

21

Quadratic Form

二次型

可视化二元、三元二次型正定性



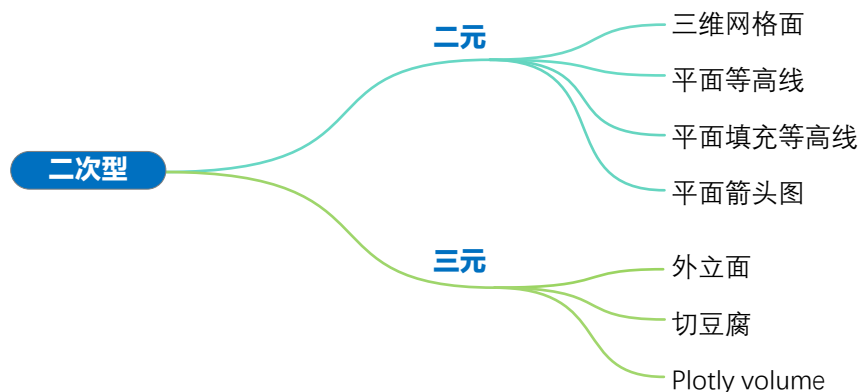
别怕完美，因为我们永远也达不到。

Have no fear of perfection - you'll never reach it.

—— 萨尔瓦多·达利 (Salvador Dali) | 西班牙超现实主义画家 | 1904 ~ 1989



- ◀ 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
- ◀ matplotlib.pyplot.quiver() 绘制箭头图
- ◀ 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
- ◀ sympy.diff() 求解符号导数和偏导解析式
- ◀ sympy.expand() 展开代数式
- ◀ sympy.lambdify() 将符号表达式转化为函数
- ◀ sympy.simplify() 简化代数式
- ◀ 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

21.1 二元二次型

读过《编程不难》的读者对**二次型** (quadratic form) 这个概念应该不陌生。

简单来说，二次型是一个二次多项式函数，通常表示为 $Q(\mathbf{x}) = \mathbf{x}^T @ \mathbf{A} @ \mathbf{x}$ ，其中 \mathbf{x} 是一个向量 $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ ， $\mathbf{A}_{n \times n}$ 是一个实对称矩阵。

举例来说，对于一元函数， $f(x) = x^2$ 就是一个二次型；对于二元函数， $f(x_1, x_2) = x_1 x_2$ 也是个二次型。

通过**特征值** (eigen value) 和**特征向量** (eigen vector) 的分析，我们揭示矩阵的性质和行为，特别是**正定性** (positive definiteness)。二次型在矩阵分析、优化方法、机器学习中具有重要意义。通过研究二次型，我们可以深入理解多元函数的形式和特点，为解决实际问题提供数学基础。



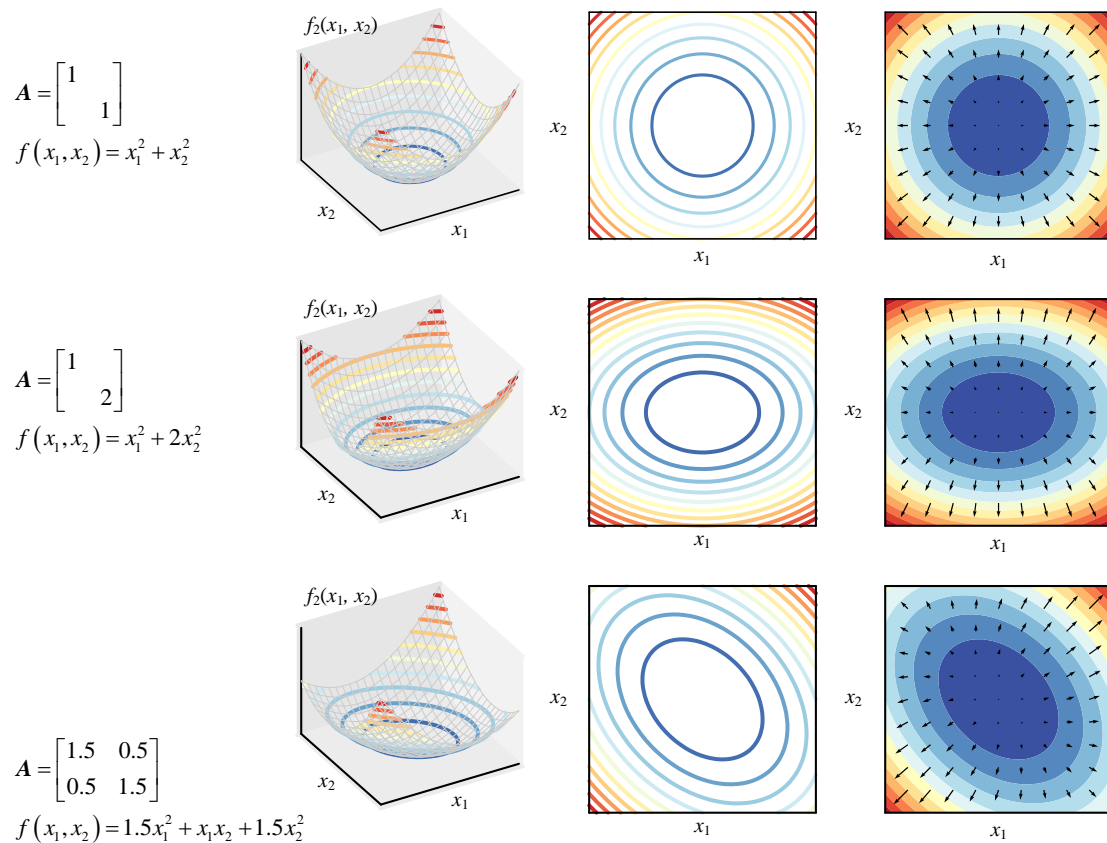
请大家回顾《编程不难》第 25 章有关二次型内容；此外，《矩阵力量》第 21 章将专门讲解二次型。

说到二次型，就必须要聊聊正定性。《编程不难》用图解法介绍过 $\mathbf{A}_{2 \times 2}$ 正定性。下面我们逐一回顾。

正定

如图 1 所示，一个矩阵 $\mathbf{A}_{2 \times 2}$ 是**正定** (positive definite)，意味着 $f(\mathbf{x}) = \mathbf{x}^T @ \mathbf{A}_{2 \times 2} @ \mathbf{x}$ 是个开口朝上的抛物面，形状像是碗。

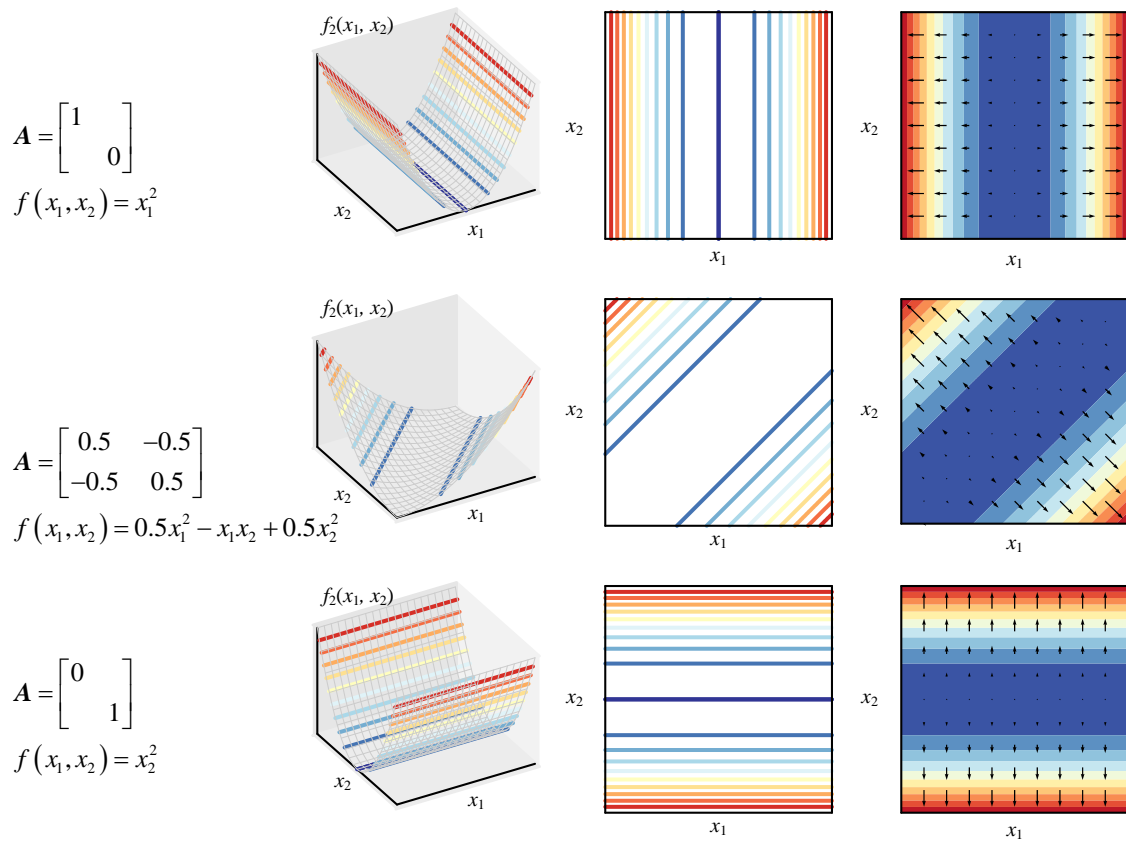
除了 $(0, 0)$ ， $f(\mathbf{x}) = \mathbf{x}^T @ \mathbf{A}_{2 \times 2} @ \mathbf{x}$ 均大于 0。 $(0, 0)$ 为最小值，图中箭头都背离 $(0, 0)$ 。

图 1. 正定, $A_{2 \times 2}$ | Bk2_Ch21_01.ipynb

半正定

如图 2 所示, 一个矩阵 $A_{2 \times 2}$ 是**半正定** (positive semi-definite), 意味着 $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ 是个开口朝上的山谷面。

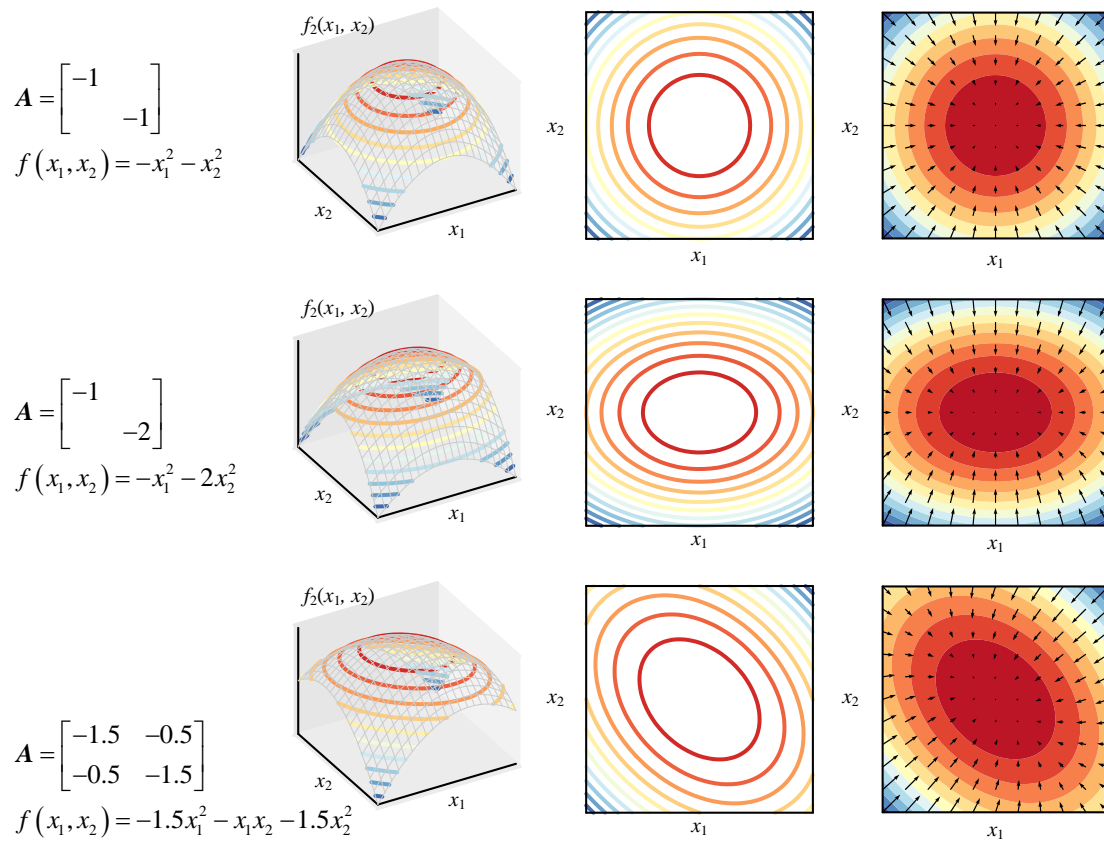
除了 $(0, 0)$, $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ 均大于等于 0。山谷的谷底都是极小值, 图中箭头都背离谷底所在直线。

