

30

Function
函数

可视化一元、二元、三元函数



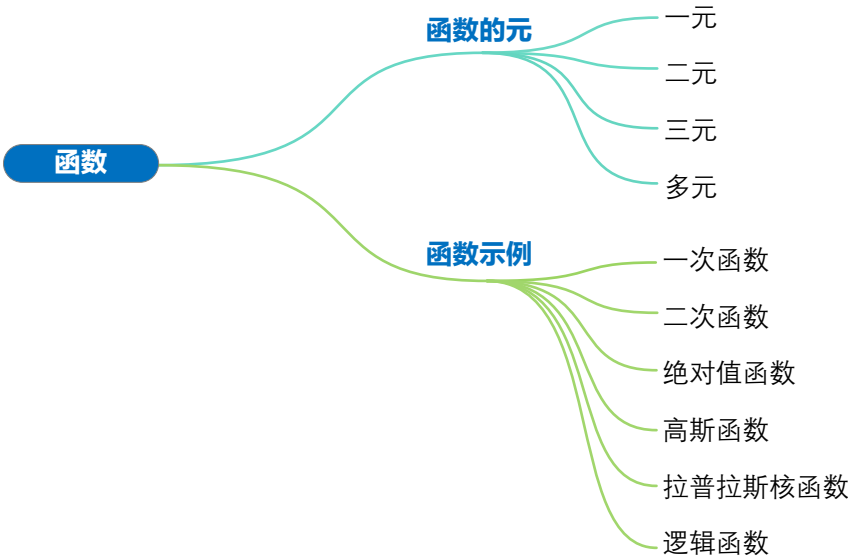
至暗深夜也必然结束，太阳终将升起。

Even the darkest night will end and the sun will rise.

—— 雨果 (Victor Hugo) | 法国文学家 | 1802 ~ 1885



- ◀ matplotlib.pyplot.contour() 绘制等高线图
- ◀ matplotlib.pyplot.contourf() 绘制填充等高线图
- ◀ matplotlib.pyplot.plot_wireframe() 绘制线框图
- ◀ numpy.linspace() 在指定的间隔内, 返回固定步长的数据
- ◀ numpy.meshgrid() 创建网格化数据



30.1 函数

在数学中，函数的**元** (arity) 指的是函数接受的参数个数。

常见的函数元数包括：

一元函数 (unary function) 接受一个参数。例如， $f_1(x) = x$ 是一个一元函数，它接受一个参数 x 。

二元函数 (binary function) 接受两个参数。例如， $f_2(x_1, x_2) = x_1 + x_2$ 是一个二元函数，它接受两个参数 x_1 和 x_2 。

三元函数 (ternary function) 接受三个参数。例如， $f_3(x_1, x_2, x_3) = x_1 + x_2 + x_3$ 是一个三元函数，它接受三个参数 x_1 、 x_2 和 x_3 。

多元函数 (n -ary function) 接受 n 个参数。多元函数的参数个数可以是任意多个，例如 $f_n(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$ 是一个多元函数，它接受任意 n 个参数 x_1 、 x_2 、 \dots 、 x_n 。



鸢尾花书《编程不难》第 8 章简述函数概念，请大家回顾。

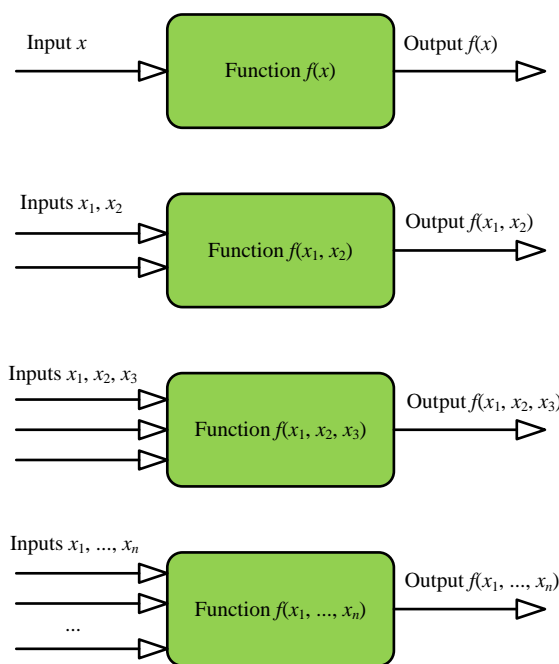


图 1. 一元、二元、三元、多元函数的映射

下面我们就来聊聊如何利用本书前文介绍的各种可视化方案展示一元、二元、三元函数。

30.2 一次函数

一元**一次函数** (linear function) 是一种形式为 $f(x) = ax + b$ 的函数, a 和 b 是实数常数。它是最简单的线性函数, 描述了一个直线的斜率和截距。图 3、图 4 所示为一元、二元、三元一次函数的几种可视化方案。



请大家思考如何可视化四元函数, 比如 $f(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3 + x_4$?

一次函数在数学和实际应用中非常有用。在统计学中, 它被广泛用于线性回归分析, 其中通过拟合一条直线来描述自变量与因变量之间的关系。这可以帮助我们预测和解释数据的变化趋势。

此外, 一次函数还在分类决策面中扮演重要角色。在机器学习和模式识别中, 一次函数可以用来划分特征空间, 将数据点分为不同的类别。



鸢尾花书《数学要素》将专门介绍一次函数。

Bk2_Ch30_01.ipynb 绘制图 3 中一元一次函数 $f(x) = x$ 。代码很简单, 下面聊聊其中可视化语句。

- a 在轴对象 `ax` 上用 `plot()` 绘制一元函数线图, `x1_array` 为横坐标, `f1_array` 为纵坐标。
- b 将坐标轴的纵横比设置为相等。
- c 隐藏四个图脊 `spines`。

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

a ax.plot(x1_array, f1_array)

ax.set_xlabel('x'); ax.set_ylabel('f(x)')
ax.set_xticks(ticks_array); ax.set_yticks(ticks_array);
b ax.set_aspect('equal', adjustable='box')
ax.set_xlim(x1_array.min(), x1_array.max());
ax.set_ylim(x1_array.min(), x1_array.max());
c ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.grid(linestyle='--', linewidth=0.25, color=[0.5, 0.5, 0.5])
plt.show()
```

代码 1. 可视化一元一次函数 | Bk2_Ch30_01.ipynb

Bk2_Ch30_02.ipynb 绘制图 3 中二元一次函数。代码 2 绘制网格面, 下面聊聊其中关键语句。

- a 在图形对象 `fig` 上用 `add_subplot(projection = '3d')` 方法增加一个三维轴对象。
- b 在三维轴对象 `ax` 上用 `plot_wireframe()` 绘制网格面, 展示二元函数 $f(x_1, x_2) = x_1 + x_2$ 。参数 `rstride` 指定网格数据行之间的跨度, 表示在行的方向上将线绘制多少行之后跳过一行。
`cstride` 指定网格数据列之间的跨度, 表示在列的方向上将线绘制多少列之后跳过一列。

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

请大家尝试将、设置为其他正整数，并观察效果。

`linewidth` 指定网格面线宽，即线的粗细。

c 利用 `set_proj_type()` 设置三维图形的投影方式为正交投影。本书前文介绍过正交投影，简单来说，正交投影不考虑“近大远小”这种视觉效果。

d 利用 `view_init()` 设定视角。参数 `elev` 为仰角，参数 `azim` 为方位角。

```
fig = plt.figure(figsize=(5,5))
a ax = fig.add_subplot(projection='3d')

b ax.plot_wireframe(xx1, xx2, f2_array,
                   rstride=1, cstride=1,
                   linewidth = 1)

ax.set_xlabel('$x_1$'); ax.set_ylabel('$x_2$')
ax.set_zlabel('$f(x_1, x_2)$'); ax.grid(False)
c ax.set_proj_type('ortho')
ax.set_xticks(ticks_array); ax.set_yticks(ticks_array)
d ax.view_init(azim=-120, elev=30)
ax.set_xlim(xx1.min(), xx1.max()); ax.set_ylim(xx2.min(), xx2.max())
plt.tight_layout()
```

代码 2. 可视化二元一次函数，网格面 | Bk2_Ch30_02.ipynb

代码 2 绘制的网格面，相当于用“经纬线”织布。而代码 3 则只绘制单一维度织线。我们在本书第 20 章介绍过这种可视化方案，请大家回顾。

a 利用 `plot_wireframe()` 时，设置 `cstride=0`，不显示数据 column 列方向织线。

b 利用 `plot_wireframe()` 时，设置 `rstride=0`，不显示数据 row 行方向织线。

```
fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(projection='3d')

