Mayavi库的安装

http://docs.enthought.com/mayavi/mayavi/index.html

mayavi 4.5.0 documentation » next | index

Mayavi

Next topic

An overview of Mayavi

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

Google Search

Go

only search Mayavi documentation

Citing Mayavi

If you publish articles using Mayavi, please **cite Mayavi**. We need these citations to justify

Mayavi: 3D scientific data visualization and plotting in Python

Using the Mayavi application

Understanding and using the Mayavi interactive application

Python scripting for 3D plotting

The simple scripting API to Mayavi

Gallery and examples

Example gallery of visualisations, with the Python code that generates them

Welcome, this is the user guide for Mayavi, a application and library for **interactive scientific data visualization** and **3D plotting in Python**.

Getting started

- You want to use an interactive application to visualize your data in 3D? Read the Mayavi application section.
- You know Python and want to use Mayavi as a Matlab or pylab replacement for 3D plotting and data visualization with *numpy*? Get started with the mlab section.
- Sources of inspiration may be found in the Example gallery, with example Python code.

Mayavi库的安装

安装Mayavi的基本要求：

- VTK
- numpy
- Traits（Traits、TraitsUI 和 TraitsBackendWX/TraitsBackendQT）

Mayavi库的安装

VTK-7.1.1-cp36-cp36m-win_amd64.whl

mayavi-4.5.0-cp36-cp36m-win_amd64.whl

PyQt4-4.11.4-cp36-cp36m-win_amd64.whl

BuildingTools

Mayavi库的安装

```
pip install mayavi-4.5.0-cp36-cp36m-win_amd64.whl
```

管理员: 命令提示符

```
C:\Tvtk>pip install mayavi-4.5.0-cp36-cp36m-win_amd64.whl
Processing c:\tvtk\mayavi-4.5.0-cp36-cp36m-win_amd64.whl
Requirement already satisfied: traitsui in c:\python36\lib\site-packages (from mayavi==4.5.0)
Requirement already satisfied: traits in c:\python36\lib\site-packages (from mayavi==4.5.0)
Requirement already satisfied: apptools in c:\python36\lib\site-packages (from mayavi==4.5.0)
Requirement already satisfied: pyface in c:\python36\lib\site-packages (from traitsui->mayavi==4.5.0)
Requirement already satisfied: configobj in c:\python36\lib\site-packages (from apptools->mayavi==4.5.0)
Requirement already satisfied: pygments in c:\python36\lib\site-packages (from pyface->traitsui->mayavi==4.5.0)
Requirement already satisfied: six in c:\python36\lib\site-packages (from configobj->apptools->mayavi==4.5.0)
Installing collected packages: mayavi
Successfully installed mayavi-4.5.0

C:\Tvtk>
```

Mayavi库的安装

代码编辑环境：Python3.6 自带的IDLE3.6

Pycharm Community Edition

Mayavi库的安装小测

```
>>> from mayavi import mlab
```

Mayavi库的基本元素

Mayavi.mlab

类 别	说明
绘图函数	barchar、contour3d、contour_surf、flow、imshow、mesh、plot3d、points3d、quiver3d、surf、triangular_mesh
图形控制函数	clf、close、draw、figure、gcf、savefig、screenshot、sync_camera
图形修饰函数	colorbar、scalarbar、xlabel、ylabel、zlabel
相机控制函数	move、pitch、roll、view、yaw
其他函数	animate、axes、get_engine、show、set_engine……
Mlab管线控制	Open、set_vtk_src、adddataset、scalar_cut_plane

Mayavi API

类 别	说明
管线基础对象	Scene、Source、Filter、ModuleManager、Module、PipelineBase、Engine
主视窗和UI对象	DecoratedScene、MayaviScene、SceneEditor、MlabSceneModel、EngineView、EngineRichView

