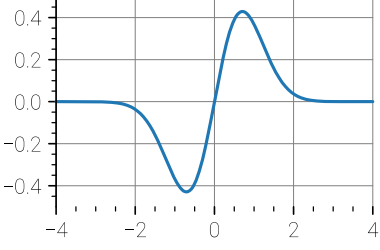
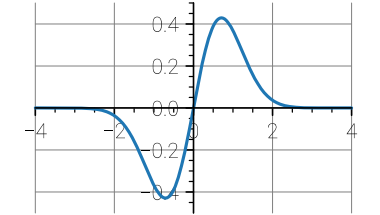
此外，`Bk_2_Ch04_02.ipynb` 还给出几种背景网格样式设计方案，如表 2 所示。

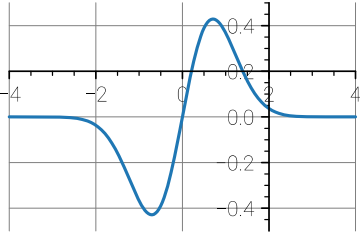
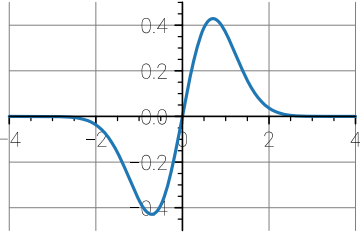
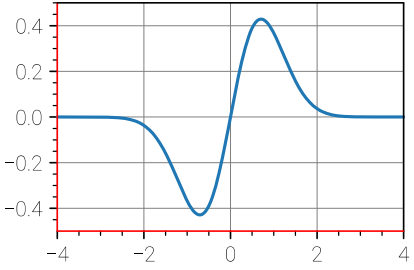
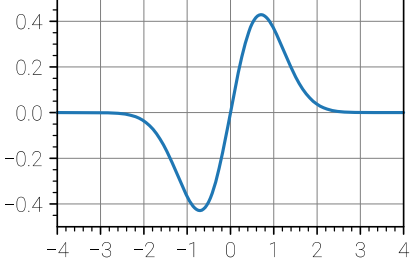
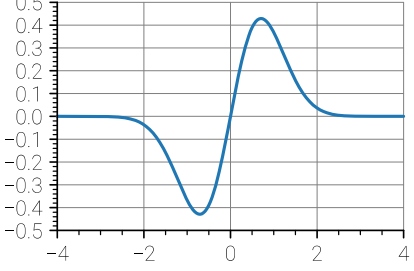
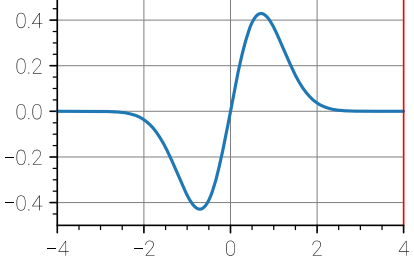
建议大家对表 1 和表 2 代码逐行注释。

表 1. 图脊设计 | [Bk_2_Ch04_01.ipynb](#)

图脊设计	代码
------	----

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

	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5);</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['top'].set_color('none') ax.spines['right'].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines[['bottom', 'left']].set_visible(False) # 也可以采用: # ax.spines['bottom'].set_color('none') # ax.spines['left'].set_color('none') ax.yaxis.set_ticks_position('right') ax.xaxis.set_ticks_position('top')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_position(('data',0)) ax.spines['right'].set_color('none') ax.spines['top'].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_color('none') ax.spines['right'].set_color('none') ax.spines['top'].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_position(('data',0)) ax.spines['left'].set_position(('data',0)) ax.spines['right'].set_color('none') ax.spines['top'].set_color('none')</pre>

	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_position(('data',0.2)) ax.spines['left'].set_position(('data',2)) ax.spines['right'].set_color('none') ax.spines['top'].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_position(('axes',0.5)) # 取值范围为 [0, 1], 0.5 代表中间 ax.spines['left'].set_position(('axes',0.5)) ax.spines['right'].set_color('none') ax.spines['top'].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_color('r') ax.spines['left'].set_color('r')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.set_xticks(np.arange(-4,5))</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['right'].set(alpha = 0.2) ax.spines['top'].set(alpha = 0.2) ax.set_yticks(np.arange(-0.5,0.6,0.1))</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['right'].set(edgecolor = 'r') ax.spines['top'].set(edgecolor = 'r')</pre>

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

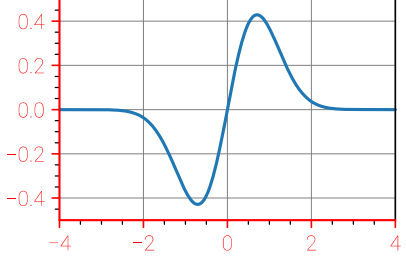
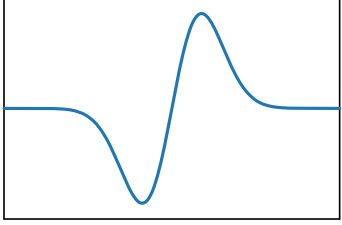
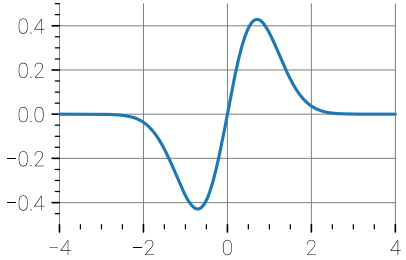
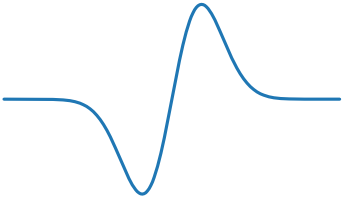
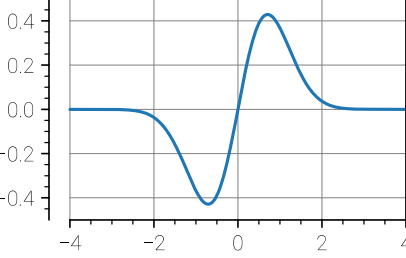
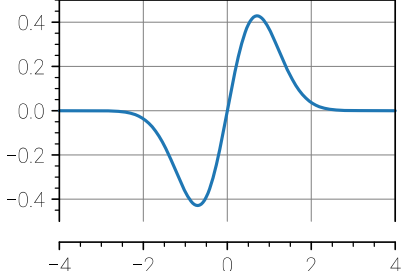

