11章-Python 3D

任务目标

- [x] 3D图形直角坐标系，极坐标，球面坐标系
- [x] 使用Matplotlib的3D Axes对象创建3D图形
- [x] 3D图形-空间曲线，空间曲面、动画
- [x] 3D模型-OpenGL

11.1 Matplotlib 3D绘图

- [x] 3D--直角坐标系-参数方程

$$\begin{cases} x = \varphi(t) \\ y = \phi(t) \\ z = \omega(t) \end{cases}$$

- [x] 3D--柱面坐标系

$$\begin{cases} x = \rho\cos(\theta) \\ y = \rho\sin(\theta) \\ z = z \end{cases}$$

- [x] 3D--球面坐标系

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

- [x] 3D图形直角坐标系，极坐标，球面坐标系

Matplotlib支持3D Axes对象

1、通过在add_subplot()函数传递projection="3d" 参数就可以创建3D Axes对象。

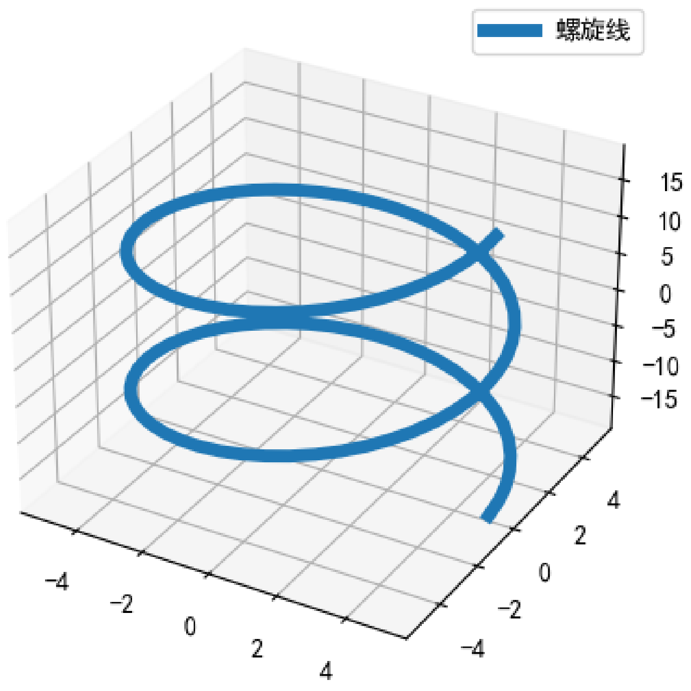
```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
```

1、绘制空间曲线

`ax.plot（x,y,z)` 绘制空间曲线，其中x、y、z变量是一维数组，plot函数通常使用参数方程。

$$\begin{cases} x = a\cos(\theta) \\ y = a\sin(\theta) \\ z = b\theta \end{cases}$$

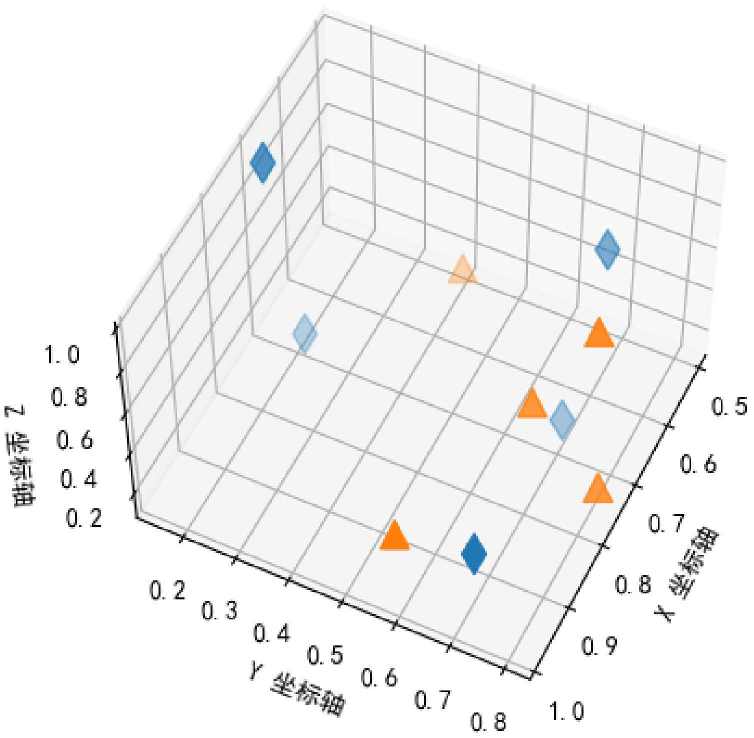
```
import random
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
plt.rcParams['xtick.direction'] = 'in' #x的刻度向内
plt.rcParams['ytick.direction'] = 'in' #y的刻度向内
plt.rcParams['lines.linewidth'] = 5
plt.rcParams['lines.color'] ='y'
if __name__=="__main__":
  fig = plt.figure();
  ax = fig.add_subplot(111, projection='3d');
  a = 5;
  b = 3;
  theta = np.linspace(-2*np.pi,2*np.pi,100);
  x = a * np.cos(theta);
  y = a * np.sin(theta);
  z  = b*theta;
  ax.plot(x,y,z,label='螺旋线');
  ax.legend();
  plt.show();
```



