

## 24

## Visualize Complex Functions

## 复数

丰富且美丽的线性、非线性变换



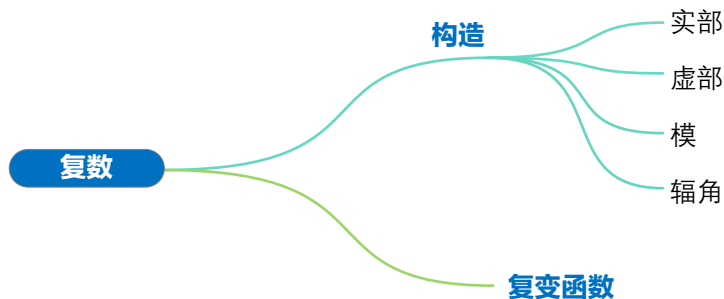
我内心燃烧着一团巨大的火焰，但没有人停下来在它身上取暖，路人只看到一缕烟雾。

*A great fire burns within me, but no one stops to warm themselves at it, and passers-by only see a wisp of smoke.*

—— 文森特·梵高 (Vincent van Gogh) | 荷兰后印象派画家 | 1853 ~ 1890



- ◀ matplotlib.pyplot.axhline() 绘制水平线
- ◀ matplotlib.pyplot.axvline() 绘制竖直线
- ◀ matplotlib.pyplot.contour() 绘制等高线图
- ◀ matplotlib.pyplot.contourf() 绘制填充等高线图
- ◀ matplotlib.pyplot.pcolormesh() 绘制二维网格数据的伪彩色图
- ◀ matplotlib.pyplot.quiver() 绘制箭头图
- ◀ matplotlib.pyplot.scatter() 绘制散点图
- ◀ numpy.abs() 计算复数模
- ◀ numpy.angle() 计算复数辐角
- ◀ numpy.meshgrid() 创建网格化数据
- ◀ numpy.zeros\_like() 用来生成和输入矩阵形状相同的零矩阵



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)

## 24.1 复数

### 什么是复数？

**复数** (complex number) 是由**实部** (real) 和**虚部** (imaginary) 组成的数。实部表示复数在实轴上的投影，虚部表示复数在虚轴上的投影。复数可以用形如  $a + bi$  的形式表示，其中  $a$  是实部， $b$  是虚部， $i$  是**虚数单位** (imaginary unit)，满足  $i^2 = -1$ 。

### 复平面

**复平面** (complex plane) 是一个用于表示复数的平面，其中实轴和虚轴分别对应平面上的横轴和纵轴。

在复平面上，每个复数可以表示为一个点  $(a, b)$ ，其中实部  $a$  对应于点在实轴上的投影，虚部  $b$  对应于点在虚轴上的投影。复数的实部和虚部可以用来确定平面上的点的位置。

不同于**笛卡尔坐标系** (Cartesian coordinate system)，复平面的横轴为实数轴，纵轴表示纯虚数轴。

通过复平面，可以直观地理解复数的性质和运算。例如，复数的加减法对应于在复平面上的向量相加和相减，复数的乘除法对应于向量的伸缩和旋转。

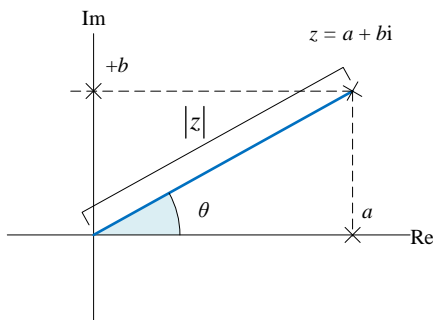


图 1. 复数

如图 2 所示，复平面上一系列点可以看成是一组“箭头”。

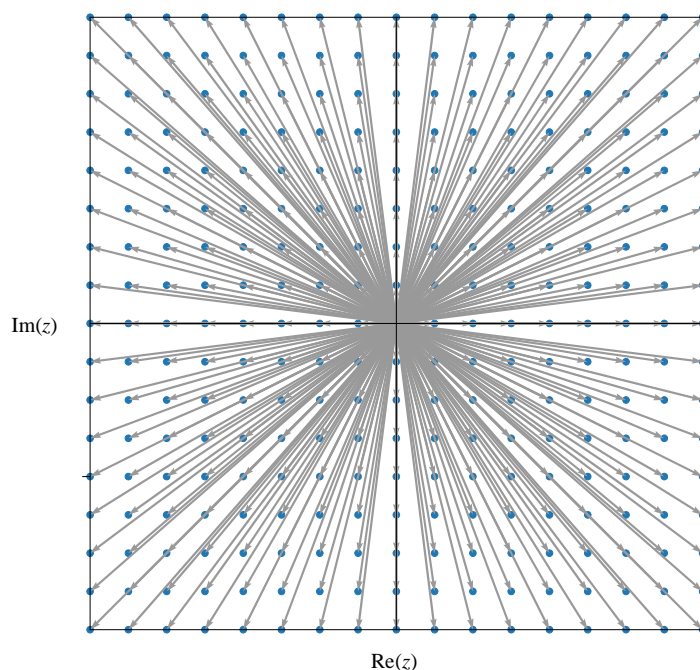



图 2. 复平面上，一系列“箭头” |  Bk2\_Ch24\_01.ipynb

下面，我们聊聊生成图 2 代码。

代码 1 用来生成复数数据。

- a** 用 `numpy.meshgrid()`，简作 `np.meshgrid()`，生成网格坐标数据。这一组网格数据颗粒度极高，用来生成细腻等高线，也用来渲染。
- b** 用来生成复数数据，`1j` 为虚数单位。实部是 `xx1`，虚部是 `xx2` 乘以虚数单位 `1j`。
- c** 使用 `numpy.angle()`，简作 `np.angle()`，计算复数的辐角。
- d** 使用 `numpy.abs()`，简作 `np.abs()`，计算复数的模。
- e** 也用 `numpy.meshgrid()` 生成网格坐标数据，不同的是颗粒度很低；这些数据用来绘制复数箭头图。为了有更好的可视化效果，我们选择了奇数 17 作为一维数组数据点个数。大家可以尝试使用其他个数，并观察图像变化。
- f** 用 `numpy.zeros_like()`，简作 `np.zeros_like()`，创建了一个与 `xx1_` 具有相同形状的全零数组。这些数据用作复数向量的起点坐标。


```

# 颗粒度高，用来颜色渲染
a xx1, xx2 = np.meshgrid(np.linspace(-2,2,2048),
                        np.linspace(-2,2,2048))
b zz = xx1 + xx2 * 1j
c zz_angle = np.angle(zz)
d zz_norm = np.abs(zz)

# 颗粒度低，用来绘制箭头图
e xx1_, xx2_ = np.meshgrid(np.linspace(-2,2,17),
                          np.linspace(-2,2,17))
zz_ = xx1_ + xx2_ * 1j
zz_angle_ = np.angle(zz_)
zz_norm_ = np.abs(zz_)

# 向量起始点 (0,0)
f zeros = np.zeros_like(xx1_)