	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set(edgecolor = 'r', linestyle = '--', linewidth = 1) ax.spines['left'].set(edgecolor = 'r', linestyle = '--', linewidth = 1) ax.tick_params(axis='x', colors='red') ax.tick_params(axis='y', colors='red')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.xaxis.set_ticks([]) ax.yaxis.set_ticks([])</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines[:].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines[:].set_color('none')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['left'].set_position(('outward', 10))</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.spines['bottom'].set_position(('outward', 10)) # 三个选择 'outward', 'axes', 'data'</pre>

表 2. 网格设计 |  Bk_2_Ch04_02.ipynb

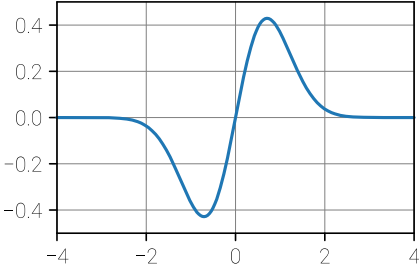
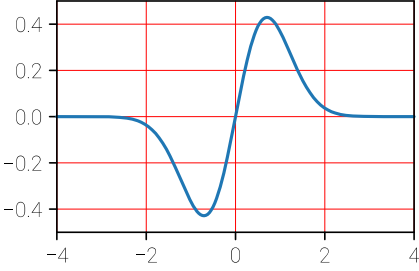
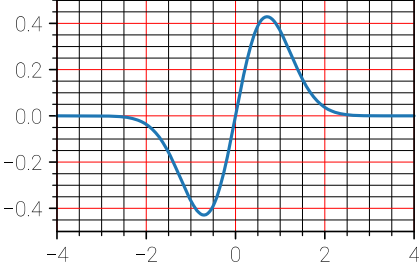
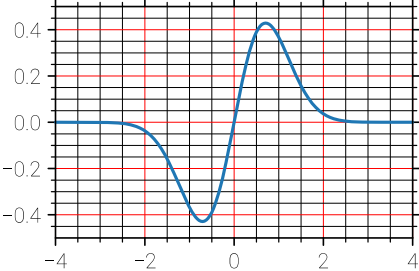
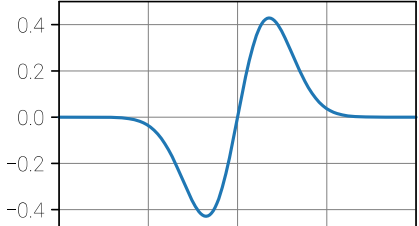
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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

图脊设计	代码
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.grid(True)</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.grid(linestyle='-', linewidth='0.5', color='red')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.set_axisbelow(True) ax.minorticks_on() ax.grid(which='major', linestyle='-', linewidth='0.5', color='red') ax.grid(which='minor', linestyle=':', linewidth='0.5', color='black')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.set_axisbelow(True) ax.minorticks_on() ax.grid(which='major', linestyle='-', linewidth='0.5', color='red') ax.grid(which='minor', linestyle=':', linewidth='0.5', color='black') ax.tick_params(which='both', top='off', left='off', right='off', bottom='off')</pre>
	<pre>ax.set_xlim(-4,4); ax.set_ylim(-0.5, 0.5); ax.grid(True) for tick in ax.xaxis.get_major_ticks(): tick.tick1line.set_visible(False) tick.tick2line.set_visible(False) tick.label1.set_visible(False) tick.label2.set_visible(False)</pre>

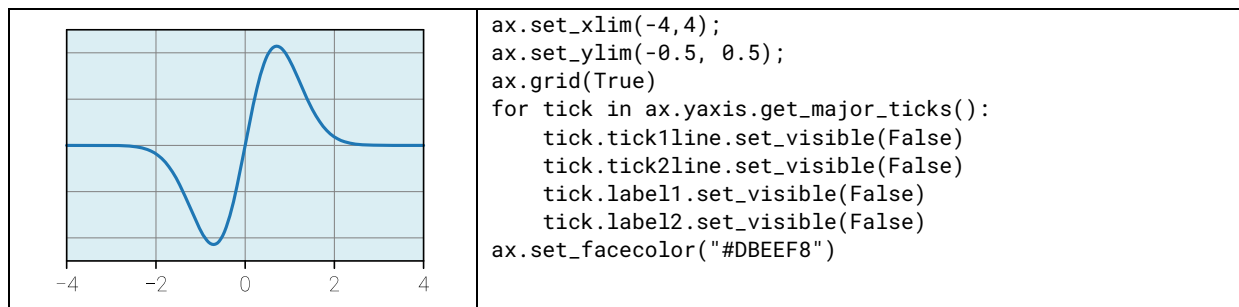
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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



4.3 图轴

在 Matplotlib 中，**图轴** (axis) 是指图形中的坐标轴，用于表示数据的数值范围和刻度。图轴包括**轴线** (axis line)、**刻度** (tick)、**轴标签** (axis label) 等艺术家。如果一个图片对象包含多个轴对象，则每个轴对象都有自己的艺术家。