快速绘图实例

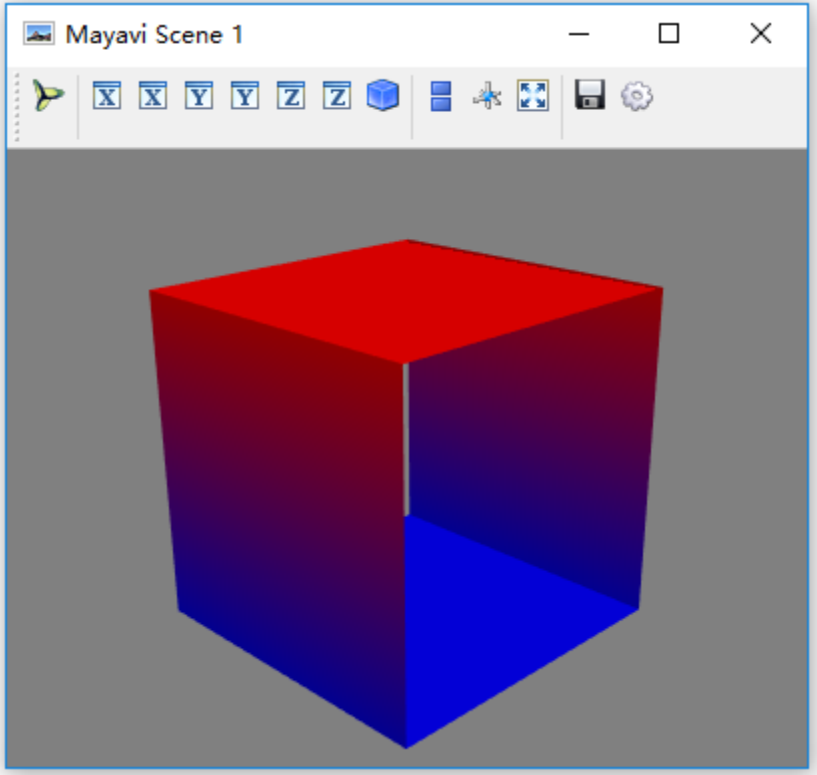
快速绘制实例1

```
>>> x=[[-1,1,1,-1,-1],[-1,1,1,-1,-1]]
>>> y=[[-1,-1,-1,-1,-1],[1,1,1,1, 1]]
>>> z=[[1,1,-1,-1,1],[1,1,-1,-1,1]]
>>> from mayavi import mlab
>>> s = mlab.mesh(x,y,z)
>>>
```

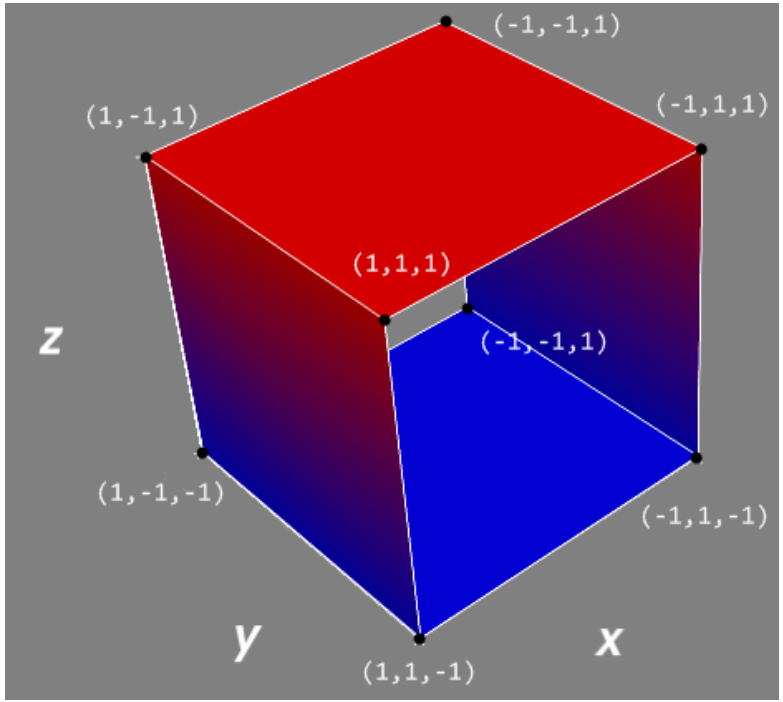
描述的坐标点为：

```
[[(-1, -1, 1), (1, -1, 1), (1, -1, -1), (-1, -1, -1), (-1, -1, 1)]
    [(-1, 1, 1), (1, 1, 1), (1, 1, -1), (-1, 1, -1), (-1, 1, 1)]]
```