```

代码 1. 产生复数数据 |  Bk2\_Ch24\_01.ipynb

代码 2 用来绘制箭头图。

**a** 用 `scatter()` 方法在轴对象 `ax` 上绘制散点图，用来代表复数  $(a, b)$  的具体位置。

**b** 用 `quiver()` 方法绘制平面箭头图。

参数 `zeros` 和 `zeros` 是箭头的起点的 `x` 和 `y` 坐标。由于这里都是零，所以箭头的起点是从原点开始的。

参数 `xx1_` 和 `xx2_` 是箭头的终点的 `x` 和 `y` 坐标。它们代表了矢量场中每个箭头的目标位置。

参数 `color=[0.6, 0.6, 0.6]` 指定箭头的颜色。这里使用 RGB 色号，`[0.6, 0.6, 0.6]` 表示灰色。

参数 `angles='xy'` 指定箭头的角度指向为  $(x, y)$  到  $(x+u, y+v)$ 。

参数 `scale_units='xy'` 指定矢量场图中箭头的比例是相对于 `x` 和 `y` 轴的。

参数 `scale=1` 指定箭头的比例因子，即箭头的长度乘以这个比例因子。

参数 `edgecolor='none'` 指定箭头的边缘颜色，设置为 `'none'` 表示没有边缘。

**c** 先用 `axhline()` 在轴对象 `ax` 上绘制水平线，水平线的高度为  $y = 0$ ，`c = 'k'` 设定颜色为黑色。半角分号 `;` 分割两句。下一句，用 `axvline()` 绘制竖直线，位置为  $x = 0$ 。

```

fig, ax = plt.subplots(figsize = (5,5))
a ax.scatter(xx1_, xx2_, marker = '.')
b ax.quiver(zeros, zeros, xx1_, xx2_,
           color = [0.6, 0.6, 0.6],
           angles='xy', scale_units='xy', scale = 1,
           edgecolor='none')
c ax.axhline(y = 0, c = 'k'); ax.axvline(x = 0, c = 'k')
ax.set_xlim(-2,2); ax.set_ylim(-2,2)
ax.set_xticks(np.arange(-2,3)); ax.set_yticks(np.arange(-2,3))
ax.set_xlabel("$Re(z)$"); ax.set_ylabel("$Im(z)$")
plt.show()

```

代码 2. 绘制复数箭头图 |  Bk2\_Ch24\_01.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)

## 复数模、辐角

复平面的**极坐标** (polar coordinate system) 形式也很常见，其中复数可以表示为**模** (modulus, magnitude, absolute value) 和**辐角** (angle) 的组合。

在极坐标形式下，复数的模对应于从原点到点的距离，辐角对应于从正实轴**逆时针** (counter-clockwise) 旋转到点的角度。

复数模是指复数的绝对值或者长度，表示从原点到复数在复平面上的距离。对于一个复数  $z = a + bi$ ，其模可以用  $|z| = \sqrt{a^2 + b^2}$  计算。图 3 所示为两种可视化复数模的方案，本书后文采用平面等高线。

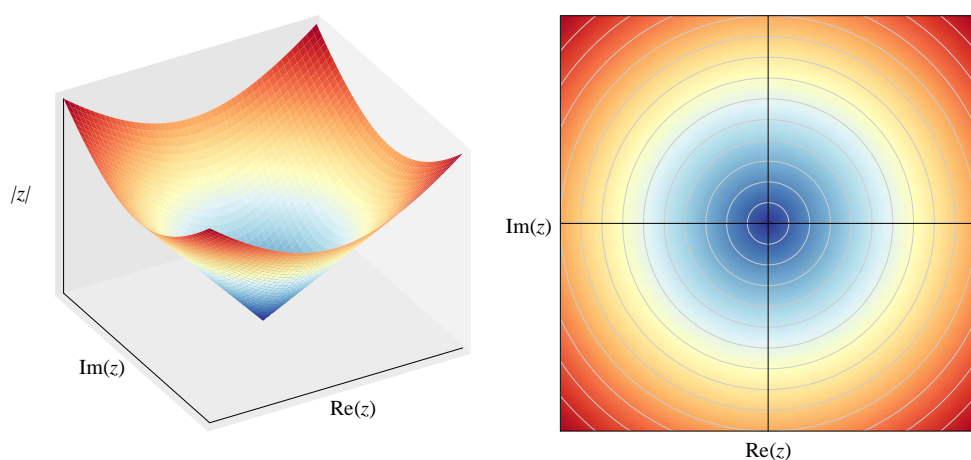



图 3. 可视化复数的模 |  Bk2\_Ch24\_01.ipynb

代码 3 绘制图 3 两幅子图，下面聊聊其中关键语句。

**a** 利用 `matplotlib.pyplot.figure()`，简作 `plt.figure()`，创建图形对象 `fig`。

大家 **b** 这一句应该很熟悉了；但是，本着“重复 + 精进”原则，我们还是要聊一下。

**b** 利用 `add_subplot()` 在 `fig` 上增加一个三维轴对象。

参数 `111` 表示将图形分成 1 行 1 列的子图网格，当前操作的是第 1 个子图。

参数 `projection="3d"` 表示这是一个三维子图。

`ax` 是返回的子图三维轴对象，我们可以使用它来进行进一步的绘图操作。

**c** 在三维轴对象 `ax` 上用 `plot_surface()` 绘制曲面图。

`xx1` 和 `xx2` 代表网格数据的 `x` 和 `y` 坐标。

`zz_norm` 为曲面图上每个点的 `z` 坐标。

`cmap="RdYlBu_r"` 指定用于渲染的颜色映射。“`RdYlBu_r`”表示红黄蓝的颜色映射，“`_r`”表示颜色映射的反转，即蓝黄红。

参数 `shade=True` 指定是否对表面进行阴影处理，以增强立体感。

参数 `alpha=1` 指定图形的透明度，`alpha=1` 代表不透明。

**d** 用 `contour()` 在平面坐标轴对象上绘制平面等高线。

参数 `levels=np.linspace(0, 5, 26)` 指定等高线高度位置。

`colors=[[0.8, 0.8, 0.8, 1]]` 指定等高线的颜色。这里使用 RGBA 表示法, `[0.8, 0.8, 0.8, 1]` 表示灰色, 最后的 1 表示完全不透明。

**e** 利用 `Pcolormesh()` 方法在二维平面上绘制一个伪彩色网格。

`zz_normy` 用来指定伪彩色网格上每个点的颜色。

`cmap='RdYlBu_r'` 指定伪彩色图的颜色映射。

`shading='auto'` 指定着色方式。使用 `'auto'` 表示由 Matplotlib 自动选择最合适的渲染方式。

`rasterized=True` 指定是否对图形进行光栅化处理。光栅化的结果为像素图, 可以提高图形的显示性能, 特别是对于大型数据集。

```
# 三维曲面渲染
a fig = plt.figure(figsize=(5,5))
b ax = fig.add_subplot(111, projection="3d")

c ax.plot_surface(xx1, xx2, zz_norm,
                  cmap="RdYlBu_r", shade=True, alpha=1)
ax.set_xlabel("$Re(z)$"); ax.set_ylabel("$Im(z)$")
ax.set_proj_type('ortho')
ax.set_xticks([]); ax.set_yticks([]); ax.set_zticks([])
ax.view_init(azim=-120, elev=30); ax.grid(False)
ax.set_xlim(xx1.min(), xx1.max()); ax.set_ylim(xx2.min(), xx2.max())
plt.show()

# 平面投影
fig, ax = plt.subplots(figsize = (5,5))

d plt.contour(xx1, xx2, zz_norm,
              levels = np.linspace(0,5,26),
              colors = [[0.8, 0.8, 0.8, 1]])
e plt.pcolormesh(xx1, xx2, zz_norm, cmap='RdYlBu_r',
                 shading = 'auto', rasterized = True)
ax.set_xlim(-2,2); ax.set_ylim(-2,2)
ax.axhline(y = 0, c = 'k'); ax.axvline(x = 0, c = 'k')
ax.set_xticks(np.arange(-2,3)); ax.set_yticks(np.arange(-2,3))
plt.show()
```