图 2. 半正定, $A_{2 \times 2}$ | Bk2_Ch21_01.ipynb

负定

如图 3 所示, 一个矩阵 $A_{2 \times 2}$ 是**负定** (negative definite), 意味着 $f(\mathbf{x}) = \mathbf{x}^T @ A_{2 \times 2} @ \mathbf{x}$ 是个开口朝下的抛物面。

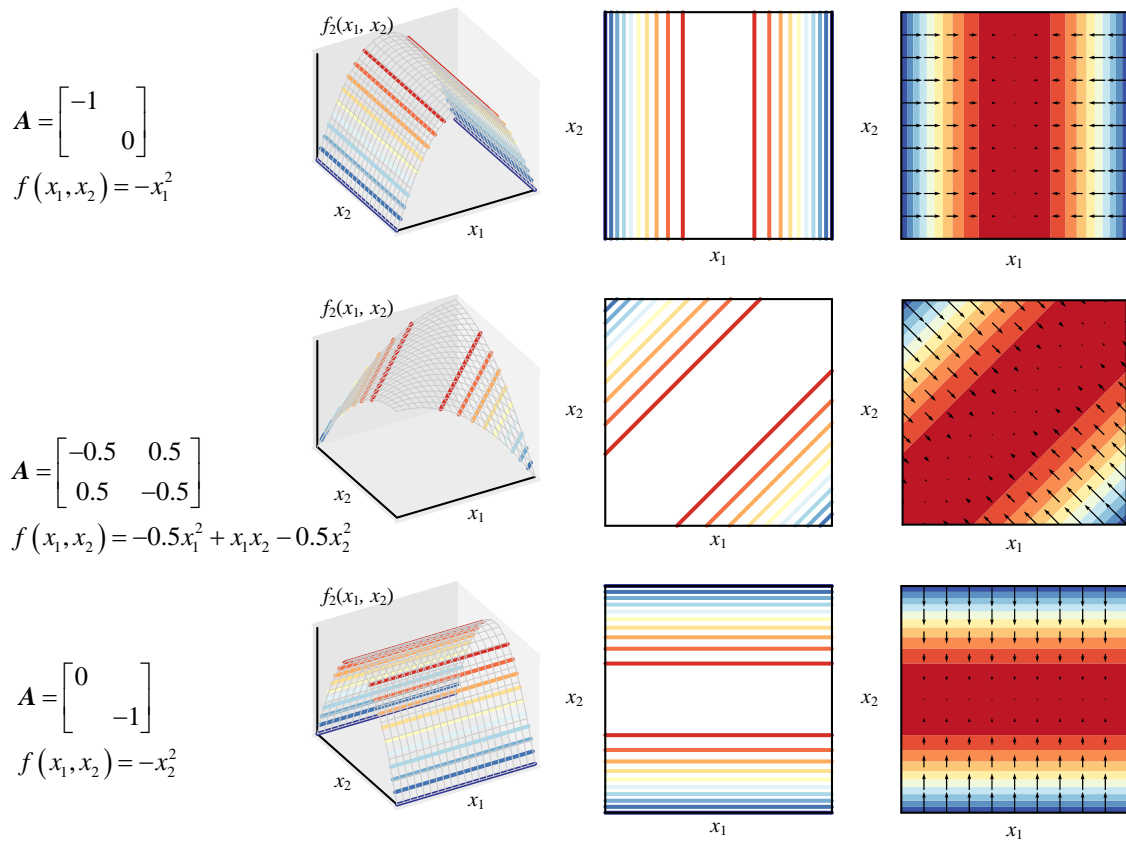
除了 $(0, 0)$, $f(\mathbf{x}) = \mathbf{x}^T @ A_{2 \times 2} @ \mathbf{x}$ 均小于 0。 $(0, 0)$ 为最大值, 图中箭头都指向 $(0, 0)$ 。

图 3. 负定, $A_{2 \times 2}$ | Bk2_Ch21_01.ipynb

半负定

如图 4 所示, 一个矩阵 $A_{2 \times 2}$ 是**半负定** (negative semi-definite), 意味着 $f(x) = x^T @ A_{2 \times 2} @ x$ 是个开口朝下的山脊面。除了 $(0, 0)$, $f(x) = x^T @ A_{2 \times 2} @ x$ 均小于等于 0。

山脊的顶端都是极大值, 图中箭头指向山脊顶端所在直线。

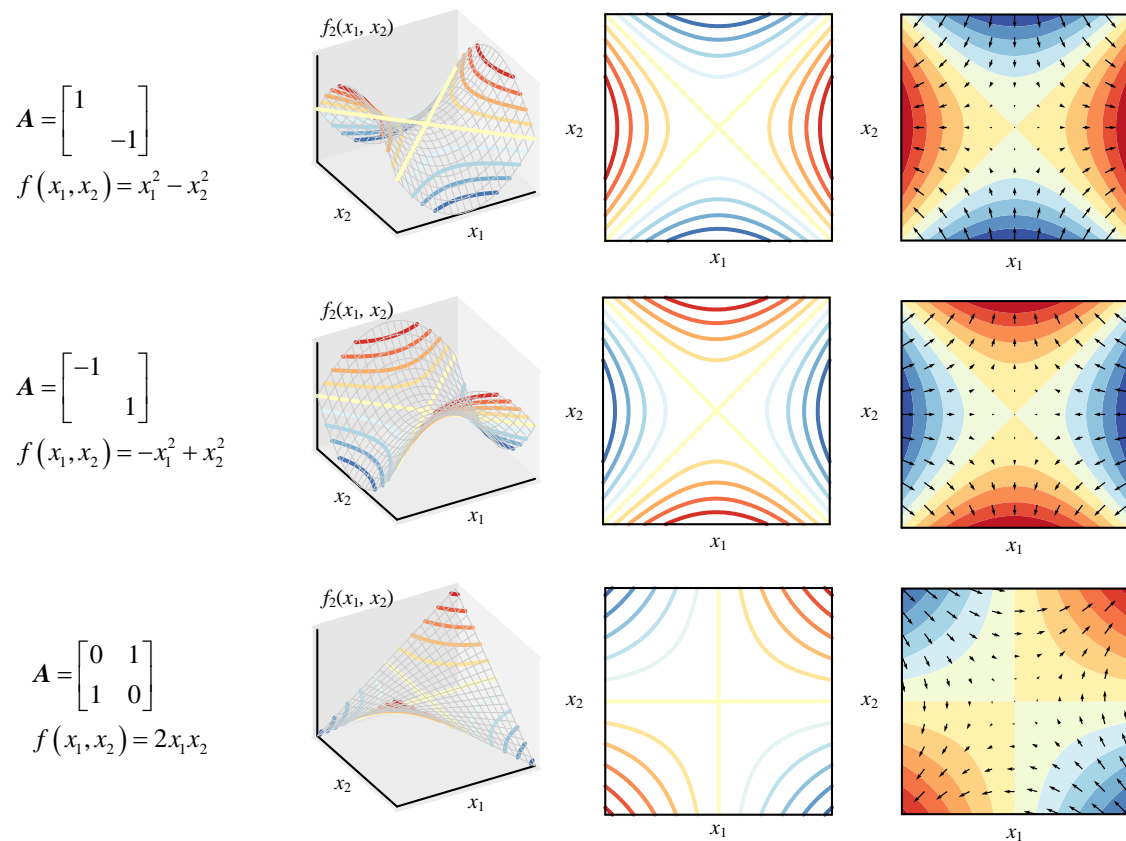
图 4. 半负定, $A_{2 \times 2}$ | Bk2_Ch21_01.ipynb

不定

如图 5 所示, 一个矩阵 $A_{2 \times 2}$ **不定** (indefinite), 意味着 $f(x) = x^T @ A_{2 \times 2} @ x$ 是个马鞍面, $(0, 0)$ 为鞍点。 $f(x) = x^T @ A_{2 \times 2} @ x$ 符号不定。

图中有些箭头背离 $(0, 0)$, 有些指向 $(0, 0)$ 。

请大家用 `numpy.linalg.eig()` 计算图 1 ~ 图 5 中不同矩阵 $A_{2 \times 2}$ 的特征值和特征向量, 并试着总结规律。

图 5. 不定, $A_{2 \times 2}$ | Bk2_Ch21_01.ipynb

文件 Bk2_Ch21_01.ipynb 绘制上述图像，请在 JupyterLab 查看完整代码。

下面，我们聊聊 Bk2_Ch21_01.ipynb 核心代码。

代码 1 自定义函数，用来计算网格坐标数据 (xx1, xx2) 的二元函数值 (ff_x) 和梯度值 (V)。

简单来说，对于二元函数，在函数曲面上任意一点的**梯度** (gradient) 告诉我们在给定点上函数增长最快的方向，也就是“上山”最陡峭的方向。如果我们沿着梯度的方向走，函数瞬时值会增加最快；反过来，梯度的反方向就是“下山”最快的方向。当然，我们每走一步，随着位置变化，梯度值一般会随时变化。

a 利用 `sympy.symbols()`，简作 `symbols()`，定义了两个符号变量 x_1 、 x_2 。

b 相当于构造符号列向量 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ 。

c 计算二次型 $\mathbf{x}^T \mathbf{A} \mathbf{x}$ 。举个例子，如果 $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ ，则 $\mathbf{x}^T \mathbf{A} \mathbf{x} = ax_1^2 + (b+c)x_1x_2 + dx_2^2$ 。f_x 的结果是二维数组，f_x[0][0] 取出其中符号代数式。

d 利用列表生成式获得梯度向量。其中，`sympy.diff()`，简作 `diff()`，完成**偏导数** (partial derivative) 计算。也就是说，这一句分别完成二元函数对 x_1 和 x_2 的偏导数，然后将它们构成一个梯度向量。