快速绘制实例1



快速绘制实例1



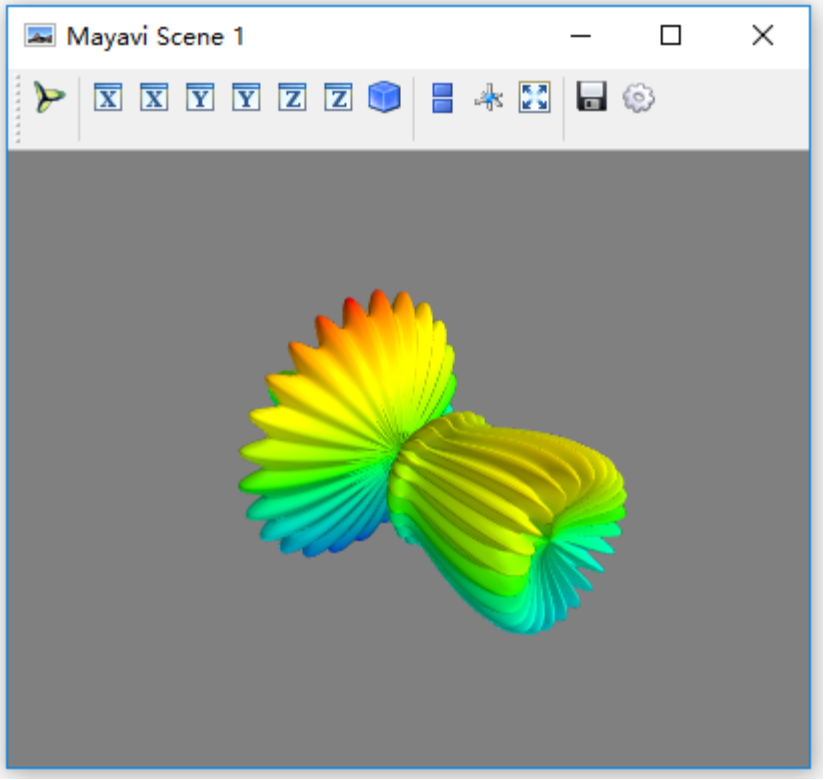
快速绘制实例2

```
from numpy import pi, sin, cos, mgrid
from mayavi import mlab
```

```
#建立数据
dphi, dtheta = pi/250.0, pi/250.0
[phi,theta] = mgrid[0:pi+dphi*1.5:dphi,0:2*pi+dtheta*1.5:dtheta]
m0 = 4; m1 = 3; m2 = 2; m3 = 3; m4 = 6; m5 = 2; m6 = 6; m7 = 4;
r = sin(m0*phi)**m1 + cos(m2*phi)**m3 + sin(m4*theta)**m5 + cos(m6*theta)**m7
x = r*sin(phi)*cos(theta)
y = r*cos(phi)
z = r*sin(phi)*sin(theta)
```

```
#对该数据进行三维可视化
s = mlab.mesh(x, y, z)
mlab.show()
```

快速绘制实例2

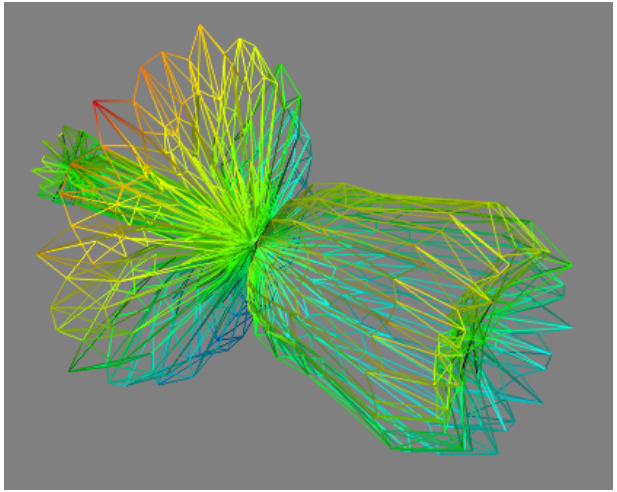


键盘鼠标对场景进行操作

- 旋转场景：左键拖动或键盘的方向键
- 平移场景：按住Shift键并使用左键拖动，shift+方向键盘
- 缩放场景：鼠标右键上下拖动或使用“+”和“-”按键
- 滚动相机：按住CTRL键并用左键拖动
- 工具栏：从坐标轴6个方向观察场景、等角投影、切换平行透视和成角透视等