表 3 总结图轴设计方案，请大家自行学习 Bk_2_Ch04_03.ipynb。

图 2 所示为对数坐标，这幅图来自 *Scientific Visualization: Python & Matplotlib*。请大家参考 Bk_2_Ch04_04.ipynb 自行学习。

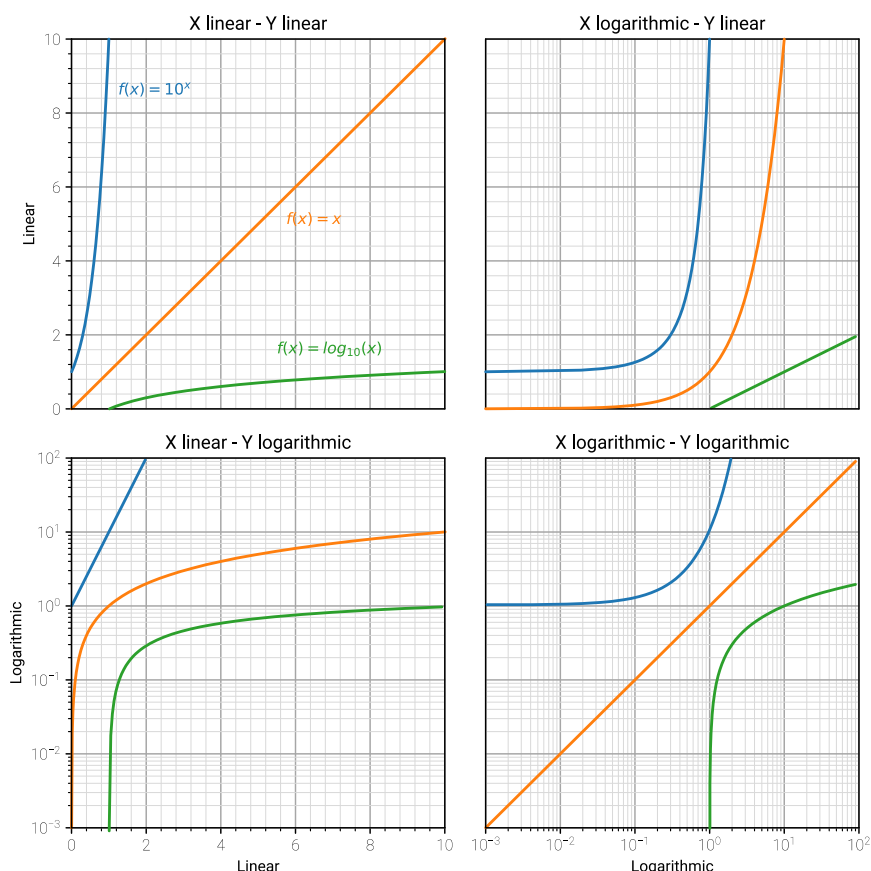



图 2. 对数坐标，代码参考 *Scientific Visualization: Python & Matplotlib* |  Bk_2_Ch04_04.ipynb


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。

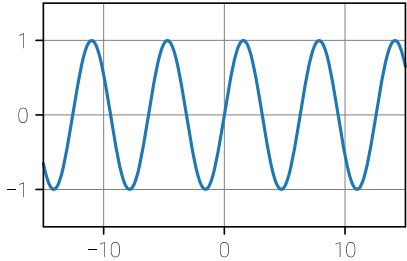
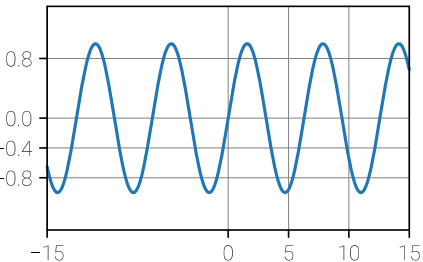
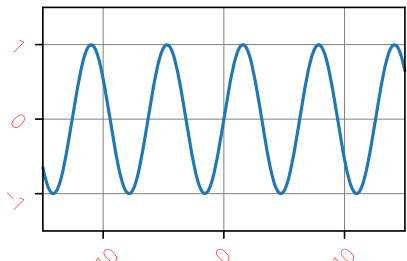
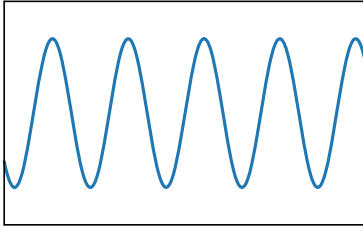
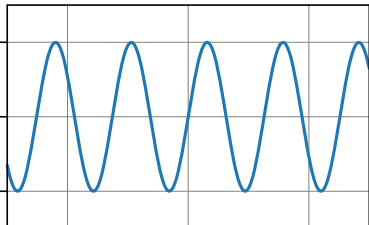
版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

表 3. 图轴设计 |  Bk_2_Ch04_03.ipynb

图轴设计	代码
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5);</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.set_xticks([-15, 0, 5, 10, 15]) # 可以用 numpy.linspace() ax.set_yticks([-0.8, -0.4, 0, 0.4, 0.8])</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.tick_params(axis='x', rotation = 45, labelcolor = 'r', labelsz = 10) ax.tick_params(axis='y', rotation = -45, labelcolor = 'r', labelsz = 10)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.tick_params(axis='x', rotation = 45, labelcolor = 'r', labelsz = 10) ax.tick_params(axis='y', rotation = -45, labelcolor = 'r', labelsz = 10)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.yaxis.set_major_formatter(plt.NullFormatter()) ax.xaxis.set_major_formatter(plt.NullFormatter())</pre>