2、绘制空间散点图

`ax.scatter（x,y,z)` 其中scatter函数的x、y、z变量是一维数组。

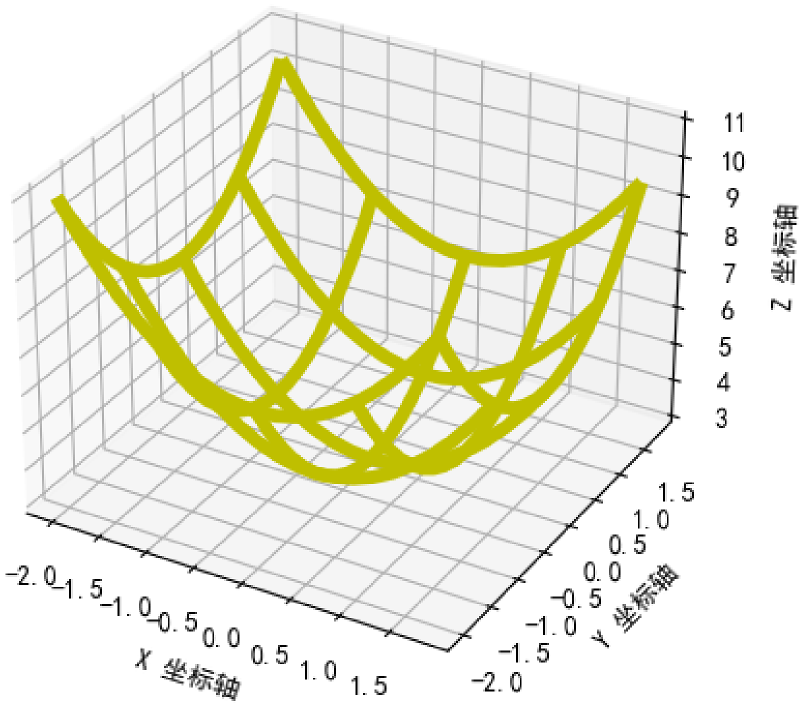
```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
plt.rcParams['xtick.direction'] = 'in' #x的刻度向内
plt.rcParams['ytick.direction'] = 'in' #y的刻度向内
plt.rcParams['lines.linewidth'] = 5
plt.rcParams['lines.color'] ='y'
np.random.seed(19680801)
if __name__=="__main__":
    fig = plt.figure();
    ax = fig.add_subplot(projection='3d');
    n = 5;
for marker in ['d','^']:
    x = np.random.rand(n);
    y = np.random.rand(n);
    z = np.random.rand(n);
    ax.scatter(x, y, z,marker=marker,s=120);
ax.set_xlabel('X 坐标轴');
ax.set_ylabel('Y 坐标轴');
ax.set_zlabel('Z 坐标轴');
plt.show();
```