简而言之，偏导数衡量了函数在某个特定点上在特定方向上的变化率，或者说切线斜率。

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



鸢尾花书《数学要素》第 16 章将专门介绍偏导数这个概念。

e 利用 `sympy.lambdify()`，简作 `lambdify()`，将符号代数式转化为 Python 函数，用来完成数值运算。具体来说，我们的符号表达式保存在 `f_x` 中，这个表达式涉及符号变量 `x1` 和 `x2`。`lambdify([x1, x2], f_x)` 将 `f_x` 转换为 Python 函数。

f 计算网格坐标 `(xx1, xx2)` 的二元函数函数值。

g 也是利用 `sympy.lambdify()`，简作 `lambdify()`，将梯度向量符号表达式也转换为 Python 函数。

h 对 NumPy Array 进行采样，每隔 20 个元素选取 1 个。



鸢尾花书《编程不难》第 14 章专门讲过 NumPy Array 索引和切片，请大家回顾。

i 计算网格坐标 `(xx1_, xx2_)` 的二元函数梯度。

j 修复梯度值。这几句检查梯度值 `V` 中的元素，如果发现其中某个元素是整数，就将其替换为与 `xx1_` 相同形状的全零数组。其中，`isinstance()` 是一个 Python 内置函数，用于检查一个对象是否是指定类或类型的实例。比如，`isinstance(1, int)` 判断 1 是否是整数 `int`，结果为 `True`；`isinstance(1.0, int)` 判断 1.0 是否是整数 `int`，结果为 `False`。

```
def fcn_n_grdnt(A, xx1, xx2):
```

```

a     x1, x2 = symbols('x1 x2')
        # 符号向量
b     x = np.array([[x1, x2]]).T
        # 二次型
c     f_x = x.T @ A @ x; f_x = f_x[0][0]

        # 计算梯度，符号
d     grad_f = [diff(f_x, var) for var in (x1, x2)]

        # 计算二元函数值 f(x1, x2)
e     f_x_fcn = lambdify([x1, x2], f_x)
f     ff_x = f_x_fcn(xx1, xx2)

        # 梯度函数
g     grad_fcn = lambdify([x1, x2], grad_f)

        # 采样，降低颗粒度
h     xx1_ = xx1[::20, ::20]; xx2_ = xx2[::20, ::20]

        # 计算梯度
i     V = grad_fcn(xx1_, xx2_)

        # 修复梯度值
j     if isinstance(V[1], int):
            V[1] = np.zeros_like(xx1_)
        if isinstance(V[0], int):
            V[0] = np.zeros_like(xx1_)

    return ff_x, V
```



代码 1. 二元二次型函数和梯度向量 |  Bk2_Ch21_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 2 定义了可视化二元二次型函数。可视化方案为一行三列子图。

a 用 `matplotlib.pyplot.figure()`，简作 `plt.figure()`，创建图形对象 `fig`。参数 `figsize=(6,3)` 指定了图形窗口的尺寸，其中 `(6, 3)` 表示宽度为 6 英寸，高度为 3 英寸。

b 使用 `add_subplot()` 方法在 `fig` 对象上添加第 1 幅子图对象。

参数 1 表示子图行数为 1 行，参数 3 表示子图列数为 3 列，参数 1 表示当前子图的索引位置，即第 1 个子图。

参数 `projection='3d'` 指定子图的投影方式为 3D，表示这是一个三维的子图。

c 在三维轴对象 `ax` 上用 `plot_wireframe()` 绘制网格曲线。

参数 `rstride` 和 `cstride` 分别表示网格线的行和列的步长，即每隔多少行和列绘制一条线。

参数 `color=[0.8,0.8,0.8]` 指定了网格线的颜色，这里用的是灰色。

参数 `linewidth=0.25` 指定了网格线的宽度。

d 在三维轴对象 `ax` 上用 `contour()` 绘制三维等高线。参数 `levels` 用来指定等高线数量。Cmap 指定了等高线的颜色映射。

e 使用 `add_subplot()` 方法在 `fig` 对象上添加第 2 幅子图对象，默认为 2D 轴对象。

f 在二维轴对象上用 `contour()` 绘制平面等高线。

g 也是使用 `add_subplot()` 方法在 `fig` 对象上添加第 3 幅子图对象。

h 在二维轴对象上用 `contourf()` 绘制平面填充等高线。

i 用 `quiver()` 绘制向量场，代表梯度。

```
# 可视化函数
def visualize(xx1,xx2,f2_array,gradient_array):
    fig = plt.figure(figsize=(6,3))
    # 第一幅子图
    ax = fig.add_subplot(1, 3, 1, projection='3d')
    ax.plot_wireframe(xx1, xx2, f2_array,
                     rstride=10, cstride=10,
                     color = [0.8,0.8,0.8],
                     linewidth = 0.25)
    ax.contour(xx1, xx2, f2_array,
               levels = 12,
               cmap = 'RdYlBu_r')
    # .....
    # 第二幅子图
    ax = fig.add_subplot(1, 3, 2)
    ax.contour(xx1, xx2, f2_array,
               levels = 12,
               cmap = 'RdYlBu_r')
    # .....
    # 第三幅子图
    ax = fig.add_subplot(1, 3, 3)
    ax.contourf(xx1, xx2, f2_array,
                levels = 12,
                cmap = 'RdYlBu_r')
    ax.quiver(xx1_, xx2_, gradient_array[0], gradient_array[1],
              angles='xy', scale_units='xy',
              edgecolor='none', alpha=0.8)
    # .....
```

代码 2. 二元二次型可视化函数 | Bk2_Ch21_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

21.2 三元二次型

二元二次型不是本章可视化的核心，本章想要聊聊三元二次型有哪些有趣的特征。图 6 ~ 图 21 给出若干种三元二次型，请大家逐个分析。特别注意等高线变化趋势、极值位置、剖面等高线的几何形状（正圆、椭圆、旋转椭圆、平行线、抛物线、双曲线），并试着解释为什么会出现这些几何形状。

 请大家也用 `numpy.linalg.eig()` 计算图 6 ~ 图 21 中不同矩阵 $A_{3 \times 3}$ 的特征值和特征向量，并试着总结规律。

文件 `Bk2_Ch21_02.ipynb` 绘制图 6 ~ 图 21，请在 JupyterLab 查看完整代码。

下面，我们聊聊 `Bk2_Ch21_02.ipynb` 核心代码。


代码 3 自定义三元二次型函数。

- a 利用 `sympy.symbols()`，简作 `symbols()`，定义了三个符号变量 x_1 、 x_2 、 x_3 。
- b 相当于构造符号列向量 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ 。
- c 计算三元二次型 $\mathbf{x}^T \mathbf{A} \mathbf{x}$ 。
- d 利用 `sympy.lambdify()`，简作 `lambdify()`，将符号代数式转化为 Python 函数。
- e 计算网格坐标 (`xxx1, xxx2, xxx3`) 的三元函数函数值。

```
# 定义三元二次型
def fcn_3(A, xxx1, xxx2, xxx3):
    a  x1, x2, x3 = symbols('x1 x2 x3')
    b  x = np.array([[x1, x2, x3]]).T
    c  f_x = x.T @ A @ x
       print(simplify(expand(f_x[0][0])))
    d  f_x_fcn = lambdify([x1, x2, x3], f_x[0][0])
    e  fff = f_x_fcn(xxx1, xxx2, xxx3)
    return fff
```

代码 3. 三元二次型函数 |  `Bk2_Ch21_02.ipynb`

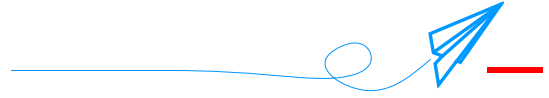
本书前文已经介绍过这种“切豆腐”的可视化方案，本章不再重复。

 请大家思考如何 Plotly 的体积图和三维等值面可视化三元二次型。答案在第 18 章。



有关 Plotly 中 volume 图，请大家参考：

<https://plotly.com/python/3d-volume-plots/>



本章在《编程不难》第 25 章基础上又深入介绍了二次型和正定性。在可视化二元二次型时，我们采用了三维网格面、三维等高线、平面等高线、平面箭头图等可视化方案。

展示三元二次型时，我们用了“切豆腐”的方法，通过在三个不同方向切片，我们可以清楚地看到三元函数的变化趋势。此外，强烈建议大家尝试使用 Plotly volume 可视化三元函数。

