

# 14

## Types of Mesh Grids

# 各种网格

等高线、三维曲面不可或缺的数据结构



艺术家用头脑绘画，而不是双手。

***A man paints with his brains and not with his hands.***

—— 米开朗琪罗 (Michelangelo) | 文艺复兴三杰之一 | 1475 ~ 1564



- ◀ matplotlib.pyplot.plot\_wireframe() 绘制线框图
- ◀ matplotlib.pyplot.scatter() 绘制散点图
- ◀ matplotlib.pyplot.triplot() 绘制三角剖分图
- ◀ matplotlib.tri.Triangulation() 用于创建三角剖分对象
- ◀ numpy.column\_stack() 将两个矩阵按列合并
- ◀ numpy.concatenate() 将多个数组进行连接
- ◀ numpy.cos() 计算余弦值
- ◀ numpy.linspace() 在指定的间隔内, 返回固定步长的数据
- ◀ numpy.meshgrid() 产生网格化数据
- ◀ numpy.ones\_like() 用来生成和输入矩阵形状相同的全 1 矩阵
- ◀ numpy.sin() 计算正弦值

# 14.1 “方方正正” 网格

相信大家已经对 `numpy.meshgrid()` 函数并不陌生。NumPy 中的 `meshgrid()` 函数用于生成网格状的坐标点矩阵，其作用是将两个或多个一维数组转换为多维数组。具体来说，`meshgrid()` 函数接受两个或多个一维数组作为参数，返回多维坐标矩阵。图 1 所示为生成二维网络状坐标原理。

