

# Mesh

Xie Yu

## 1 介绍

Baffalo\_Mesh 是划分3D网格的模块，主要功能是：

- 将2D的网格转换为3D的网格。
- 建立基本的3D网格体
- 雕刻3D网格

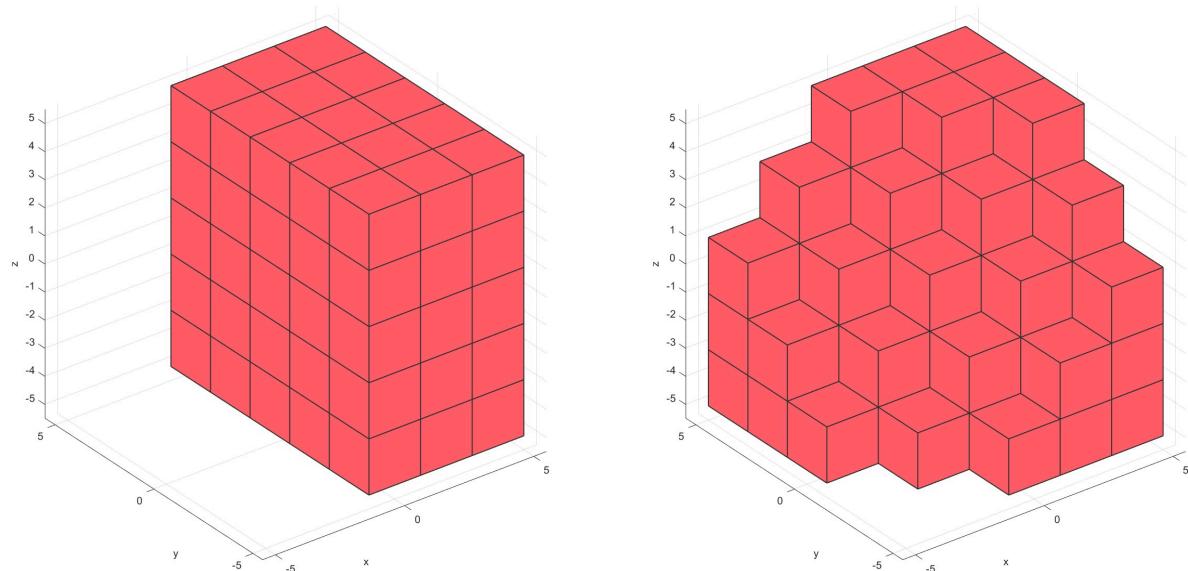
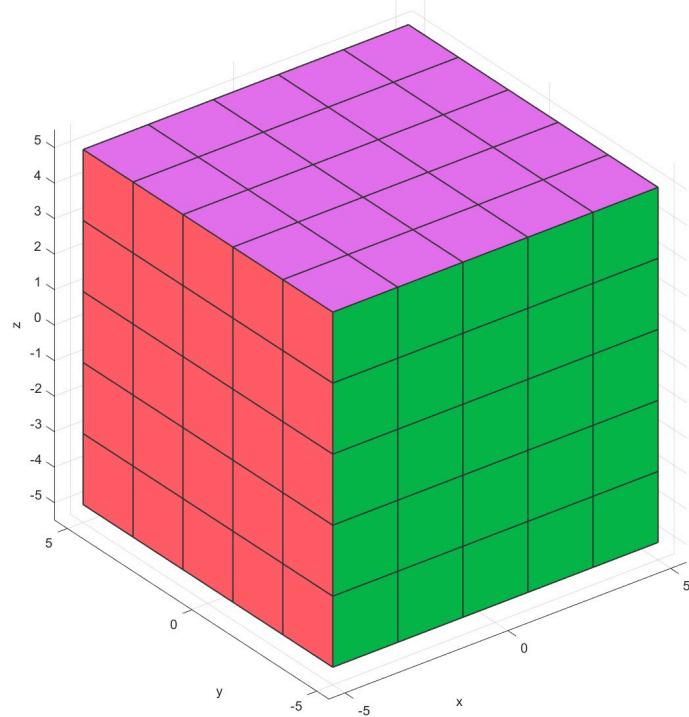
主要参考的工具箱或代码：Gibbon<sup>[1]</sup>[2],MRST<sup>[3]</sup>,TetGen<sup>[4]</sup>,voronoi3d<sup>[5]</sup>

## 2 案例

### 2.1 Create cube mesh (Flag=1)

```
1 %% Specifying dimensions and number of elements
2 m=Mesh('Mesh');
3 %% Create a box with hexahedral elements
4 cubeDimensions=[10 10 10]; %Dimensions
5 cubeElementNumbers=[5 5 5]; %Number of elements
6 m=MeshCube(m,cubeDimensions,cubeElementNumbers);
7 %% Plot elements
8 PlotFace(m,'face_alpha',1);
9 sec.pos=[0,0,0];
10 sec.vec=[1,0,0];
11 PlotElement(m,'section',sec);
12 sec.vec=[1,1,-1];
13 PlotElement(m,'section',sec);
```

建立cube体网格：



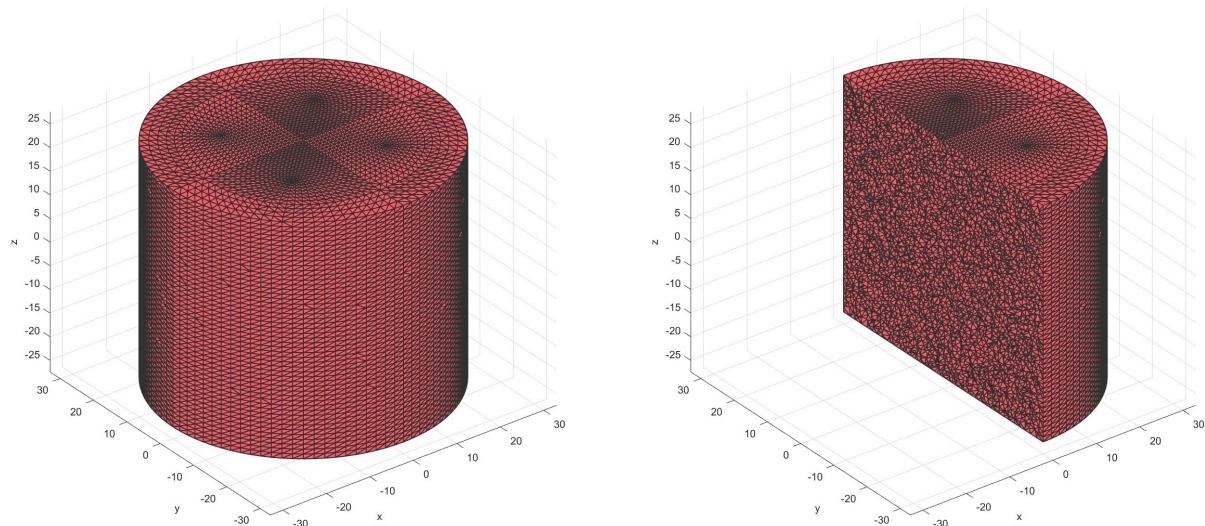
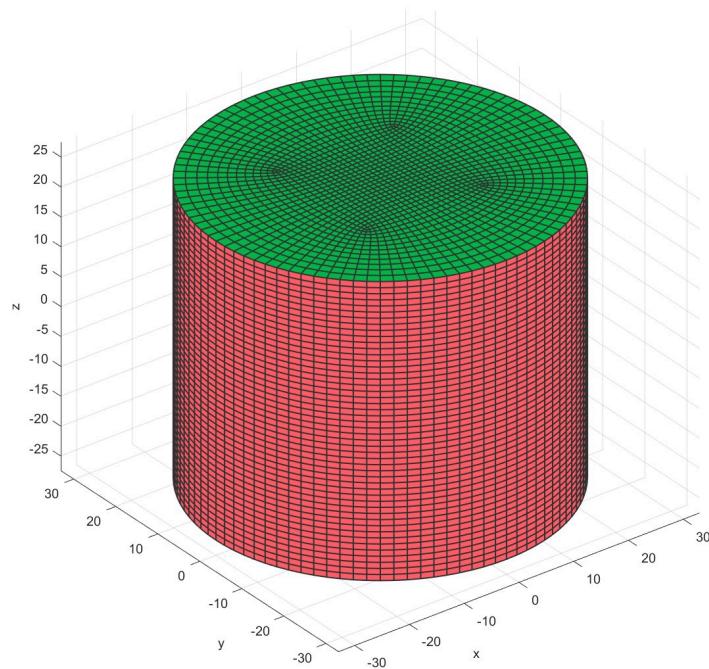
## 2.2 Create cylinder mesh (Flag=2)

```

1 %Specifying dimensions and number of elements
2 mm=Mesh('Cylinder Mesh');
3 mm=MeshCylinder(mm,1,30,50,'ElementType','quad');
4 PlotFace(mm);
5 % The cylinder is meshed using tetrahedral elements using tetgen
6 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
7 PlotElement(mm);
8 sec.pos=[0,0,0];
9 sec.vec=[1,0,0];
10 PlotElement(mm,'section',sec);

```

建立圆柱体网格：



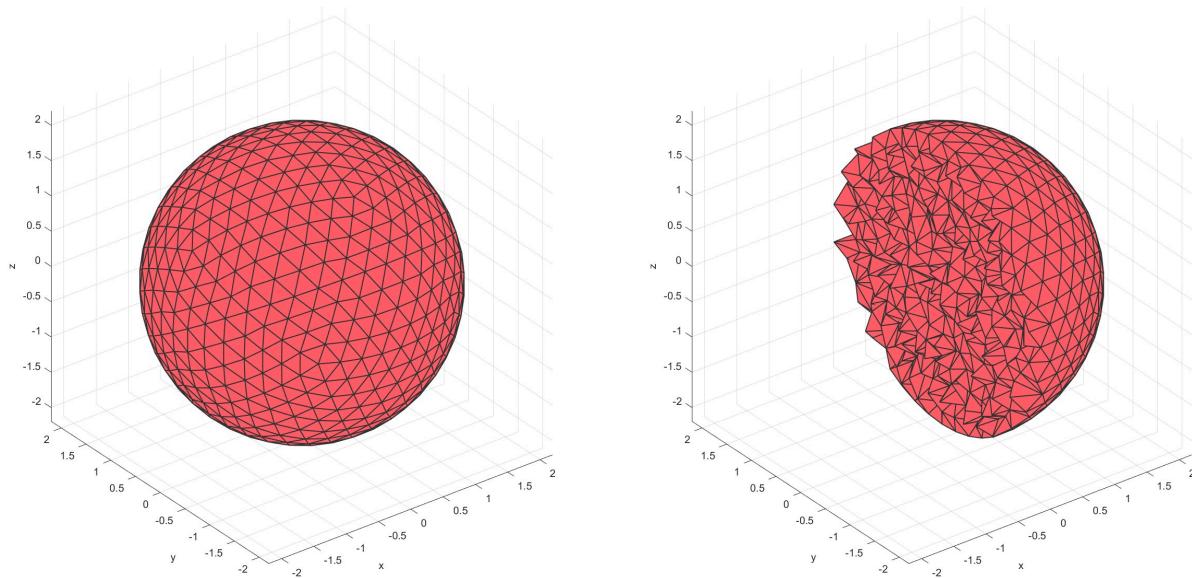
### 2.3 Create sphere mesh (Flag=3)

```

1 %Sphere parameters
2 numRefineStepsSphere=3;
3 sphereRadius=2;
4 mm=Mesh('Demo Sphere Mesh');
5 mm=MeshSphere(mm,numRefineStepsSphere,sphereRadius);
6 PlotFace(mm);
7 sec.pos=[0,0,0];
8 sec.vec=[1,0,0];
9 PlotElement(mm,'section',sec);