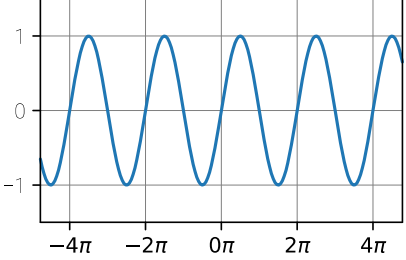
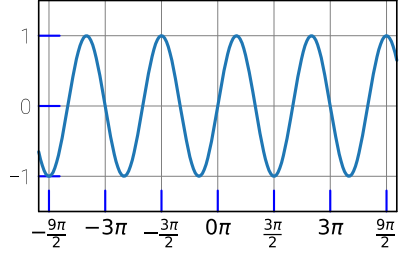
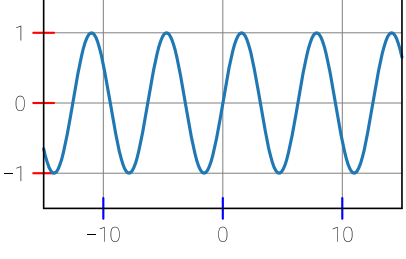
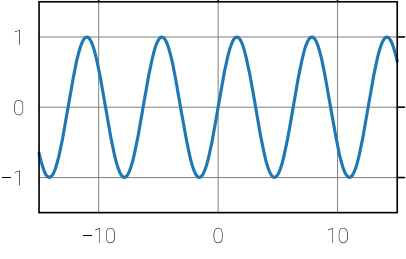
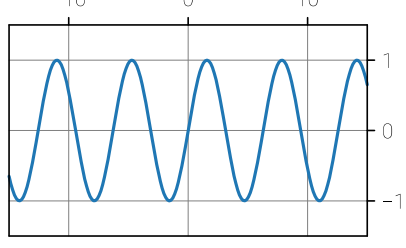
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

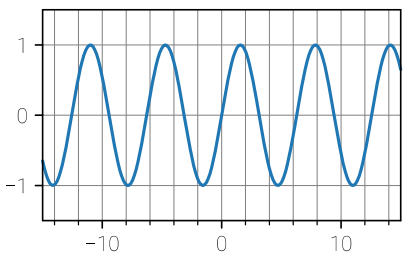
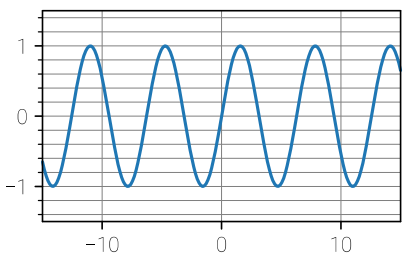
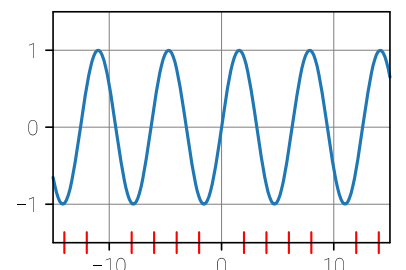
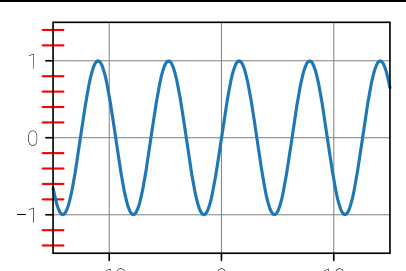
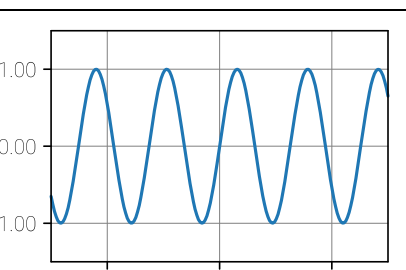
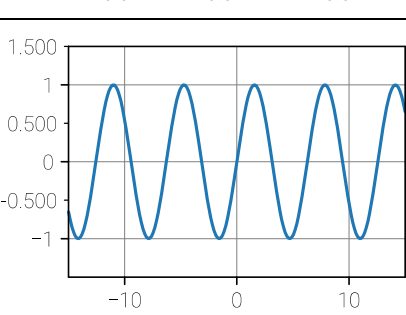
版权归清华大学出版社所有，请勿商用，引用请注明出处。