当然，要想真正理解二次型和正定性，我们就需要利用线性代数工具，这是《矩阵力量》要解决的问题。

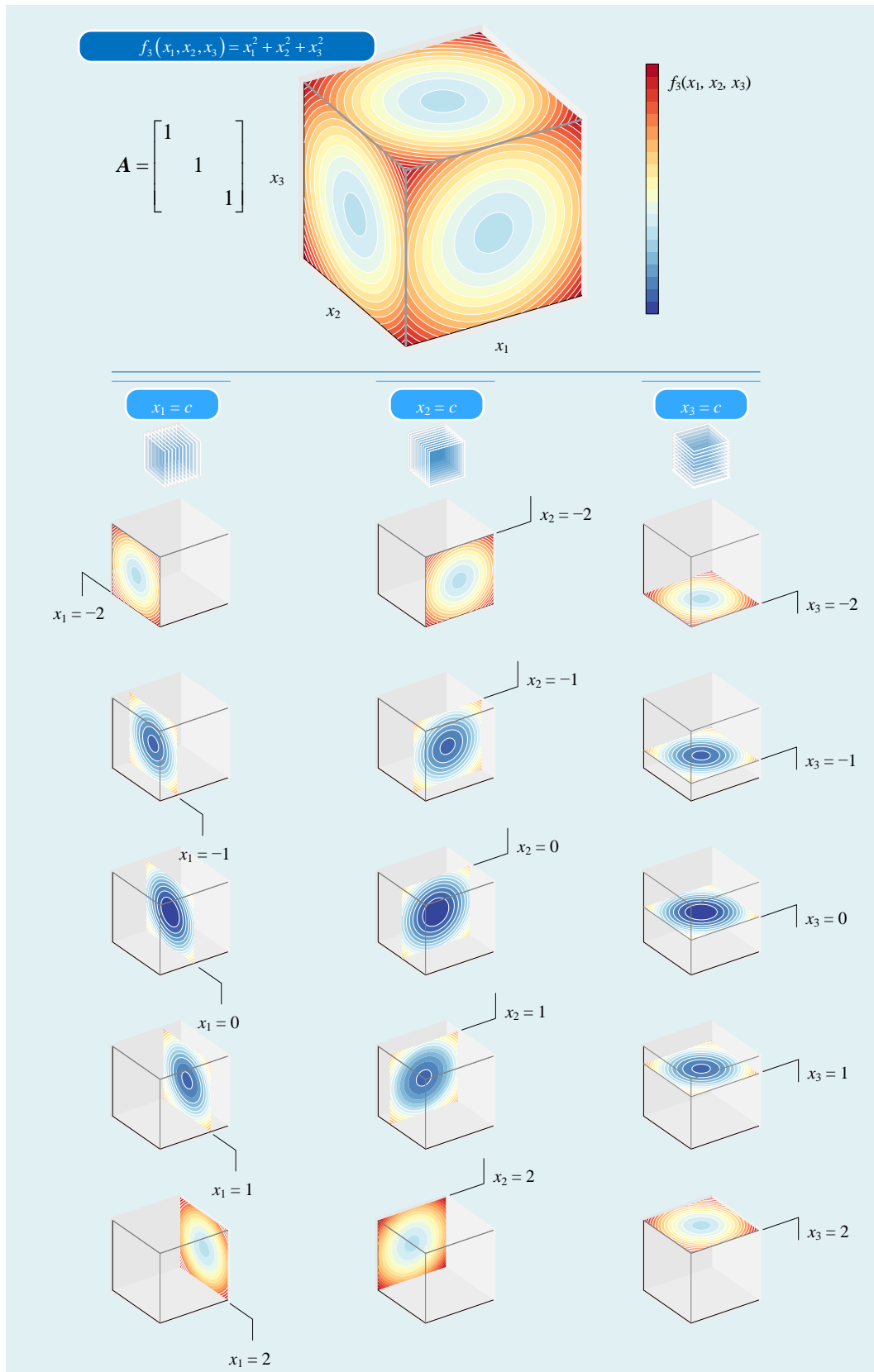


图 6. 三元二次型，正定，情况 A | Bk2_Ch21_02.ipynb

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

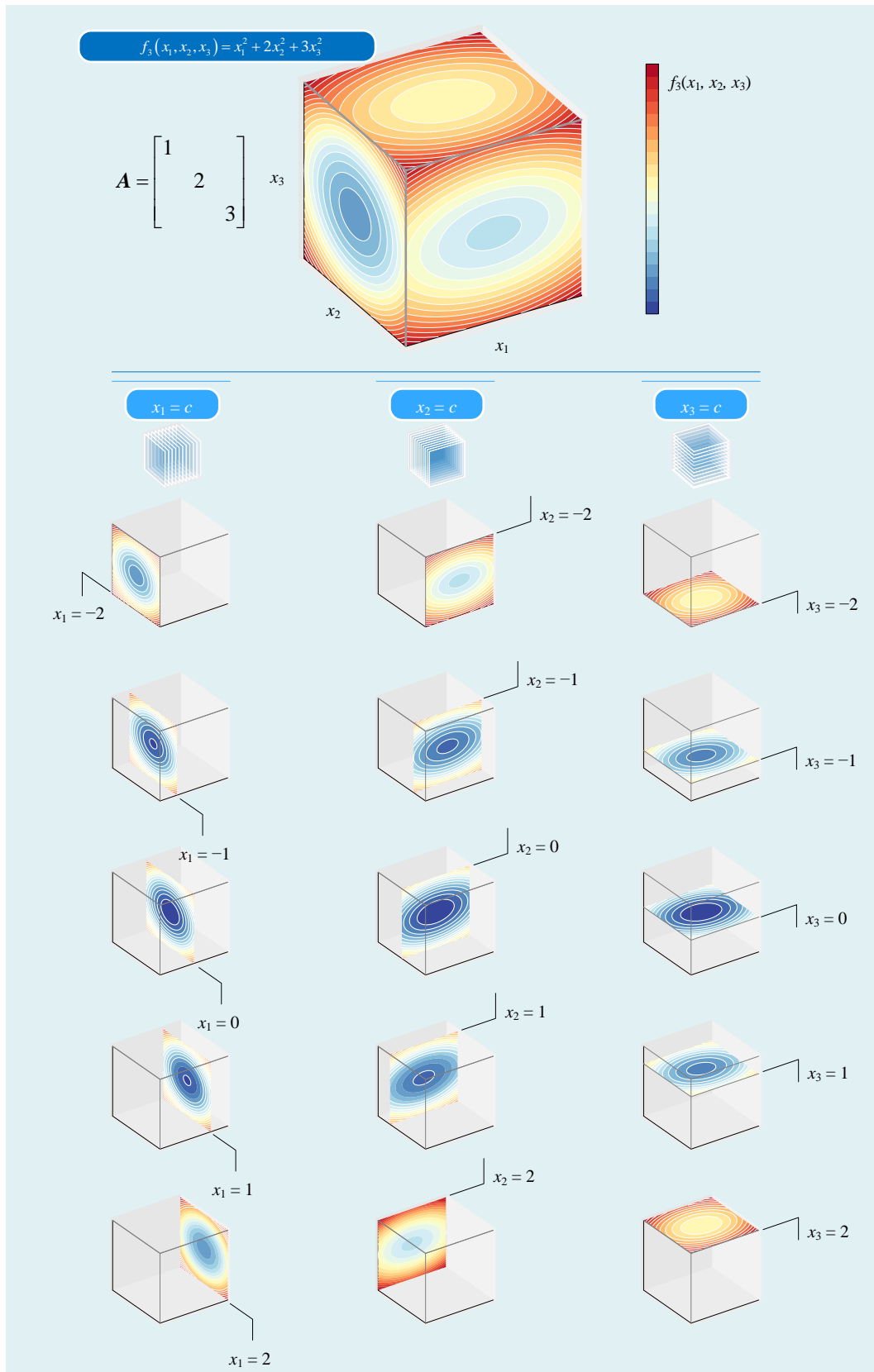


图 7. 三元二次型，正定，情况 B | Bk2_Ch21_02.ipynb

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

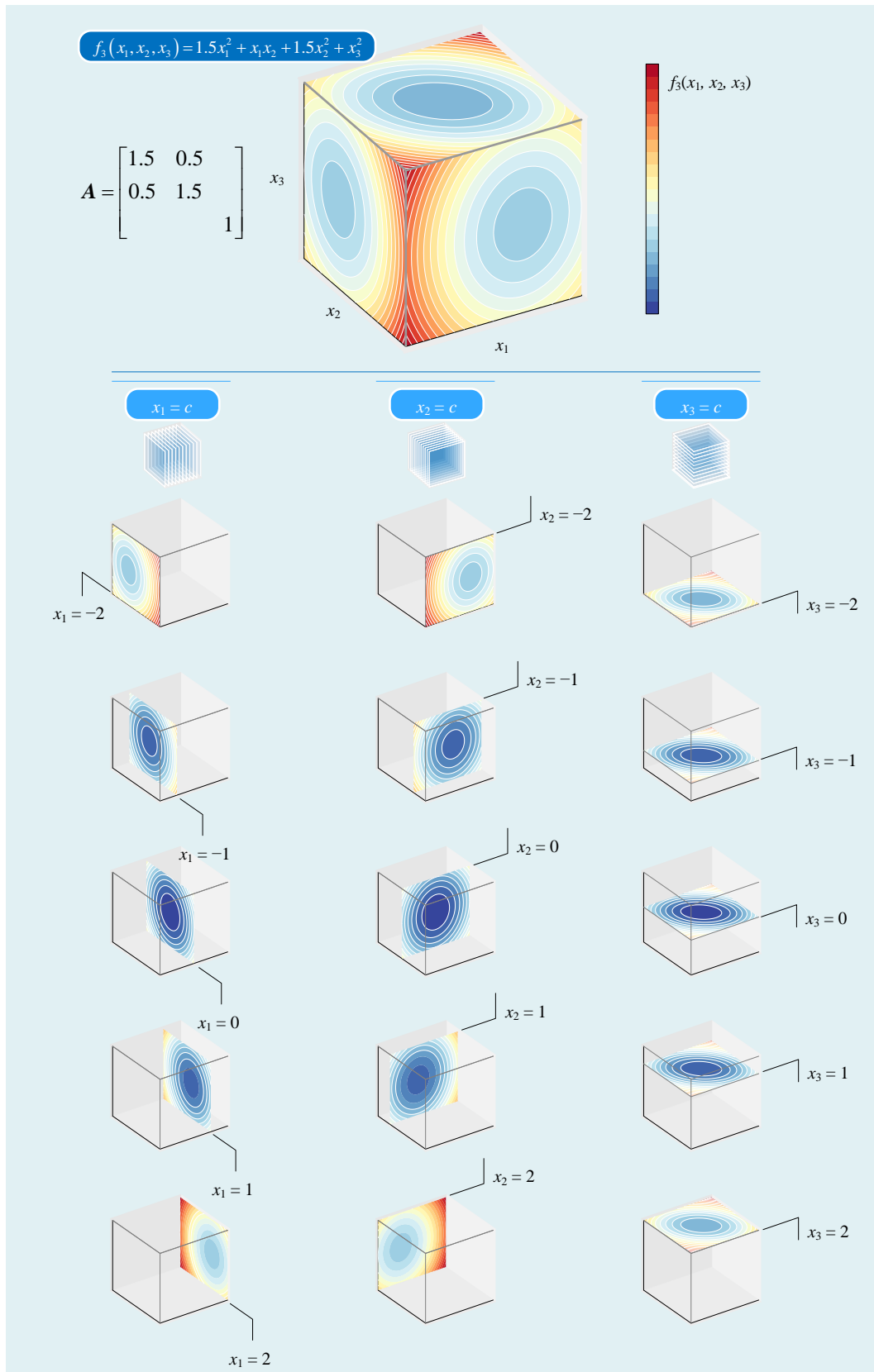


图 8. 三元二次型，正定，情况 C | Bk2_Ch21_02.ipynb

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