```

建立球体网格：



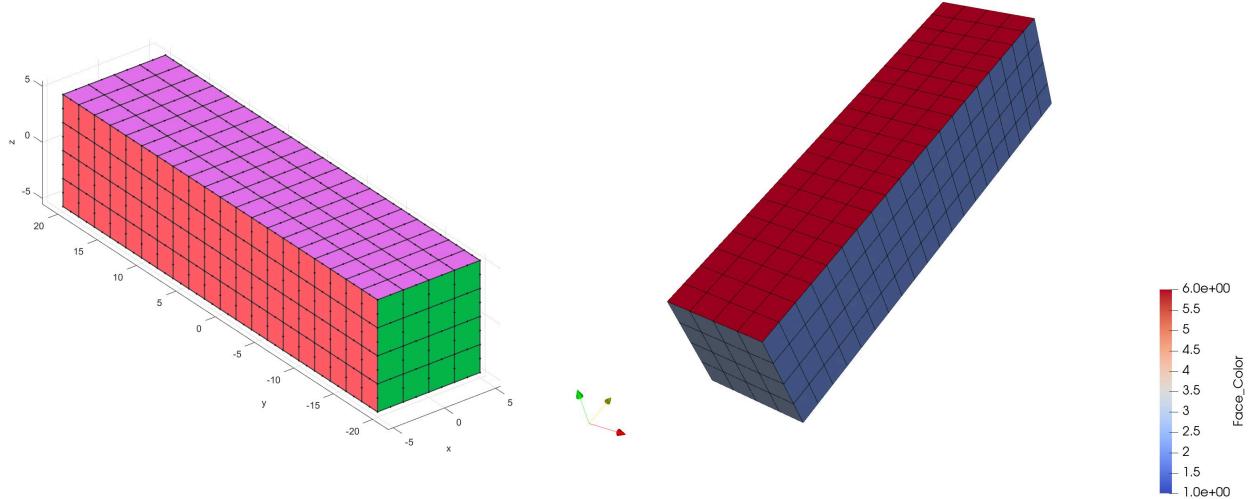
## 2.4 Hex20 Plot (Flag=4)

```

1 % Specifying dimensions and number of elements
2 m=Mesh('Mesh');
3 % Create a box with hexahedral elements
4 beamDimensions=[10 40 10]; %Dimensions
5 beamElementNumbers=[4 20 4]; %Number of elements
6 m=MeshCube(m,beamDimensions,beamElementNumbers,'ElementType','hex20');
7 PlotFace(m,'marker',1);
8 PlotFace2(m);

```

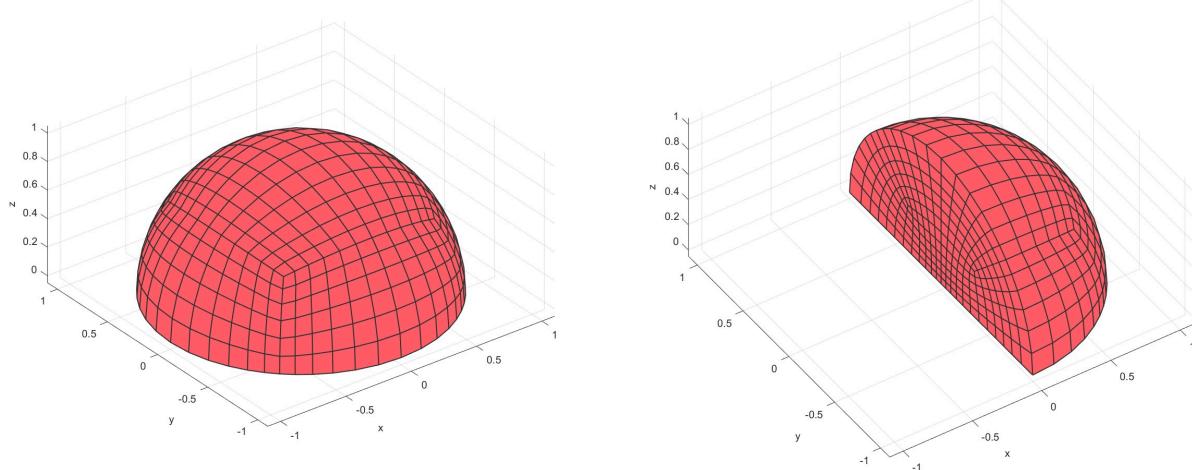
二阶六面体网格绘制：



## 2.5 MeshHemiSphere (Flag=5)

```
1 % Control settings
2 sphereRadius=1;
3 numElementsMantel=6;
4 m=Mesh('Mesh');
5 m=MeshHemiSphere(m,sphereRadius, numElementsMantel);
6 PlotFace(m);
7 sec.pos=[0,0,0];
8 sec.vec=[1,0,0];
9 PlotElement(m,'section',sec);
```

半球体网格绘制：



## 2.6 Mesh plate1 (Flag=6)

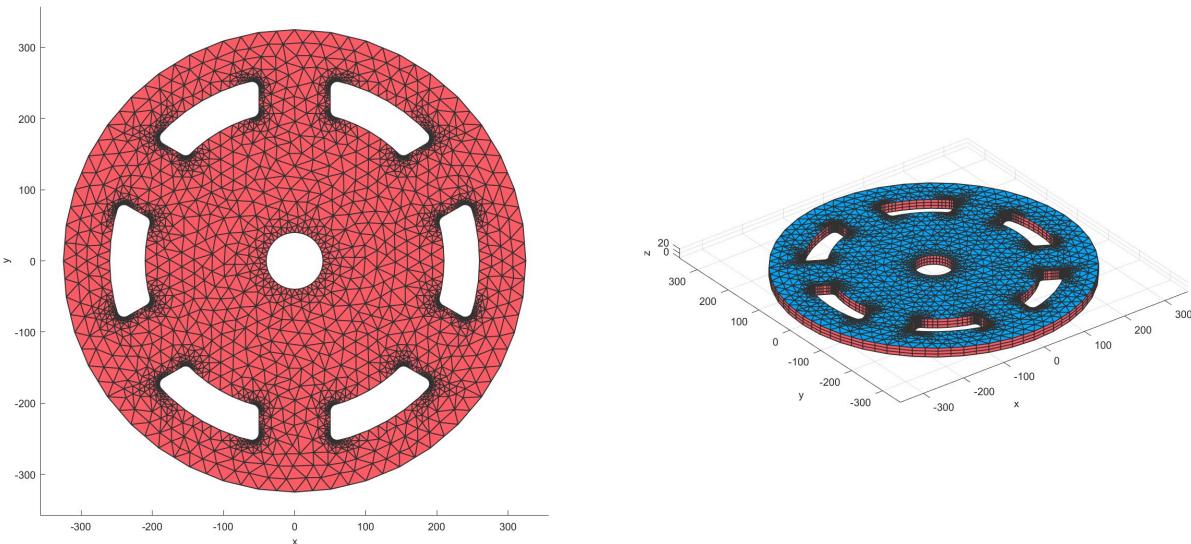
```
1 % IR=211/2;
2 % OR=801/2;
3 % radius1=300/2;
4 % radius2=600/2;
5 % num=6;
6 % par1=100;
7 % Thickness=15;
8 IR=80/2;
9 OR=650/2;
10 radius1=420/2;
11 radius2=520/2;
12 num=6;
13 par1=100;
14 Thickness=20;
15 ang1=360/num-asin(par1/2/radius1)/pi*180*2;
16 ang2=360/num-asin(par1/2/radius2)/pi*180*2;
17 a=Point2D('Point Ass1');
18 b=Line2D('Line Ass1');
19 a>AddPoint(a,0,0);
20 a>AddPoint(a,par1,0);
21 a>AddPoint(a,[radius1*cos(ang1/2/180*pi);radius2*cos(ang2/2/180*pi)],...
22 [radius1*sin(ang1/2/180*pi);radius2*sin(ang2/2/180*pi)]);
```

```

23 a=AddPoint(a,[radius2*cos(-ang2/2/180*pi);radius1*cos(-ang1/2/180*pi)],...
24 [radius2*sin(-ang2/2/180*pi);radius1*sin(-ang1/2/180*pi)]);
25 b=AddCircle(b,OR,a,1);
26 S=Surface2D(b);
27 h1=Line2D('Hole Group1');
28 h1=AddCircle(h1,IR,a,1);
29 S>AddHole(S,h1);
30 h2=Line2D('Hole Group2','Dtolt',0.1);
31 h2=AddCircle(h2,radius1,a,1,'sang',-ang1/2,'ang',ang1);
32 h2>AddLine(h2,a,3);
33 h2=AddCircle(h2,radius2,a,1,'sang',ang2/2,'ang',-ang2);
34 h2>AddLine(h2,a,4);
35 h2>CreateRadius(h2,1,10);
36 h2>CreateRadius(h2,3,10);
37 h2>CreateRadius(h2,5,10);
38 h2>CreateRadius(h2,7,10);
39 for i=1:num
40 S>AddHole(S,h2,'rot',360/num*(i-1));
41 end
42 m=Mesh2D('Mesh1');
43 m>AddSurface(m,S);
44 m>SetSize(m,20);
45 m=Mesh(m);
46 Plot(m);
47 mm=Mesh('Mesh');
48 mm>Extrude2Solid(mm,m,Thickness,3);
49 PlotFace(mm);