代码 3. 绘制复数箭头图 | Bk2\_Ch24\_01.ipynb

辐角, 或幅角, 是指从正实轴逆时针旋转到复数所在位置的角度。可以使用三角函数来计算角度。对于一个非零复数  $z = a + bi$ , 其角度可以用  $\theta = \arctan(b, a)$  计算。`arctan()` 是一个可以返回带符号角度的反正切函数。角度的单位通常是弧度。

图 4 所示为在复平面上用 `hsv` 色谱可视化辐角, 这是因为 `hsv` 色谱为循环色谱, 首尾颜色咬合。

复数的模和角度提供了对复数在复平面上的位置和特征的描述。它们在复数的运算、变换和表示中都起着重要的作用。可以使用模和角度来进行复数的乘法、除法、幂运算等操作, 并且它们还可以方便地表示复数的极坐标形式。

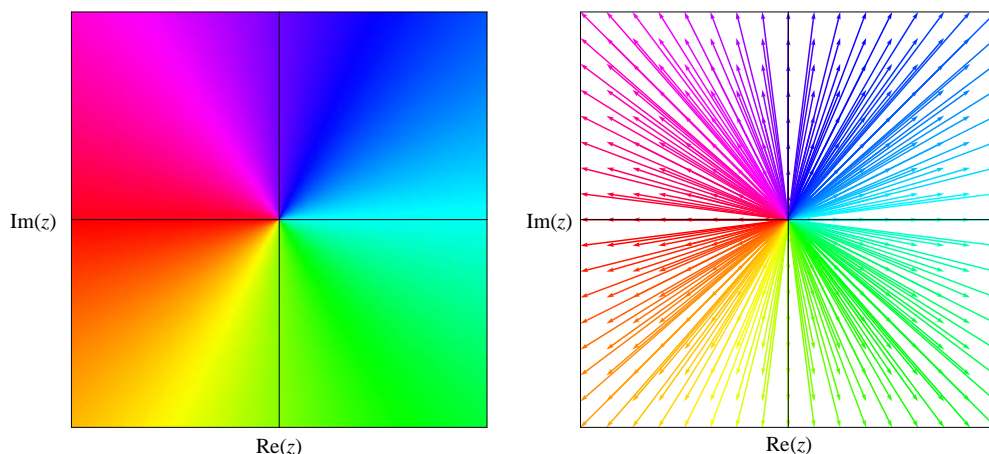


图 4. 复平面上, 用 hsv 色谱可视化辐角 | Bk2\_Ch24\_01.ipynb

代码 4 绘制图 4, 下面聊聊其中关键语句。

**a** 用 `pcolormesh()` 绘制伪彩色网格图展示辐角。

辐角 `zz_angle` 决定了伪彩色网格上每个点的颜色。

`cmap='hsv'` 指定颜色映射。

**b** 用 `quiver()` 绘制平面箭头图。不同的是, 我们用 `zz_angle_` 确定每条箭头的颜色, 指定的颜色映射是 `'hsv'`。

```
# 用伪彩色网格展示辐角
fig, ax = plt.subplots(figsize = (5,5))
ax.pcolormesh(xx1, xx2, zz_angle,
              cmap='hsv', shading = 'auto',
              rasterized = True)
ax.set_xlim(-2,2); ax.set_ylim(-2,2)
ax.set_xticks(np.arange(-2,3)); ax.set_yticks(np.arange(-2,3))
ax.axhline(y = 0, c = 'k'); ax.axvline(x = 0, c = 'k')
plt.show()

# 渲染向量
fig, ax = plt.subplots(figsize = (5,5))
ax.quiver(zeros, zeros, xx1_, xx2_, zz_angle_,
          angles='xy', scale_units='xy', scale = 1,
          edgecolor='none', alpha=0.8, cmap = 'hsv')

ax.set_xlim(-2,2); ax.set_ylim(-2,2)
ax.axhline(y = 0, c = 'k'); ax.axvline(x = 0, c = 'k')
ax.set_xticks(np.arange(-2,3)); ax.set_yticks(np.arange(-2,3))
plt.show()
```

代码 4. 可视化辐角 | Bk2\_Ch24\_01.ipynb

Bk2\_Ch24\_01.ipynb 还给出可视化复数的其他方案, 请大家自行学习。



## 24.2 复变函数

复变函数是指接受复数作为输入并返回复数作为输出的函数。复变函数可以将一个或多个复数作为参数，然后根据特定的规则进行计算，并返回一个或多个复数作为结果。

复变函数可以涉及各种数学运算和操作，例如加法、减法、乘法、除法、幂运算、三角函数、指数函数、对数函数等。这些函数在复数领域中扮演着重要的角色，因为复数具有实部和虚部，可以进行各种运算和转换。

一些常见的复变函数包括。

- ▶ 复数加法和减法函数：将两个或多个复数相加或相减。
- ▶ 复数乘法和除法函数：将两个或多个复数相乘或相除。
- ▶ 复数幂函数：计算一个复数的幂，其中指数可以是实数或复数。
- ▶ 复数三角函数：包括正弦、余弦、正切等，可以用来计算复数的三角函数值。
- ▶ 复数指数函数和对数函数：计算复数的指数和对数值。

本章重点内容就是可视化复变函数。本章选定图 5 所示两个可视化方案展示复变函数。图 5 (a) 背景是辐角，等高线为复数模；图 5 (b) 的黑色网格用来可视化复变函数导致的线性或非线性几何变换。本书前文介绍过如何用等高线绘制这种网格。

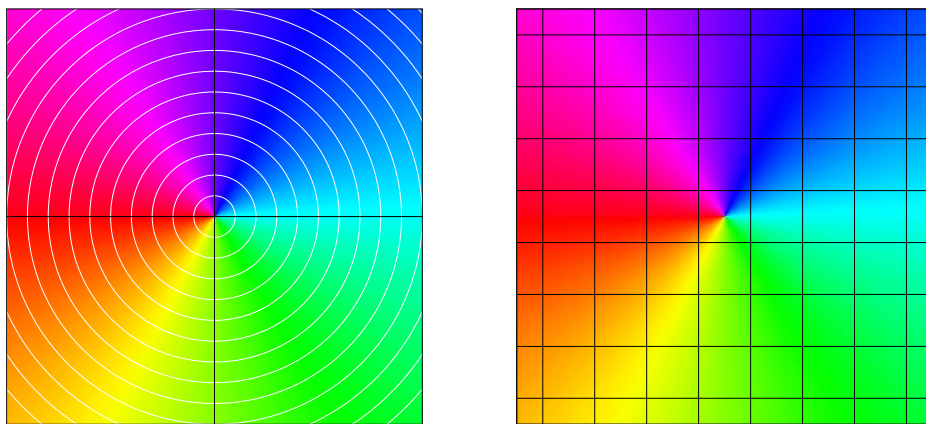


图 5. 复变函数可视化方案 |  Bk2\_Ch24\_01.ipynb

Bk2\_Ch24\_02.ipynb 绘制图 6 ~ 图 14。下面聊聊其中关键语句。

- a 这两句分别计算复数模和辐角。
- b 分离复数的实部和虚部。
- c 用 `pcolormesh()` 绘制为彩色网格可视化复数辐角。
- d 用 `contour()` 绘制等高线可视化复数模。
- e 还是用 `pcolormesh()` 绘制为彩色网格可视化复数辐角。
- f 用黑色网格可视化线性、非线性几何变换。

