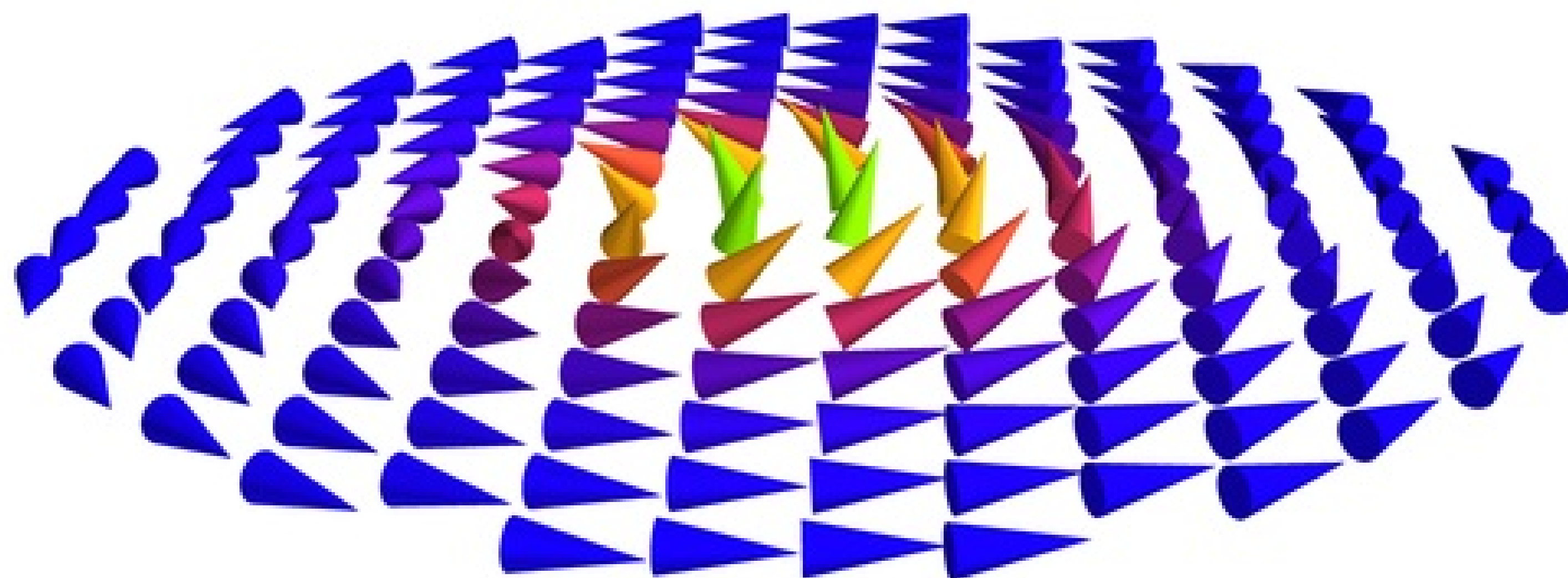


Nmag微磁学仿真流程 V3.0.1



Tang Hong

2015 年 8 月 1日于 固体物理研究所 @SICNU



0. 简介

0.1 Nmag 是一款微磁学仿真计算的工具包。

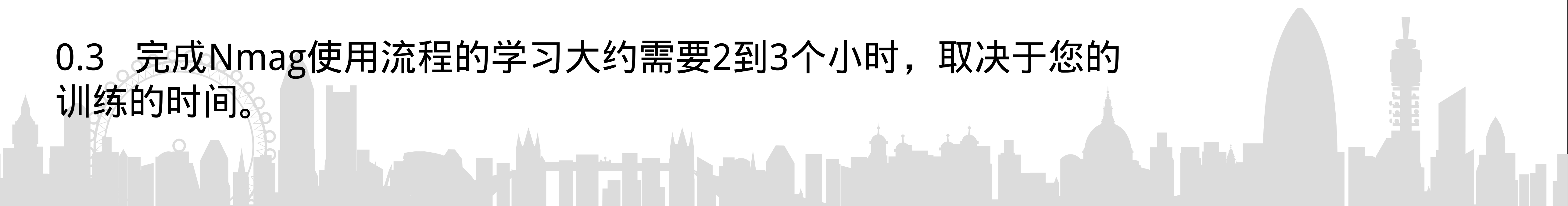
它由英国南安普顿大学的 Hans Fangohr 教授等人开发并维护。

它是一款基于 GNU GPL协议的开源软件。

目前的最新版本为 2012年1月13日发布的 Nmag 0.2.1 。

0.2 限于篇幅，本教程的目标是将介绍使用Nmag计算的5个流程。掌握好这5个流程，您就能很容易的使用Nmang 来仿真计算一般磁性材料的磁学性质。

0.3 完成Nmag使用流程的学习大约需要2到3个小时，取决于您的训练的时间。



0.4

5个流程

第五步：处理计算结果

第四步：生成计算结果

第三步：提交计算任务

第二步：设置计算脚本

第一步：生成几何模型



第一步：生成几何模型

使用的工具有：.geo 文件， Netgen软件， nmeshimport 命令行工具