图 2 所示为从三维空间视角看二维网络状散点。

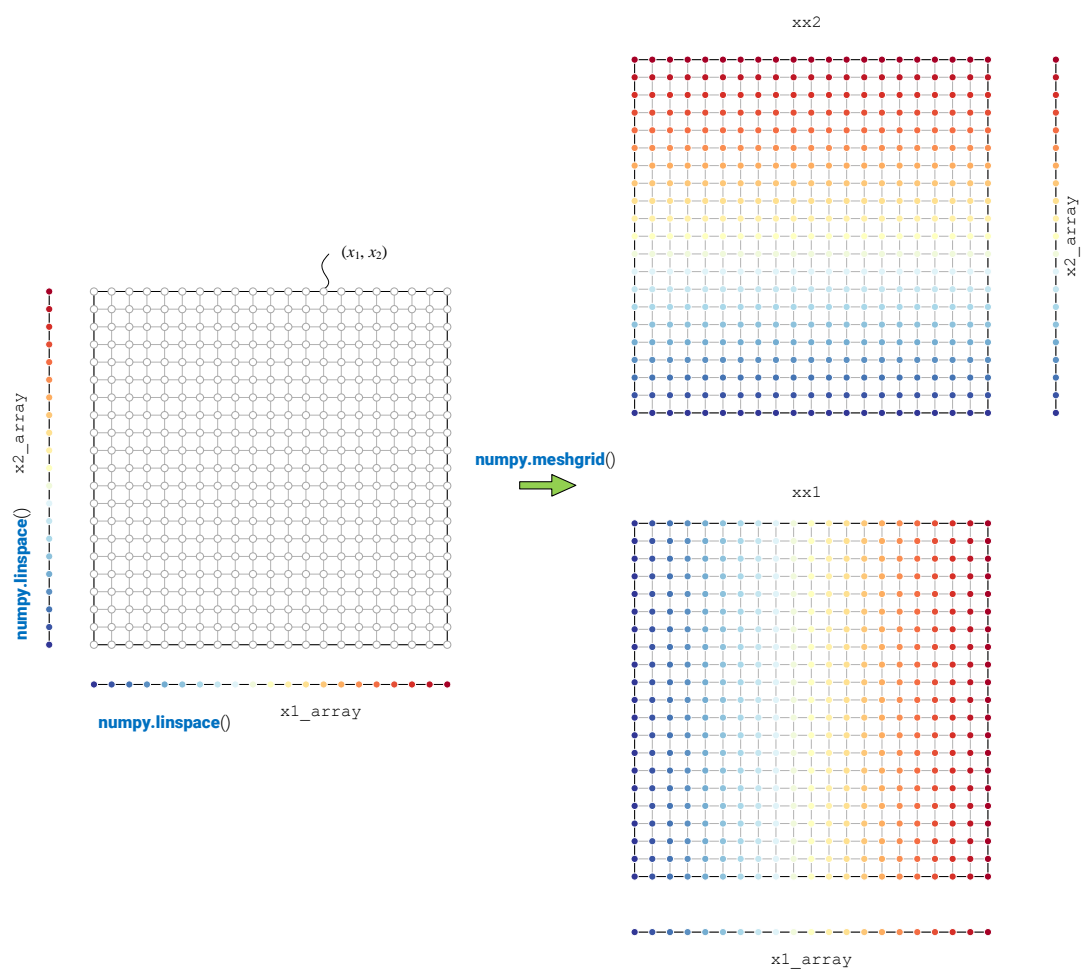


图 1. 用 `numpy.meshgrid()` 生成二维网络数据

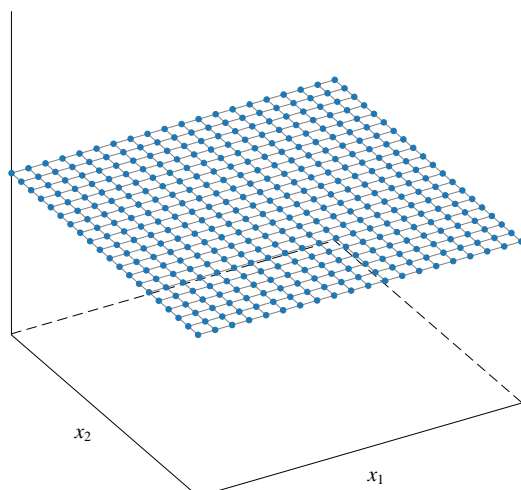


图 2. 三维空间看二维网络状坐标

### 网格状坐标的用途

`meshgrid()` 产生的二维网络状坐标通常用于绘制网格曲面、等高线等场景。图 3 所示为用二维网络状坐标可视化二元函数  $f(x_1, x_2)$ 。

本书后续，大家会看到我们用网格状坐标绘制等高线。

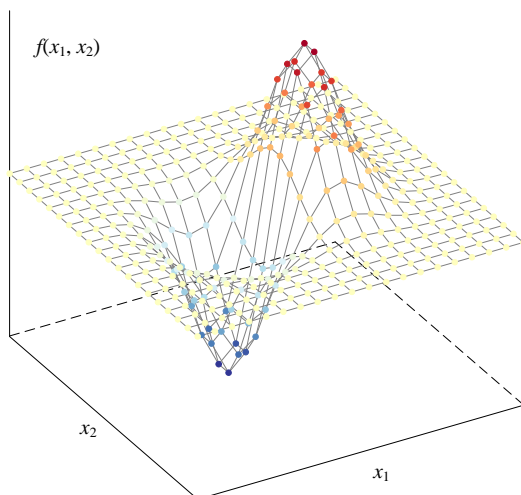


图 3. 二维网络状坐标可视化二元函数

### 颗粒度

类似平面线图，利用网络状坐标可视化数据时，也会遇到颗粒度的问题。如图 4、图 5 所示，颗粒度过低、过高都会导致可视化效果不理想。本书后文将分别从等高线、网格曲面等几个角度继续颗粒度这个话题。