3、绘制空间网格图

`ax.plot_wireframe`(*self*, *X*, *Y*, *Z*, args, kwargs)，其中X, Y, Z必须是2D arrays。

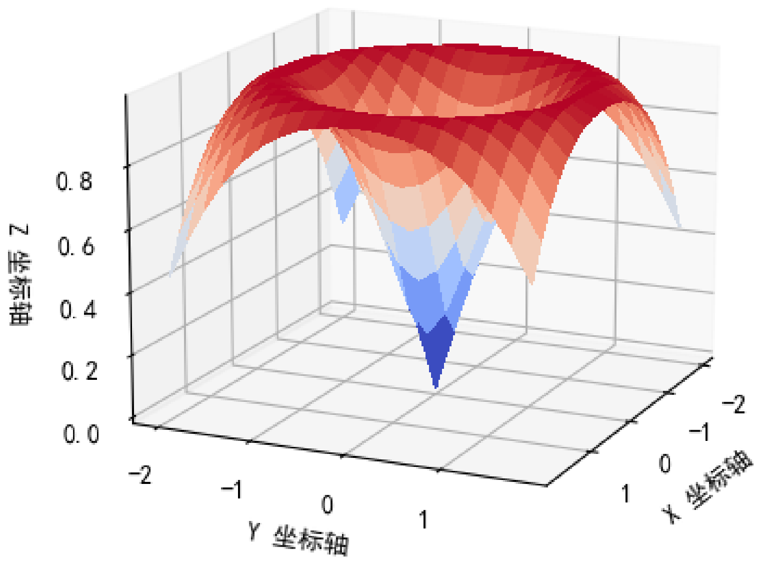
```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
plt.rcParams['xtick.direction'] = 'in' #x的刻度向内
plt.rcParams['ytick.direction'] = 'in' #y的刻度向内
plt.rcParams['lines.linewidth'] = 5
plt.rcParams['lines.color'] ='y'
np.random.seed(19680801)
if __name__=="__main__":
    fig = plt.figure();
    ax = fig.add_subplot(projection='3d');
    n = 5;
    x = np.arange(-2,2,0.2);
    y = np.arange(-2,2,0.2);
    X, Y = np.meshgrid(x,y);
    Z = X*X + Y*Y +3;
    ax.plot_wireframe(X,Y,Z,rstride=5, cstride=5);
ax.set_xlabel('X 坐标轴');
ax.set_ylabel('Y 坐标轴');
ax.set_zlabel('Z 坐标轴');
plt.show();
```



4、绘制空间曲面

`ax.plot_surface`(*self*, args, color=None, *norm=None*, *vmin=None*, *vmax=None*, lightsource=None, kwargs)，其中X, Y, Z是 2D arrays。

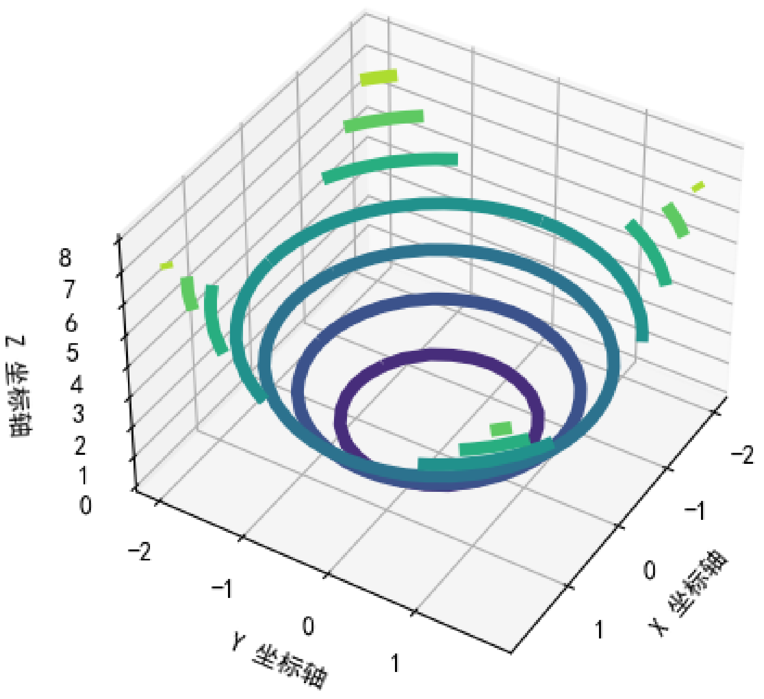
```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
plt.rcParams['xtick.direction'] = 'in' #x的刻度向内
plt.rcParams['ytick.direction'] = 'in' #y的刻度向内
plt.rcParams['lines.linewidth'] = 5
plt.rcParams['lines.color'] ='y'
np.random.seed(19680801)
if __name__=="__main__":
    fig = plt.figure();
    ax = fig.add_subplot(projection='3d');
    n = 5;
    x = np.arange(-2,2,0.2);
    y = np.arange(-2,2,0.2);
    X, Y = np.meshgrid(x,y);
    Z = np.sin(np.sqrt(X**2+Y**2));
    ax.plot_surface(X,Y,Z,cmap=cm.coolwarm,linewidth=0, antialiased=False);
ax.set_xlabel('X 坐标轴');
ax.set_ylabel('Y 坐标轴');
ax.set_zlabel('Z 坐标轴');
plt.show();
```