# 只绘制row方向织线
a ax.plot_wireframe(xx1, xx2, f2_array,
                   rstride=1, cstride=0)

fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(projection='3d')

# 只绘制column方向织线
b ax.plot_wireframe(xx1, xx2, f2_array,
                   rstride=0, cstride=1)
```

代码 3. 可视化二元一次函数，网格面单一维度织线 | Bk2_Ch30_02.ipynb

代码 4 采用“网格面 + 三维等高线”方式可视化二元函数。请大家自行分析代码。

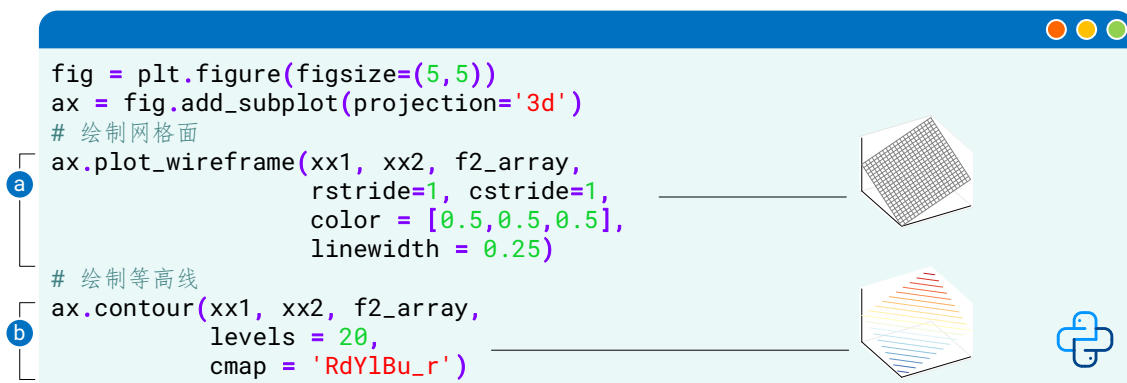

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

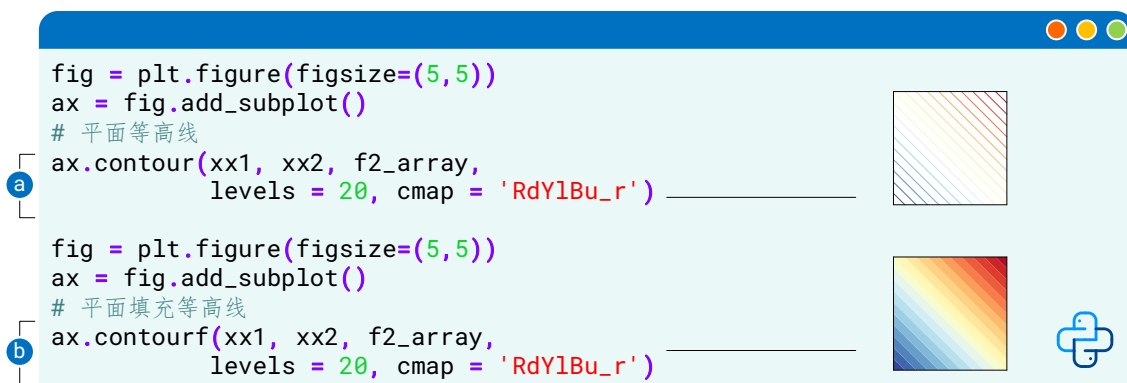

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

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

代码 4. 可视化二元一次函数，网格面 + 三维等高线 |  Bk2_Ch30_02.ipynb

代码 5 分别采用平面等高线、平面填充等高线可视化二元函数，请大家自行分析。
Bk2_Ch30_02.ipynb 还给出其他几种可视化方案，请大家自行分析。

代码 5. 可视化二元一次函数，平面等高线，平面填充等高线 |  Bk2_Ch30_02.ipynb

Bk2_Ch30_03.ipynb 主要用“切豆腐”方法可视化三元函数，我们已经在本书前文介绍过相关代码，请大家自行分析。

图 2 所示利用 Plotly 的 volume 图（体积图）可视化三元函数。在 JupyterLab 中，大家可以旋转、缩放这幅图。

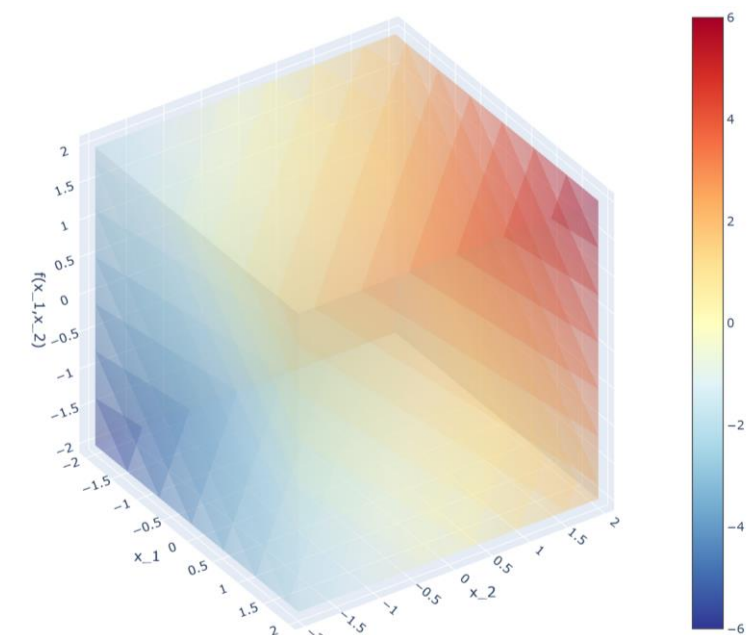



图 2. 用 Plotly 的 volume 图可视化三元一次函数 |  Bk2_Ch30_03.ipynb

下面，我们聊聊绘制图 2 的核心代码。

a 用 `plotly.graph_objects.Volume()`，简作 `go.Volume()`，绘制体积图。

`x=xxx1.flatten()` 表示 x 轴上的坐标数据。其中，`xxx1.flatten()` 是将 `xxx1` 中的三维数组展平成一维数组。

`y=xxx2.flatten()` 表示 y 轴上的坐标数据。

`z=xxx3.flatten()` 表示 z 轴上的坐标数据。

`value=f3_array.flatten()` 表示体积图上每个点的三元函数数值。

`isomin=f3_array.min()` 指定体积图的数据范围的下限，对应三元函数最小值。

`isomax=f3_array.max()` 指定体积图的数据范围的上限，对应三元函数最大值。

`opacity=0.25` 设置体积图的透明度，`0.25` 表示相对较透明。

`colorscale='RdYlBu_r'` 指定体积图的颜色映射。

`surface_count=20` 表示体积图中显示的曲面数量。

b 用 `plotly.graph_objects.Layout()`，简作 `go.Layout()`，设置图形属性。

`width=800` 设定图形的宽度为 800 像素。

`height=800` 设定图形的高度为 800 像素。

`scene=dict(...)` 对图形中场景的设置，包含了 x 轴、 y 轴、 z 轴的标题等信息。

`xaxis_title='x_1'` 设置 x 轴的标题。

`yaxis_title='x_2'` 设置 y 轴的标题。

`zaxis_title='f(x_1,x_2)'` 设置 z 轴的标题。

c 用 `plotly.graph_objects.Figure()`，简作 `go.Figure()`，创建图形对象。

`data=data` 指定图形的数据，即之前创建的体积图数据。

`layout=layout` 指定图形的布局。

d 设置投影方式为正交投影。

e 和 **f** 设置视角位置。

```

import plotly.graph_objects as go
data=go.Volume(
    x=xxx1.flatten(),
    y=xxx2.flatten(),
    z=xxx3.flatten(),
    value=f3_array.flatten(),
    isomin=f3_array.min(),
    isomax=f3_array.max(),
    opacity=0.25,
    colorscale = 'RdYlBu_r',
    surface_count=20)

# 布局设置
layout = go.Layout(
    width=800, height=800,
    scene=dict(
        xaxis_title='x_1',
        yaxis_title='x_2',
        zaxis_title='f(x_1,x_2)'))