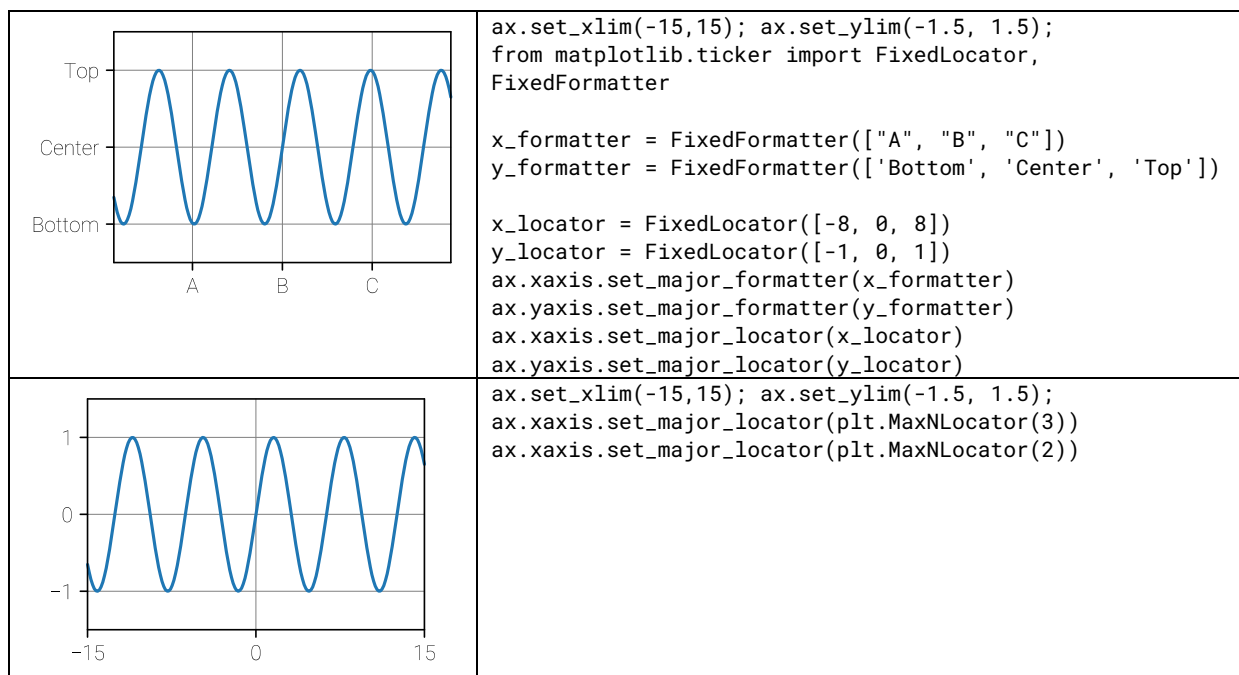
代码及 PDF 文件下载：<https://github.com/Visualize-ML>

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

	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.set_xticks(np.linspace(-4*np.pi, 4*np.pi, 5)) ax.set_xticklabels([r'\$-4\pi\$', r'\$-2\pi\$', r'\$0\pi\$', r'\$2\pi\$', r'\$4\pi\$'])</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.set_xticks(np.linspace(-9/2*np.pi, 9/2*np.pi, 7)) ax.tick_params(axis = 'x', direction = 'in', color = 'b', width = 1, length = 10) ax.tick_params(axis = 'y', direction = 'in', color = 'b', width = 1, length = 10) ax.set_xticklabels([r'\$-\frac{9\pi}{2}\$', r'\$-3\pi\$', r'\$-\frac{3\pi}{2}\$', r'\$0\pi\$', r'\$\frac{3\pi}{2}\$', r'\$3\pi\$', r'\$\frac{9\pi}{2}\$'])</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.tick_params(axis = 'x', direction = 'inout', color = 'b', width = 1, length = 10) ax.tick_params(axis = 'y', direction = 'inout', color = 'r', width = 1, length = 10)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.tick_params(bottom = False, top = True, left = False, right = True)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.tick_params(bottom = False, top = True, left = False, right = True) ax.tick_params(labelbottom = False, labeltop = True, labelleft = False, labelright = True)</pre>

	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); minor_locator = AutoMinorLocator(5) ax.xaxis.set_minor_locator(minor_locator) plt.grid(which='minor')</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); minor_locator = AutoMinorLocator(5) ax.xaxis.set_minor_locator(minor_locator) plt.grid(which='minor')</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); minor_locator = AutoMinorLocator(5) ax.xaxis.set_minor_locator(minor_locator) ax.tick_params(which="minor", axis="x", direction="inout", color = 'r', length = 10, width = 1)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); minor_locator = AutoMinorLocator(5) ax.yaxis.set_minor_locator(minor_locator) ax.tick_params(which="minor", axis="y", direction="inout", color = 'r', length = 10, width = 1)</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.yaxis.set_major_formatter(FormatStrFormatter('%0.2f')) ax.xaxis.set_major_formatter(FormatStrFormatter('%0.1f'))</pre>
	<pre>ax.set_xlim(-15,15); ax.set_ylim(-1.5, 1.5); ax.set_yticks([-1, 0, 1]) minor_locator = AutoMinorLocator(2) ax.yaxis.set_minor_locator(minor_locator) ax.yaxis.set_minor_formatter(FormatStrFormatter('%0.3f'))</pre>



4.4 注释

一幅图的注释可以如下常见类型。

- ▶ **标题** (title), 对应函数为 `matplotlib.pyplot.title()` 或 `ax.set_title()`;
- ▶ **横轴标题** (x axis label), 对应函数为 `matplotlib.pyplot.xlabel()` 或 `ax.set_xlabel()`;
- ▶ **纵轴标题** (y axis label), 对应函数为 `matplotlib.pyplot.ylabel()` 或 `ax.set_ylabel()`;
- ▶ **图例** (legend), 对应函数为 `matplotlib.pyplot.legend()`;
- ▶ **文字** (text), 对应函数为 `matplotlib.pyplot.text()`;
- ▶ **注解** (annotation), 对应函数为 `matplotlib.pyplot.annotate()`。

图 5 给出的几种标题、轴标题、图例布置方案，请大家自行学习 `Bk_2_Ch04_05.ipynb`。

本书不特别展开讲解如何通过编程添加文字或符号注释，请大家自行学习。

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.text.html

<https://matplotlib.org/stable/users/explain/text/annotations.html>

《编程不难》介绍过，考虑时间成本、美观效果、可编辑性等因素，如果后期制作加入各种注释更为方便的话，不建议大家耗费精力通过编程方式在图片中添加注释。

如若用平面散点、三维散点可视化样本数据，需要添加标签的话，建议使用 `plotly.express.scatter()` 和 `plotly.express.scatter_3d()`。

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

4.5 视角

三维可视化方案是本书重要的组成部分。大家在本书中看到我们利用三维散点图、线图、网格曲面、等高线展示各种数学概念。

《编程不难》介绍过，在 Matplotlib 中，`ax.view_init(elev, azim, roll)` 方法用于设置三维坐标轴的视角，也叫相机照相位置。这个方法接受三个参数：`elev`、`azim` 和 `roll`，它们分别表示仰角、方位角和滚动角。