5、绘制等高线（Contour）

`ax.contour`(*self*, *X*, *Y*, *Z*, **args*, *extend3d=False*, *stride=5*, *zdir='z'*, *offset=None*, ***kwargs*),

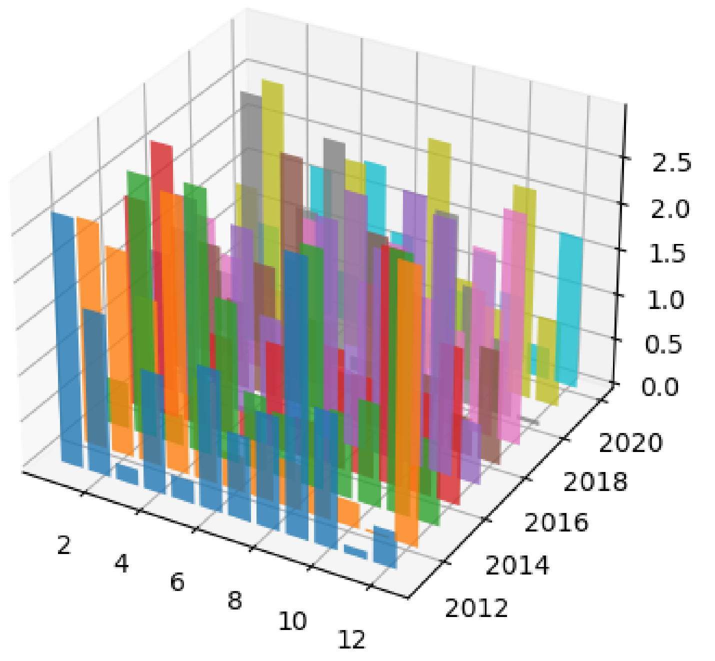
```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
plt.rcParams['xtick.direction'] = 'in' #x的刻度向内
plt.rcParams['ytick.direction'] = 'in' #y的刻度向内
plt.rcParams['lines.linewidth'] = 5
plt.rcParams['lines.color'] ='y'
np.random.seed(19680801)
if __name__=="__main__":
    fig = plt.figure();
    ax = fig.add_subplot(projection='3d');
    n = 5;
    x = np.arange(-2,2,0.2);
    y = np.arange(-2,2,0.2);
    X, Y = np.meshgrid(x,y);
    Z = X*X + Y*Y;
    ax.contour(X,Y,Z,zdir='z');
ax.set_xlabel('X 坐标轴');
ax.set_ylabel('Y 坐标轴');
ax.set_zlabel('Z 坐标轴');
plt.show();
```