分3小步：

1.1 在 .geo文件中写入几何模型的参数

用文本编辑器打开或者新建一个格式固定的后缀名为.geo的文件。

它的内容类似如下文所示：

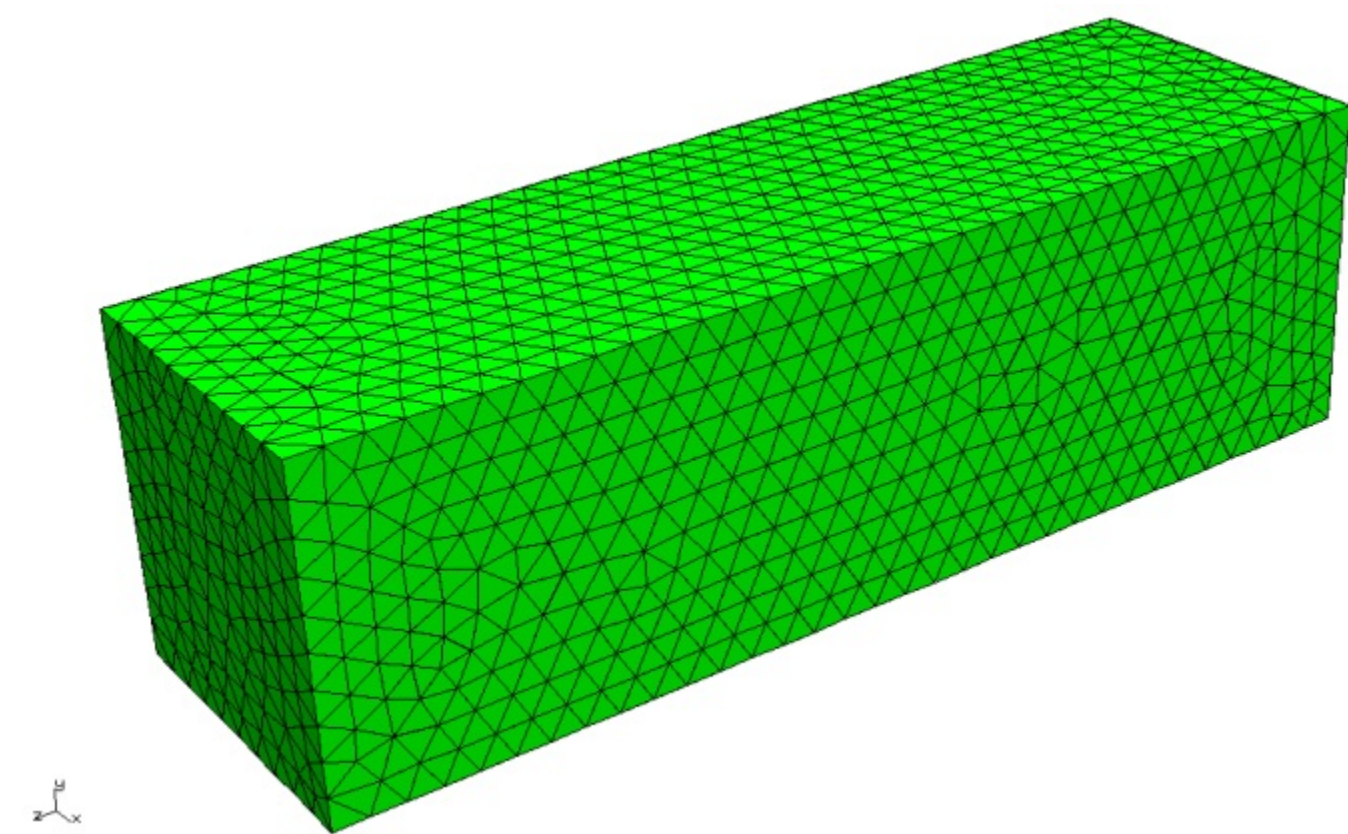
六面立方体：

```
algebraic3d
```

```
# cube consisting of 6 planes:
```

```
solid cube = plane (0, 0, 0; 0, 0, -1)
               and plane (0, 0, 0; 0, -1, 0)
               and plane (0, 0, 0; -1, 0, 0)
               and plane (30, 30, 100; 0, 0, 1)
               and plane (30, 30, 100; 0, 1, 0)
               and plane (30, 30, 100; 1, 0, 0) -maxh=3.0;
```

```
tlo cube;
```