```

绘制平板1



## 2.7 Mesh plate2 (Flag=7)

```

1 % Parameter
2 IR=640/2;
3 OR=768/2;
4 par=442;
5 radius=70;

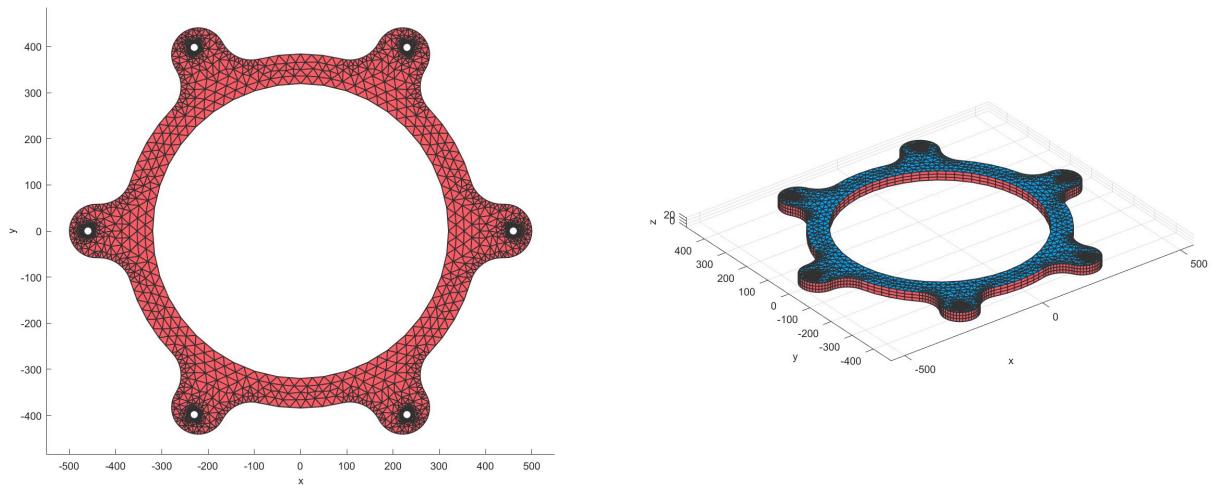
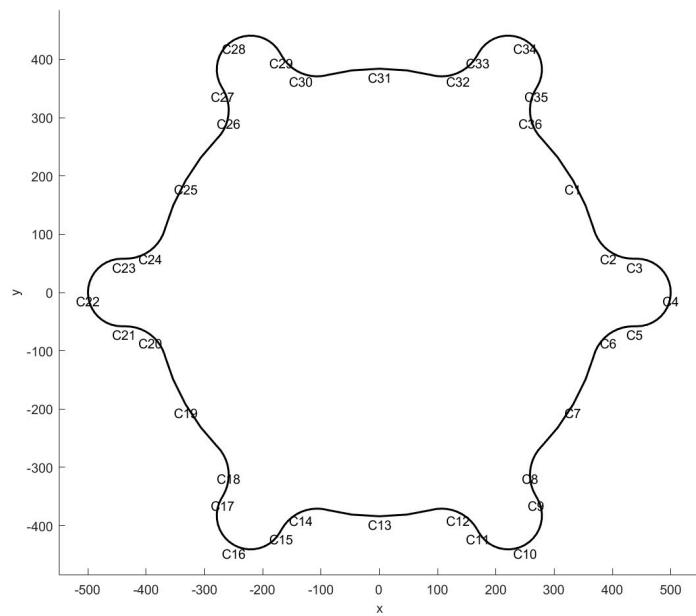
```

```

6 R=116/2;
7 num=6;
8 Rp=460;
9 t=30;
10 hd=17.5;
11 sang=asin(R/OR)/pi*180;
12 ang=360/num-2*sang;
13 a=Point2D('Point Ass1');
14 b=Line2D('Line Ass1','Dtol',1);
15 a=AddPoint(a,0,0);
16 for i=1:num
17 b=AddCircle(b,OR,a,1,'sang',sang+ang-360/num*(i-1),'ang',-ang);
18 theta=-360/num/180*pi*(i-1);
19 mat=[cos(theta),-sin(theta);sin(theta),cos(theta)];
20 p1=mat*[sqrt(OR^2-R^2),par;R,R];
21 p1=p1';
22 a=AddPoint(a,p1(:,1),p1(:,2));
23 b=AddLine(b,a,2+3*(i-1));
24 p2=mat*[par;0];
25 p2=p2';
26 a=AddPoint(a,p2(1,1),p2(1,2));
27 b=AddCircle(b,R,a,3+3*(i-1),'sang',90+theta/pi*180,'ang',-180);
28 p3=mat*[par,sqrt(OR^2-R^2);-R,-R];
29 p3=p3';
30 a=AddPoint(a,p3(:,1),p3(:,2));
31 b=AddLine(b,a,4+3*(i-1));
32 end
33 for i=1:num
34 b=CreateRadius(b,1+6*(i-1),radius);
35 b=CreateRadius(b,5+6*(i-1),radius);
36 end
37 Plot(b,'clabel',1,'equal',1)
38 S=Surface2D(b);
39 h1=Line2D('Hole Group1');
40 h1=AddCircle(h1,IR,a,1);
41 S=AddHole(S,h1);
42 a1=Point2D('Point Ass2');
43 a1=AddPoint(a1,Rp,0);
44 h2=Line2D('Hole Group2');
45 h2=AddCircle(h2,hd/2,a1,1);
46 for i=1:num
47 S=AddHole(S,h2,'rot',360/num*(i-1));
48 end
49 m=Mesh2D('Mesh1');
50 m=AddSurface(m,S);
51 m=SetSize(m,20);
52 m=Mesh(m);
53 Plot(m);
54 mm=Mesh('Mesh');
55 mm=Extrude2Solid(mm,m,t,3);
56 PlotFace(mm);

```

绘制平板2:



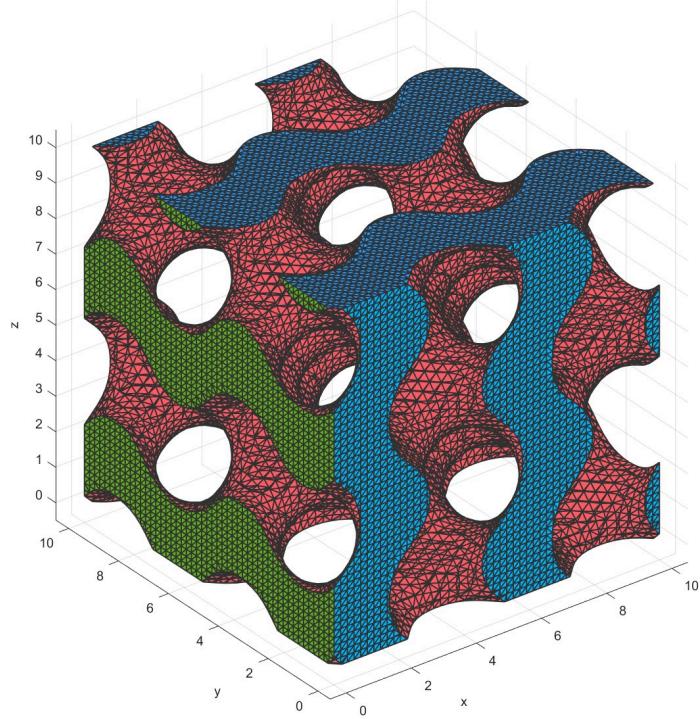
## 2.8 Mesh TriplyPeriodicMinimalSurface (Flag=8)

```

1 %Gyroid
2 Length=10; % Length of the sample
3 mm=Mesh('Demo Gyroid Mesh');
4 mm=MeshTriplyPeriodicMinimalSurface(mm,Length);
5 PlotFace(mm);

```

建立围观元胞：



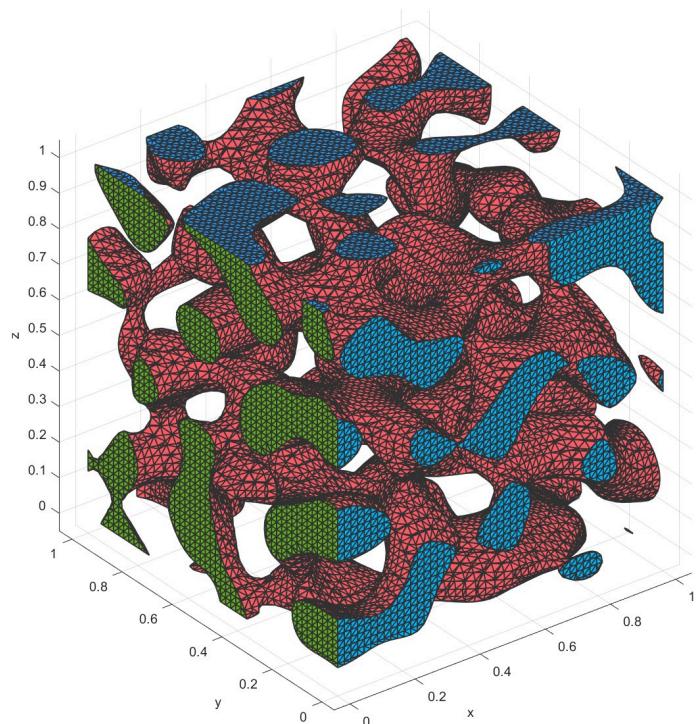
## 2.9 Mesh StochasticMicroStructure (Flag=9)

```

1 %Stochastic structure
2 Length=10; % Length of the sample
3 paticleLength=1;
4 mm=Mesh('Demo Stochastic structure Mesh');
5 mm=MeshStochasticMicroStructure(mm,Length,paticleLength);
6 PlotFace(mm);

```

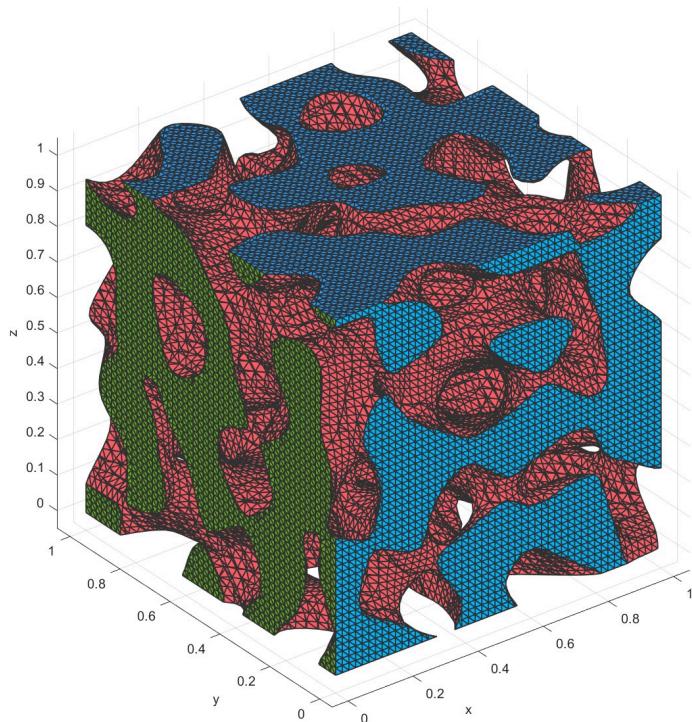
建立微观元胞：



## 2.10 Mesh Spinodoid (Flag=10)

```
1 %spinodoid
2 Length=10; % Length of the sample
3 paticleLength=1;
4 mm=Mesh('Demo spinodoid Mesh');
5 mm=MeshSpinodoid(mm,Length,paticleLength);
6 PlotFace(mm);
```

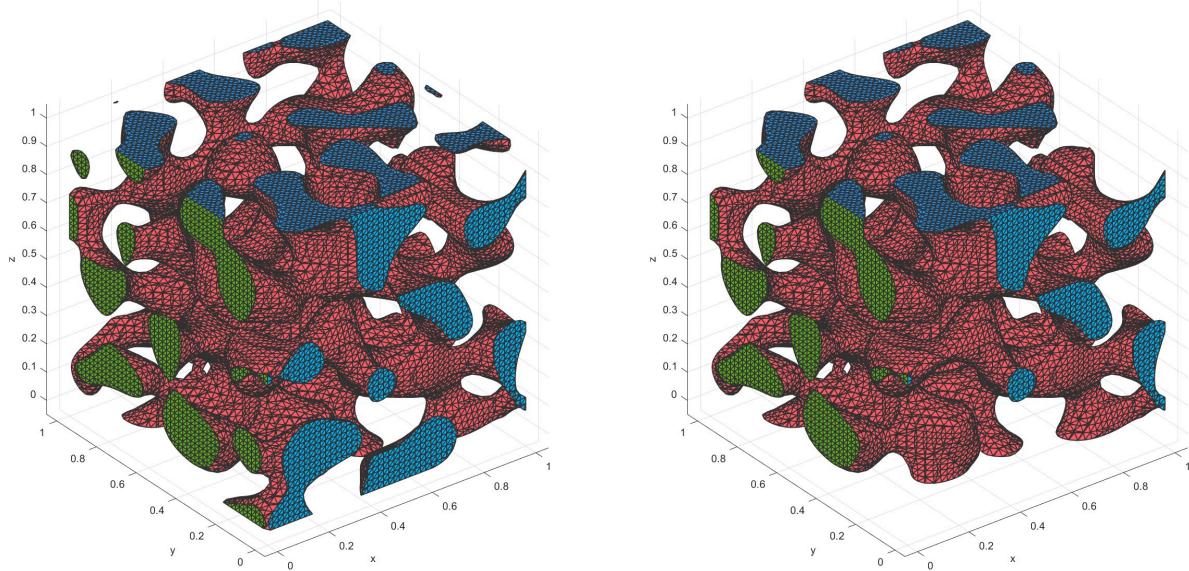
建立微观元胞：



## 2.11 Group Test (Flag=11)

```
1 %Stochastic structure
2 Length=10; % Length of the sample
3 paticleLength=1;
4 mm=Mesh('Demo Stochastic structure Mesh');
5 mm=MeshStochasticMicroStructure(mm,Length,paticleLength);
6 PlotFace(mm);
7 [G,~]=GroupTest(mm);
8 disp(G);
9 mm=KeepGroup(mm,1);
10 PlotFace(mm);
```