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)



```
def visualize(xx1, xx2, cc):
    # 计算复数模、辐角
    cc_norm = np.abs(cc); cc_angle = np.angle(cc)
    # 分离实部、虚部
    cc_xx1 = cc.real; cc_xx2 = cc.imag
    levels = np.linspace(0, np.mean(cc_norm) + 5 * np.std(cc_norm), 31)
    fig = plt.figure(figsize=(6, 3))
    # 第一幅子图
    ax = fig.add_subplot(1, 2, 1)
    ax.pcolormesh(xx1, xx2, cc_angle, cmap='hsv',
                  shading = 'auto', rasterized = True)
    ax.contour(xx1, xx2, cc_norm,
               levels = levels,
               colors = 'w', linewidths = 0.25)
    ax.set_xlim(-2, 2); ax.set_ylim(-2, 2)
    ax.set_xticks([]); ax.set_yticks([])
    ax.axhline(y = 0, c = 'k'); ax.axvline(x = 0, c = 'k')
    ax.set_aspect('equal')
    # 第二幅子图
    ax = fig.add_subplot(1, 2, 2)
    ax.pcolormesh(cc_angle, cmap='hsv',
                  shading = 'auto', rasterized = True)
    ax.contour(np.abs(cc_xx1 - np.round(cc_xx1)),
               levels = 1, colors="black", linewidths=0.25)
    ax.contour(np.abs(cc_xx2 - np.round(cc_xx2)),
               levels = 1, colors="black", linewidths=0.25)
    ax.set_xticks([]); ax.set_yticks([])
    ax.set_aspect('equal')
```

代码 5. 可视化复变函数 | Bk2\_Ch24\_02.ipynb

有关复数基本运算，请大家参考。

<https://mathworld.wolfram.com/ComplexNumber.html>

复数在许多领域有广泛的应用。在物理学中，复数用于描述交流电路中的电压和电流。在工程中，复数在信号处理和控制系统中起着重要作用。在数学领域，复数广泛用于解析几何和复变函数等领域。

在机器学习领域，复数也有一些应用。例如，复数神经网络是一种利用复数权重和激活函数的神经网络模型。这种模型可以更好地处理复杂的信号和数据，特别是在音频和图像处理等领域。复数神经网络在语音识别、图像处理和模式识别等任务中取得了一定的成功。

相信读完本章，哪怕之前对复变函数不感兴趣的读者，也不得不为复变函数展现出来的数学之美所震撼。数学之美其实无处不在，我们需要的丰富的创意和合适的可视化工具，就可以把数学之美立体地展现在大家面前。

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)