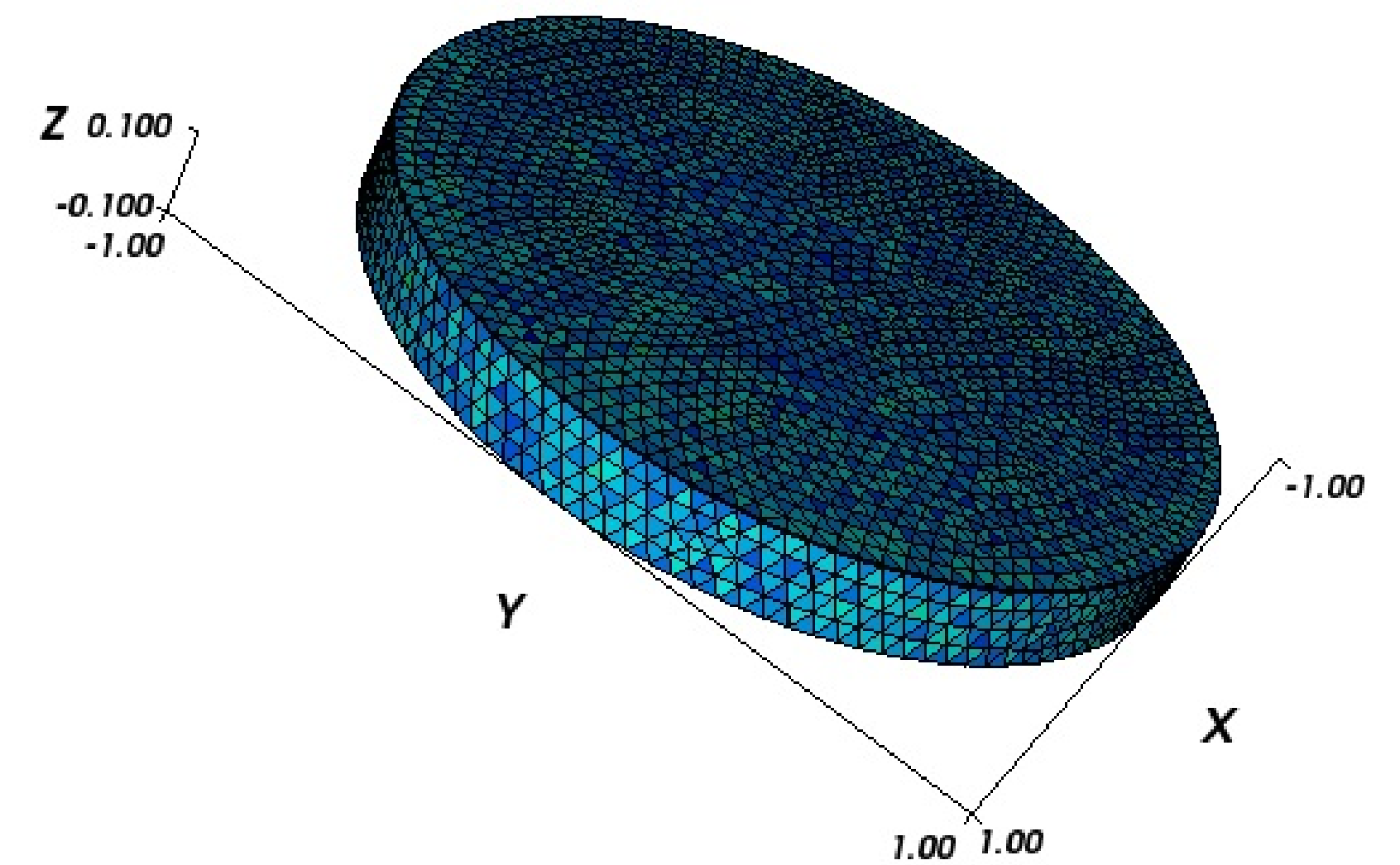


圆盘：

algebraic3d

```
solid fincyl = cylinder ( 0, 0, 1; 0, 0, -1; 1.0 )  
                  and plane (0, 0, -0.1; 0, 0, -1)  
                  and plane (0, 0, 0.1; 0, 0, 1) -maxh = 0.05;
```

```
tlo fincyl;
```