用Group Test找到独立的网格，并保留其中一块独立的区域：



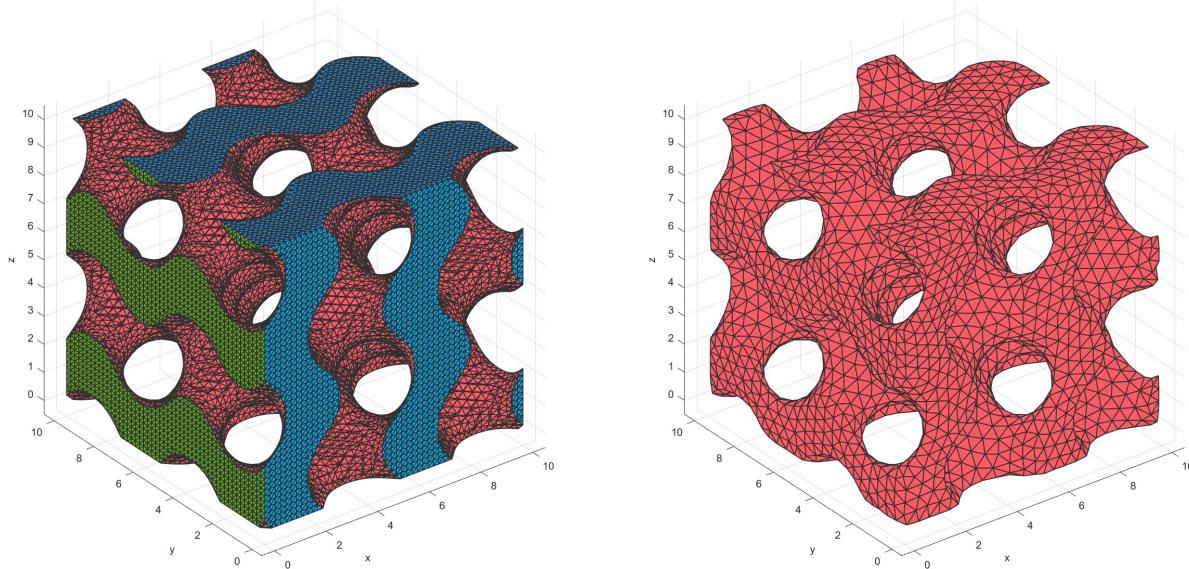
## 2.12 Remesh (Flag=12)

```

1 %Gyroid
2 Length=10; % Length of the sample
3 mm=Mesh('Demo Gyroid Mesh');
4 mm=MeshTriplyPeriodicMinimalSurface(mm,Length);
5 PlotFace(mm);
6 mm=Remesh(mm,0.5);
7 PlotFace(mm);

```

网格重新绘制：



## 2.13 Drill holes (Flag=13)

```

1 % Derive patch data for a cylinder
2 mm=Mesh('Cylinder Mesh');
3 mm=MeshCylinder(mm,3,30,50);
4 PlotFace(mm);
5 % The cylinder is meshed using tetrahedral elements using tetgen

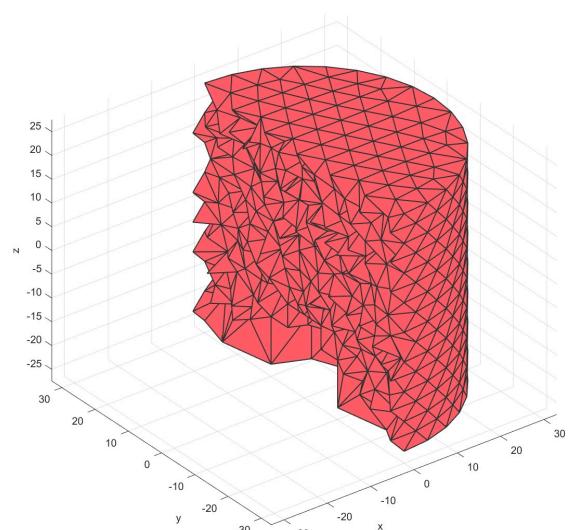
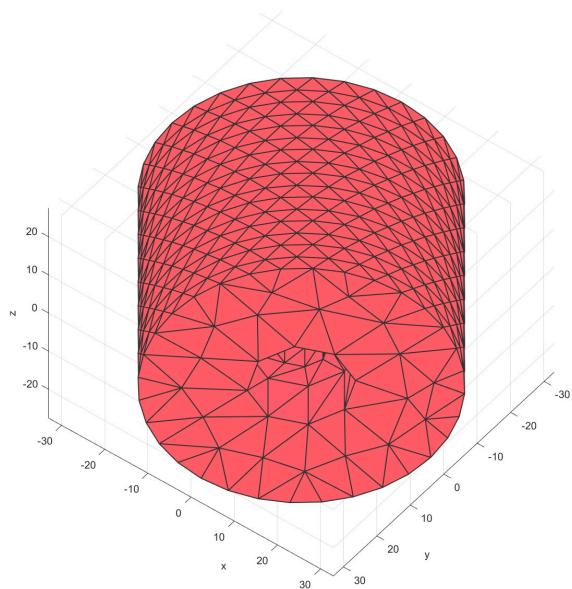
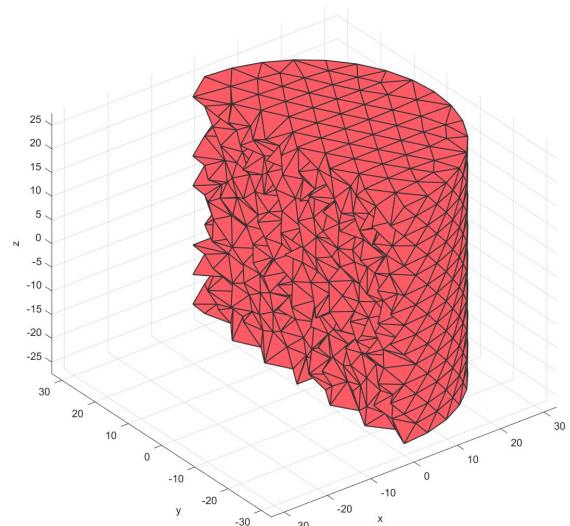
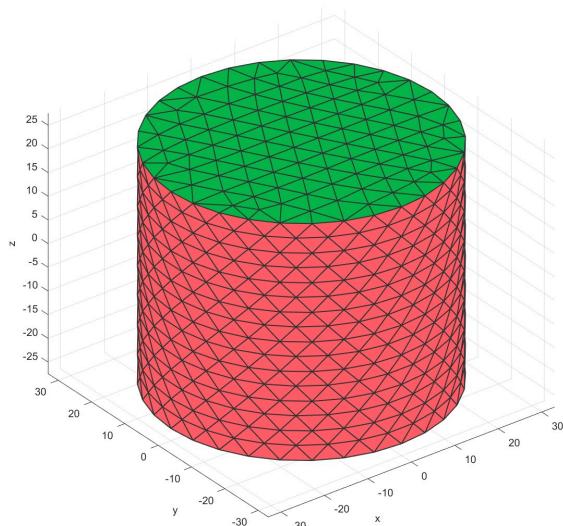
```

```

6 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
7 sec.pos=[0,0,0];
8 sec.vec=[1,0,0];
9 PlotElement(mm,'section',sec);
10 mm=DrillHole(mm,[10;5],[10;15],[0,0],3,'type',5,'slice',8);
11 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
12 PlotFace(mm);
13 sec.pos=[0,0,0];
14 sec.vec=[1,0,0];
15 PlotElement(mm,'section',sec);

```

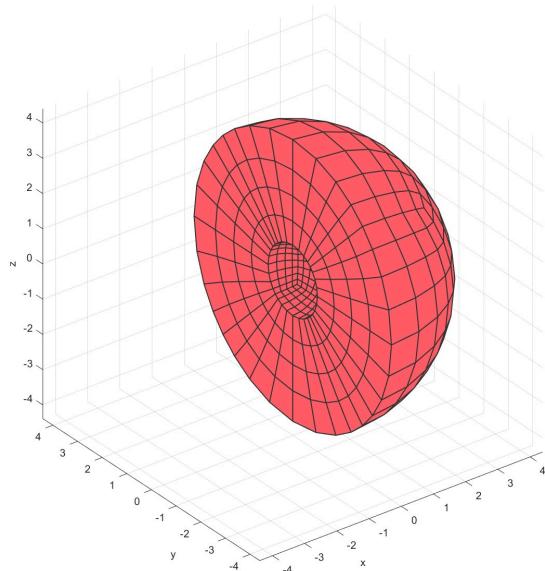
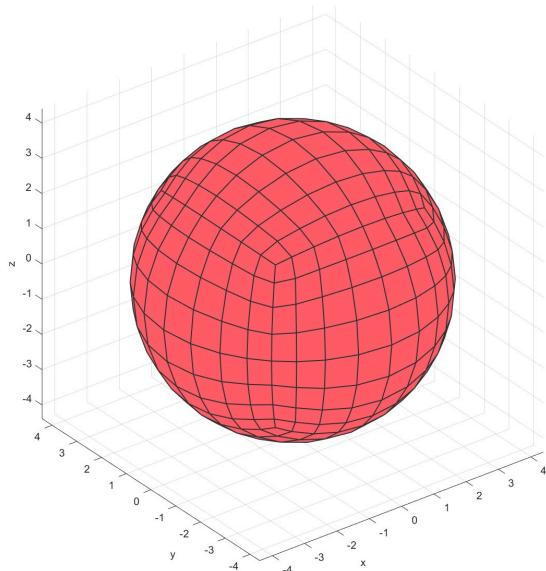
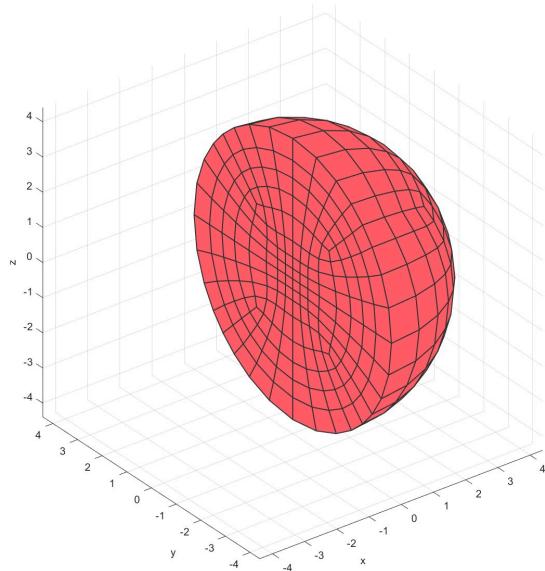
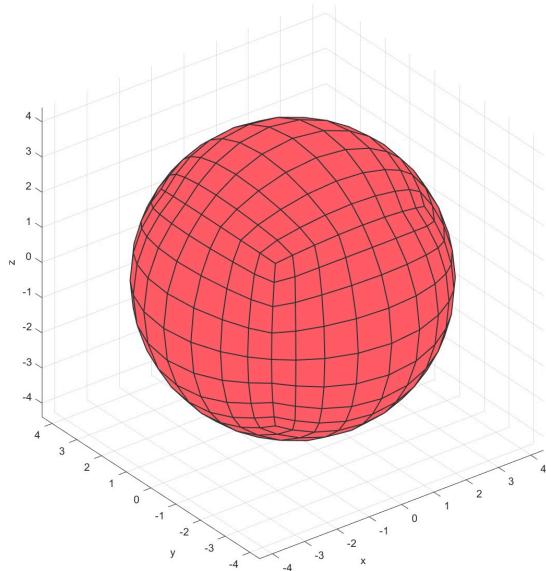
在指定的面上钻盲孔，并且可定义盲孔的尺寸。