# 创建图形对象
fig = go.Figure(data=data, layout=layout)
fig.layout.scene.camera.projection.type = "orthographic"
camera = dict(eye=dict(x=0.6, y=-1, z=0.5))
fig.update_layout(scene_camera=camera)
fig.show()

```

代码 6. 用 Plotly 的 volume | Bk2_Ch30_03.ipynb

30.3 其他几个函数示例

请大家自行修改 Bk2_Ch30_01.ipynb、Bk2_Ch30_02.ipynb、Bk2_Ch30_03.ipynb，自行绘制如下几个函数的可视化方案。

二次函数

一元**二次函数** (quadratic function) 是一种形式为 $f(x) = ax^2 + bx + c$ 的函数，其中 a 、 b 、 c 是实数常数且 a 不为零。图 5、图 6 所示为一元、二元、三元二次函数的几种可视化方案。

二次函数在数学和实际应用中有着广泛的应用。在数学领域，二次函数是研究代数和解析几何的重要对象，它们具有丰富的性质和特征。

在实际应用中，二次函数可以用于建模和预测各种现象。例如，在物理学中，二次函数可以描述自由落体运动的高度随时间变化的关系。此外，二次函数还用于优化问题和曲线拟合。

例如，在最优化领域，通过分析二次函数的性质，可以找到最小值或最大值的位置；在曲线拟合中，可以利用二次函数来逼近实际数据，并进行预测和插值。



鸢尾花书《数学要素》将专门介绍二次函数。

绝对值函数

绝对值函数 (absolute value function) 是一种形式为 $f(x) = a|x - b|$ 的函数, a 和 b 是实数常数。图 7、图 8 所示为一元、二元、三元绝对值函数的几种可视化方案。

绝对值函数在数学和实际应用中都具有重要作用。在数学中, 绝对值函数是一种特殊的分段函数, 具有简单的图像和性质。它在不等式和绝对值相关的问题时经常出现。

绝对值函数可以用于测量误差或距离的绝对值, 使得结果不受正负号影响。在优化问题中, 绝对值函数常用于表示目标函数或约束条件, 帮助找到最优解。此外, 绝对值函数还在统计学和数据分析中发挥作用。它可以用于计算绝对误差或残差, 评估模型的拟合程度。



鸢尾花书《数学要素》将专门介绍不同绝对值函数。

高斯函数

图 9、图 10 所示为一元、二元、三元**高斯函数** (Gaussian function) 的几种可视化方案。

高斯函数常用于高斯分布, 因此高斯函数在统计学和概率论中广泛应用。高斯函数在数据分析和模型拟合中非常有用。它可以用来描述数据集的分布情况, 并进行概率推断和统计推断。

