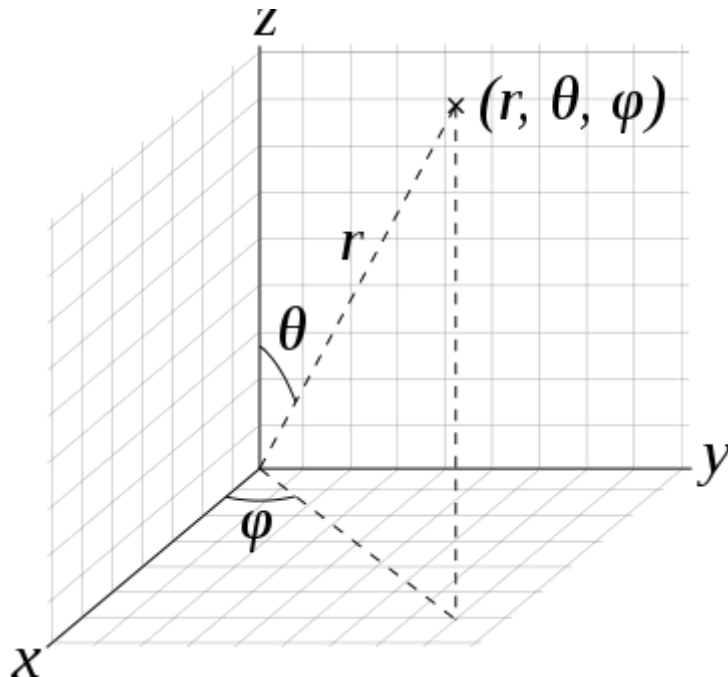


# 使用Python绘制空间球体

## 点阵的生成和球体的显示

### 数学依据



球坐标系参数方程

$$\begin{cases} x = r \sin \theta \cos \phi \\ y = r \sin \theta \sin \phi \\ z = r \cos \theta \end{cases}$$

### 代码依据

依赖包: matplotlib, numpy

API清单:

```
1  numpy.linspace()           #生成一维采样序列
2  numpy.meshgrid()          #生成网格
3  numpy.array()             #生成numpy数组
4  numpy.ones()              #生成全1数组
5  numpy.sin()               #数组的正弦函数
6  numpy.cos()               #数组的余弦函数
7
8  pyplot.figure()           #初始化窗口, 返回句柄
9  pyplot.show()             #显示图像
10 pyplot.gca()              #取得坐标轴句柄
11
```

```
12 Axes3D.plot_surface()      #绘制曲面
13 Axes3D.set_xlim()         #设置x轴范围
14 Axes3D.set_ylim()         #设置y轴范围
15 Axes3D.set_zlim()         #设置z轴范围
```