## 2.14 MeshQuadSphere (Flag=14)

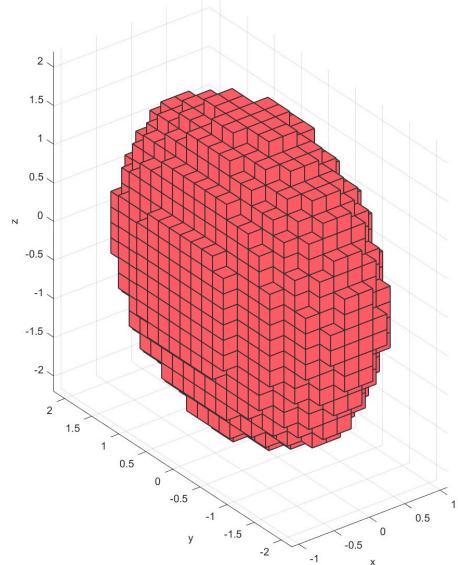
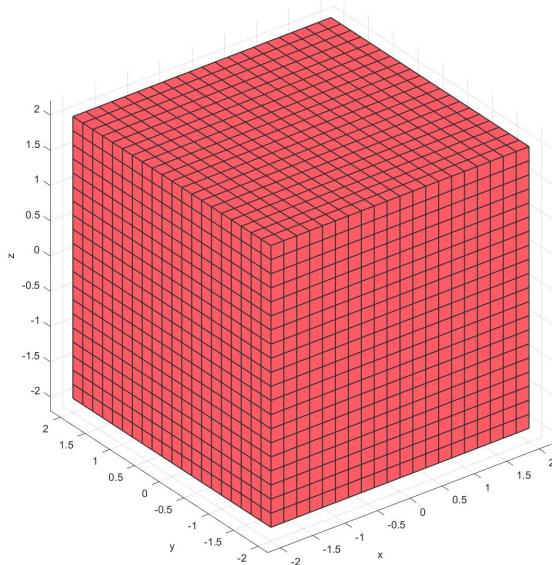
```
1 %Sphere parameters
2 numRefineStepsSphere=4;
3 sphereRadius=4;
4 mm=Mesh('Demo Quad Sphere Mesh');
5 mm=MeshQuadSphere(mm,numRefineStepsSphere,sphereRadius);
6 PlotFace(mm);
7 sec.pos=[0,0,0];
8 sec.vec=[1,0,0];
9 PlotElement(mm,'section',sec);
10 mm1=Mesh('Demo Quad hollow Sphere Mesh');
11 mm1=MeshQuadSphere(mm1,numRefineStepsSphere,sphereRadius,'hollow',1,'coreRadius',sphereRadius/4);
12 PlotFace(mm1);
13 PlotElement(mm1,'section',sec);
```

绘制六面体圆网格。



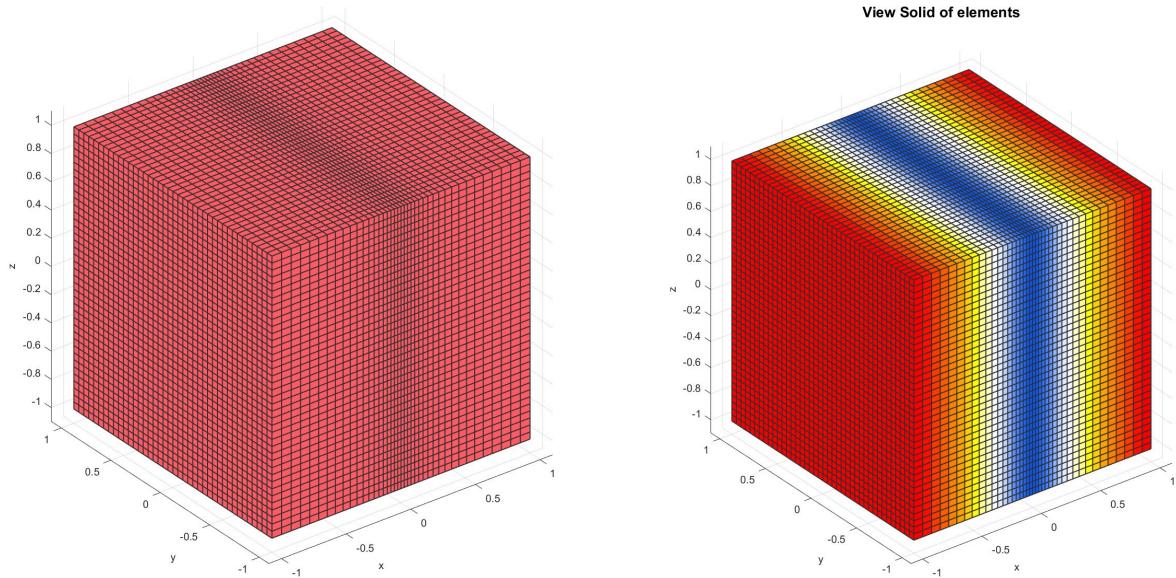
## 2.15 Mesh TensorGrid (Flag=15)

```
1 x = linspace(-2,2,21);
2 mm=Mesh('Demo TensorGrid');
3 mm=MeshTensorGrid(mm,x,x,x);
4 PlotFace(mm);
5 mm= ComputeGeometryG(mm);
6 c = mm.G.cells.centroids;
7 r = c(:,1).^2 + 0.25*c(:,2).^2+0.25*c(:,3).^2;
8 mm = RemoveCells(mm, r>1);
9 PlotFace(mm);
```



## 2.16 PlotG (Flag=16)

```
1 x=-1:0.05:1;
2 xx=x-1/2/pi.*sin(pi.*x);
3 yy=-1:0.05:1;
4 zz=-1:0.05:1;
5 mm=Mesh('Demo TensorGrid');
6 mm=MeshTensorGrid(mm,xx,yy,zz);
7 PlotFace(mm);
8 mm= ComputeGeometryG(mm);
9 PlotG(mm,'volume',1);
```



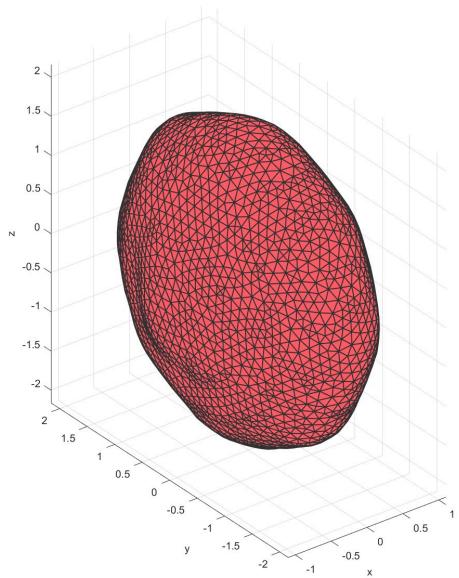
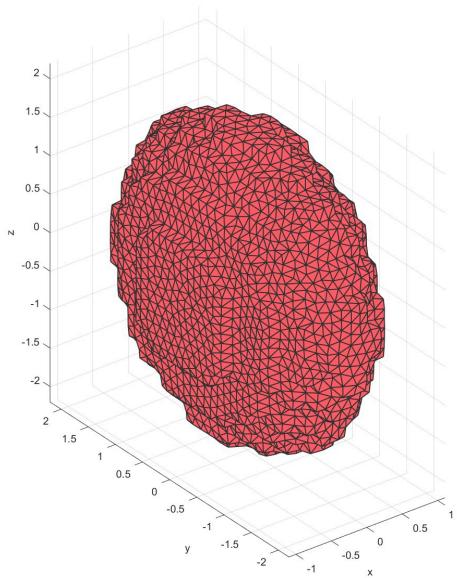
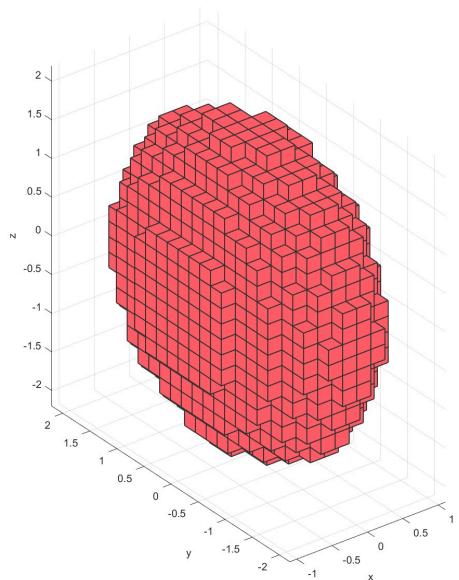
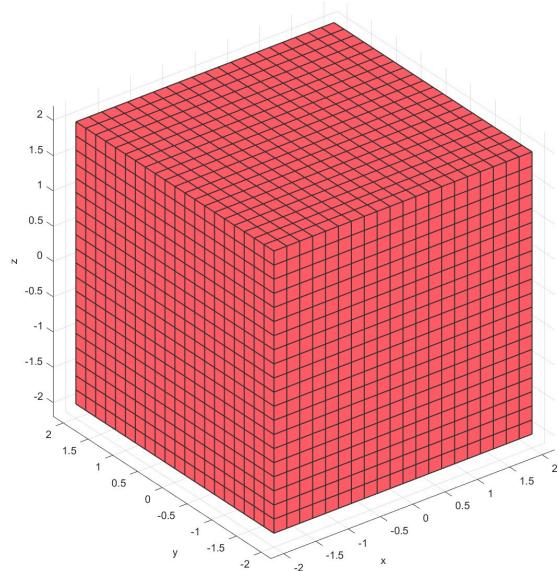
## 2.17 Simple sculpture Mesh (Flag=17)

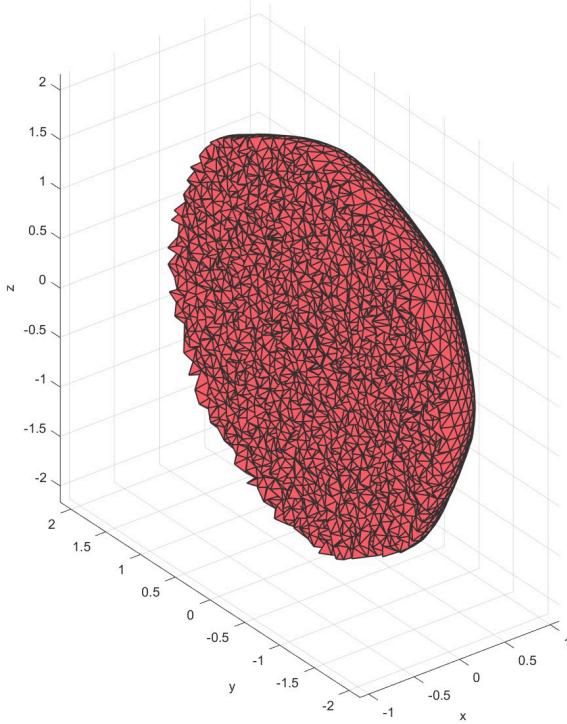
```

1 x = linspace(-2,2,21);
2 mm=Mesh('Demo TensorGrid');
3 mm=MeshTensorGrid(mm,x,x,x);
4 PlotFace(mm);
5 mm= ComputeGeometryG(mm);
6 c = mm.G.cells.centroids;
7 r = c(:,1).^2 + 0.25*c(:,2).^2+0.25*c(:,3).^2;
8 mm = RemoveCells(mm, r>1);
9 PlotFace(mm);
10 mm=Remesh(mm,0.2);
11 PlotFace(mm);
12 mm= Smoothface(mm,10);
13 mm=Mesh3D(mm);
14 PlotFace(mm);
15 sec.pos=[0,0,0];
16 sec.vec=[1,0,0];
17 PlotElement(mm,'section',sec);