快速绘制实例2

```
Mlab.mesh(x,y,z,representation=‘wireframe’,line_width=1.0)
```

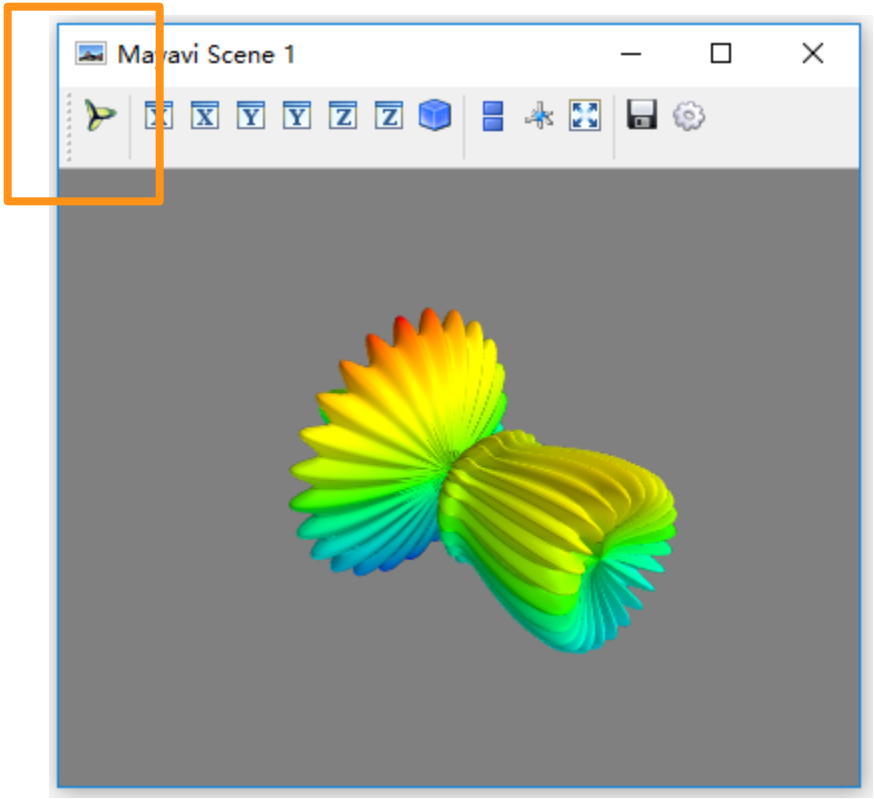


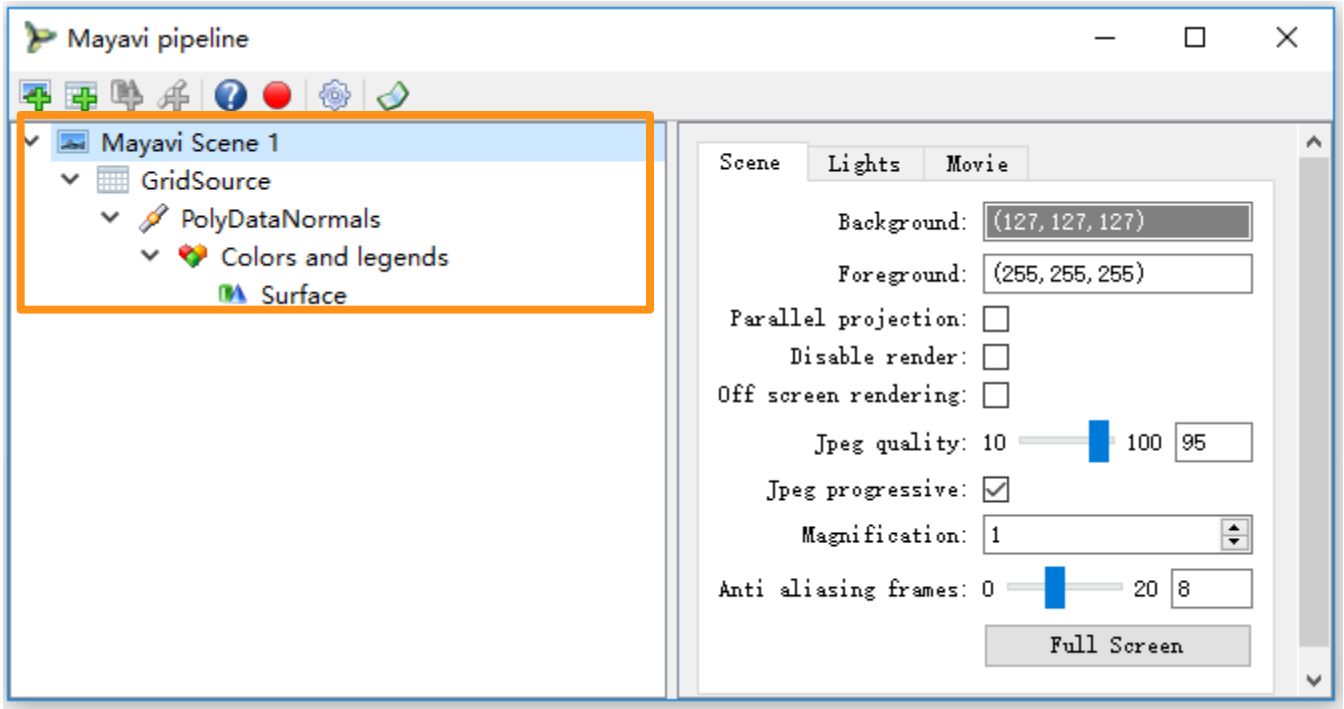
Mayavi管线

Mayavi管线的层级

- **Engine**：建立和销毁Scenes
- **Scenes**：多个数据集合Sources
- **Filters**：对数据进行变换
- **Module Manager**：控制颜色，Colors and Legends
- **Modules**：最终数据的表示，如线条、平面等

```
mlab.show_pipeline()
```




Mayavi pipeline
Mayavi Scene 1
GridSource
PolyDataNormals
Colors and legends
Surface
Scene
Lights
Movie
Background:
(127, 127, 127)
Foreground:
(255, 255, 255)
Parallel projection:
Disable render:
Off screen rendering:
Jpeg quality:
10
100
95
Jpeg progressive:
Magnification:
1
Anti aliasing frames:
0
20
8
Full Screen