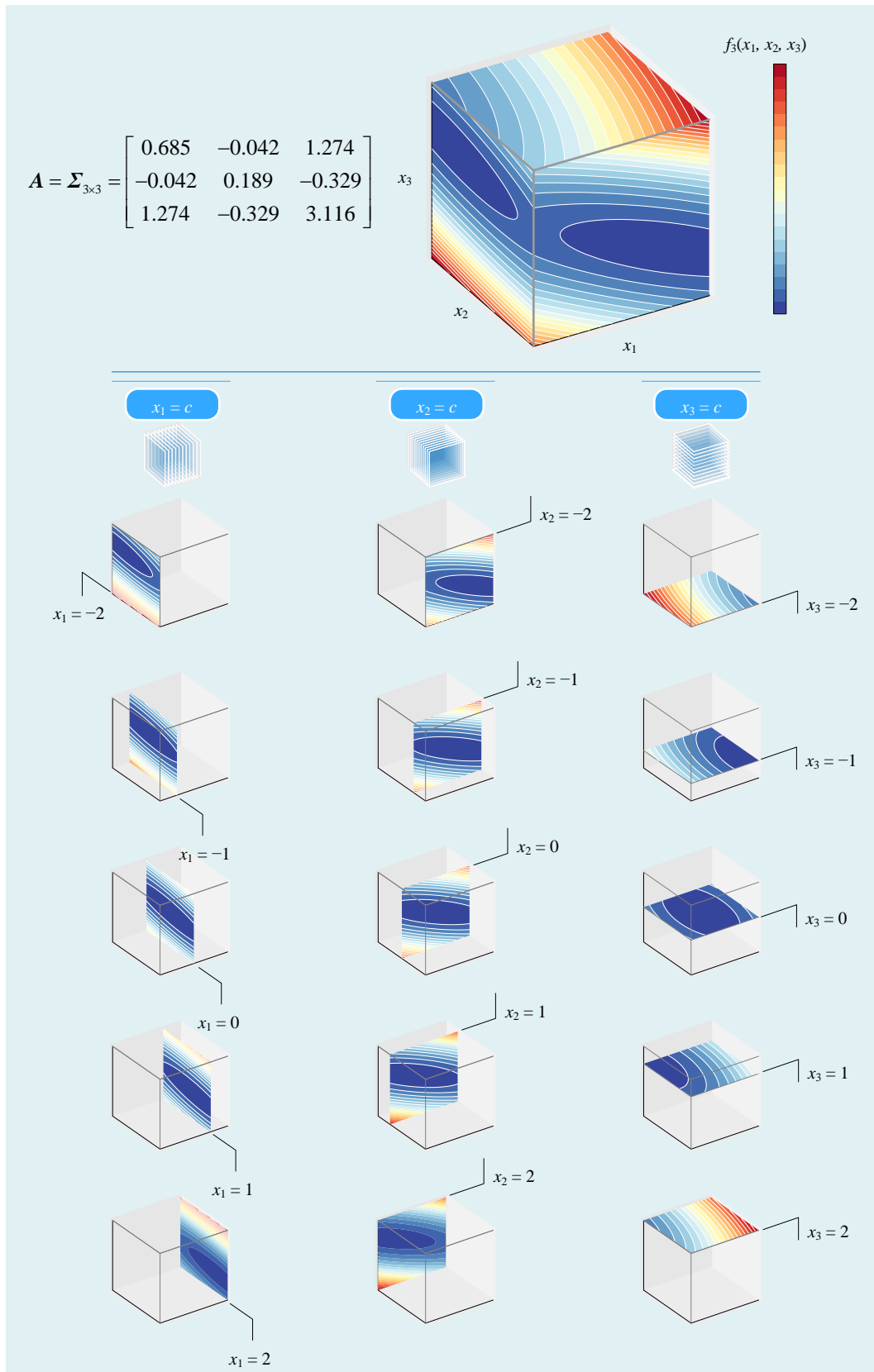


图 9. 三元二次型，正定，情况 D. (鸢尾花数据前三个特征的协方差矩阵) | Bk2_Ch21_02.ipynb

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

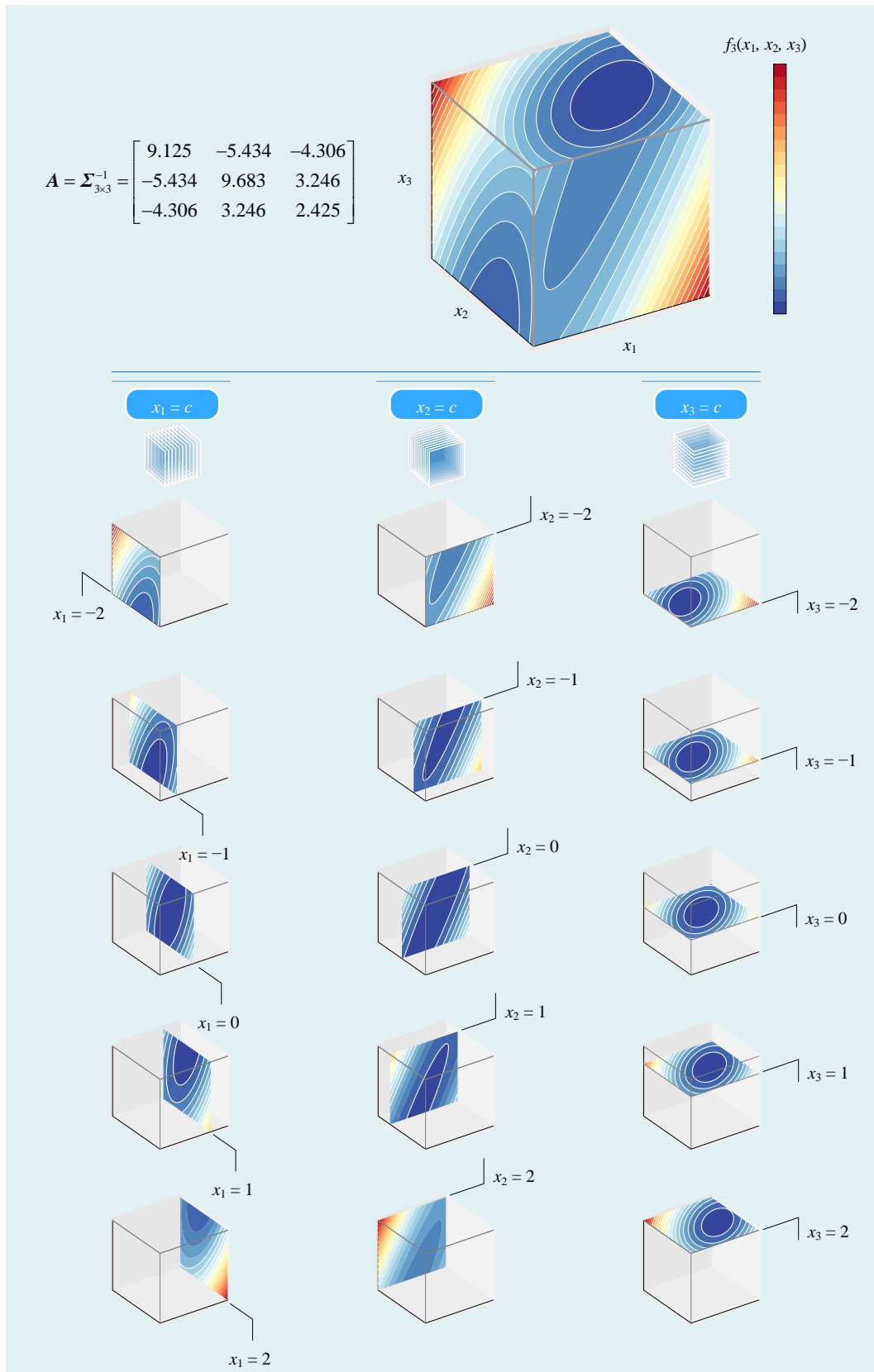


图 10. 三元二次型，正定，情况 E，（鸢尾花数据前三个特征的协方差矩阵逆矩阵） | Bk2_Ch21_02.ipynb

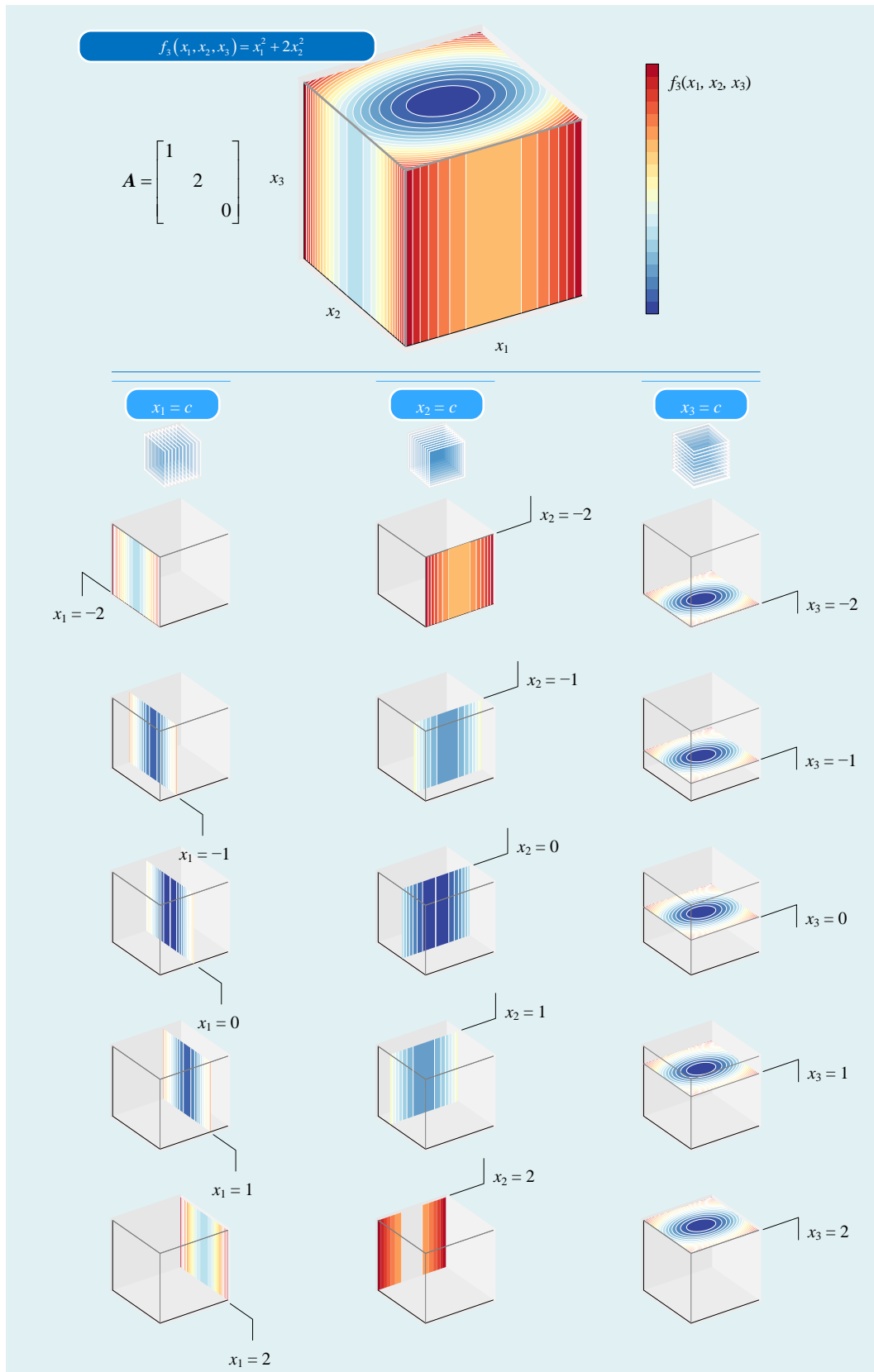

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

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 11. 三元二次型, 半正定, 情况 F |  Bk2_Ch21_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