```

在六面体网格中雕刻出椭球体：





## 2.18 Check G center inner mesh (Flag=18)

```

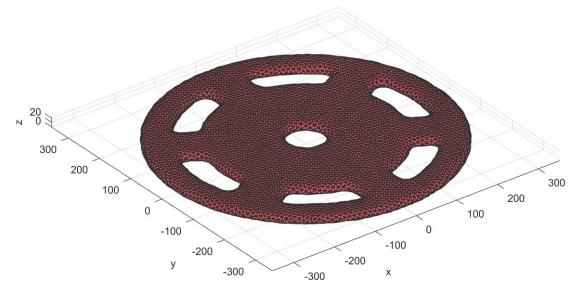
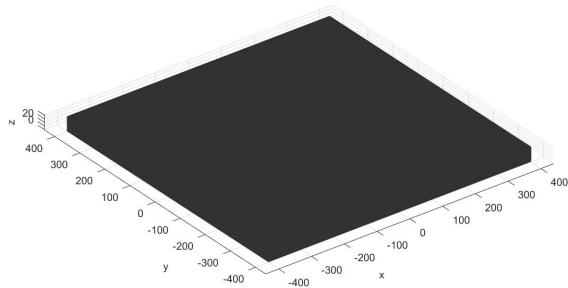
1 % Original geometry
2 x = linspace(-400,400,200);
3 y=linspace(-400,400,200);
4 z=linspace(-10,30,40);
5 mm=Mesh('Demo TensorGrid');
6 mm=MeshTensorGrid(mm,x,y,z);
7 mm= ComputeGeometryG(mm);
8 PlotFace(mm);
9 % Plate
10 IR=80/2;
11 OR=650/2;
12 radius1=420/2;
13 radius2=520/2;
14 num=6;
15 par1=100;
16 Thickness=20;
17 ang1=360/num-asin(par1/2/radius1)/pi*180*2;
18 ang2=360/num-asin(par1/2/radius2)/pi*180*2;
19 a=Point2D('Point Ass1');
20 b=Line2D('Line Ass1');
21 a=AddPoint(a,0,0);
22 a=AddPoint(a,par1,0);
23 a=AddPoint(a,[radius1*cos(ang1/2/180*pi);radius2*cos(ang2/2/180*pi)],...
24 [radius1*sin(ang1/2/180*pi);radius2*sin(ang2/2/180*pi)]);
25 a=AddPoint(a,[radius2*cos(-ang2/2/180*pi);radius1*cos(-ang1/2/180*pi)],...
26 [radius2*sin(-ang2/2/180*pi);radius1*sin(-ang1/2/180*pi)]);
27 b=AddCircle(b,OR,a,1);
28 S=Surface2D(b);
29 h1=Line2D('Hole Group1');
30 h1=AddCircle(h1,IR,a,1);
31 S=AddHole(S,h1);

```

```

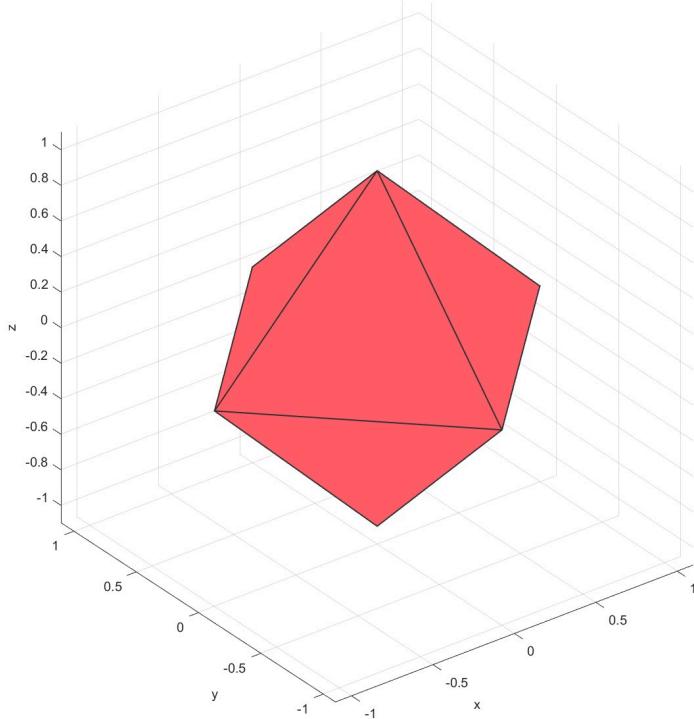
32 h2=Line2D('Hole Group2','Dtolt',0.1);
33 h2>AddCircle(h2, radius1, a, 1, 'sang', -ang1/2, 'ang', ang1);
34 h2>AddLine(h2, a, 3);
35 h2>AddCircle(h2, radius2, a, 1, 'sang', ang2/2, 'ang', -ang2);
36 h2>AddLine(h2, a, 4);
37 h2>CreateRadius(h2, 1, 10);
38 h2>CreateRadius(h2, 3, 10);
39 h2>CreateRadius(h2, 5, 10);
40 h2>CreateRadius(h2, 7, 10);
41 for i=1:num
42 S>AddHole(S, h2, 'rot', 360/num*(i-1));
43 end
44 m=Mesh2D('Mesh1');
45 m>AddSurface(m, S);
46 m/SetSize(m, 20);
47 m=Mesh(m);
48 Plot(m);
49 mm1=Mesh('Mesh');
50 mm1=Extrude2Solid(mm1, m, Thickness, 3);
51 PlotFace(mm1);
52 % Check inner
53 in = IsInMesh(mm, mm1);
54 mm = RemoveCells(mm, ~in);
55 PlotFace(mm);
56 mm=Remesh(mm, 10);
57 PlotFace(mm);
58 mm= Smoothface(mm, 10);
59 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
60 PlotFace(mm);
61 mm=Remesh(mm, 10);
62 PlotFace(mm);

```



## 2.19 MeshOctahedron (Flag=19)

```
1 dimension=[2,2,2];
2 mm1=Mesh('Mesh');
3 mm1=MeshOctahedron(mm1,dimension);
4 PlotFace(mm1);
```

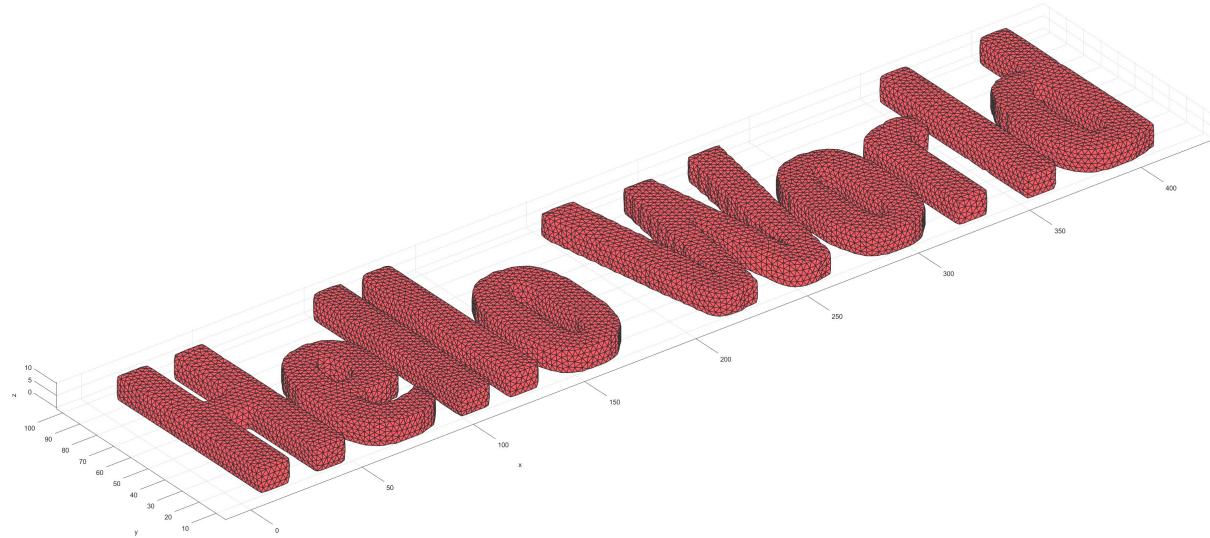
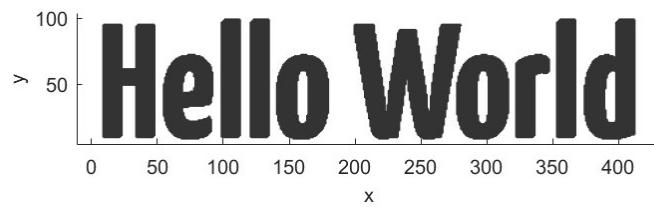


## 2.20 Image read (Flag=20)

```
1 hw=logical(imread('helloworld.png'));
2 imshow(hw);
3 m=Mesh2D('Mesh1');
4 nx=size(hw,2);ny=size(hw,1);
5 lx=size(hw,2)-1;ly=size(hw,1)-1;
6 m=MeshGrid(m,[nx, ny],[lx,ly]);
7 m= ComputeGeometryG(m);
8 PlotG(m);
9 m = RemoveCells(m,fliplr(hw'));
10 m = Quad2Tri(m);
11 PlotG(m);
12 mm=Mesh('Demo Hello World');
13 mm=Extrude2Solid(mm,m,10,3);
14 mm=Remesh(mm,5);
15 PlotFace(mm);
```

读取图片绘制网格：

Hello World



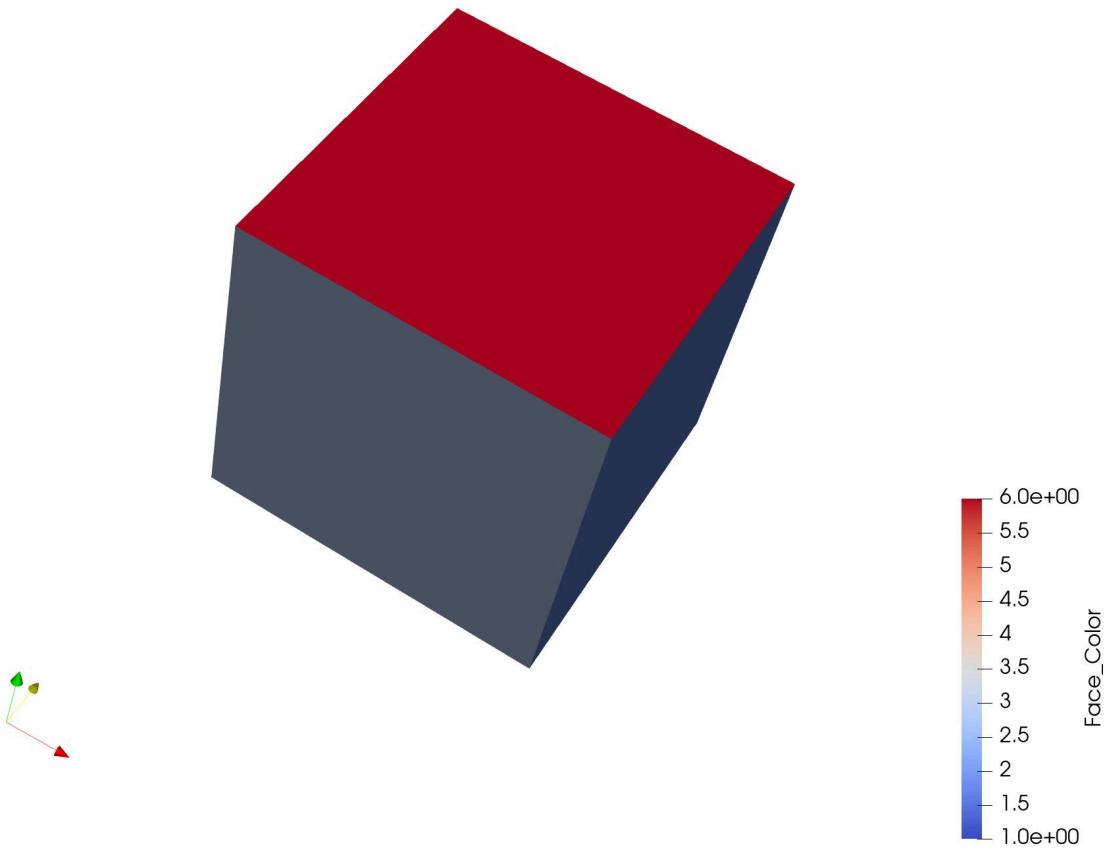
## 2.21 Plot Face in ParaView (Flag=21)

```

1  %% Specifying dimensions and number of elements
2  m=Mesh('Mesh');
3  %% Create a box with hexahedral elements
4  cubeDimensions=[10 10 10]; %Dimensions
5  cubeElementNumbers=[5 5 5]; %Number of elements
6  m=MeshCube(m,cubeDimensions,cubeElementNumbers);
7  %% Plot Face
8  PlotFace2(m);