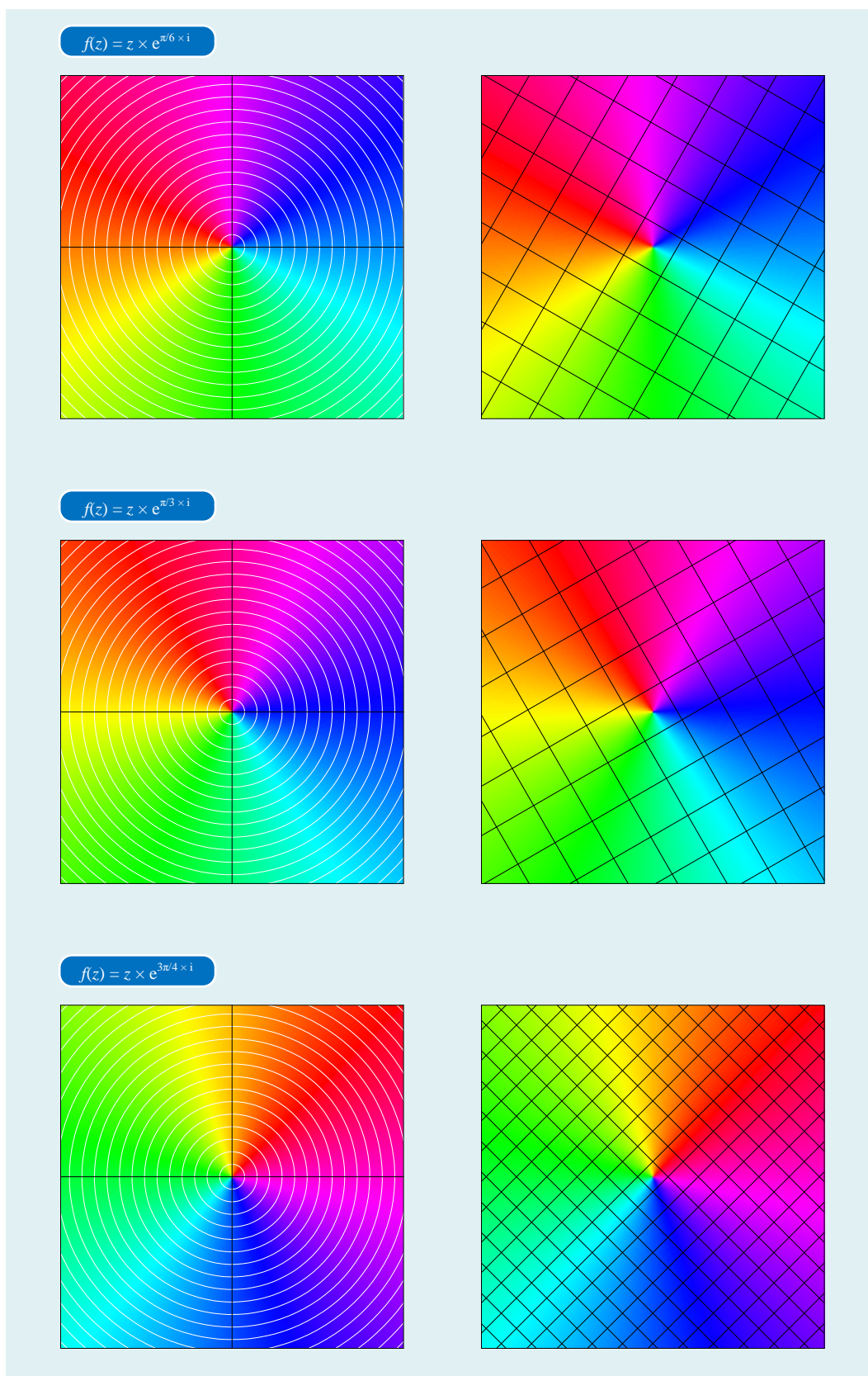


图 6. 旋转、缩放 | Bk2\_Ch24\_02.ipynb

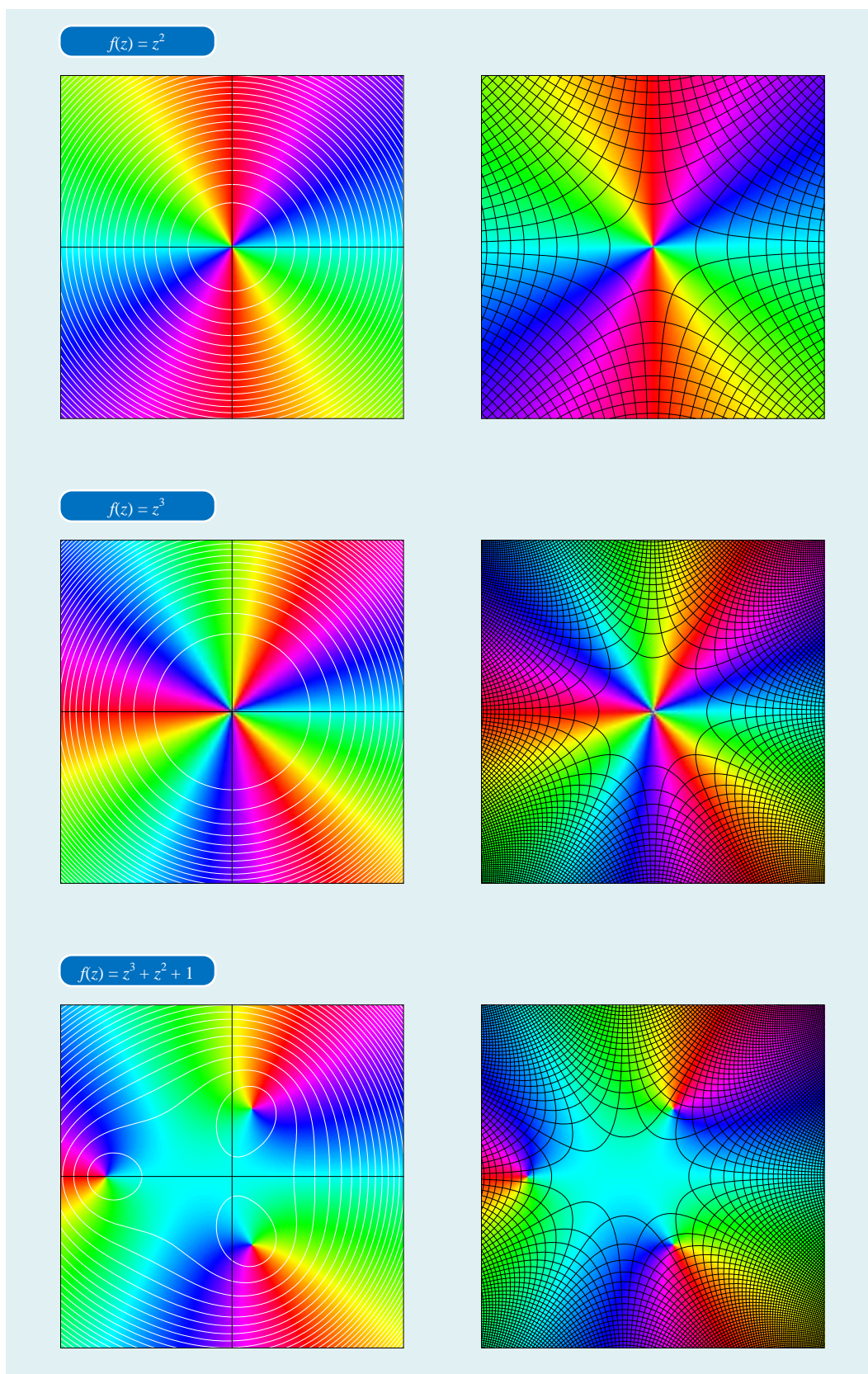


图 7. 多项式 | Bk2\_Ch24\_02.ipynb

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](mailto:jiang.visualize.ml@gmail.com)

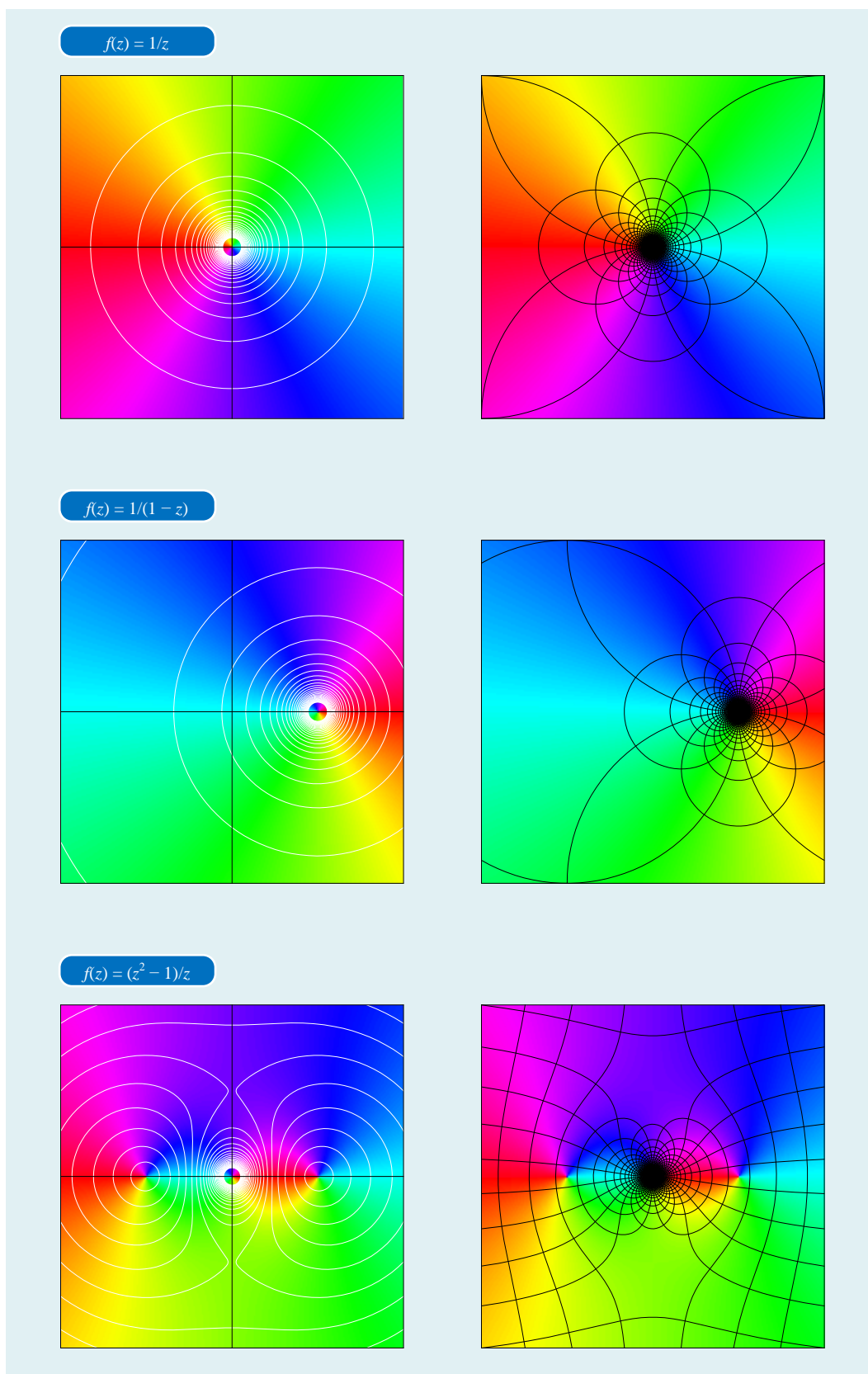


图 8. 分式, 第 1 组 | Bk2\_Ch24\_02.ipynb

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)