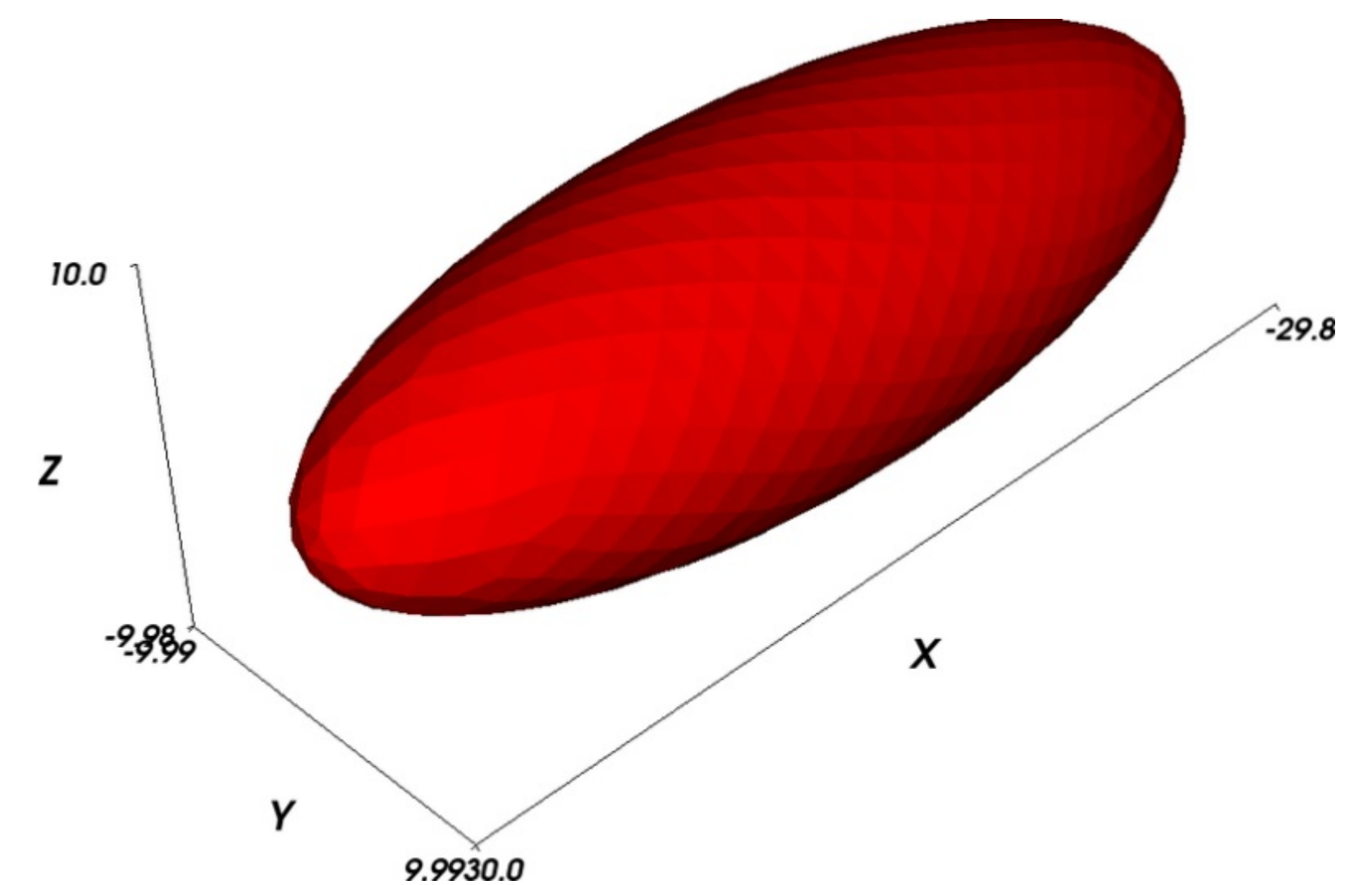


椭球：

algebraic3d

```
solid test = ellipsoid (0, 0, 0; 30.0, 0, 0; 0, 10.0, 0; 0, 0, 10.0)  
-maxh=3;
```

```
tlo test;
```