```

在paraview显示表面网格：



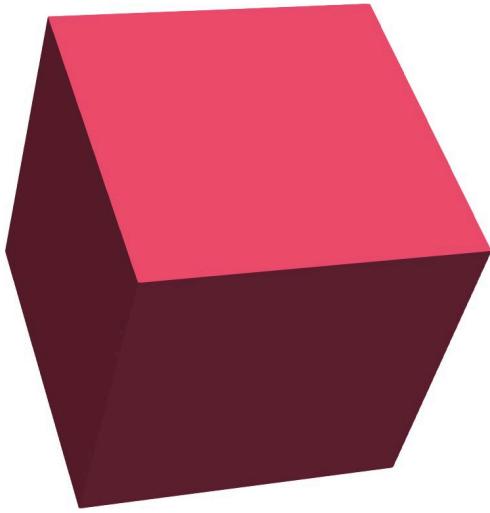
## 2.22 Plot Element in ParaView (Flag=22)

```

1 %% Specifying dimensions and number of elements
2 m=Mesh('Mesh');
3 %% Create a box with hexahedral elements
4 cubeDimensions=[10 10 10]; %Dimensions
5 cubeElementNumbers=[5 5 5]; %Number of elements
6 m=MeshCube(m,cubeDimensions,cubeElementNumbers);
7 %% Plot Element
8 PlotElement2(m)

```

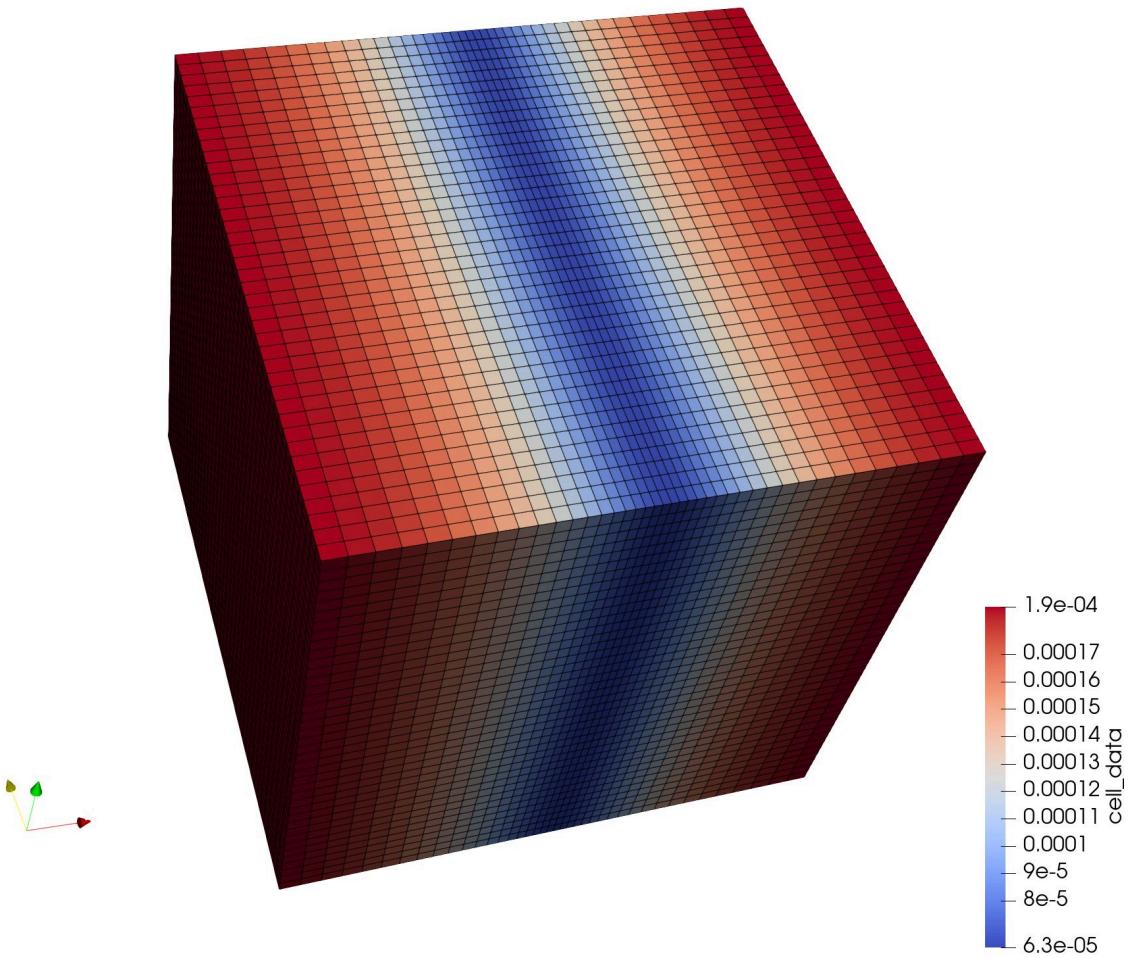
在paraview中显示实体网格：



## 2.23 Plot G and cell data in ParaView (Flag=23)

```
1 x=-1:0.05:1;
2 xx=x-1/2/pi.*sin(pi.*x);
3 yy=-1:0.05:1;
4 zz=-1:0.05:1;
5 mm=Mesh('Demo TensorGrid');
6 mm=MeshTensorGrid(mm,xx,yy,zz);
7 PlotFace(mm);
8 mm= ComputeGeometryG(mm);
9 mm=AddCellData(mm,mm.G.cells.volumes);
10 PlotG2(mm);
```

在Mesh中加入celldata



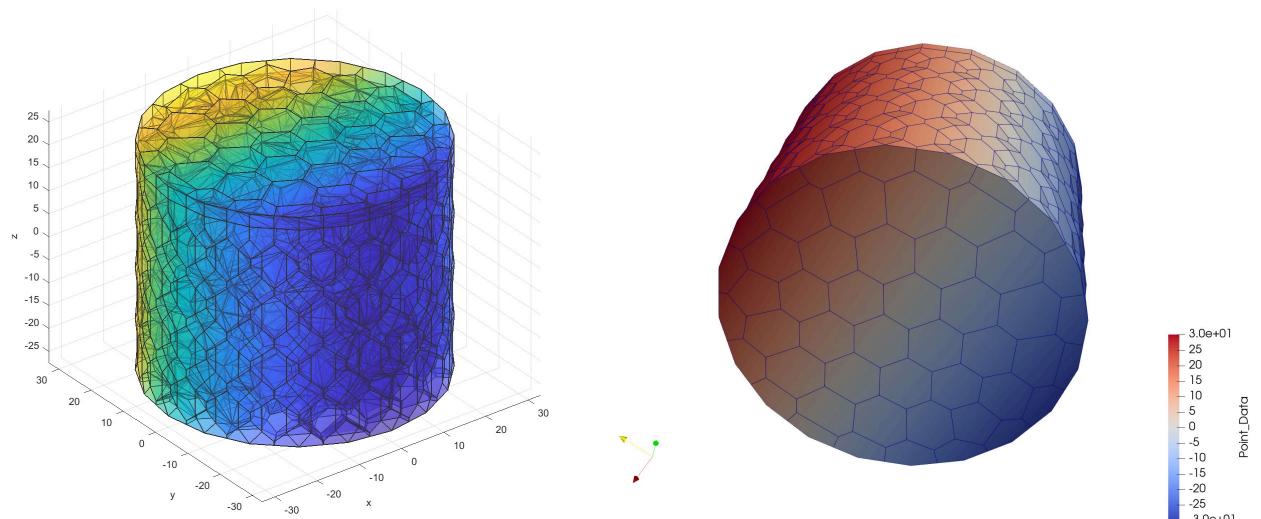
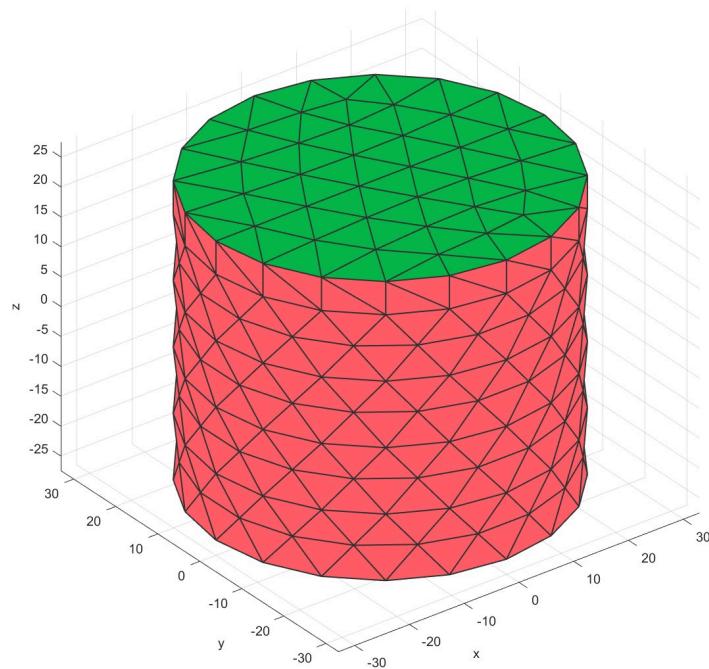
## 2.24 Tri3 to voronoi (Flag=24)

```

1 %Specifying dimensions and number of elements
2 mm=Mesh('Cylinder Mesh');
3 mm=MeshCylinder(mm,5,30,50);
4 PlotFace(mm);
5 % The cyclinder is meshed using tetrahedral elements using tetgen
6 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
7 mm=Tri2Voronoi(mm);
8 mm=AddPointData(mm,mm.Voronoi.Nodes(:,2));
9 PlotVoronoi(mm);
10 PlotVoronoi2(mm);