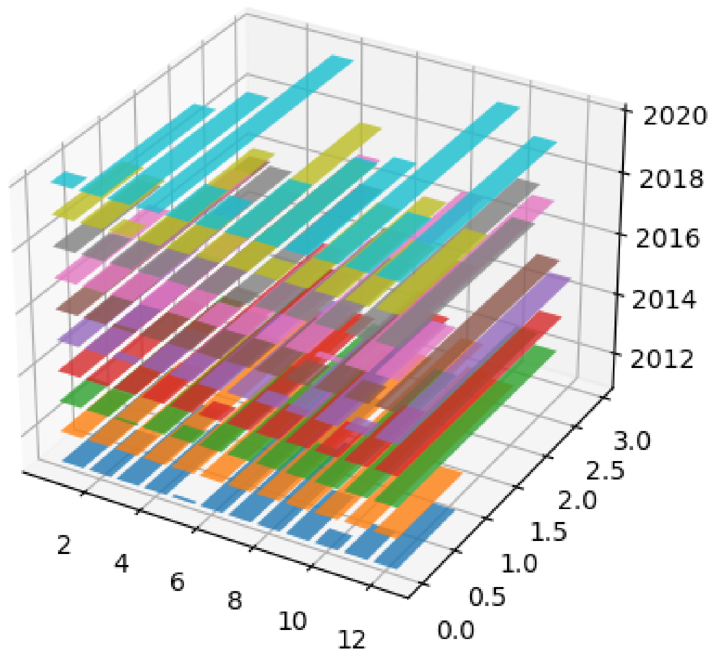
6、绘制空间柱状图

`ax.bar（xs,ys,zs）` x轴、y轴、z轴坐标值、zdir决定哪个坐标轴作为z轴的维度。通过Axes3D绘制3D柱状图，ax.bar（xs,ys,zs,zdir) x轴、y轴、z轴坐标值、zdir决定哪个坐标轴作为z轴的维度。

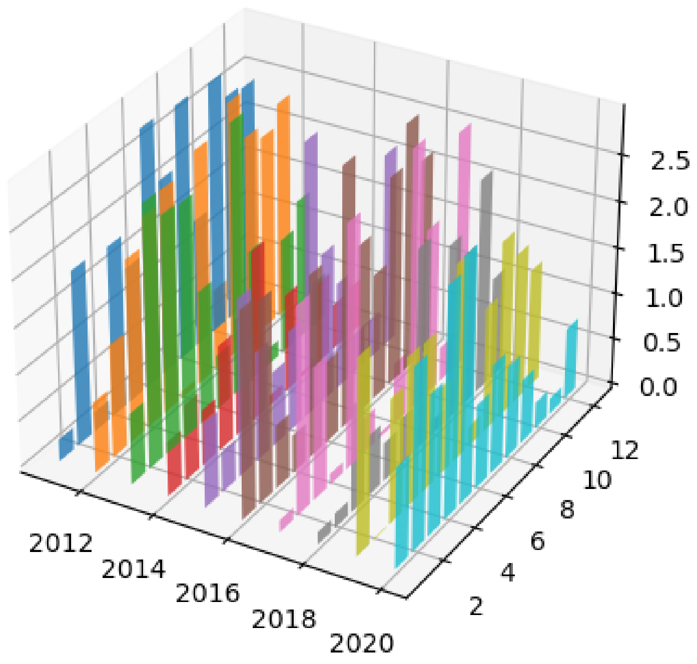
```
import random
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

if __name__=="__main__":
  mpl.rcParams['font.size'] = 10;
  fig = plt.figure();
  ax = fig.add_subplot(111, projection='3d');
  for z in [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019,2020]:
    xs = range(1,13);
    ys = 3*np.random.rand(12);
    ax.bar(xs, ys, zs=z, zdir='y', alpha=0.8);
  plt.show();
```



```
ax.bar(xs, ys, zs=z, zdir='z', alpha=0.8);
```



```
ax.bar(xs, ys, zs=z, zdir='x', alpha=0.8);
```



7、绘制Pringle函数的三翼面图

`ax.plot_trisurf（x,y,z)` x轴、y轴、z轴坐标值、zdir决定哪个坐标轴作为z轴的维度。

```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
import matplotlib.pyplot as plt
import numpy as np

n_angles = 36
n_radii = 8
radii = np.linspace(0.125, 1.0, n_radii)
angles = np.linspace(0, 2 * np.pi, n_angles, endpoint=False)
angles = np.repeat(angles[..., np.newaxis], n_radii, axis=1)
x = np.append(0, (radii * np.cos(angles)).flatten())
y = np.append(0, (radii * np.sin(angles)).flatten())
z = np.sin(-x * y)
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_trisurf(x, y, z, cmap=cm.jet, linewidth=0.2)
plt.show()
```

8、创建3D直方图

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
mpl.rcParams['font.size'] = 10
samples = 25
x = np.random.normal(5, 1, samples)
```