在机器学习中, 高斯函数常用于聚类算法、异常检测和生成模型。此外, 高斯函数还在信号处理中广泛应用, 如图像处理和滤波。它可以用来平滑信号、降噪和边缘检测。



鸢尾花书《统计至简》将专门介绍高斯分布。

拉普拉斯核函数

拉普拉斯核函数 (Laplacian kernel function) 是一种常用的非线性核函数, 用于**支持向量机** (Support Vector Machine, SVM) 和其他机器学习算法中的非线性分类和回归任务。图 11、图 12 所示为一元、二元、三元拉普拉斯核函数的几种可视化方案。



鸢尾花书《机器学习》将专门介绍支持向量机。

拉普拉斯核函数可以将数据映射到高维特征空间, 从而在低维空间中实现非线性分类。与其他核函数相比, 拉普拉斯核函数在决策边界附近具有更陡峭的变化, 因此对异常点更敏感, 使其能够更好地捕捉数据中的局部结构。

拉普拉斯核函数在图像识别、文本分类、生物信息学等领域广泛应用。它具有良好的分类性能, 并且在处理高维数据和处理小样本问题时表现出色。

逻辑函数

逻辑函数 (logistic function) 是一种常用的非线性函数, 也称为 Sigmoid 函数或逻辑曲线。逻辑函数的取值范围在 0 到 1 之间, 具有平滑的 S 形曲线。图 13、图 14 所示为一元、二元、三元逻辑函数的几种可视化方案。

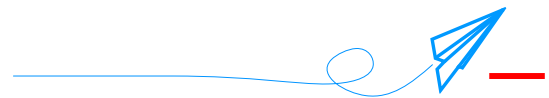