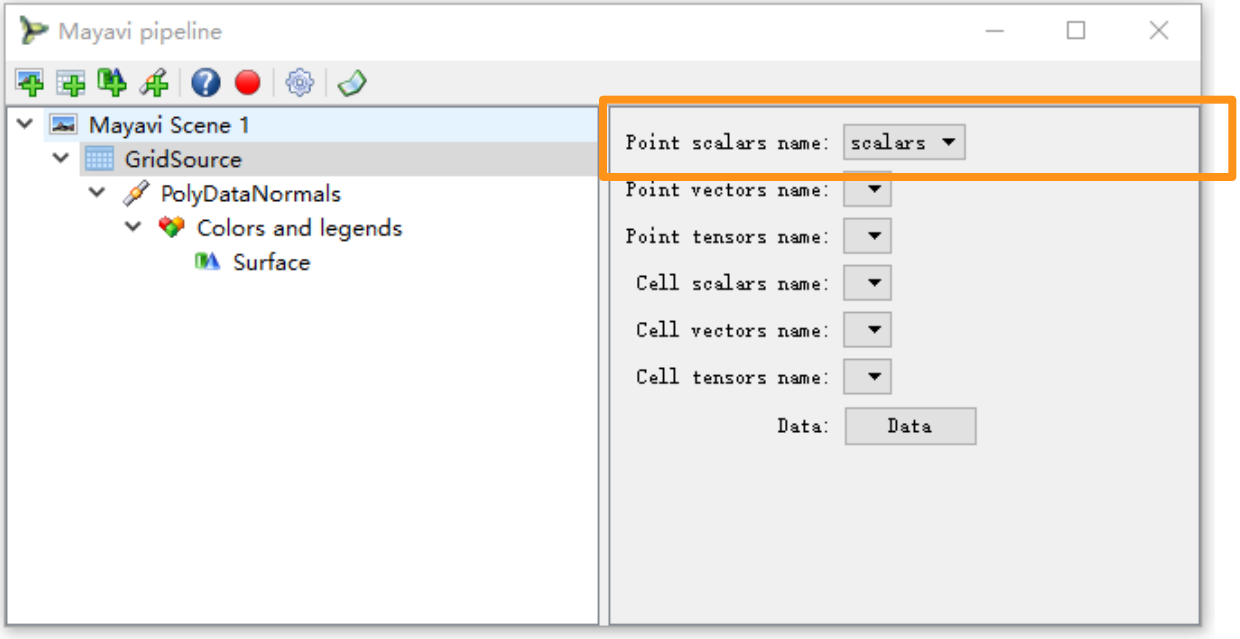
```python
from numpy import pi, sin, cos, mgrid
from mayavi import mlab

#建立数据
dphi, dtheta = pi/250.0, pi/250.0
[phi,theta] = mgrid[0:pi+dphi*1.5:dphi,0:2*pi+dtheta*1.5:dtheta]
m0 = 4; m1 = 3; m2 = 2; m3 = 3; m4 = 6; m5 = 2; m6 = 6; m7 = 4;
r = sin(m0*phi)**m1 + cos(m2*phi)**m3 + sin(m4*theta)**m5 + cos(m6*theta)**m7
x = r*sin(phi)*cos(theta)
y = r*cos(phi)
z = r*sin(phi)*sin(theta)

#对该数据进行三维可视化
s = mlab.mesh(x, y, z)
#mlab.show()
```

管线中的对象scene

Mayavi Scene：处于树的最顶层的对象，表示场景。

```
>>> s = mlab.gcf()
>>> print(s)
<mayavi.core.scene.Scene object at 0x00000249726A6E08>
>>> print(s.scene.background)
(0.5, 0.5, 0.5)
```