```
y = np.random.normal(3, .5, samples)
fig = plt.figure()
ax = fig.add_subplot(211, projection='3d')
hist, xedges, yedges = np.histogram2d(x, y, bins=10)
elements = (len(xedges) - 1) * (len(yedges) - 1)
xpos, ypos = np.meshgrid(xedges[:-1]+.25, yedges[:-1]+.25)
xpos = xpos.flatten()
ypos = ypos.flatten()
zpos = np.zeros(elements)
dx = .1 * np.ones_like(zpos)
dy = dx.copy()
dz = hist.flatten()
ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color='b', alpha=0.4)
ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')
ax.set_zlabel('Z Axis')
ax2 = fig.add_subplot(212)
ax2.scatter(x, y)
ax2.set_xlabel('X Axis')
ax2.set_ylabel('Y Axis')
plt.show()
```

9、在matplotlib中创建动画

1、下载ffmpeg-win64-static.zip文件，并将ffmpeg.exe文件放置在代码文件夹中，

```
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import animation
fig = plt.figure()
ax = plt.axes(xlim=(0, 2), ylim=(-2, 2))
line, = ax.plot([], [], lw=2)
def init():
    line.set_data([], [])
    return line,
def animate(i):
    x = np.linspace(0, 2, 1000)
    y = np.sin(2 * np.pi * (x - 0.01 * i)) * np.cos(22 * np.pi * (x - 0.01 * i))
    line.set_data(x, y)
    return line,
animator = animation.FuncAnimation(fig, animate, init_func=init,frames=200,
interval=20, blit=True)
animator.save('basic_animation1.mp4', fps=30, extra_args=['-vcodec',
'libx264'],writer='ffmpeg_file')
plt.show()
```

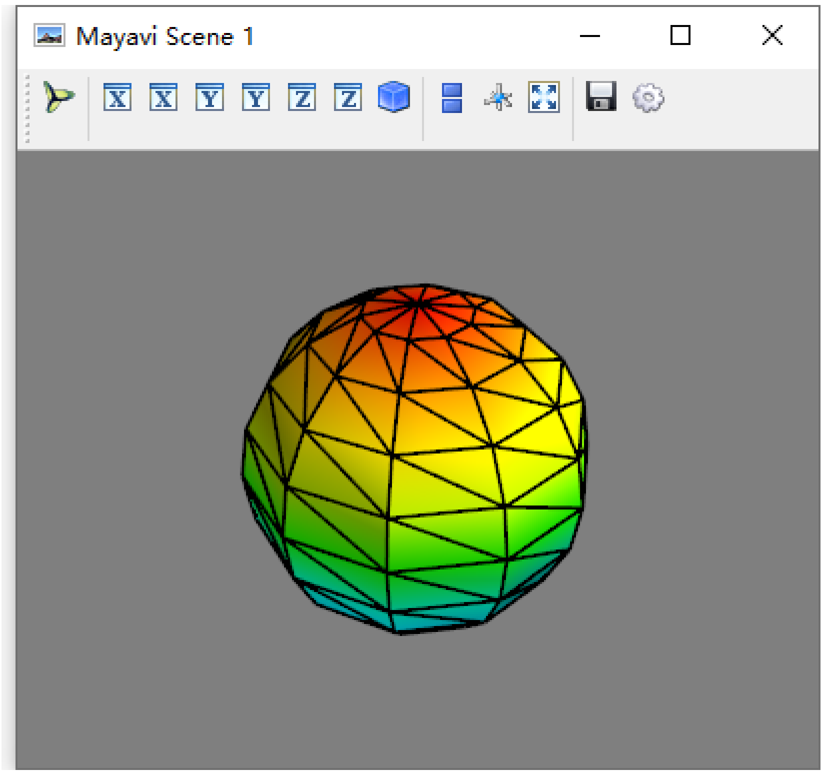
11.2 OpenGL 3D绘图

Python支持OpenGL 3D对象绘图