逻辑函数在机器学习、神经网络和统计学中广泛应用。它主要用于将输入值映射到概率值，将连续变量转换为概率分布。在二元分类任务中，逻辑函数可以将输入的线性组合转化为 0 和 1 之间的概率值，用于判断样本属于某一类别的概率。

逻辑函数在逻辑回归模型中扮演重要角色，用于建立分类模型和进行概率预测。它能够对数据进行非线性建模，拟合复杂的分类决策边界。

此外，逻辑函数还用于激活函数，如在神经网络中，逻辑函数可用于将神经元的输出转化为非线性响应，提高模型的表达能力。



鸢尾花书《数据有道》将专门介绍逻辑回归。



函数是代数、机器学习中极为重要的数学概念。通常，我们看到最多的是一元函数的图像；本章利用各种可视化方案，向大家展示二元、三元函数图像。

下一章，我们还会用类似本章的可视化方案展示二次型，这个在优化问题、机器学习算法中重要的数学概念。

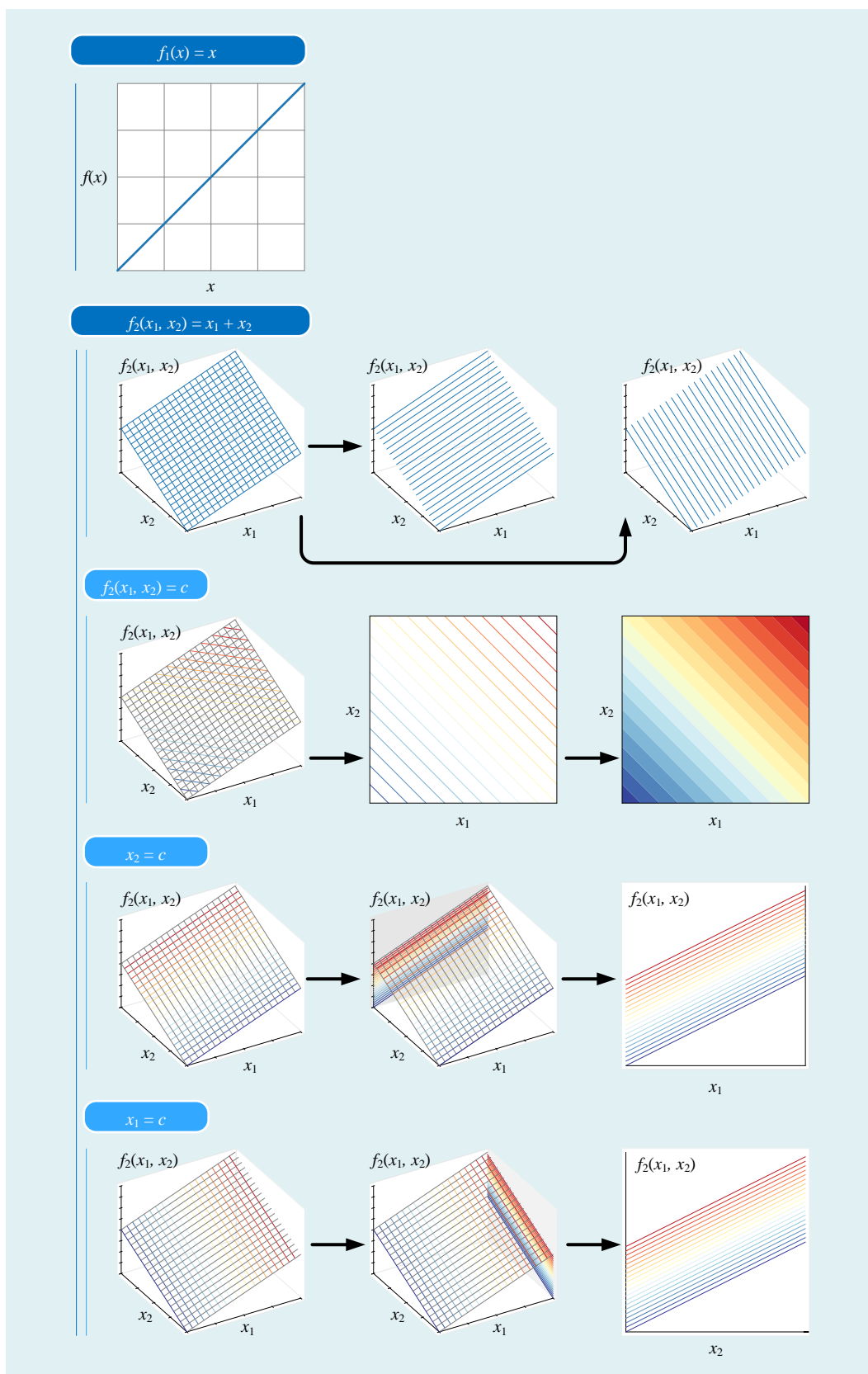



图 3. 一次函数，一元、二元 |  Bk2_Ch30_01.ipynb | Bk2_Ch30_03.ipynb

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

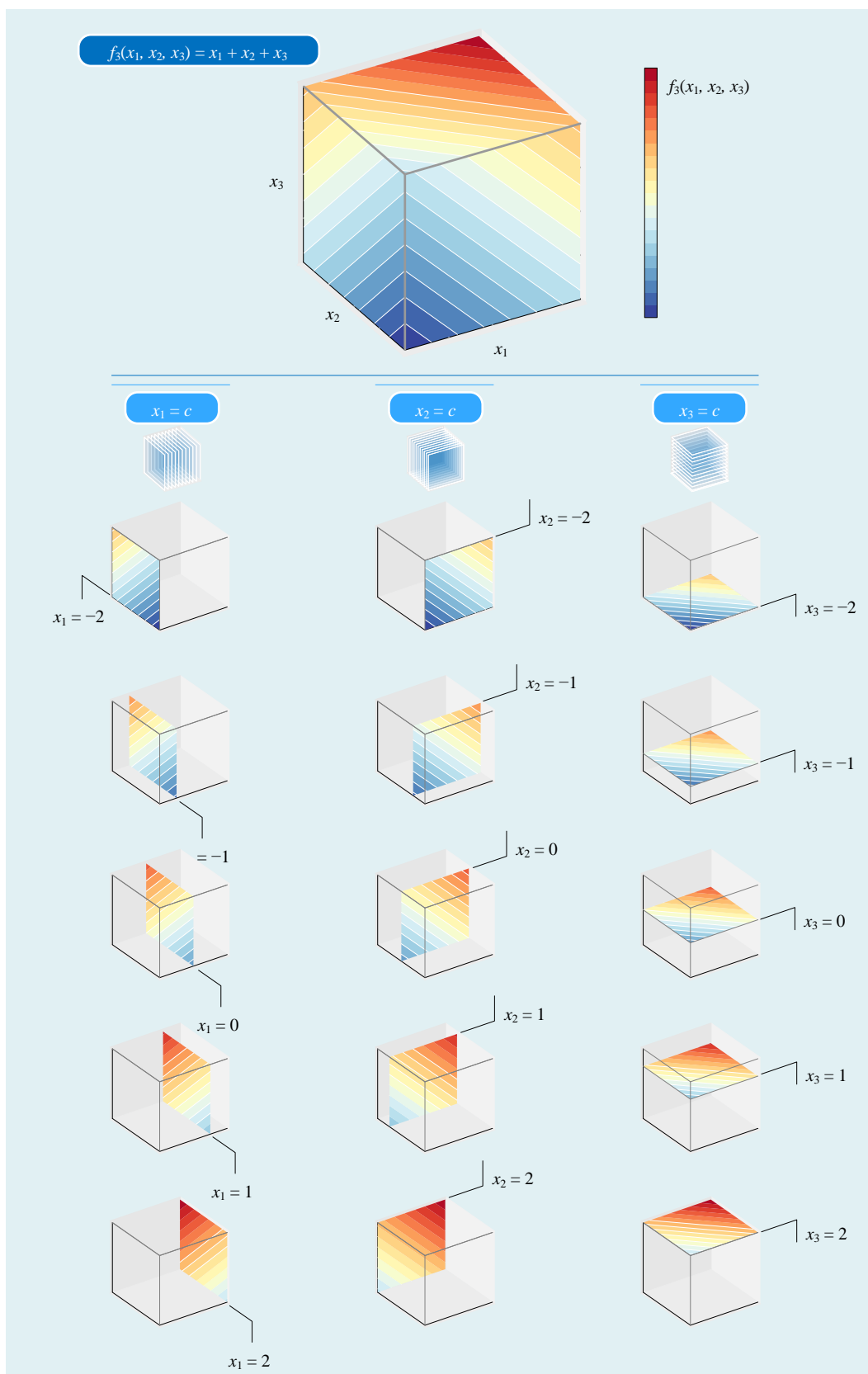