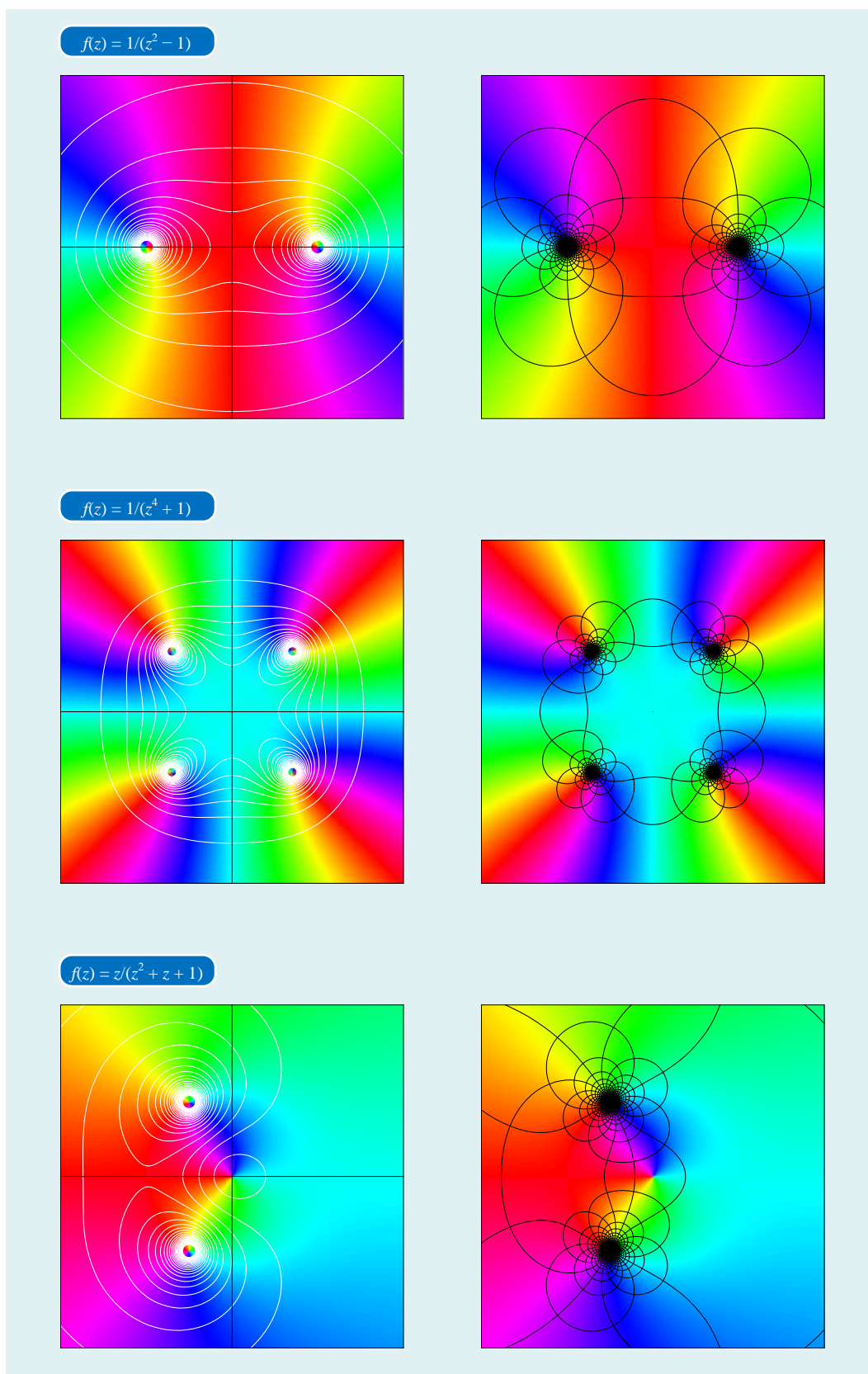
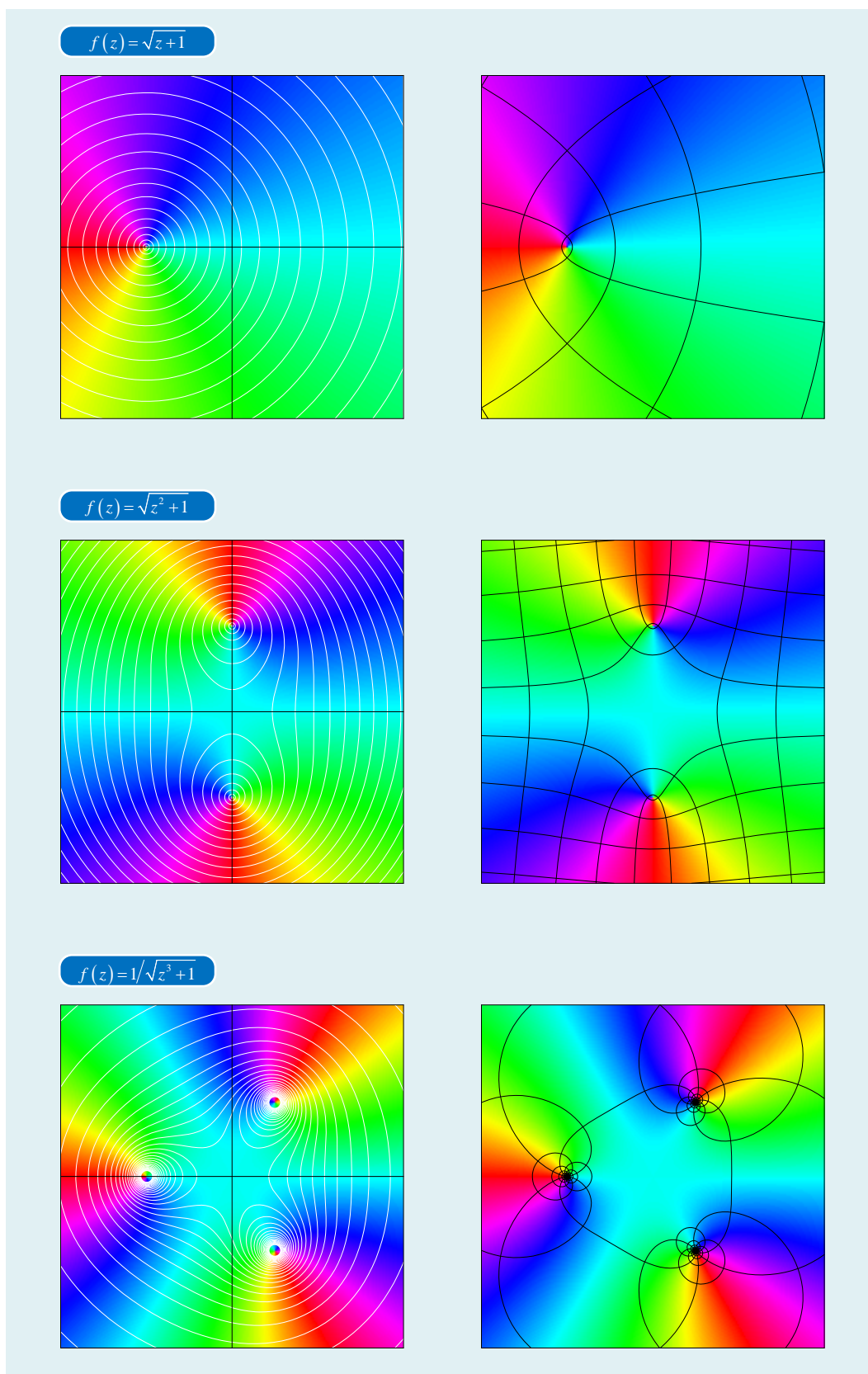