管线中的对象GridSource



管线中的对象GridSource

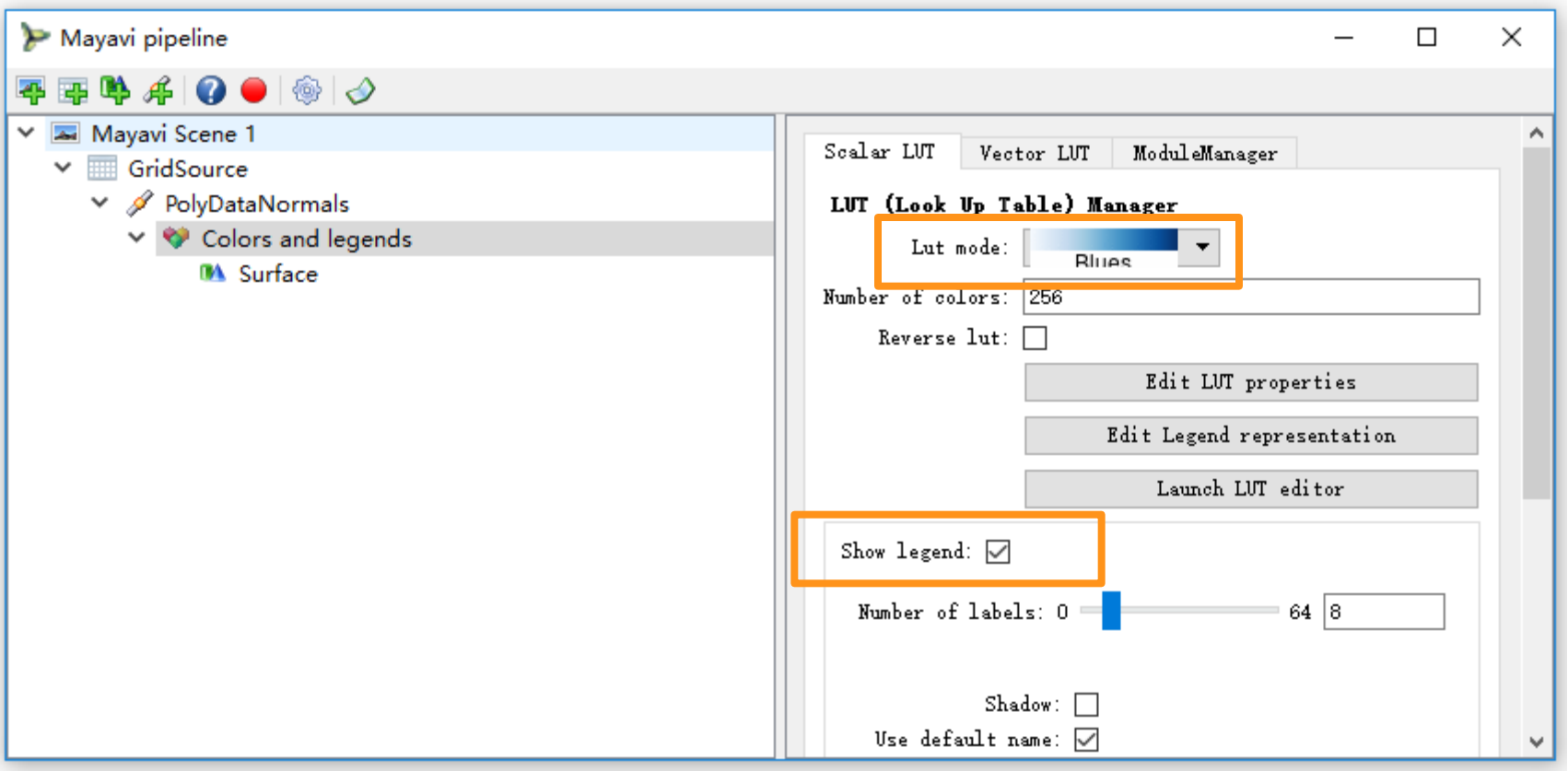
```
>>> source = s.children[0]
>>> print(repr(source))
<mayavi.sources.vtk_data_source.VTKDataSource object at 0x0000024972A508E0>
>>> print(source.name)
GridSource
>>> print(repr(source.data.points))
[(0.0, 2.0, 0.0), ..., (-0.025048897296365225, -1.9933803751132322, -0.0003
1479029697865414)], length = 126504
>>> print(repr(source.data.point_data.scalars))
[0.0, ..., -0.00031479029697865414], length = 126504
>>>
```

管线中的对象PlolyDataNormals

PolyDataNormals：数据源的法向量

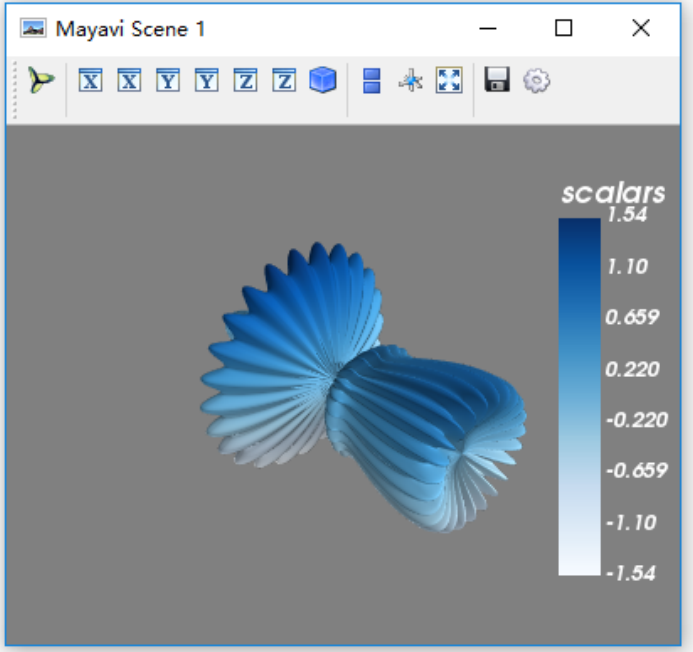
```
>>> manager = source.children[0]
>>> print(manager)
<mayavi.filters.poly_data_normals.PolyDataNormals object at 0x0000024972570
938>
```