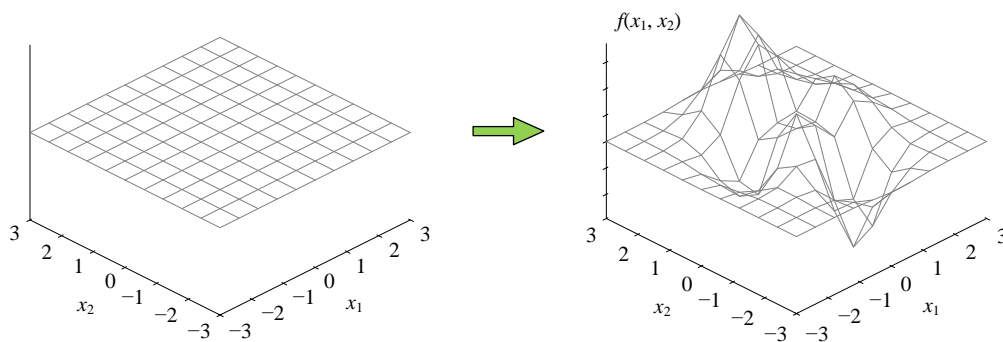


图 4. 颗粒度低

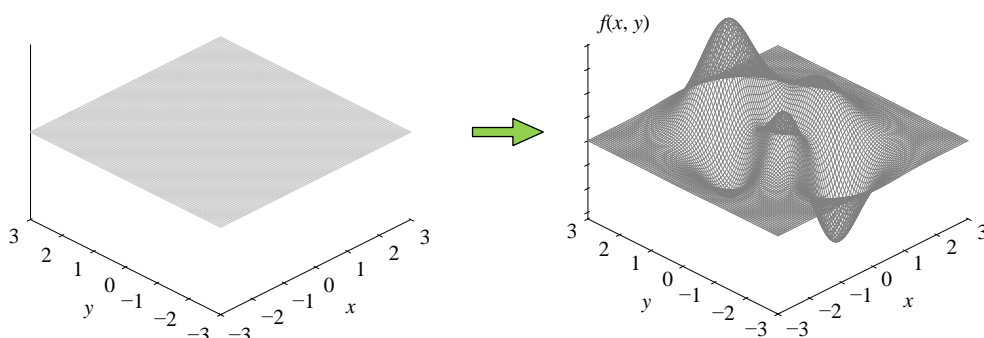


图 5. 颗粒度过高

### 使用面具

类似前文线图，对于网格我们也可以使用面具 (mask)。图 6 所示的两个例子为满足特定条件的的部分网格数据。

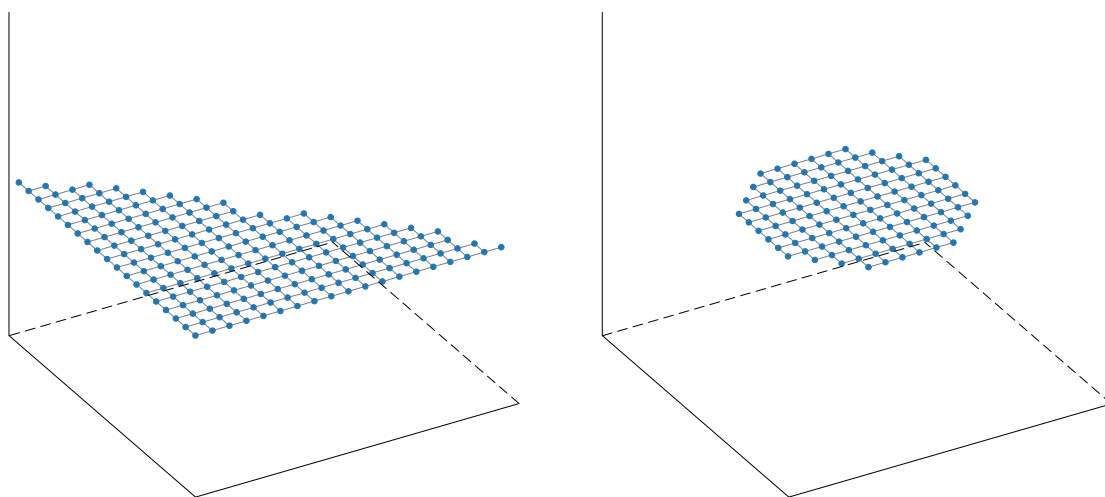


图 6. 使用面具

### 三维网格

此外，大家对图 7 所示三维网格也应该不陌生。我们在色彩模型中用过它。此外，本书后文还会继续用三维网格散点提供更为丰富的可视化方案。

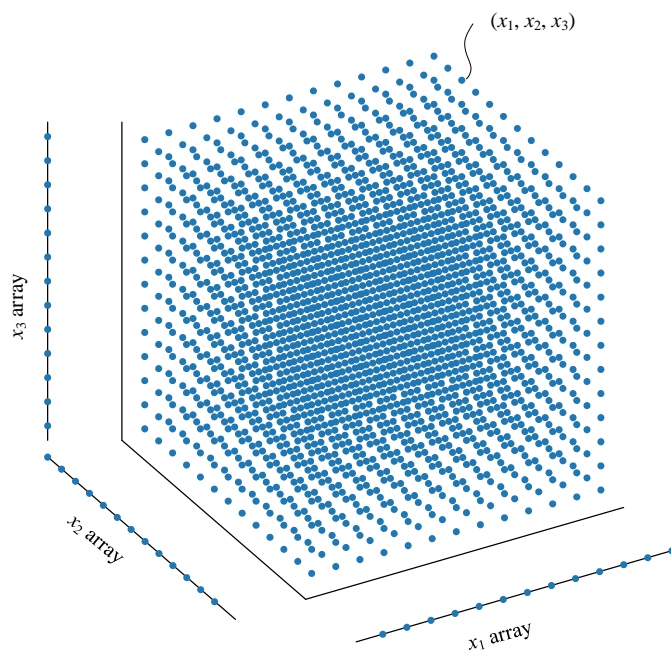


图 7. 三维网格