- **仰角 (elevation)**: `elev` 参数定义了观察者与 xy 平面之间的夹角，也就是观察者与 xy 平面之间的旋转角度。当 `elev` 为正值时，观察者向上倾斜，负值则表示向下倾斜。
- **方位角 (azimuth)**: `azim` 参数定义了观察者绕 z 轴旋转的角度。它决定了观察者在 xy 平面上的位置。`azim` 的角度范围是 -180 到 180 度，其中正值表示逆时针旋转，负值表示顺时针旋转。
- **滚动角 (roll)**: `roll` 参数定义了绕观察者视线方向旋转的角度。它决定了观察者的头部倾斜程度。正值表示向右侧倾斜，负值表示向左侧倾斜。

如图 3 所示，鸢尾花书中调整三维视图视角一般只会用 `elev`、`azim`，几乎不使用 `roll`。

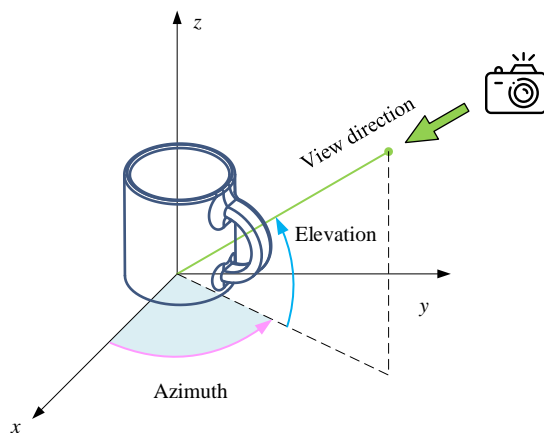


图 3. 仰角和方位角示意图

图 6 展示各种常见视角，供大家参考。Bk_2_Ch04_06.ipynb 绘制图 6，请大家自行学习。

此外，我们还在《编程不难》提过，鸢尾花书一般采用正交投影，即不考虑“近大远小”。对于三维轴对象会通过 `ax.set_proj_type('ortho')` 设置正交投影。

4.6 风格

图像的风格指的是图表的整体外观和样式，包括配色方案、线条类型、图脊、图轴、线条宽度、字体、标记符号等等设计元素。

Matplotlib 提供了一系列的预定义风格，可以通过设置来改变图表的外观。以下是 Matplotlib 中常用的一些图像风格类型：

- ▶ "default" 是 Matplotlib 的默认风格，使用蓝色线条和绿色网格。
- ▶ "classic" 是一种经典的 Matplotlib 风格，使用黑色线条和白色背景，类似于传统的 Matplotlib 版本。
- ▶ "ggplot" 模仿了 R 语言中的 ggplot 库的外观，使用灰色网格和彩色线条。
- ▶ "fivethirtyeight" 模仿了流行的数据新闻网站 FiveThirtyEight 的外观，使用红色和蓝色线条，以及灰色网格。
- ▶ "dark_background" 使用深色背景和亮色线条，适合用于暗色主题的环境。
- ▶ "seaborn" 模仿了 Seaborn 库的外观，使用柔和的颜色和灰色网格。

关于 Matplotlib 绘图风格，请大家参考：

https://matplotlib.org/stable/gallery/style_sheets/style_sheets_reference.html

使用过 MATLAB 绘图的读者一定忘不了其严肃，甚至有些呆板的图像风格。而用过 R 语言的 ggplot 的读者，换成 Python 的 Matplotlib 绘图时肯定会有各种视觉上的不适。

Python 中 Plotnine 库的出图风格类似 ggplot，本书不展开介绍，请大家自行学习。

<https://plotnine.readthedocs.io/>

对于科技制图风格设定，请大家关注 SciencePlots 这个开源项目。

<https://pypi.org/project/SciencePlots/>

有必要介绍 ProPlot 提供的可视化方案，因为这个绘图库的出图风格特别“像”科技三大刊——Cell、Nature 和 Science。

ProPlot 建立在 Matplotlib 之上，提供了更简洁、更便于使用的科学绘图包。ProPlot 支持各种常见的绘图类型，包括线图、散点图、等高线图、柱状图等，并且支持创建多个子图和面板图。

ProPlot 支持高分辨率的输出，可以生成矢量图形（如 PDF、SVG）和栅格图像（如 PNG、JPEG）等多种格式。

大家需要在科技期刊发表文章的话，可以学习使用 ProPlot。ProPlot 官网地址如下：

<https://proplot.readthedocs.io/en/stable/>

为了节省篇幅，本书不展示 ProPlot 官网提供的范例。大家可以在本章配套代码中找到范例和对应代码，请大家自行学习实践。

需要注意的是，ProPlot 还在开发中，使用时可能会报错。目前这个可视化库在 <https://snyk.io/advisor/python> 打分低于 80 分。