图 4. 一次函数, 三元 | Bk2_Ch30_03.ipynb

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

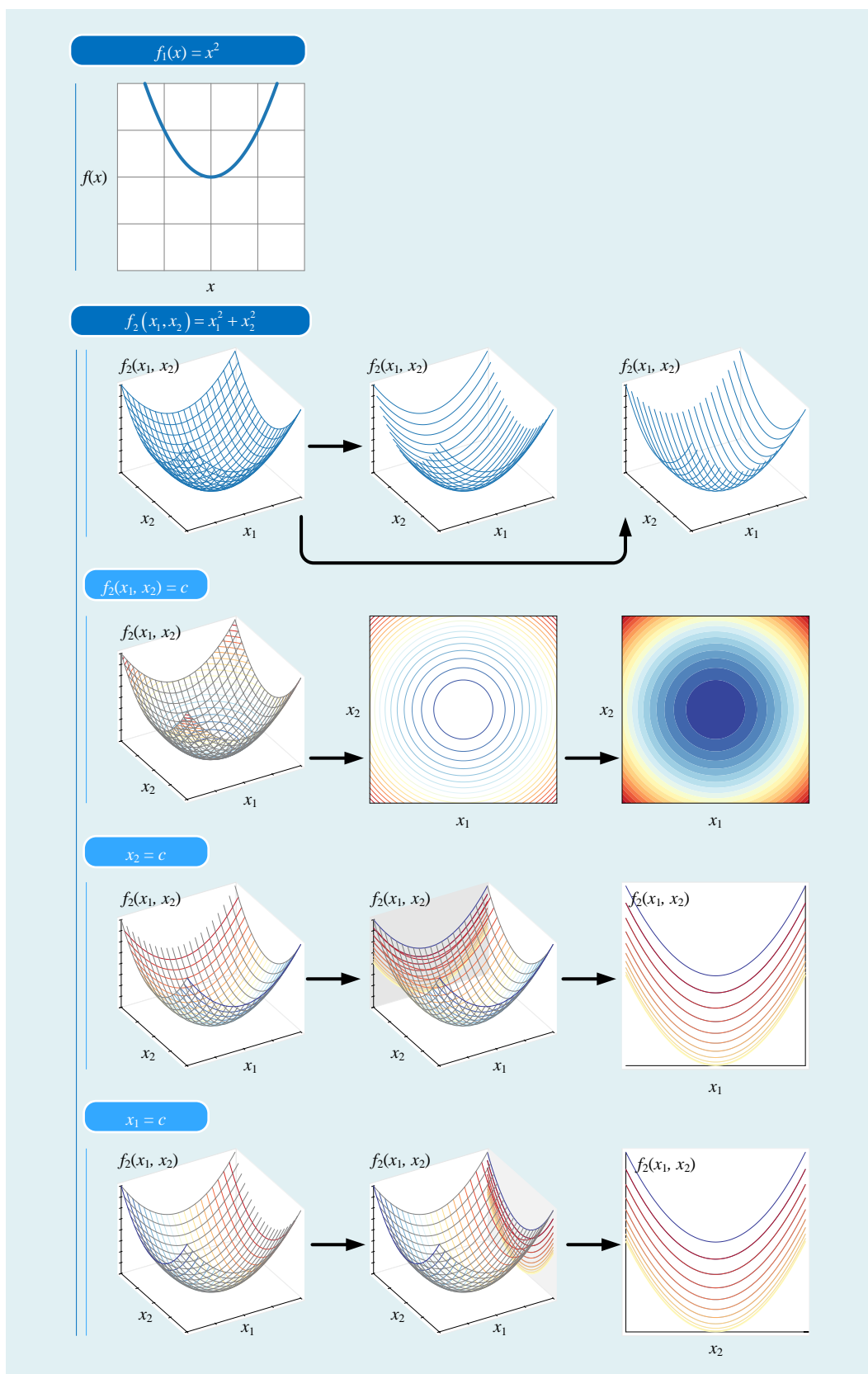


图 5. 二次函数，一元、二元

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

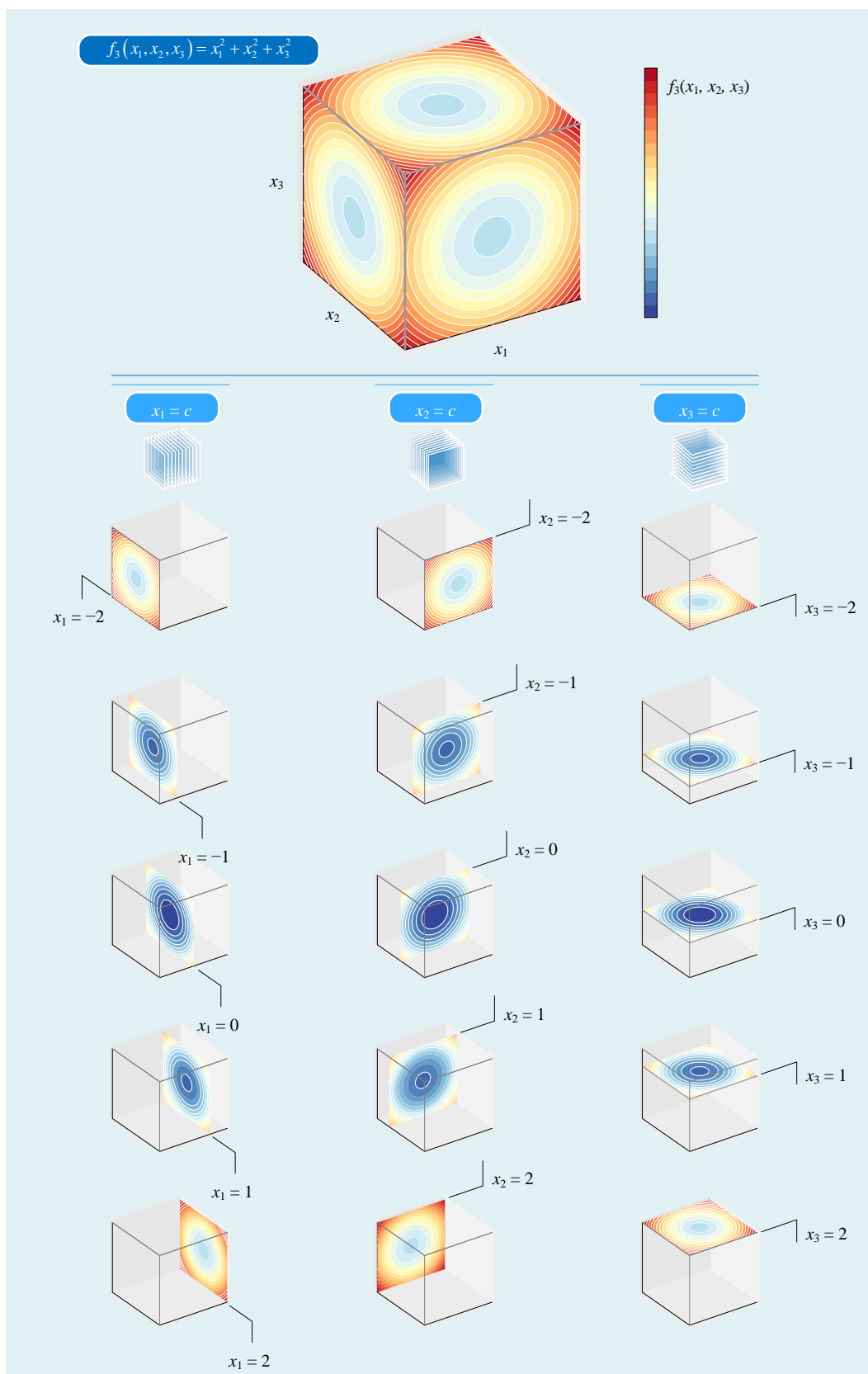


图 6. 二次函数, 三元

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

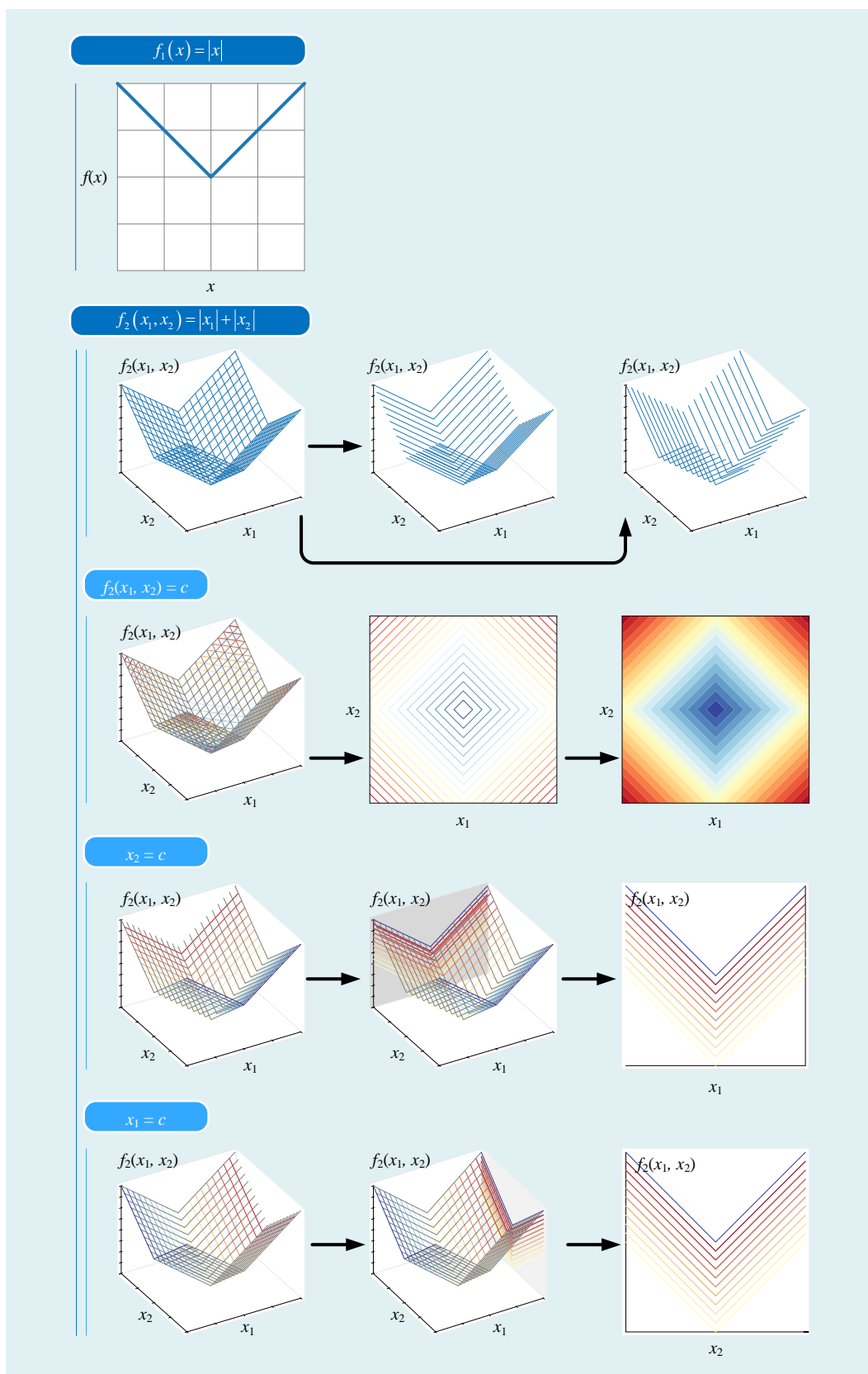


图 7. 绝对值函数, 一元、二元

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

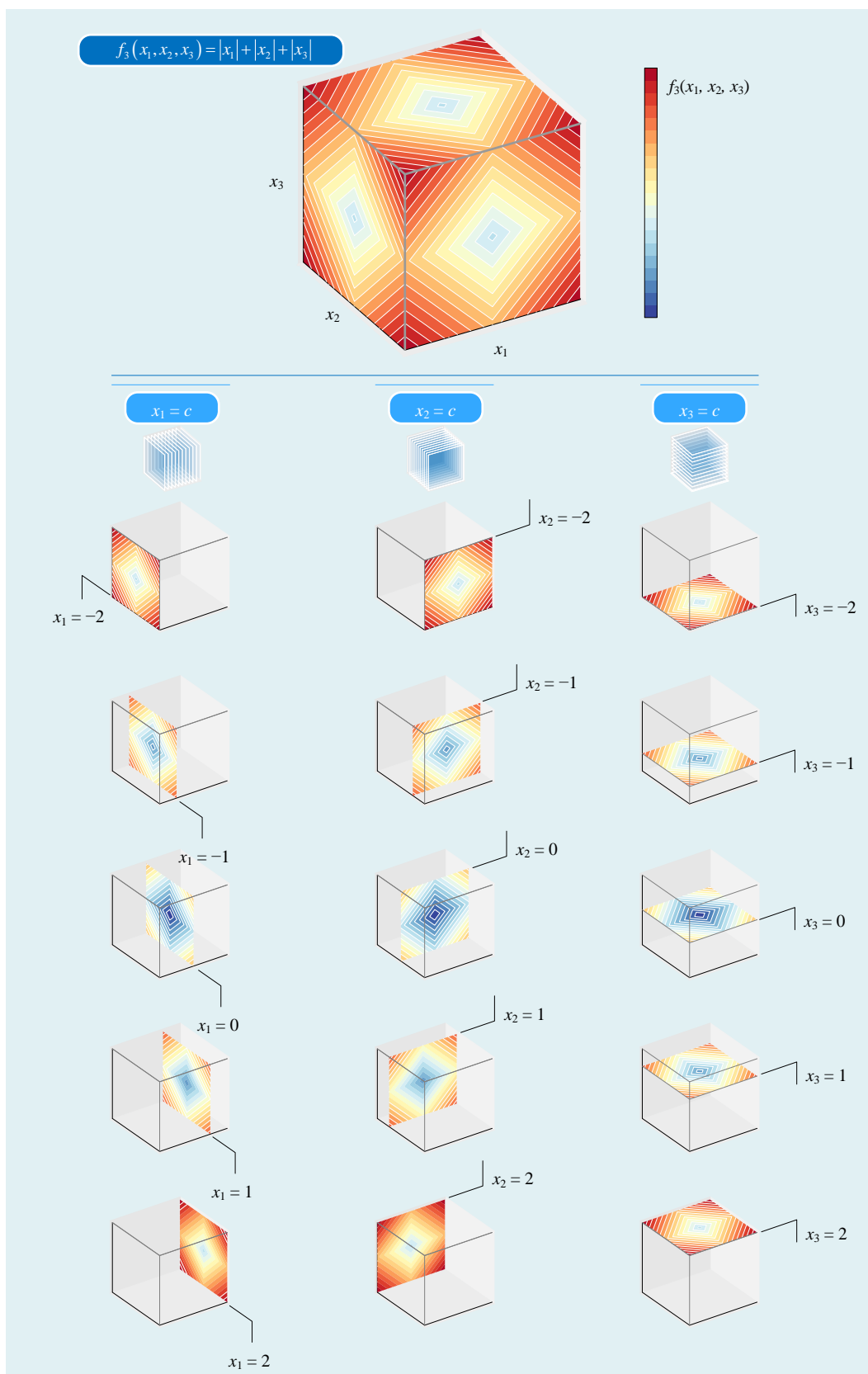


图 8. 绝对值函数, 三元

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

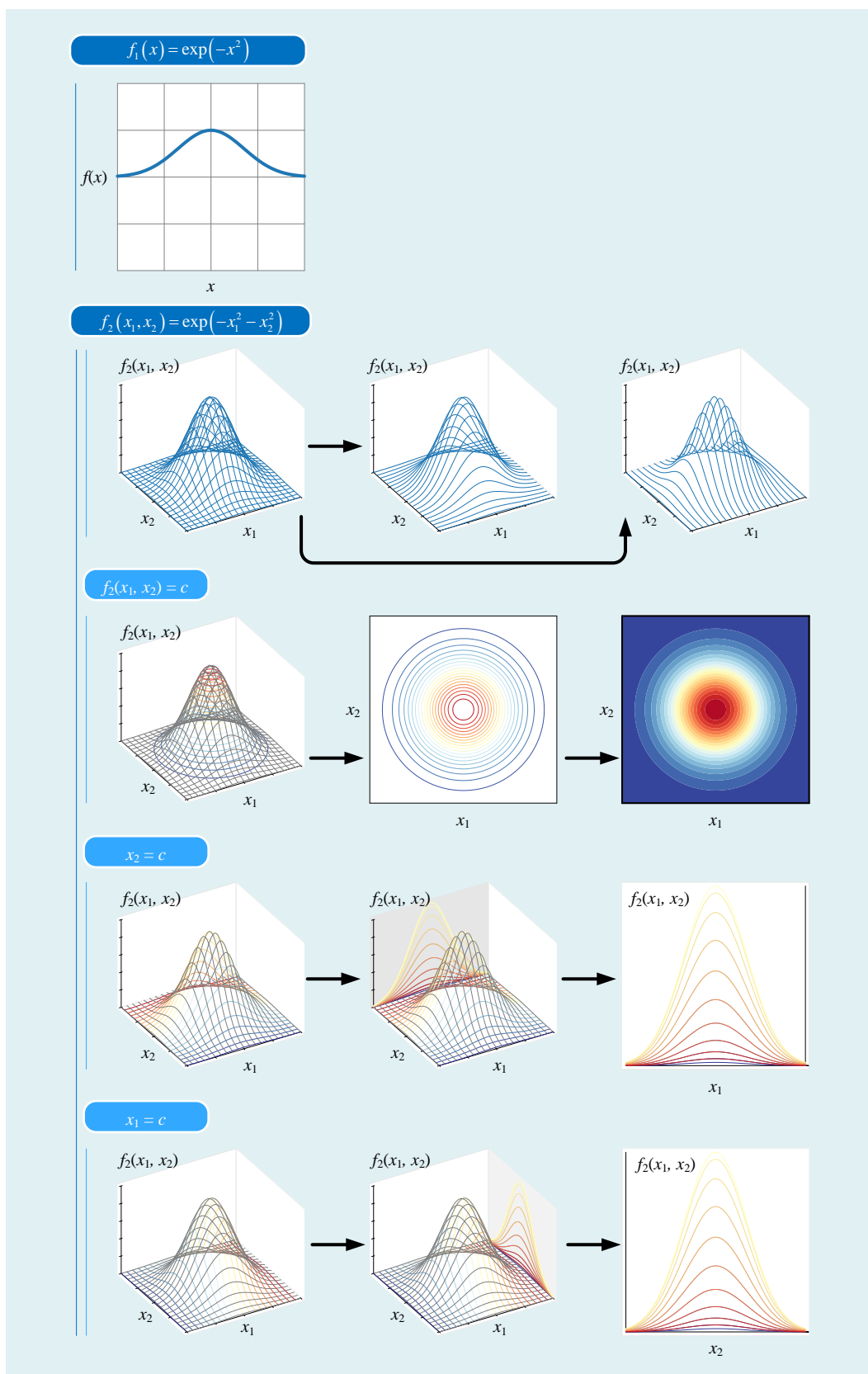


图 9. 高斯函数，一元、二元