1.2 使用Netgen 软件

(下载<http://sourceforge.net/projects/netgen-mesher/files/netgen-mesher/6.0/>)

导入.geo文件,
设置网格的大小

Mesh->Meshing Options->Mesh Size

导出生成 .neutral后缀名的网格文件

1.3 使用nmeshimport 将 .neutral 文件转换为 .nmesh.h5文件

```
$ nmeshimport --netgen bar30_30_100.neutral bar30_30_100.nmesh.h5
```

(unix环境中)步骤1.2 和1.3 可以用如下形式的脚本完成：

```
netgen -geofile=bar30_30_100.geo -meshfiletype="Neutral Format" -  
meshfile=bar30_30_100.neutral -batchmode  
nmeshimport --netgen bar30_30_100.neutral bar30_30_100.nmesh.h5
```

得到几何模型文件.h5后, 也就确定了仿真材料的形状结构, 接下来 就可以进入下一个步骤,

2. 设置计算脚本



第二步：设置计算脚本

.py脚本如下例：

```
import nmag

from nmag import SI

mat_Py = nmag.MagMaterial(name="Py",
                           Ms=SI(0.86e6, "A/m"),
                           exchange_coupling=SI(13.0e-12, "J/m"),
                           llg_damping=0.5)

sim = nmag.Simulation("bar")

sim.load_mesh("bar30_30_100.nmesh.h5",
              [("Py", mat_Py)],
              unit_length=SI(1e-9, "m"))

sim.set_m([1, 0, 1])

dt = SI(5e-12, "s")

for i in range(0, 61):
    sim.advance_time(dt*i)                                #compute time development

    if i % 10 == 0:                                         #every 10 loop iterations,
        sim.save_data(fields='all')                         #save averages and all
                                                            #fields spatially resolved
    else:
        sim.save_data()                                     #otherwise just save averages
```



第三步：提交计算任务

用到的工具： nsim

命令行形式：

```
$ nsim sphere1.py
```

```
$ nsim sphere1.py --clean
```

也可以用脚本文件来运行：

```
#!/bin/bash  
cd ~/nmag/ex1
```

```
# Set up the environment for thread enabled Tcl/Tk  
source /lustre/home/zhaogp2015/.bashrc
```

```
# Finally run the simulation  
nsim ~/nmag/ex1/sphere1.py --clean
```



第四步：生成计算结果

工具：ncol nmagpp

第三步的计算将会生成 后缀为.ndt和 .h5结果文件，分别可以使用ncol 和 nmagpp处理生成计算结果数据

生成文本数据：

```
$ ncol ellipsoid H_ext_0 m_Py_0 > plot.dat
```

```
$ nmagpp --dump sphere1
```

生成后缀为 .vtk的可视化结果文件

```
$ nmagpp --vtk sphere1.vtk sphere1
```



第五步：处理计算结果

工具：gnuplot, Mayavi (或者其他数值绘图软件)

绘制磁滞回线

使用gnuplot

```
$ gnuplot make_plot.gnu
```

.gun文件内容

```
set term postscript eps enhanced color
set out 'hysteresis.eps'
set xlabel 'Applied field H_x (A/m)'
set ylabel 'M_x / M_s'
set xrange [-1050000:1050000]
set yrange [-1.2:1.2]
plot 'plot.dat' u 1:2 ti 'ellipsoid example' w lp 3
```



使用nmagpp 生成三维可视化结果

```
$ nmagpp --range 0 --vtk bar_initial.vtk bar
```

```
$ nmagpp --range 60 --vtk bar_final.vtk bar
```

```
$ nmagpp --vtk bar.vtk bar
```

生成文件

bar-000000.vtk

bar-000010.vtk

bar-000020.vtk

bar-000030.vtk

bar-000040.vtk

bar-000050.vtk

bar-000060.vtk



使用 Mayavi 处理结果文件

```
mayavi -d sphere1-000000.vtk
```

设置矢量图

Visualize -> Modules -> VelocityVector

添加 坐标轴线

Visualize -> Modules -> Axes