本章主要介绍了 Matplotlib 一些常见的装饰操作。不需要大家死记硬背，掌握这些技巧的唯一途径就是多尝试、多实践。

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

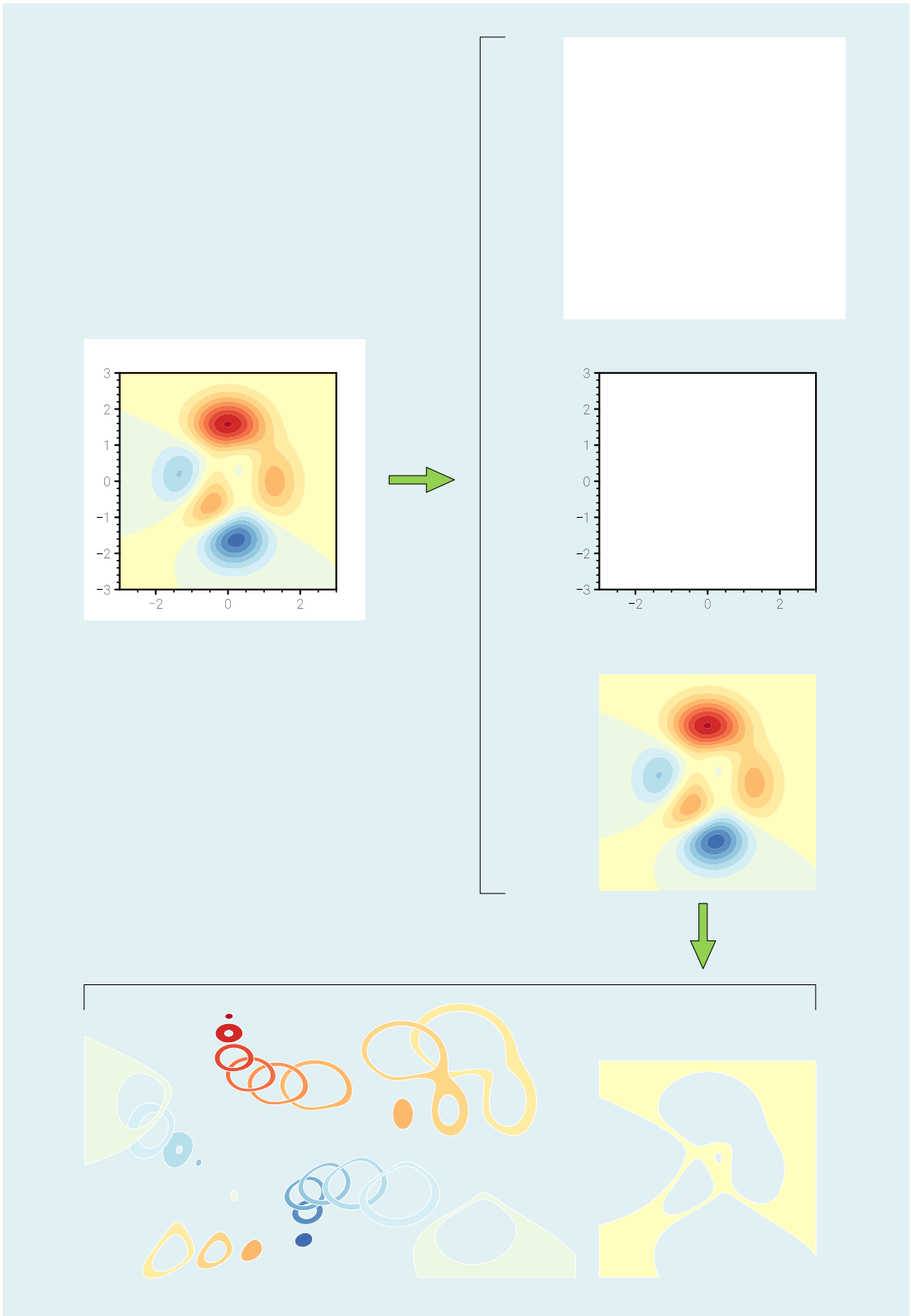


图 4. 一幅等高线的层级结构和艺术家们

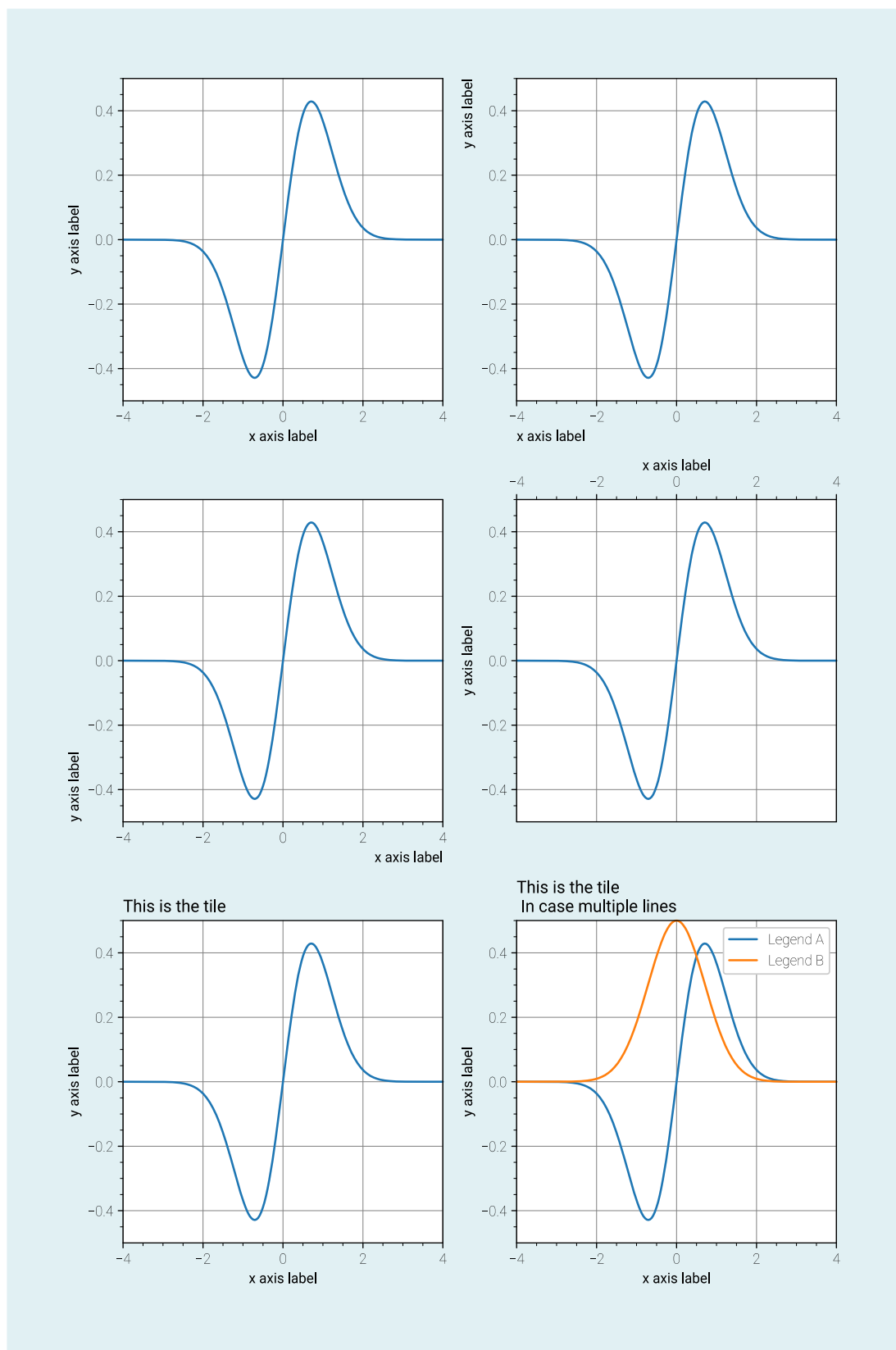