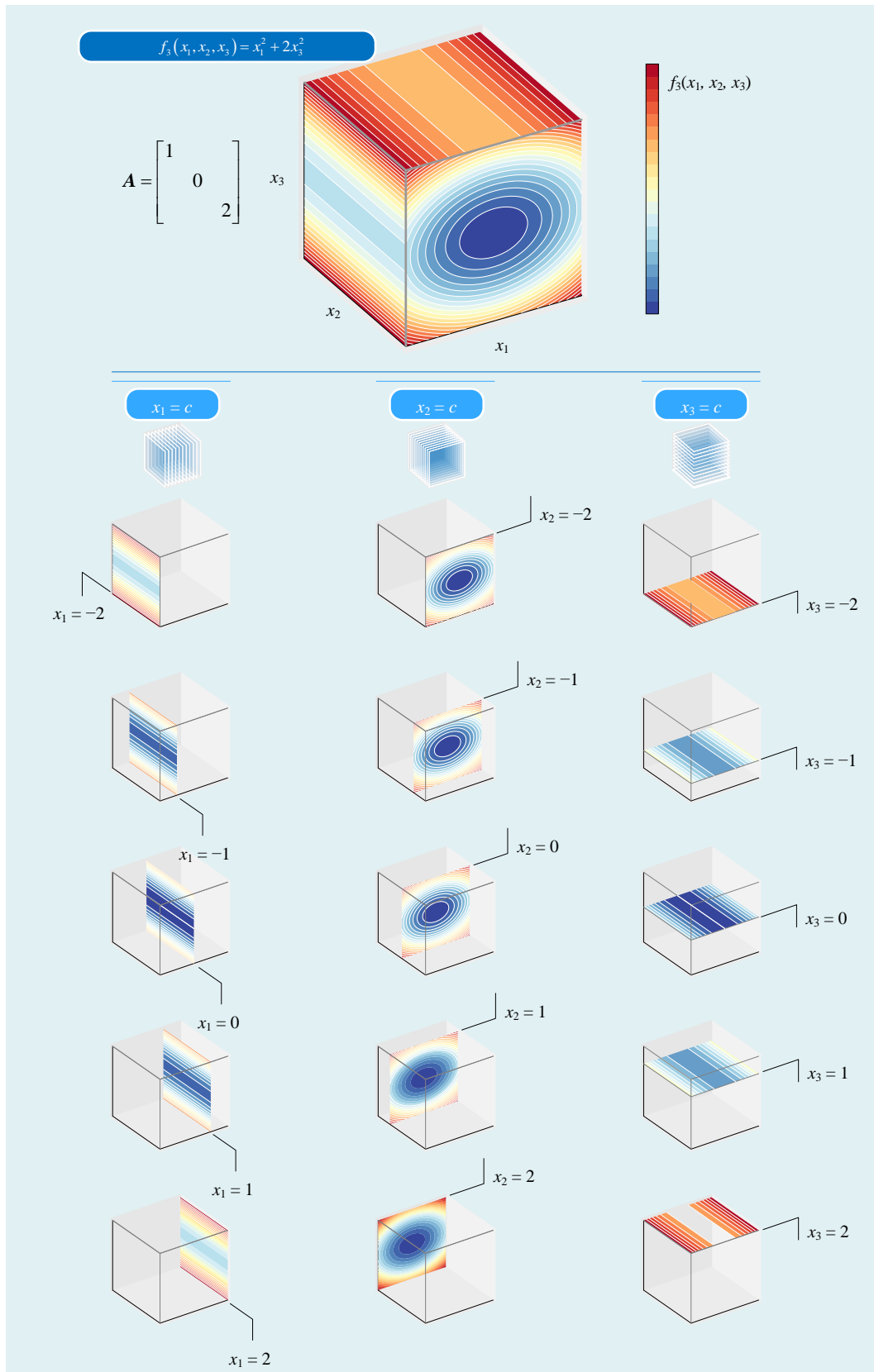


图 12. 三元二次型, 半正定, 情况 G | Bk2_Ch21_02.ipynb

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

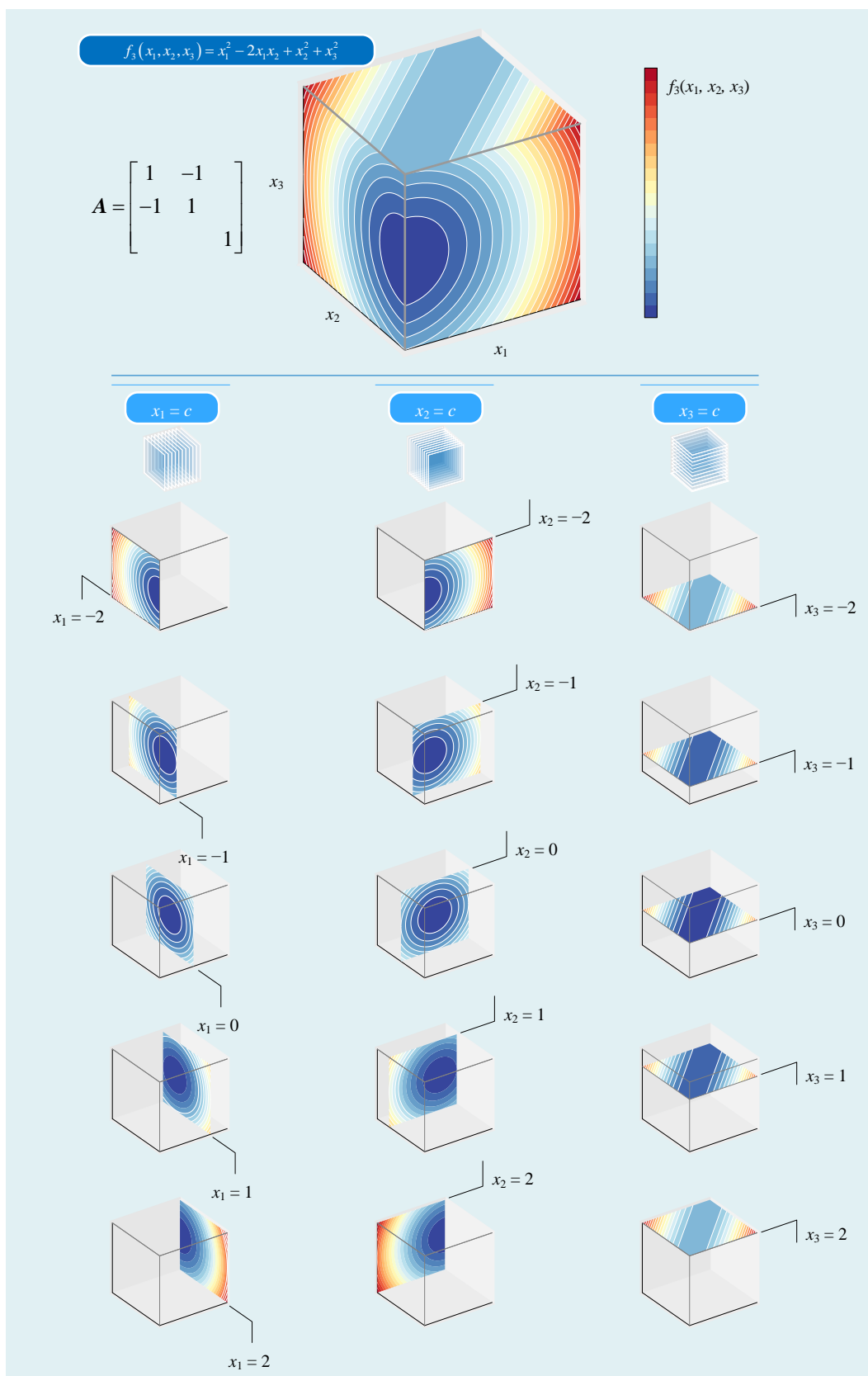


图 13. 三元二次型, 半正定, 情况 H | Bk2_Ch21_02.ipynb

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

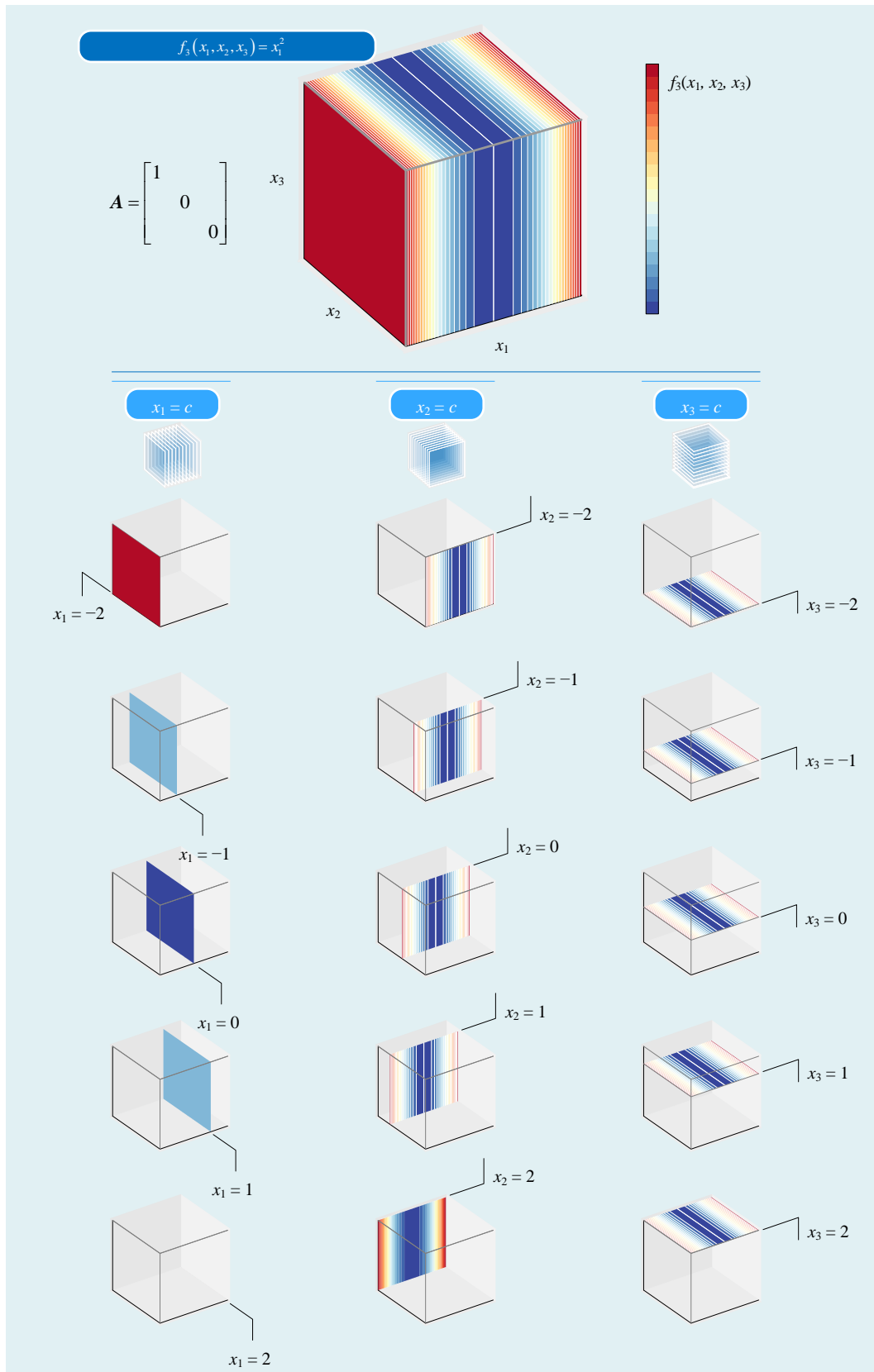


图 14. 三元二次型, 半正定, 情况 I | Bk2_Ch21_02.ipynb

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

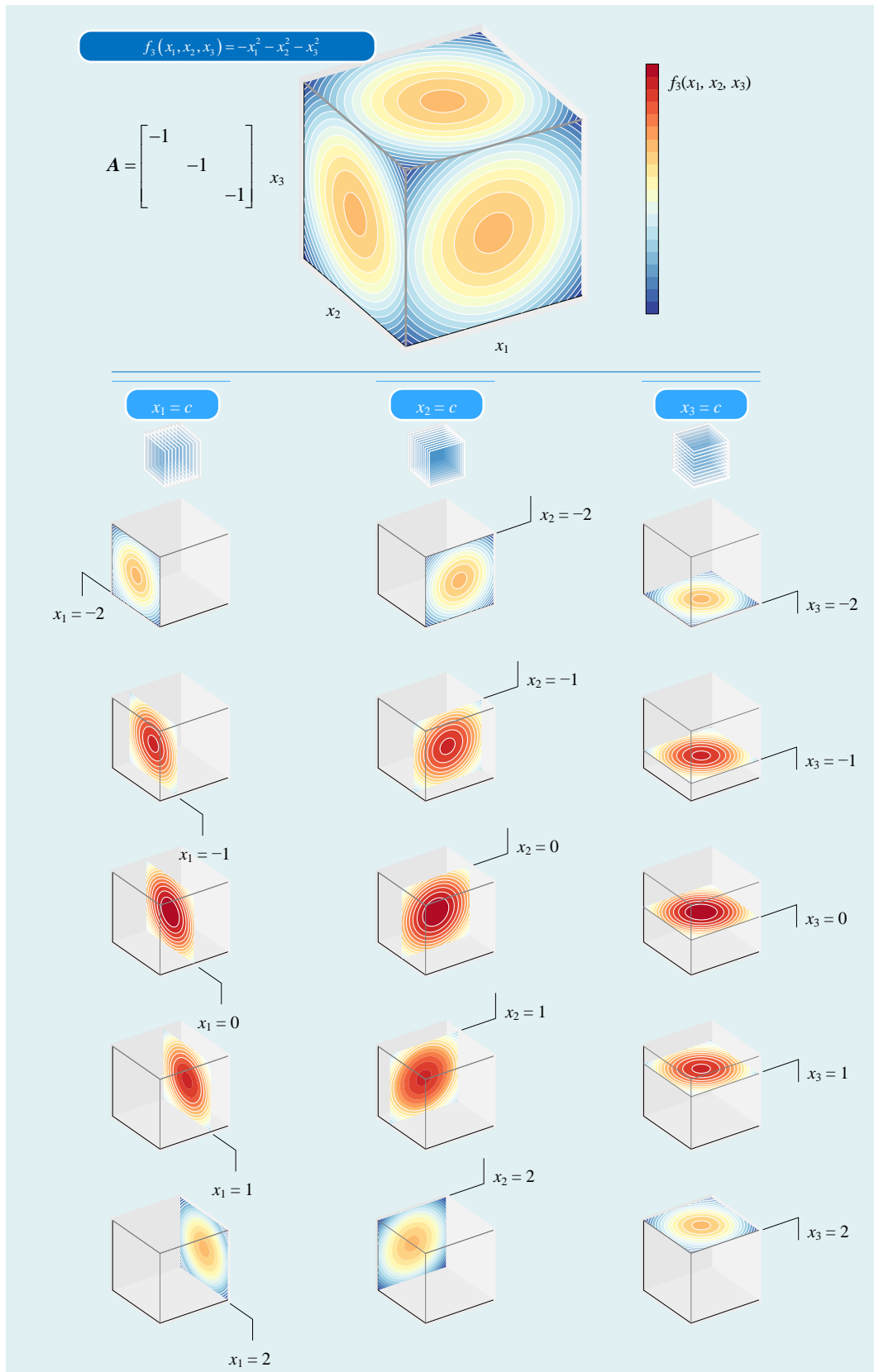