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

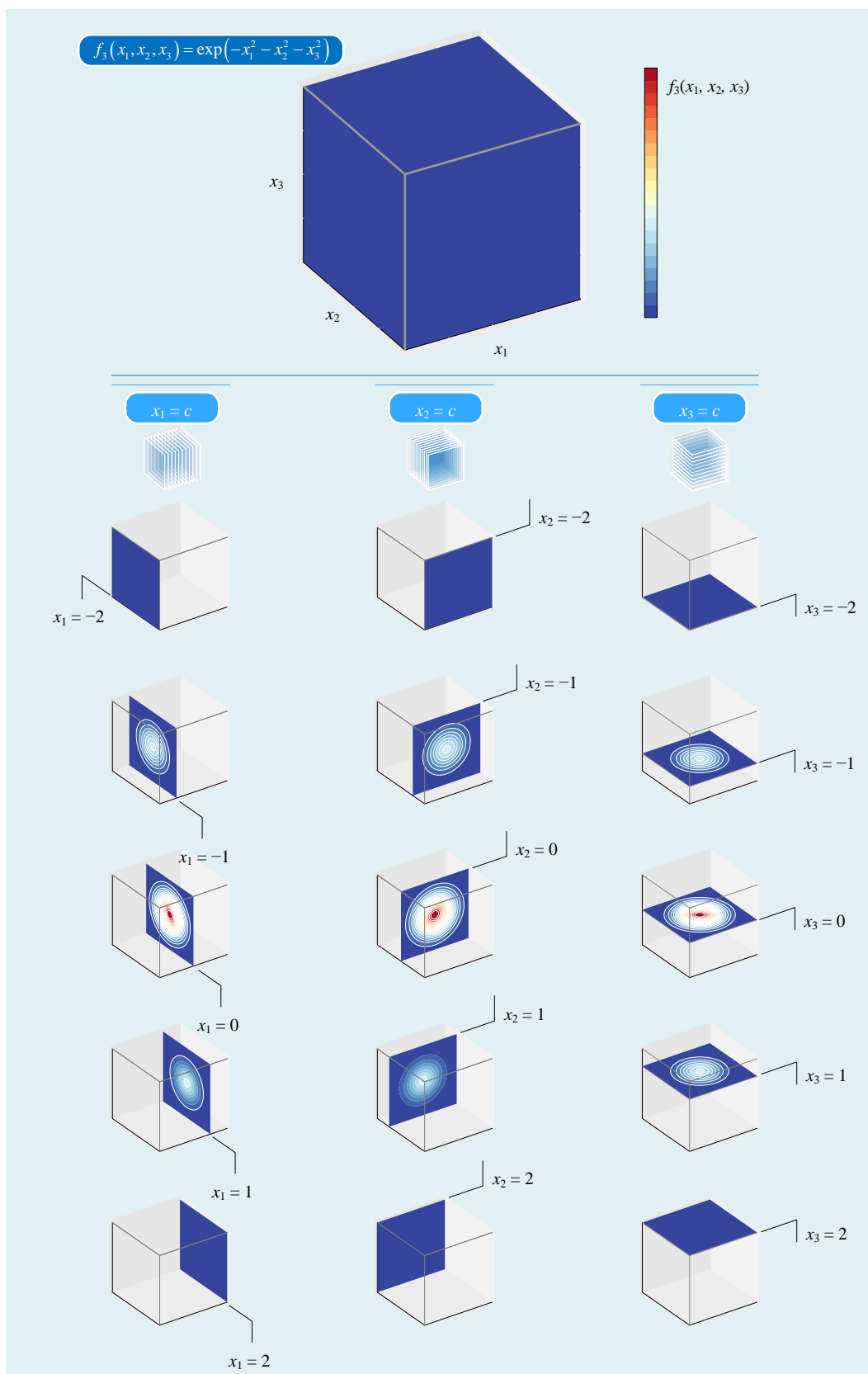


图 10. 高斯函数，三元

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

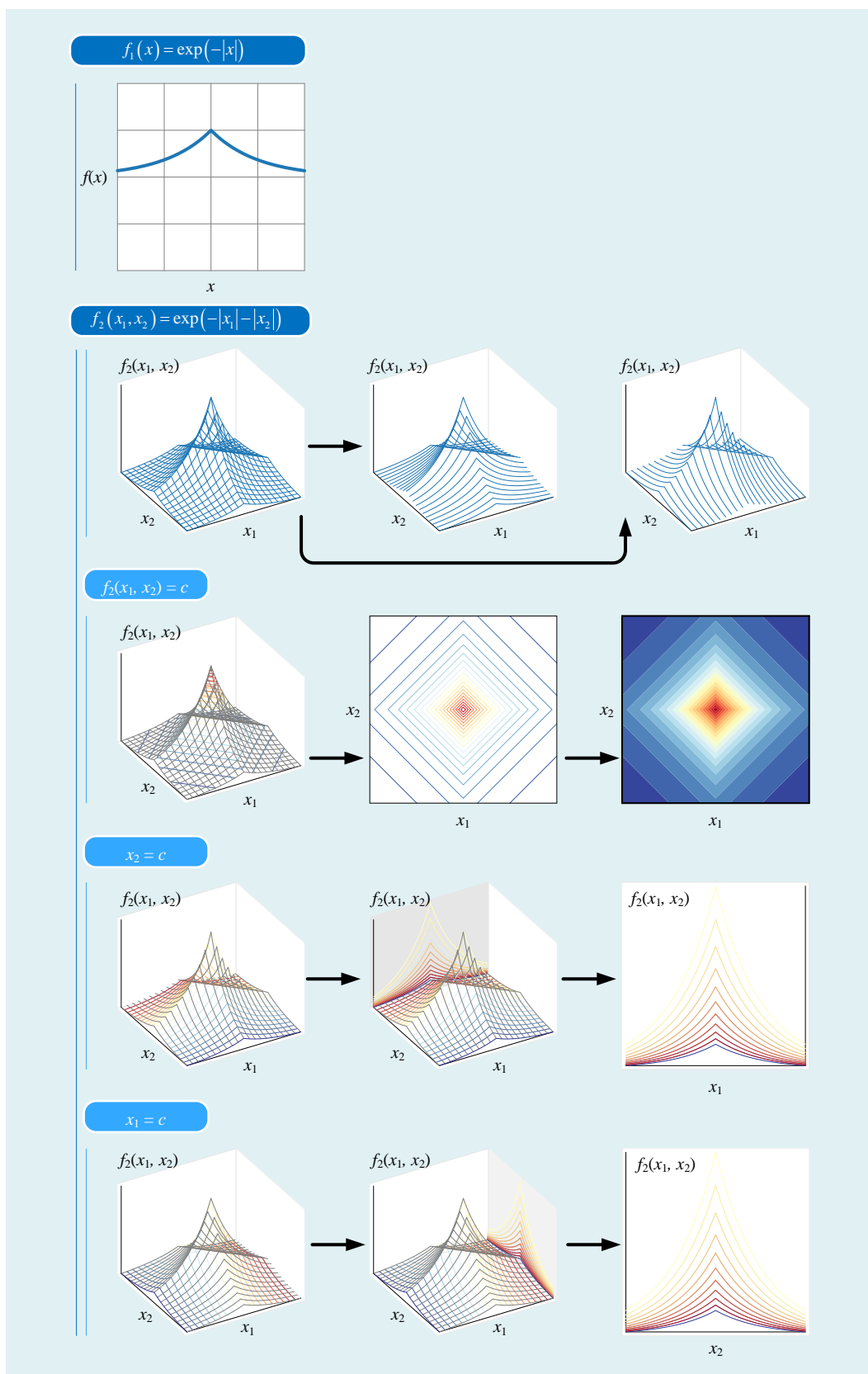


图 11. 拉普拉斯核函数，一元、二元

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

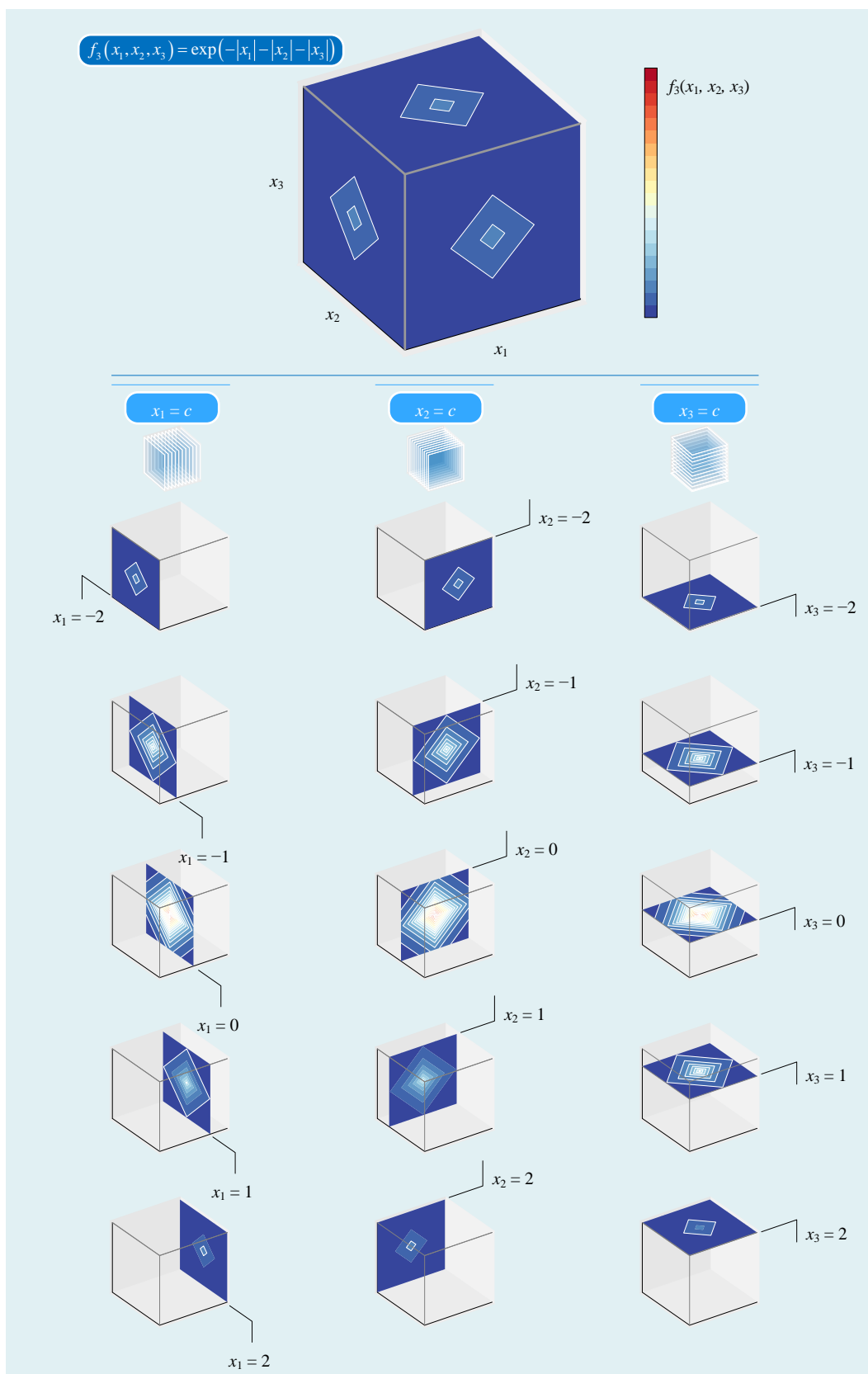


图 12. 拉普拉斯核函数，三元

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

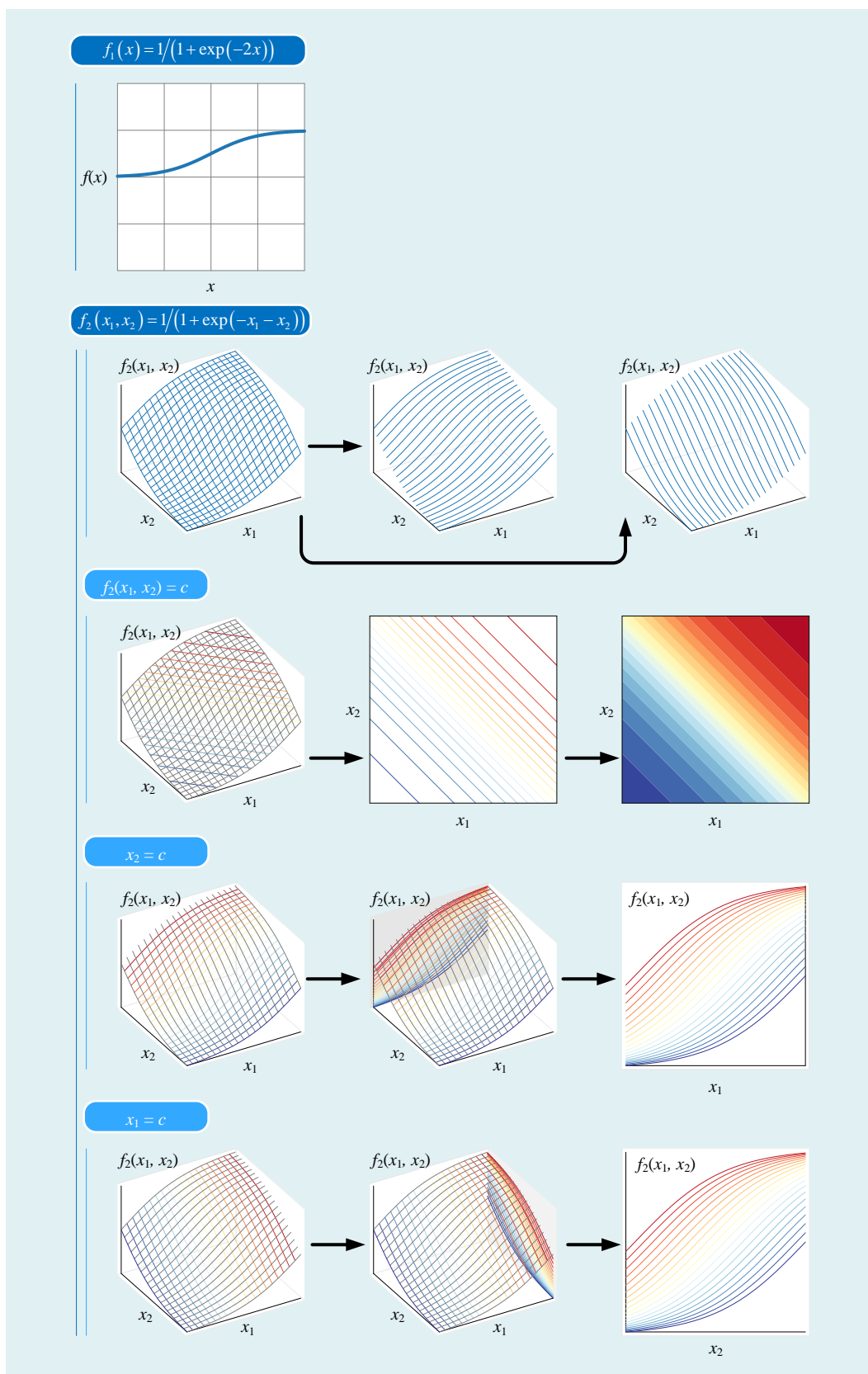


图 13. 逻辑函数，一元、二元

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

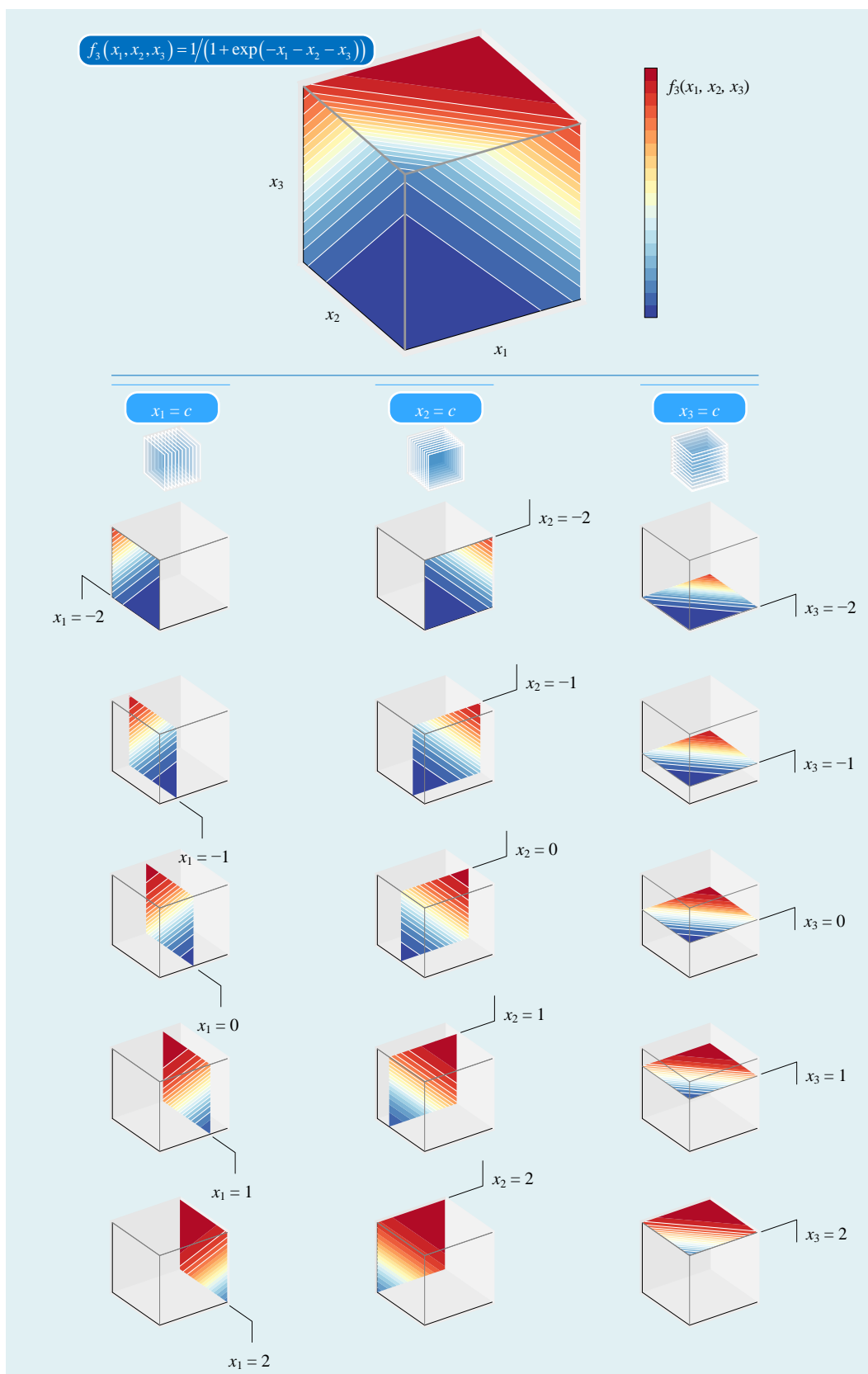


图 14. 逻辑函数，三元

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