```
pip3 install traits-6.0.0-cp37-cp37m-win_amd64.whl
pip3 install PyQt4-4.11.4-cp37-cp37m-win_amd64.whl
pip3 install VTK-8.2.0-cp37-cp37m-win_amd64.whl
pip3 install mayavi-4.7.1+vtk82-cp37-cp37m-win_amd64.whl
```

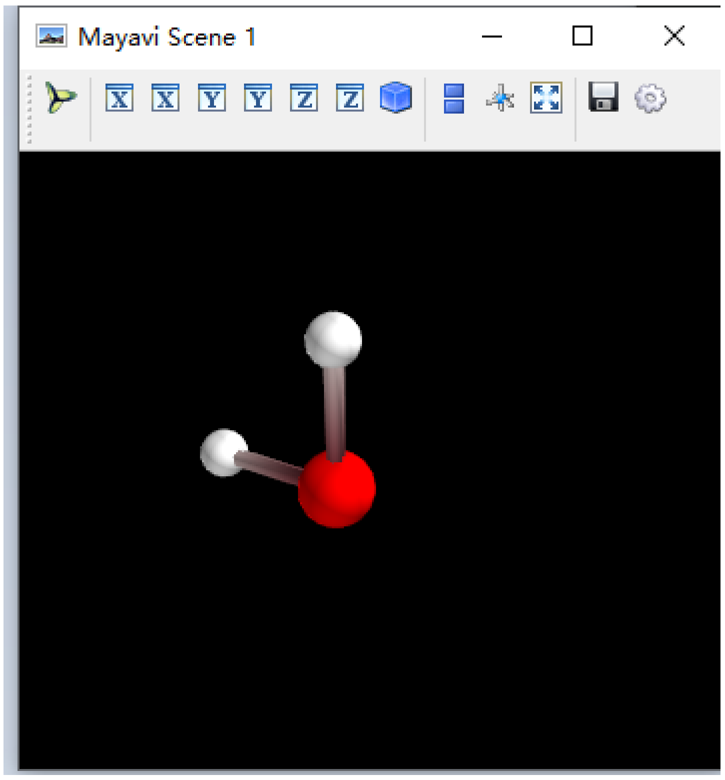
1、绘制三维球面

```
import numpy as np
from mayavi import mlab
mlab.clf()
phi, theta = np.mgrid[0:np.pi:11j, 0:2*np.pi:11j]
x = np.sin(phi) * np.cos(theta)
y = np.sin(phi) * np.sin(theta)
z = np.cos(phi)
mlab.mesh(x, y, z)
mlab.mesh(x, y, z, representation='wireframe', color=(0, 0, 0))
mlab.show()
```



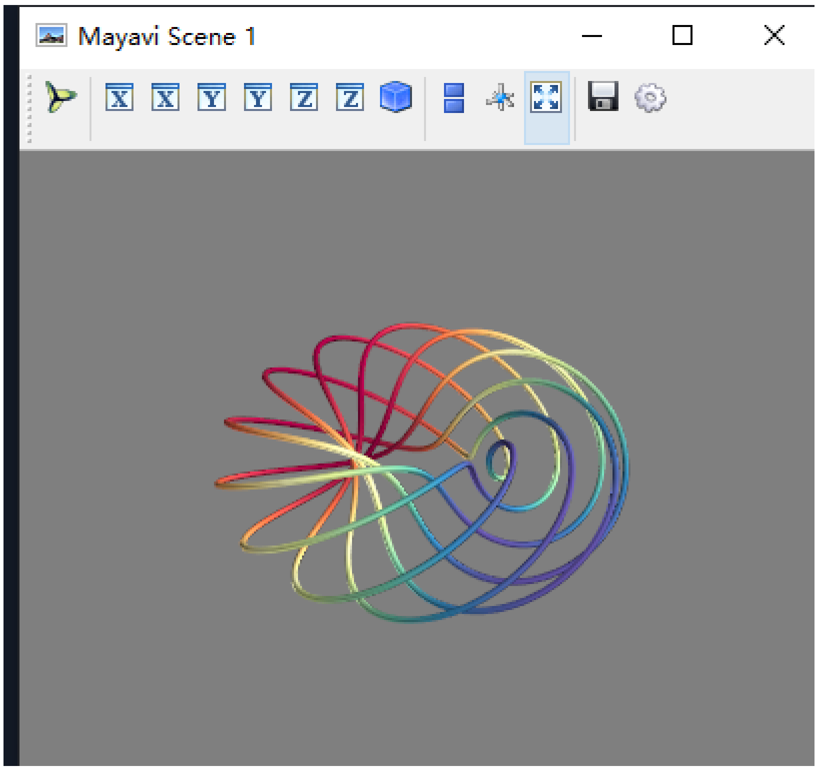
2、绘制分子结构

```
import numpy as np
from mayavi import mlab
mlab.figure(1, bgcolor=(0, 0, 0), size=(350, 350))
mlab.clf()

atoms_x = np.array([2.9, 2.9, 3.8]) * 40 / 5.5
atoms_y = np.array([3.0, 3.0, 3.0]) * 40 / 5.5
atoms_z = np.array([3.8, 2.9, 2.7]) * 40 / 5.5