图 15. 三元二次型, 负定, 情况 J | Bk2_Ch21_02.ipynb

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

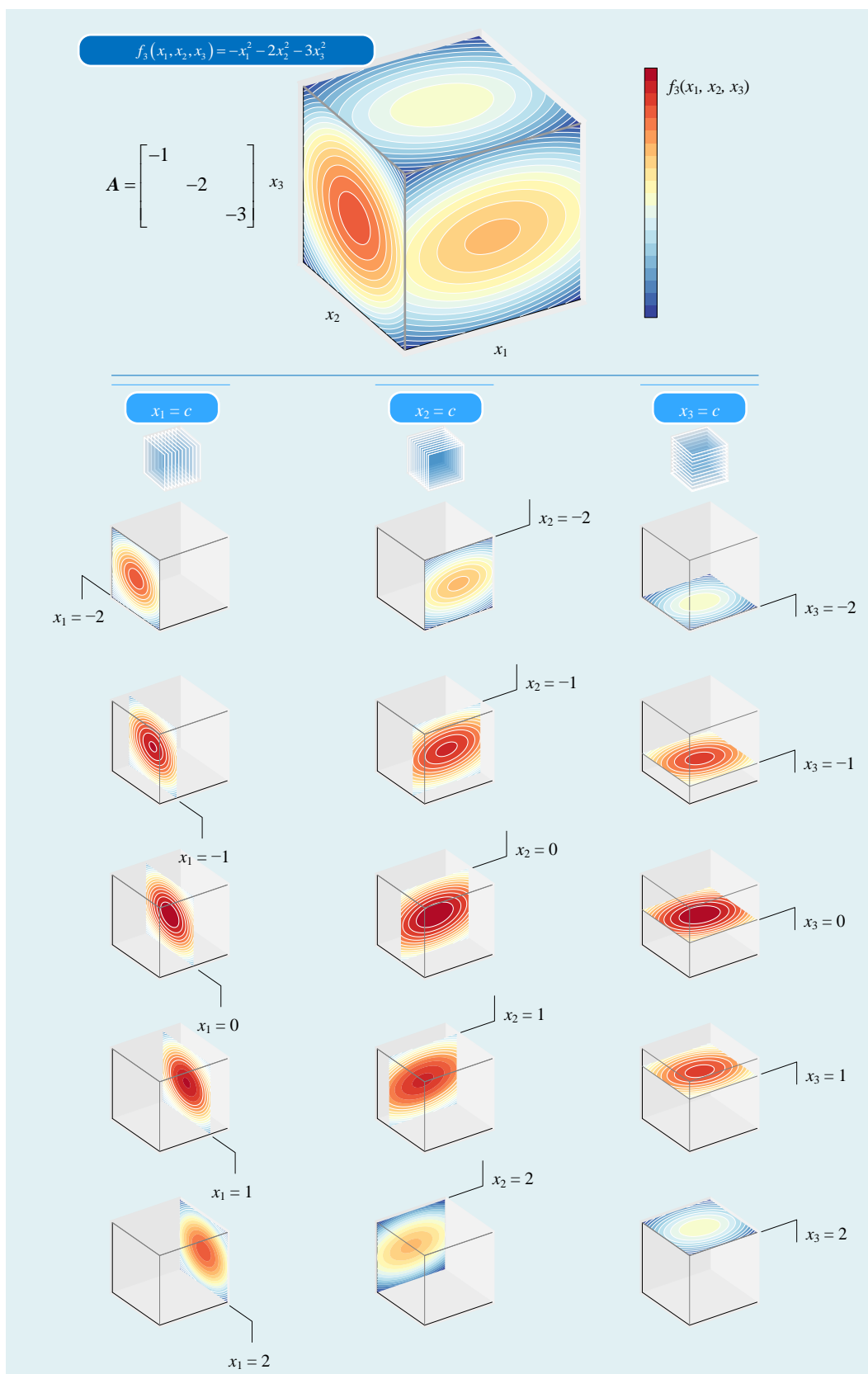


图 16. 三元二次型, 负定, 情况 K | Bk2_Ch21_02.ipynb

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

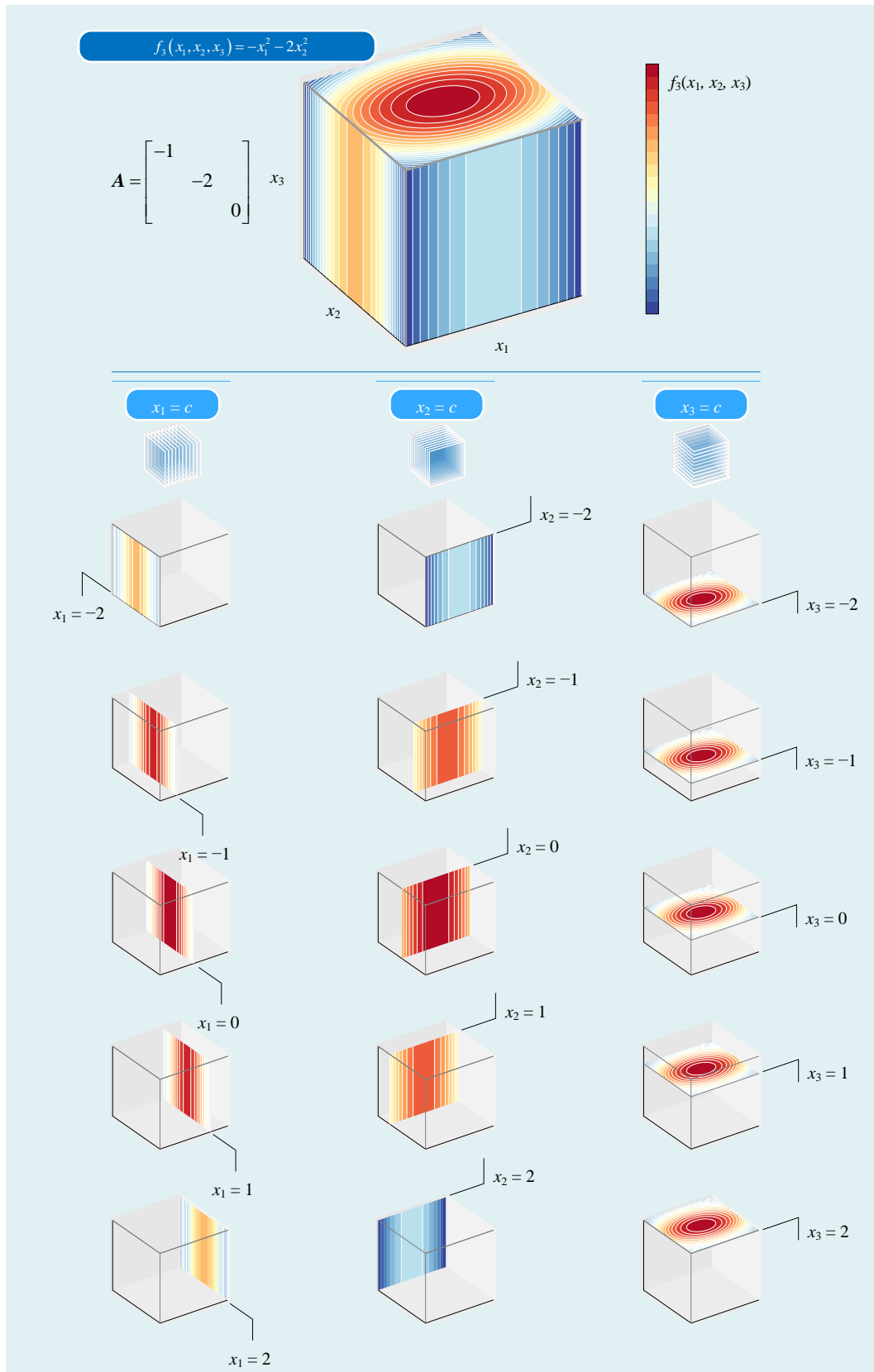


图 17. 三元二次型, 半负定, 情况 L | Bk2_Ch21_02.ipynb

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

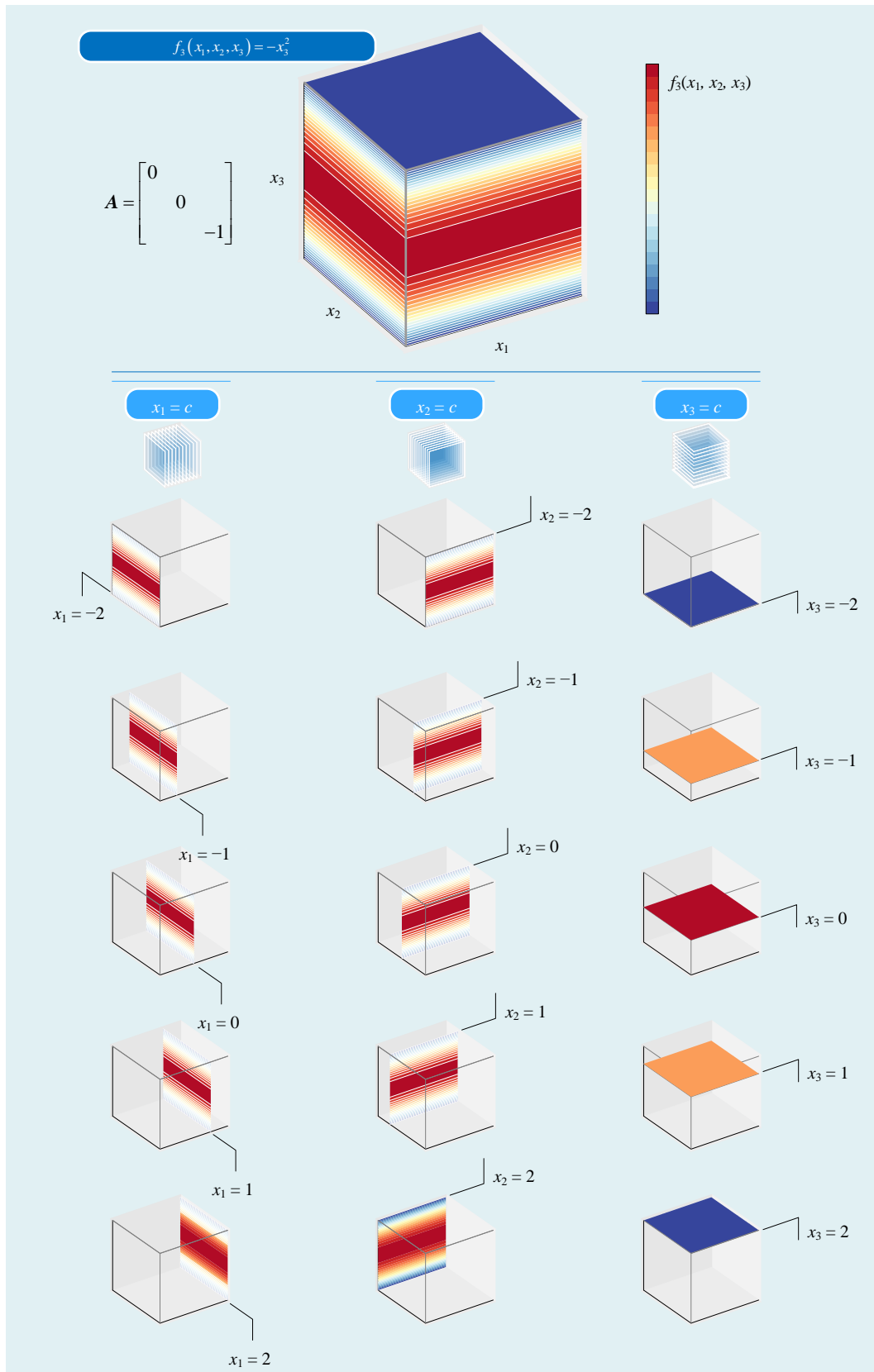


图 18. 三元二次型, 半负定, 情况 M | Bk2_Ch21_02.ipynb