图 9. 分式, 第 2 组 | Bk2\_Ch24\_02.ipynb

图 10. 根式 |  Bk2\_Ch24\_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。

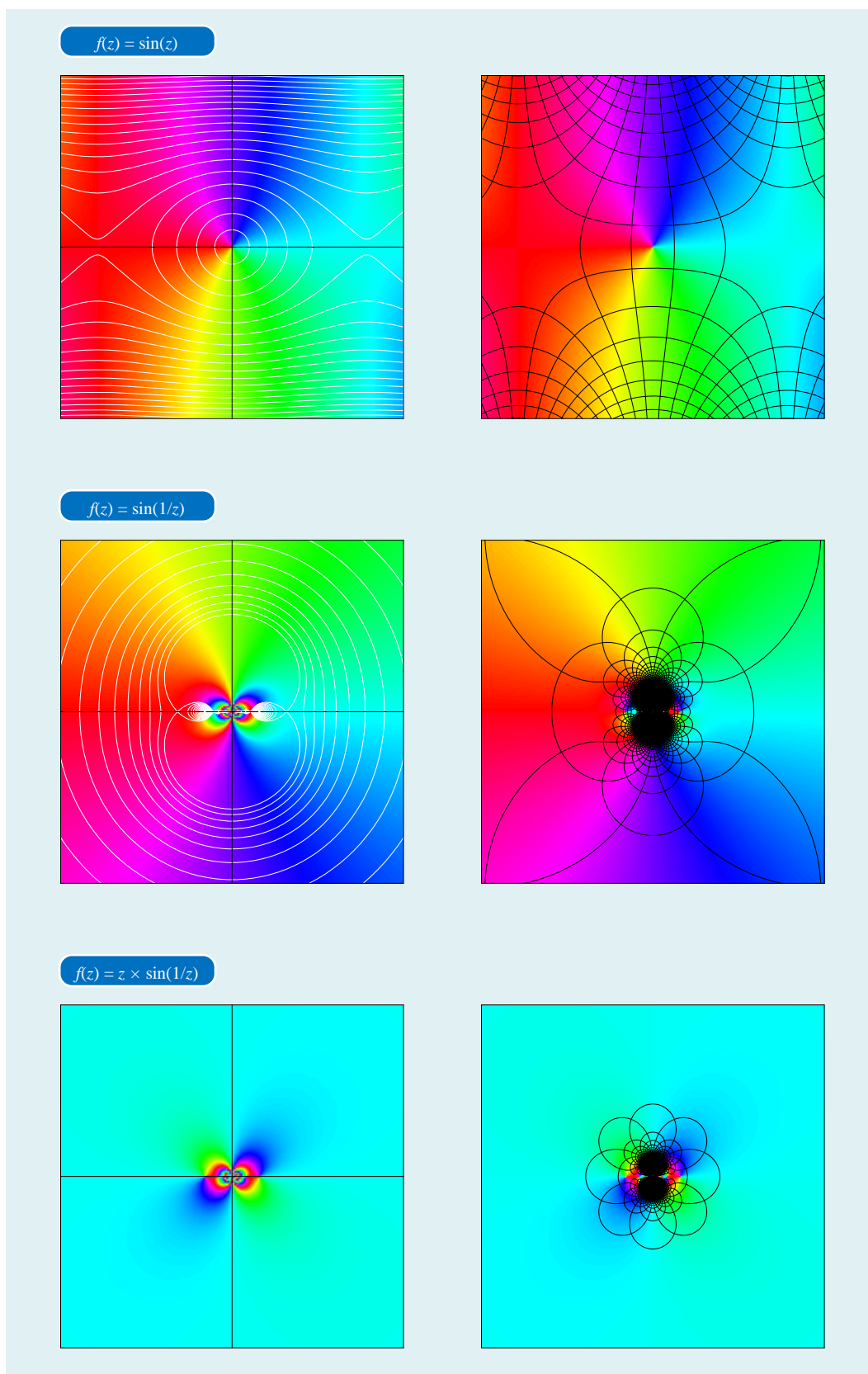
版权归清华大学出版社所有，请勿商用，引用请注明出处。

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](mailto:jiang.visualize.ml@gmail.com)



图 11. 三角正弦 |  Bk2\_Ch24\_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](mailto:jiang.visualize.ml@gmail.com)