O = mlab.points3d(atoms_x[1:-1], atoms_y[1:-1], atoms_z[1:-1],
                  scale_factor=3,
                  resolution=20,
                  color=(1, 0, 0),
                  scale_mode='none')

H1 = mlab.points3d(atoms_x[:1], atoms_y[:1], atoms_z[:1],
                   scale_factor=2,
                   resolution=20,
                   color=(1, 1, 1),
                     scale_mode='none')

H2 = mlab.points3d(atoms_x[-1:], atoms_y[-1:], atoms_z[-1:],
                     scale_factor=2,
                     resolution=20,
                     color=(1, 1, 1),
                     scale_mode='none')

mlab.plot3d(atoms_x, atoms_y, atoms_z, [1, 2, 1],tube_radius=0.4,
colormap='Reds')
mlab.view(132, 54, 45, [21, 20, 21.5])
mlab.show()
```



3、绘制旋转图形

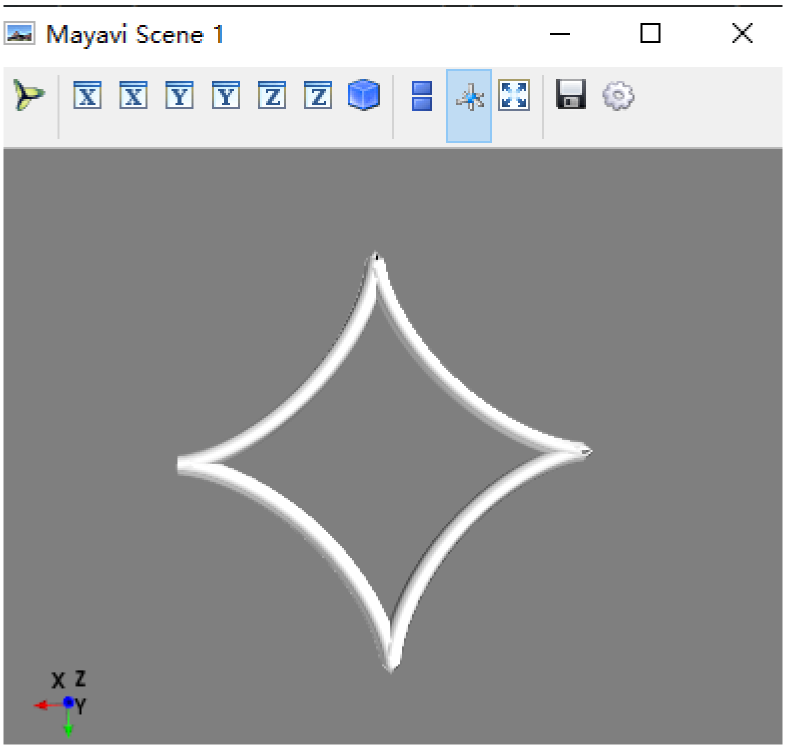
```
import numpy
import mayavi.mlab as mlab
n_mer, n_long = 6, 11
pi = numpy.pi
dphi = pi/1000.0
phi = numpy.arange(0.0, 2*pi + 0.5*dphi, dphi, 'd')
mu = phi*n_mer
x = numpy.cos(mu)*(1+numpy.cos(n_long*mu/n_mer)*0.5)
y = numpy.sin(mu)*(1+numpy.cos(n_long*mu/n_mer)*0.5)
z = numpy.sin(n_long*mu/n_mer)*0.5
l = mlab.plot3d(x, y, z, numpy.sin(mu), tube_radius=0.025, colormap='Spectral')
ms = l.mlab_source
for i in range(100):
    x = numpy.cos(mu)*(1+numpy.cos(n_long*mu/n_mer + numpy.pi*(i+1)/5.)*0.5)
    scalars = numpy.sin(mu + numpy.pi*(i+1)/5)
    ms.set(x=x, scalars=scalars)
mlab.show()
```



4、绘制星形线

根据以下公式，绘制星形线（$a=2$）。

$$\begin{cases} x = acos^3(\theta) & 0 \le \theta \le 2\pi \\ y = asin^3(\theta) & 0 \le \theta \le 2\pi \end{cases}$$


Mayavi Scene 1
X Z
Y