可以在python交互式环境下导入模块 `import numpy` 后

使用 `help()` 命令查看函数原型

例如:

```
1 import numpy
2 help(numpy.array)
```

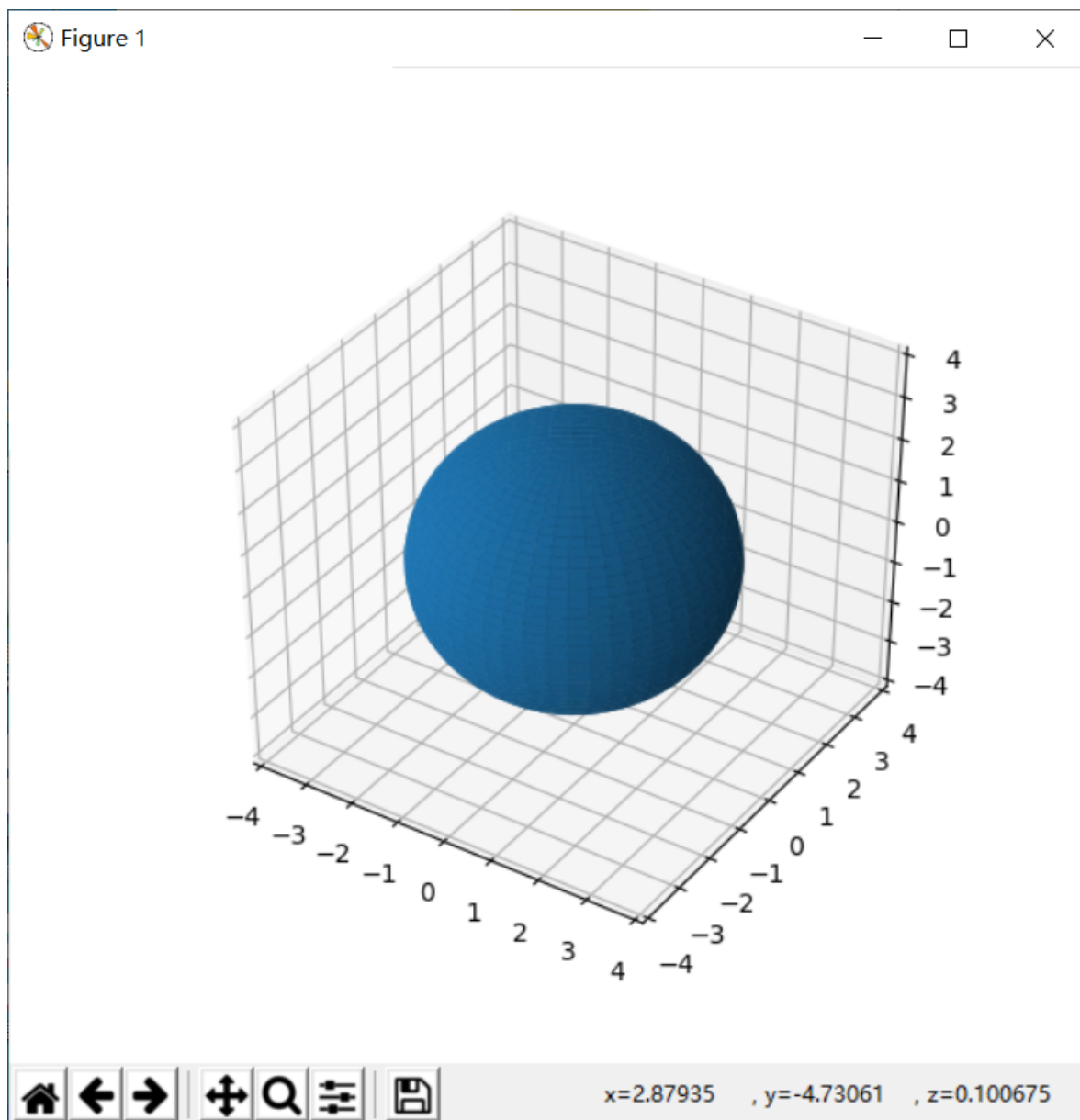
熟悉matlab的小伙伴们肯定会发现python的绘图接口和matlab非常相似，几乎一模一样。

———尼斯湖水怪

## 具体实现

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 theta = np.linspace(0,np.pi,50)
6 phi = np.linspace(0,2*np.pi,50)
7 r = 3
8
9 s_range = 4
10
11 fig = plt.figure()
12 ax = fig.gca(projection="3d")
13 ax.set_xlim(-s_range,s_range)
14 ax.set_ylim(-s_range,s_range)
15 ax.set_zlim(-s_range,s_range)
16
17 (theta_grid,phi_grid) = np.meshgrid(theta,phi)
18 x = r*np.sin(theta_grid)*np.cos(phi_grid)
19 y = r*np.sin(theta_grid)*np.sin(phi_grid)
20 z = r*np.cos(theta_grid)
21
22 ax.plot_surface(x,y,z)
23 plt.show()
```

## 运行效果演示



## 深入理解绘图原理

一切都是基于 `Axes3D.plot_surface(x,y,z)` 的参数形式而建立的3D描述

下面举例说明该函数的使用

核心算法 `meshgrid()` 将生成两类点阵

```
1 xx = np.array(range(1,5))
2 yy = np.array(range(1,9))
3 (X,Y) = np.meshgrid(xx,yy)
```

此时

$$x = (1 \quad 2 \quad 3 \quad 4)$$

$$y = (1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8)$$

## x类二维点阵

$$\mathbf{X} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

## y类二维点阵

$$\mathbf{Y} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 \\ 8 & 8 & 8 & 8 \end{pmatrix}$$

## 点阵运算

### 引入对点阵运算的定义（不听不听王八念经）

若  $\mathbf{Z} = (z_{ij})_{m \times n}$ ,  $\mathbf{X} = (x_{ij})_{m \times n}$ ,  $\mathbf{Y} = (y_{ij})_{m \times n}$ , 其中  $1 \leq i \leq m, 1 \leq j \leq n$ , 则

$$z_{ij} = f(x_{ij}, y_{ij})$$

记作

$$\mathbf{Z} = \mathbf{F}(\mathbf{X}, \mathbf{Y})$$

查阅手册可知, `numpy` 库中的数学函数基本上都符合以上的  $\mathbf{F}$  的定义。

简单来说都是对**相同位置**上的元素进行运算。

—————尼斯湖水怪

此时  $z_{ij}$  就表示二维平面上  $(x_{ij}, y_{ij})$  处的**高度**。

通过这样的运算, 确定了三维空间的所有采样点的位置

$$P_n : (x_{ij}, y_{ij}, z_{ij})$$

在调用 `Axes3D.plot_surface()` 的时候, 它的三个参数分别是上述的  $\mathbf{X}$ 、 $\mathbf{Y}$ 、 $\mathbf{Z}$ 。