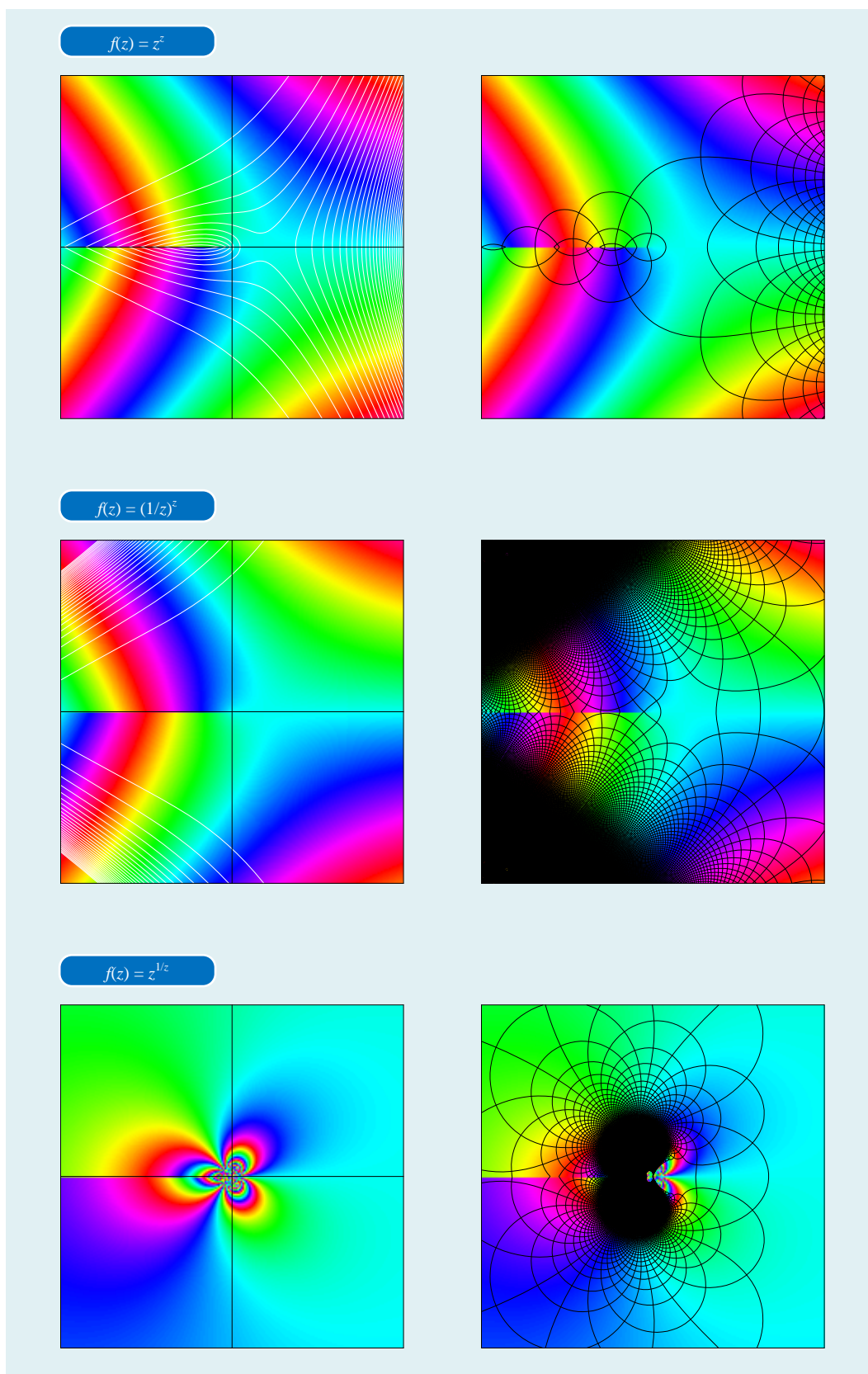


图 12. 乘幂 | Bk2\_Ch24\_02.ipynb

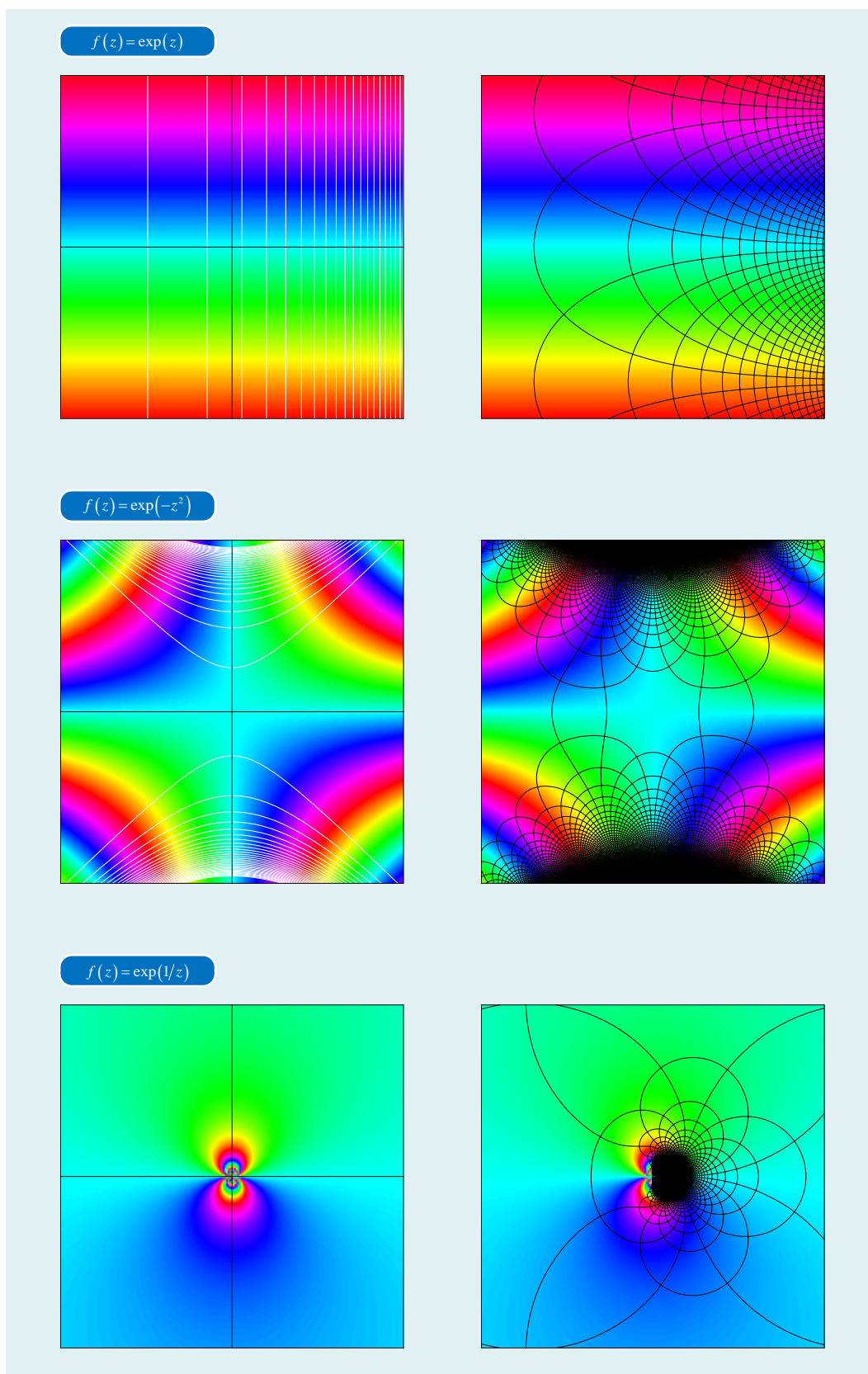

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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](mailto:jiang.visualize.ml@gmail.com)

图 13. 指数 |  Bk2\_Ch24\_02.ipynb

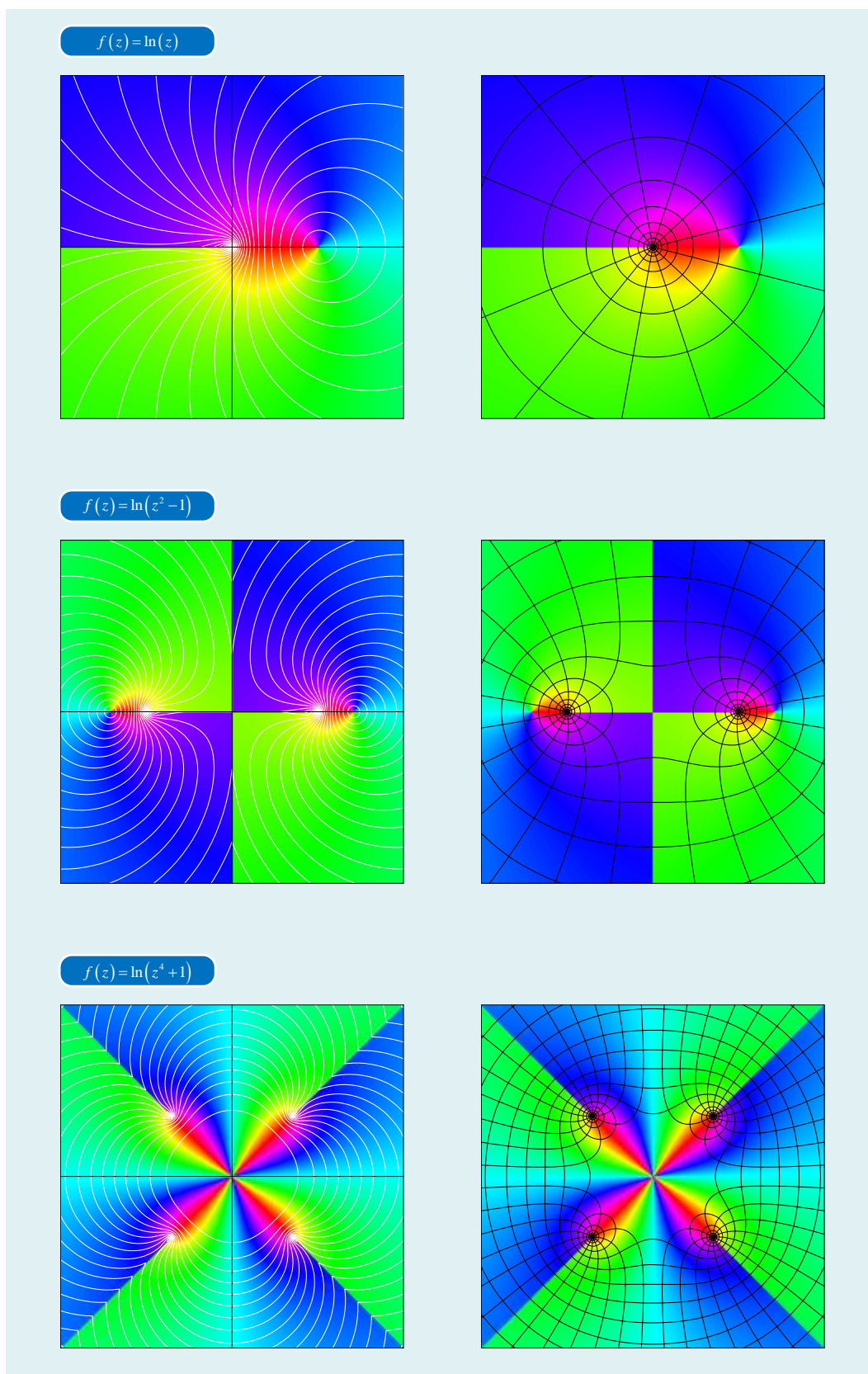

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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](mailto:jiang.visualize.ml@gmail.com)

图 14. 对数 |  Bk2\_Ch24\_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](mailto:jiang.visualize.ml@gmail.com)