管线中的对象Colors and legends

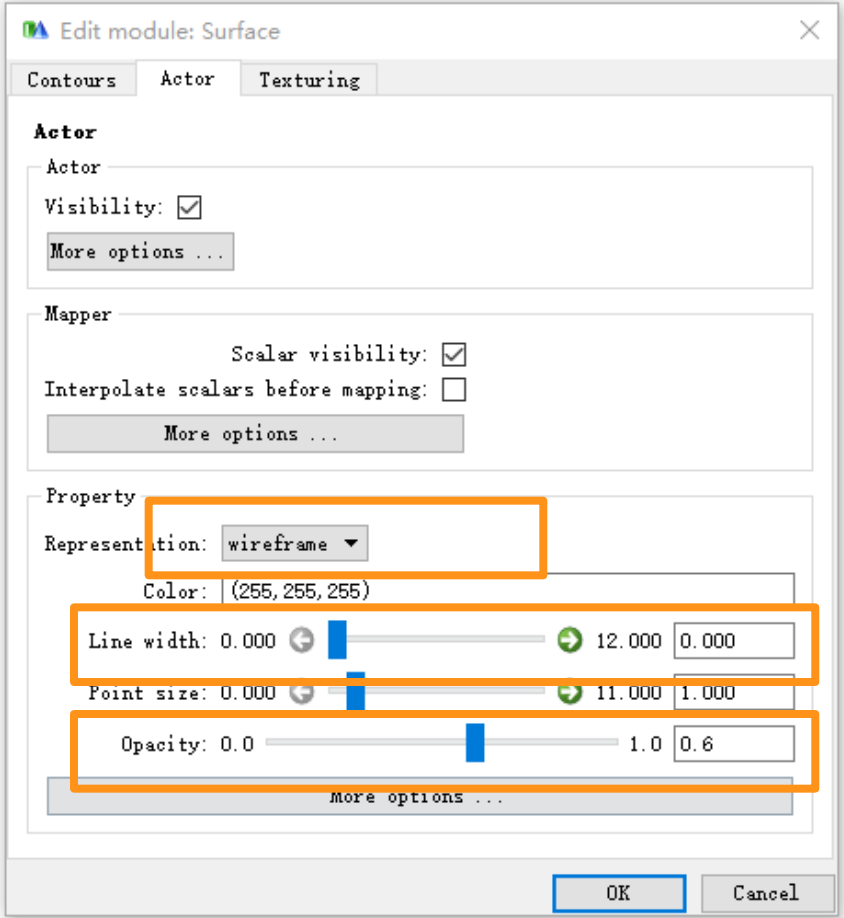


管线中的对象Colors and legends

```
>>> colors = manager.children[0]
>>> colors.scalar_lut_manager.lut_mode = 'Blues'
>>> colors.scalar_lut_manager.show_legend = True
```

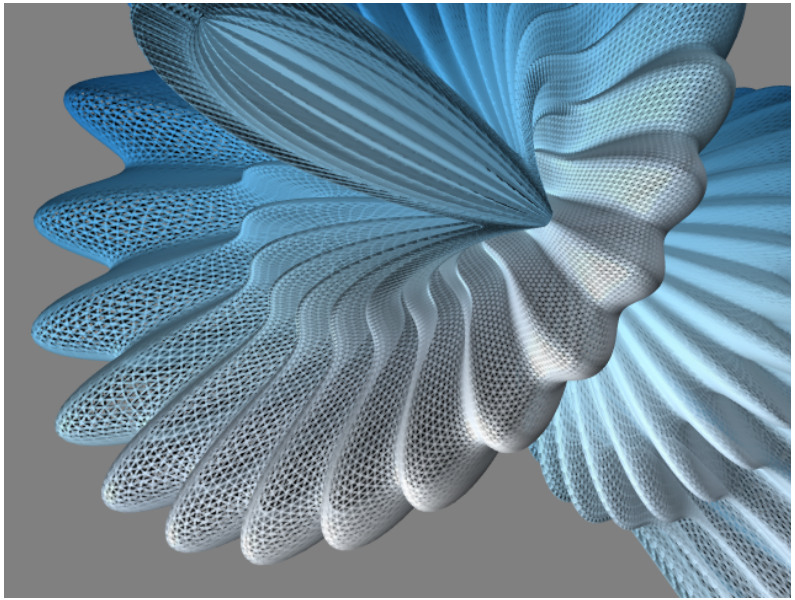


管线中的对象Surface



管线中的对象Surface

```
>>> surface = colors.children[0]
>>> surface.actor.property.representation = 'wireframe'
>>> surface.actor.property.opacity = 0.6

>>> mlab.show()
```



程序配置属性的步骤

1、获得场景对象，mlab.gcf()