## 14.2 极坐标网格

除了方方正正的网格，本系列丛书还会用到极坐标网格。产生如图 8 所示的极坐标网格很容易。首先利用 `numpy.linspace()` 生成极角、极轴的数组，然后用 `numpy.meshgrid()` 生成极坐标网格坐标，最后再将其从极坐标转化为平面直角坐标系坐标。

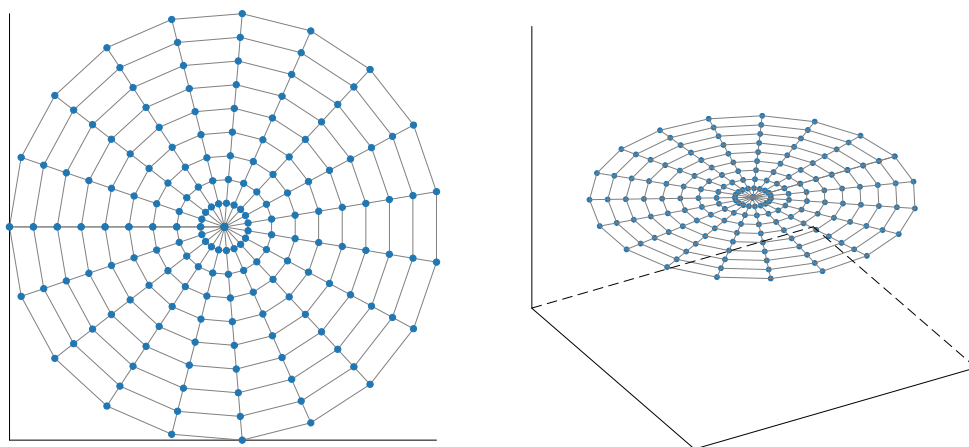


图 8. 极坐标网格

## 14.3 三角网格

本书后续还会使用三角网格完成特定的可视化方案。三角形网格是由一系列三角形所组成的网格结构。在计算机图形学和计算机模拟等领域，三角形网格常被用于表示复杂的几何体，如曲面、体细胞等，它可以通过三角形边界的拼接来逼近这些复杂的几何形状。三角形网格也常被用于数值计算中，如有限元分析等，因为三角形具有良好的性质，如易于计算、几何尺寸不变等。