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

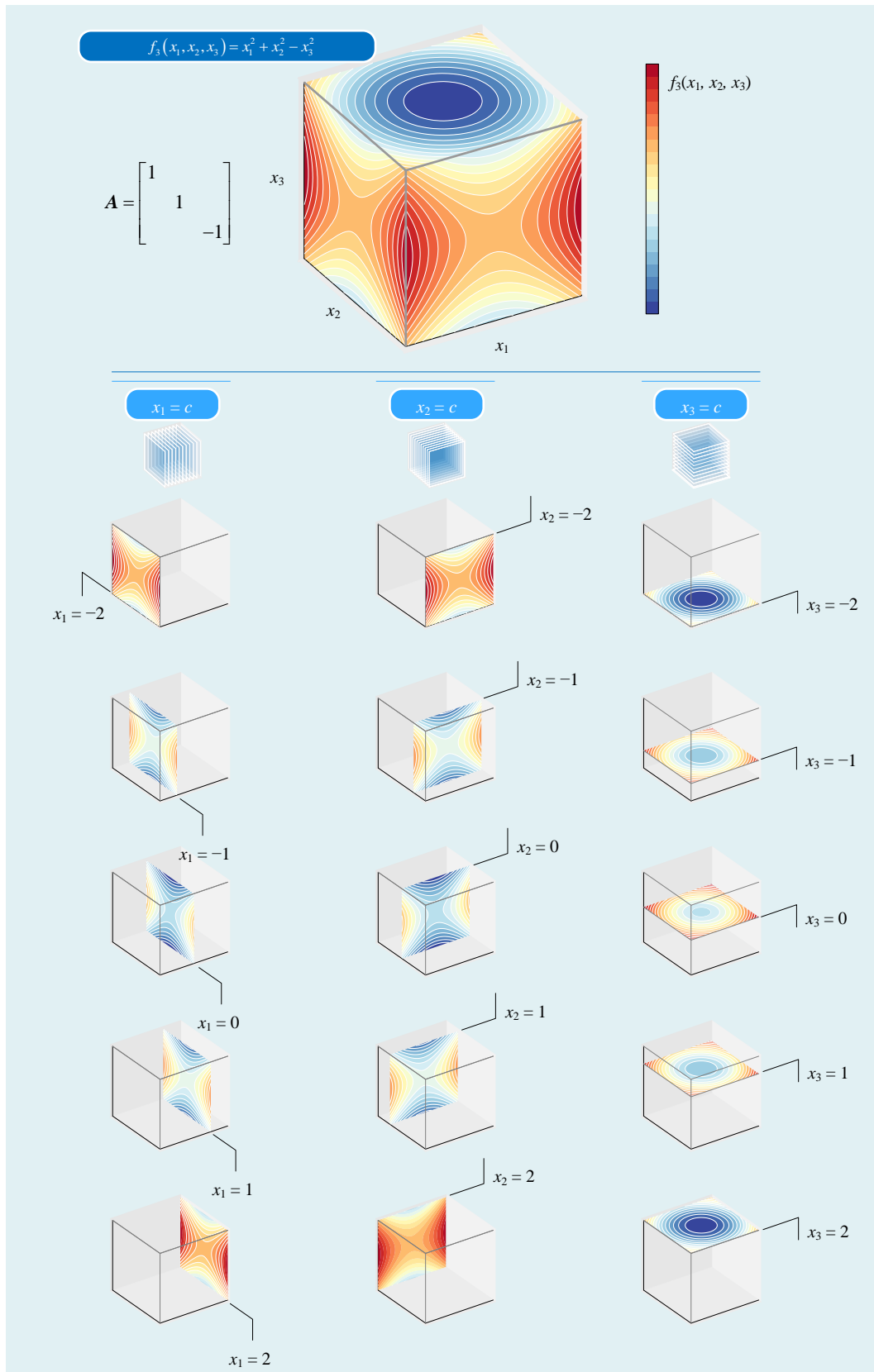


图 19. 三元二次型, 不定, 情况 N | Bk2_Ch21_02.ipynb

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

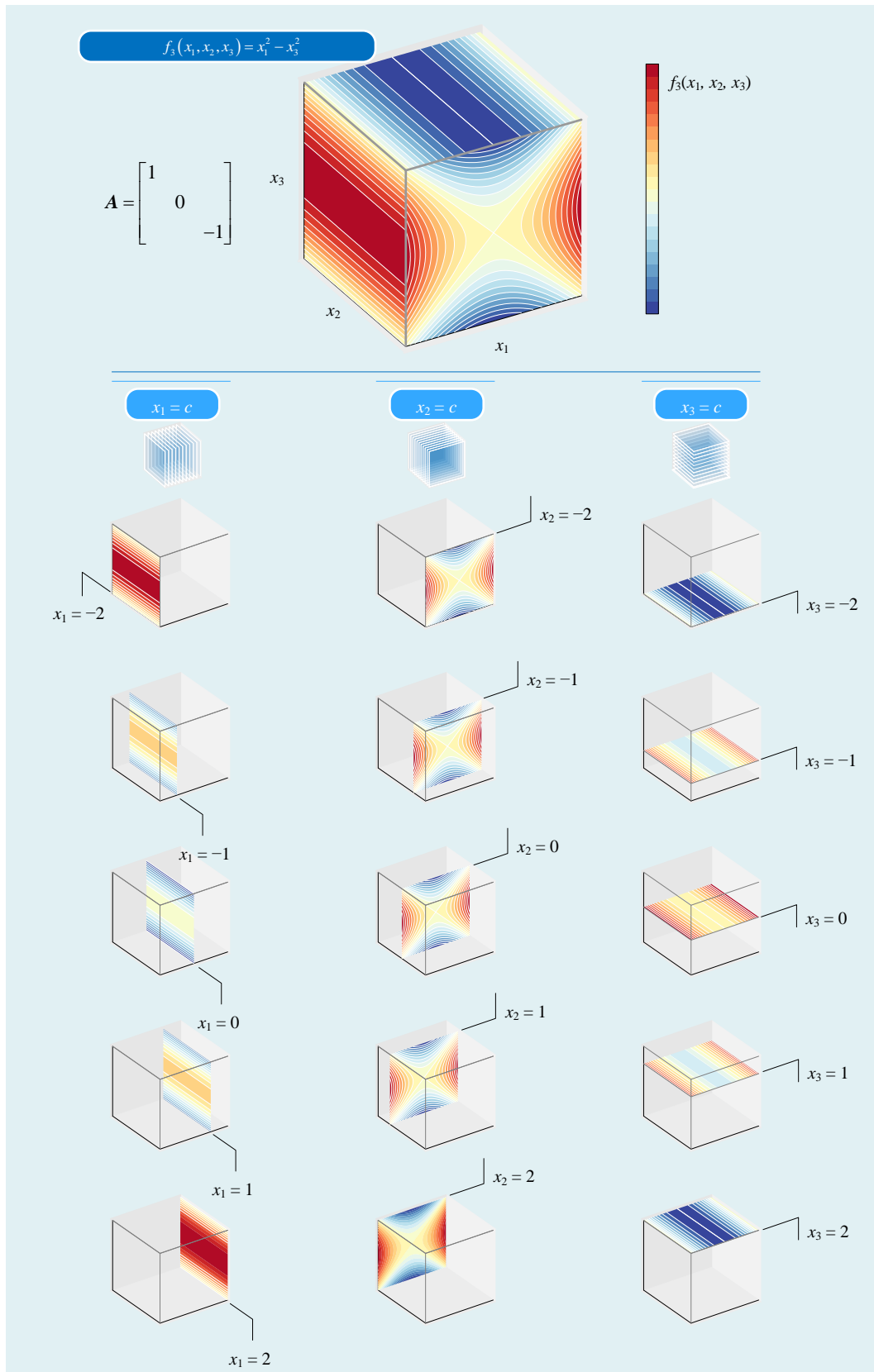


图 20. 三元二次型，不定，情况 0 | Bk2_Ch21_02.ipynb

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

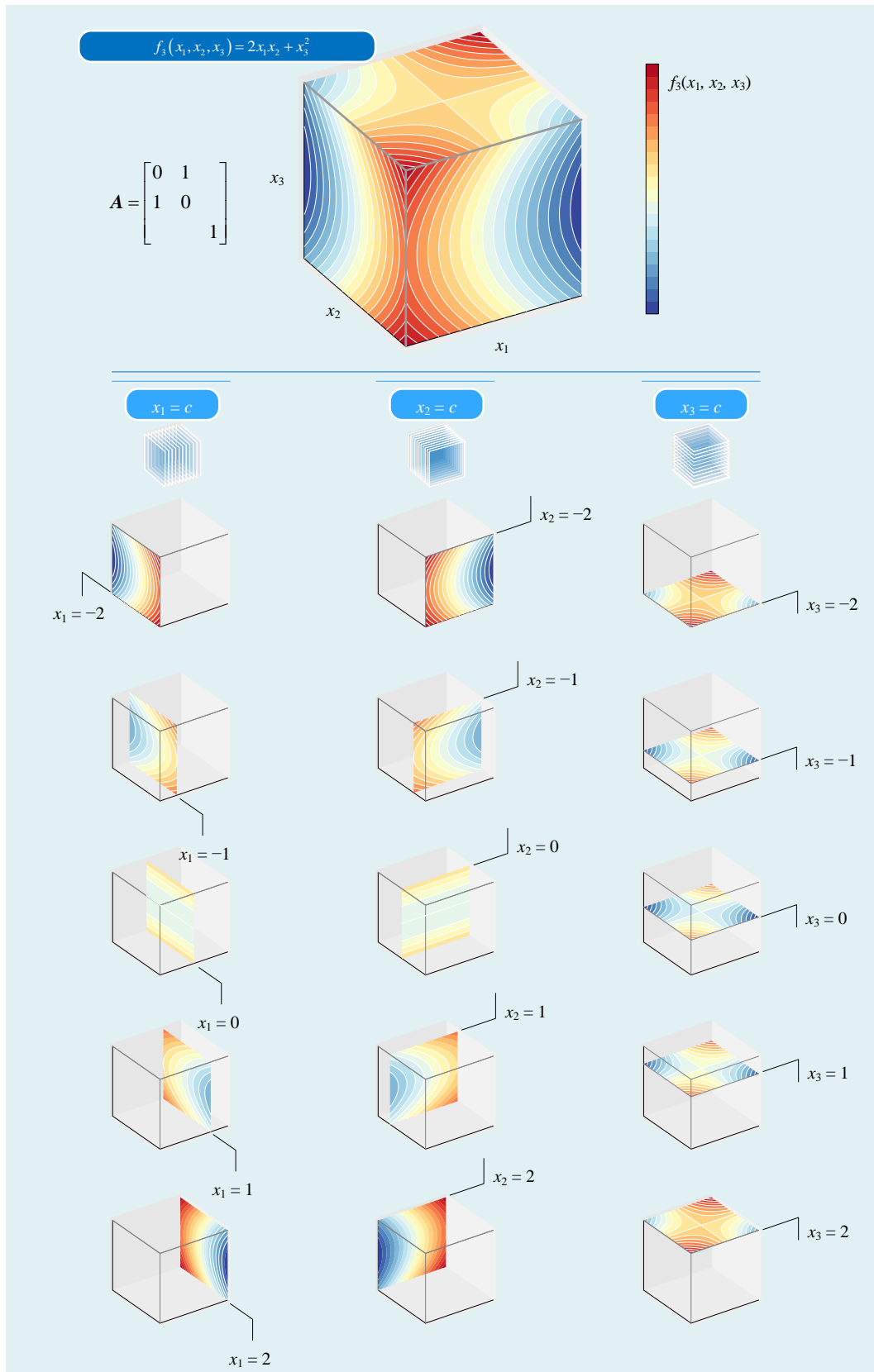


图 21. 三元二次型, 不定, 情况 P | Bk2_Ch21_02.ipynb

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