图 5. 标注 | Bk_2_Ch04_05.ipynb

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

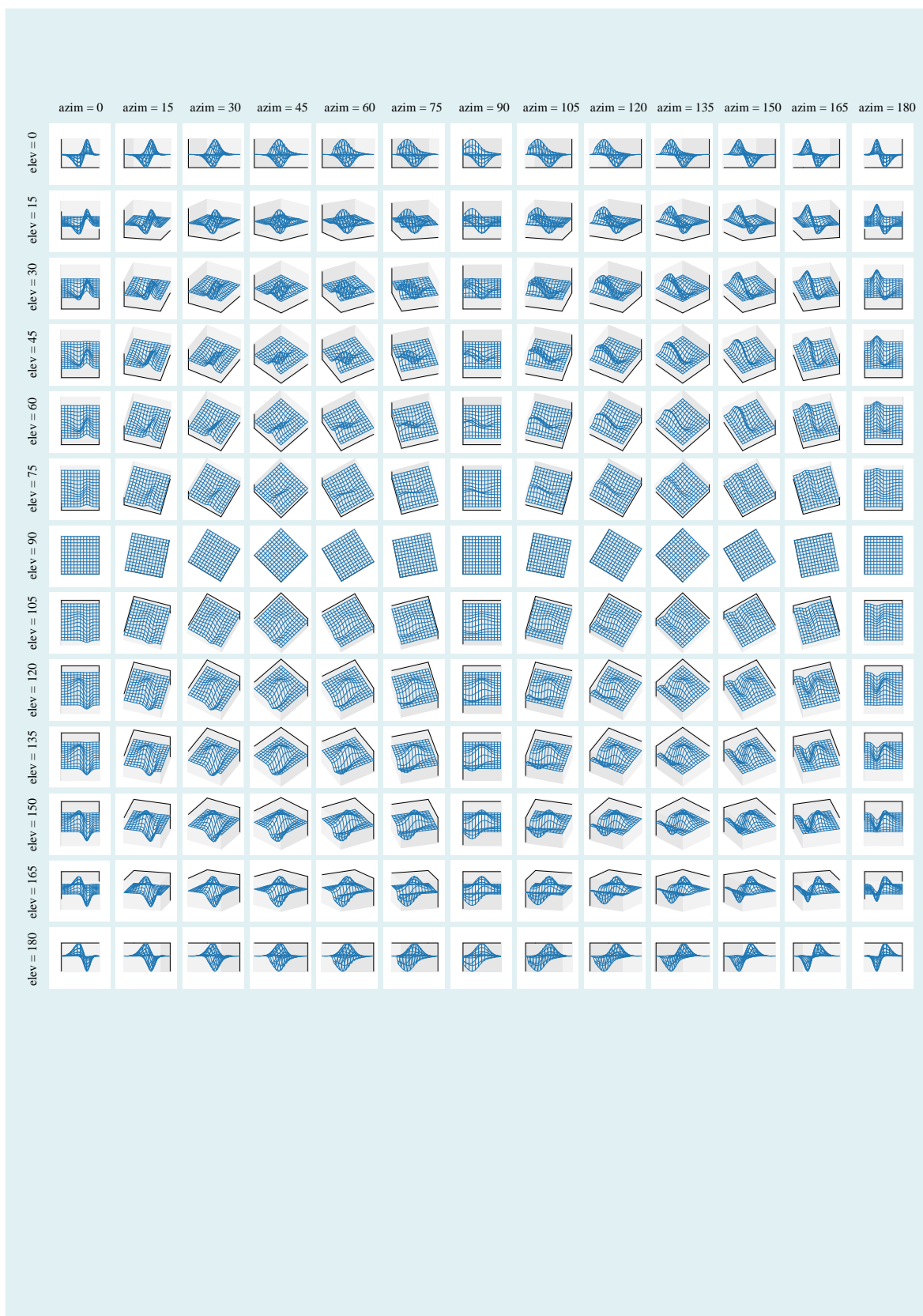



图 6. 利用 `add_gridspec()` 函数绘制子图展示三维网格面随视角变化 |  Bk_2_Ch04_06.ipynb

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