2、通过children属性，在管线中找到需要修改的对象

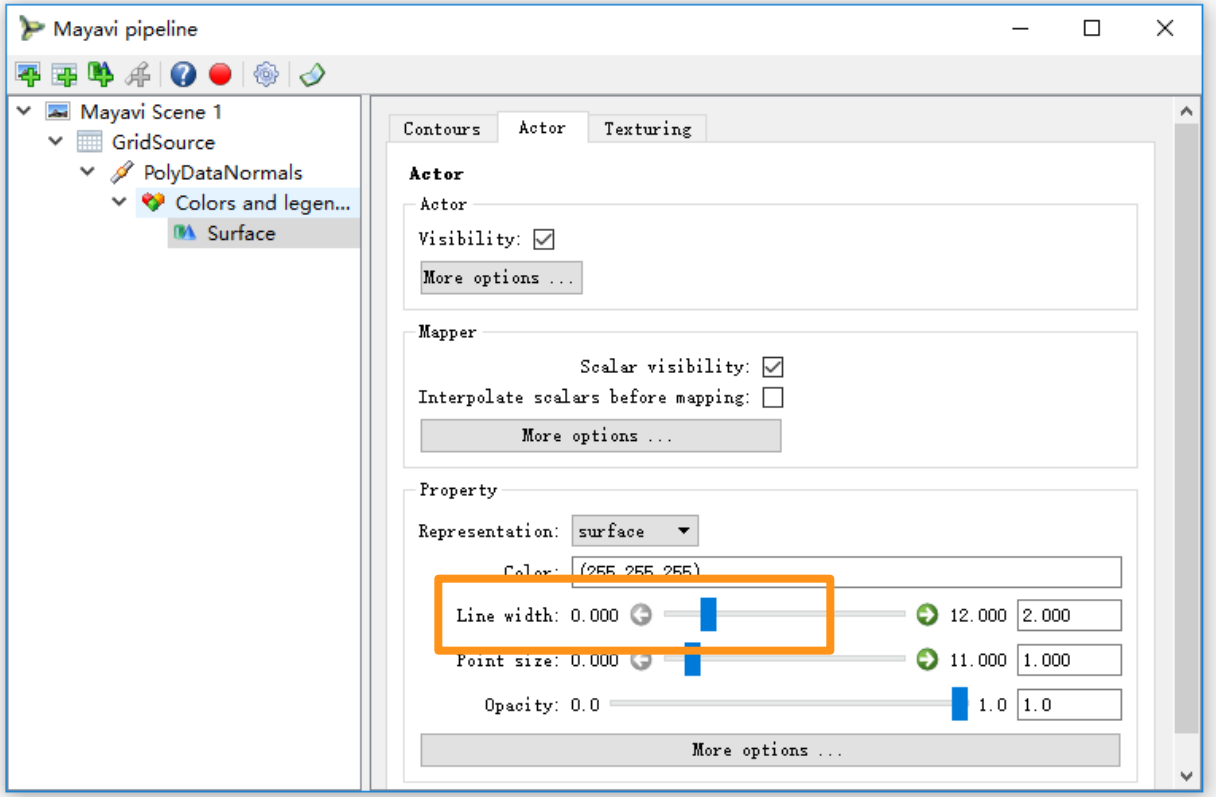
3、配置窗口有多个选项卡，属性需要一级一级获得

程序配置属性的步骤

界面上文字与对象属性名的转换关系：

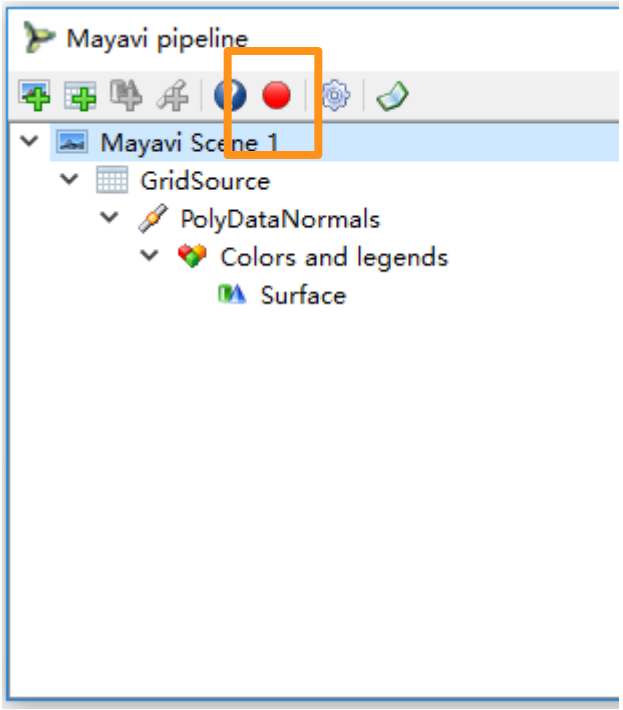
首字母变大写、下划线变空格

程度配置属性的步骤



surface.actor.property.line_width

程序配置属性


Mayavi pipeline
Mayavi Scene 1
GridSource
PolyDataNormals
Colors and legends
Surface