三角形网格可以由多种方式生成，其中最常见的是 Delaunay 三角剖分，该方法可以将给定的点集分割成一组不重叠、不交叉的三角形。在 Delaunay 三角剖分中，对于任意三角形，其外接圆不包含其他点，这种性质可以保证三角形的质量较高，从而使得数值计算更加准确和稳定。

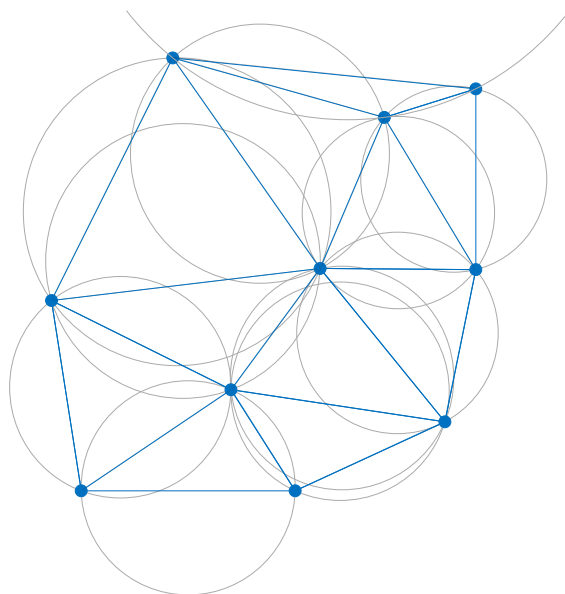


图 9. Delaunay 三角剖分法

`matplotlib.tri` 是一个 Python 库，用于创建和操作三角形网格。它提供了许多用于可视化和分析三角形网格的功能。

`matplotlib.tri` 可以创建三角形网格。可以使用 `Triangulation` 类从给定的点集中创建一个三角形网格，也可以使用其他函数生成各种类型的网格，如 Delaunay 三角剖分等。

`matplotlib.tri` 还可以可视化三角形网格。可以使用 `tripcolor`、`tricontour` 等函数在三角形网格上绘制颜色填充、等高线图等。

`matplotlib.tri` 也可以操作三角形网格。比如，可以使用 `TriAnalyzer`、`TriInterpolator` 等类对三角形网格进行分析、插值等操作。

总的来说，`matplotlib.tri` 为处理三角形网格提供了很多方便的工具和函数，使得用户可以方便地进行可视化和分析。

图 10 所示为常见三种三角网格。本书后续还将深入介绍三角形网格及其应用场景。

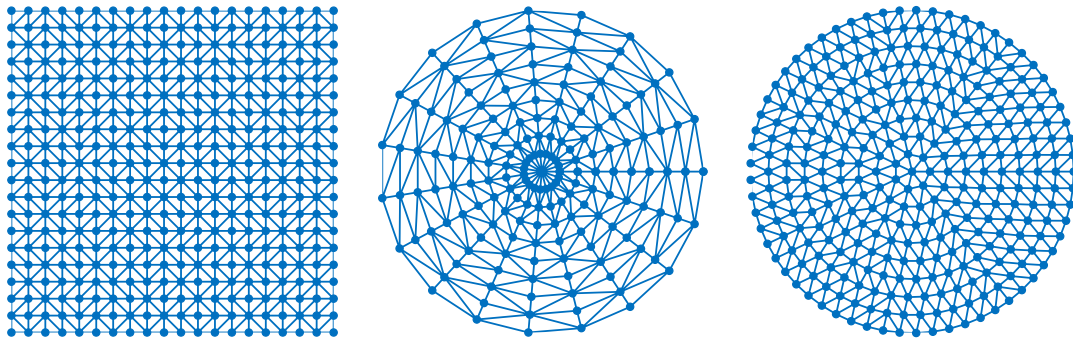
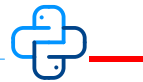


图 10. 三角网格

大家如果对 Delaunay 三角剖分法感兴趣的话，可以参考：

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



Jupyter 笔记 BK\_2\_Ch14\_1.ipynb 绘制本章大部分图片。