```

四面体网格转化为voronoi网格

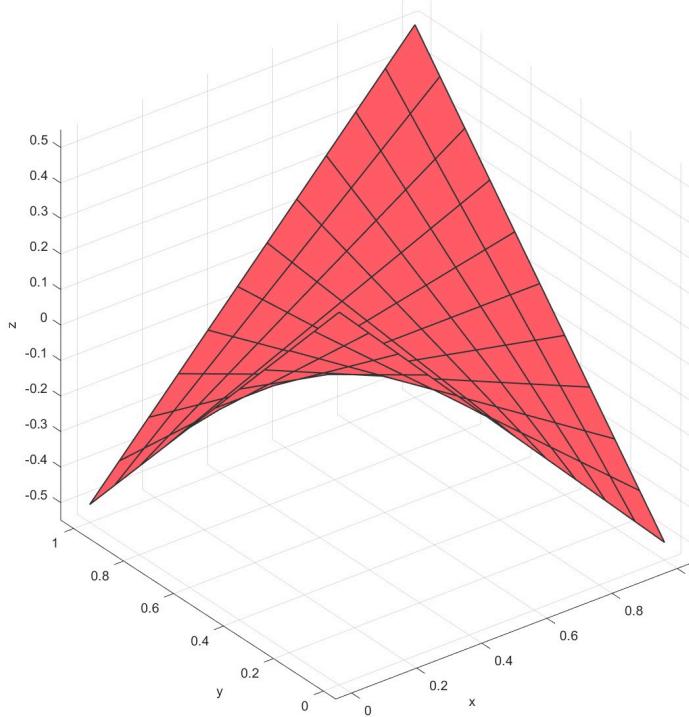


## 2.25 Nurb4Surf (Flag=25)

```

1 % Define 4 corner points of surface
2 mm=Mesh('Nurb surface');
3 Points=[0,0,0.5;1,0,-0.5;0,1,-0.5;1,1,0.5];
4 mm = Nurb4Surf(mm,Points,'subd',[11,10]);
5 PlotFace(mm);

```

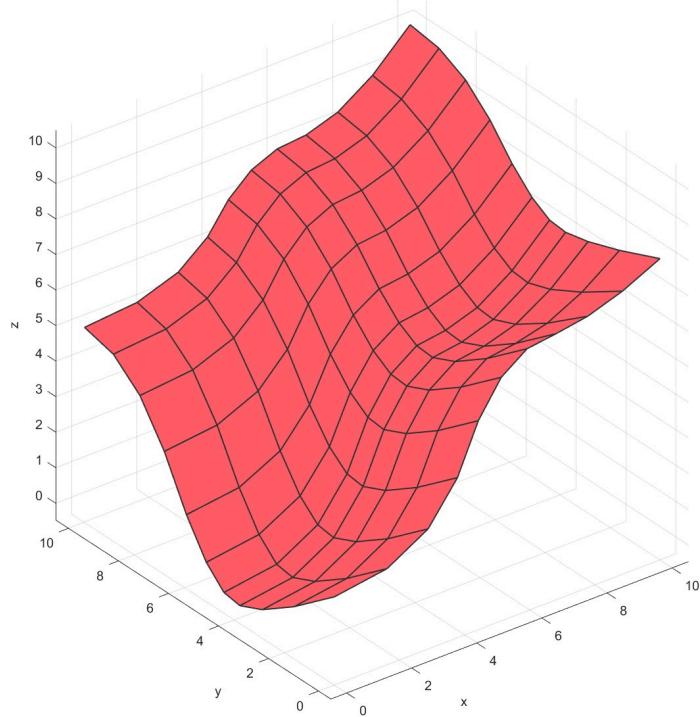


## 2.26 Nurb Surface (Flag=26)

```

1 % allocate multi-dimensional array of control points
2 pnts = zeros(3,5,5);
3 % define a grid of control points
4 % in this case a regular grid of u,v points
5 % pnts(3,u,v)
6 %
7 pnts(:,:,1) = [ 0.0 3.0 5.0 8.0 10.0; % w*x
8 0.0 0.0 0.0 0.0 0.0; % w*y
9 2.0 2.0 7.0 7.0 8.0]; % w*z
10 pnts(:,:,2) = [ 0.0 3.0 5.0 8.0 10.0;
11 3.0 3.0 3.0 3.0 3.0;
12 0.0 0.0 5.0 5.0 7.0];
13 pnts(:,:,3) = [ 0.0 3.0 5.0 8.0 10.0;
14 5.0 5.0 5.0 5.0 5.0;
15 0.0 0.0 5.0 5.0 7.0];
16 pnts(:,:,4) = [ 0.0 3.0 5.0 8.0 10.0;
17 8.0 8.0 8.0 8.0 8.0;
18 5.0 5.0 8.0 8.0 10.0];
19 pnts(:,:,5) = [ 0.0 3.0 5.0 8.0 10.0;
20 10.0 10.0 10.0 10.0 10.0;
21 5.0 5.0 8.0 8.0 10.0];
22 % knots
23 knots{1} = [0 0 0 1/3 2/3 1 1 1]; % knots along u
24 knots{2} = [0 0 0 1/3 2/3 1 1 1]; % knots along v
25 mm=Mesh('Nurb surface');
26 mm = NurbSurf(mm,pnts,knots);
27 PlotFace(mm);

```

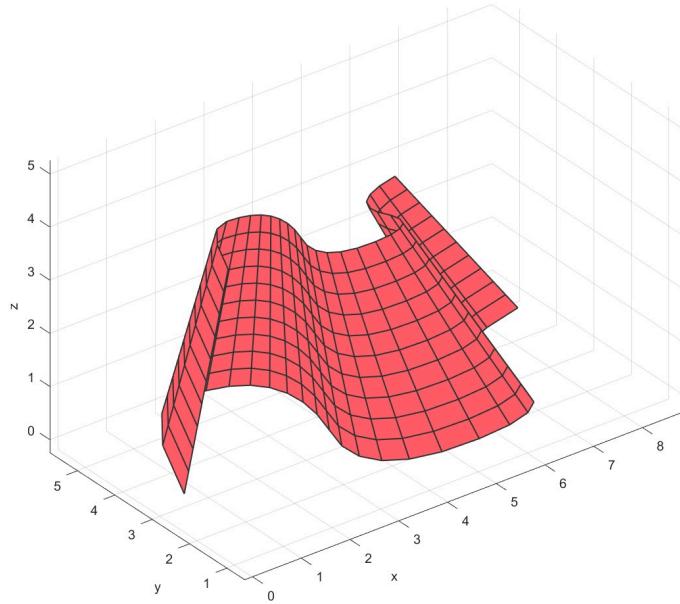


## 2.27 NurbRuled (Flag=27)

```

1 coefs = [0.5 1.5 4.5 3.0 7.5 6.0 8.5;
2 3.0 5.5 5.5 1.5 1.5 4.0 4.5;
3 0.0 0.0 0.0 0.0 0.0 0.0 0.0];
4 knots=[0 0 0 1/10 2/5 3/5 4/5 1 1 1];
5 b=Line('Line Ass1');
6 b=AddNurb(b,coefs,knots);
7 b=Move(b,1,[0,0,5], 'new',1);
8 b=Scale(b,2,[0.5,0.5,1]);
9 mm=Mesh('Nurb surface');
10 mm = NurbRuled(mm,b,1,2);
11 PlotFace(mm);

```



## 2.28 Extrude2Solid (Flag=28)

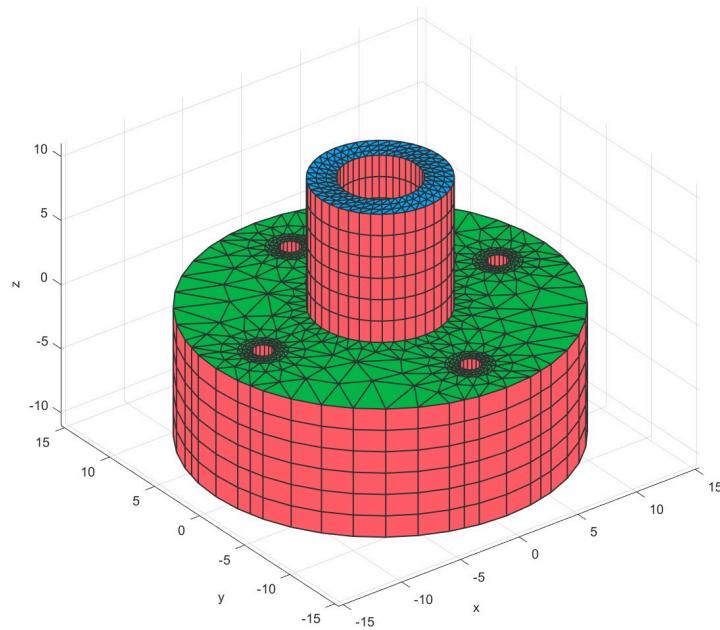
```

1 a=Point2D('Temp');
2 a=AddPoint(a,0,0);
3 % Add outline
4 b1=Line2D('Out');
5 b1=AddCircle(b1,14,a,1);
6 % Add innerline
7 b2=Line2D('Inner');
8 b2=AddCircle(b2,5,a,1,'seg',40);
9 % Add assembly hole
10 h=Line2D('Hole');
11 h=AddCircle(h,3.5/2,a,1,'seg',16);
12 S=Surface2D(b1);
13 S=AddHole(S,b2);
14 for i=1:4
15 S=AddHole(S,h,'dis',[10*cos(pi/2*(i-1)),10*sin(pi/2*(i-1))]);
16 end
17 m=Mesh2D('Temp');
18 m=AddSurface(m,S);
19 m=SetSize(m,5);
20 m=Mesh(m);
21 Plot(m)
22 % Mesh Layer edge
23 m=MeshLayerEdge(m,2,2);
24 % Mesh Layer edge
25 for i=3:6
26 m=MeshLayerEdge(m,2,1);
27 end
28 Plot(m)
29 mm=Mesh('Mesh1');
30 mm=Extrude2Solid(mm,m,-10,6);
31 PlotFace(mm);

```

```
32 mm=Extrude2Solid(mm,m,10,6,'Cb',2);
33 PlotFace(mm);
```

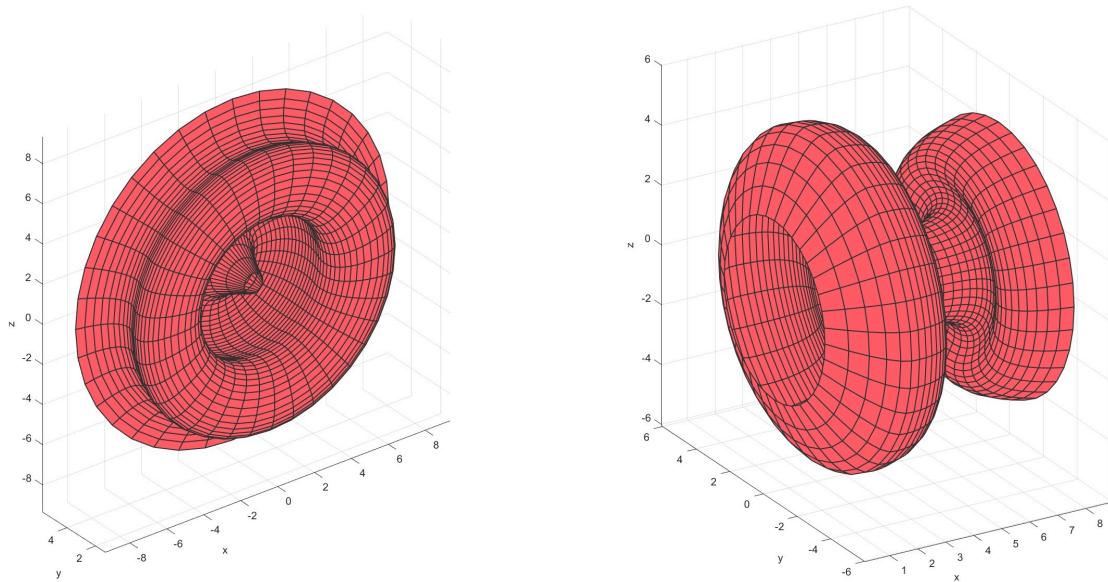
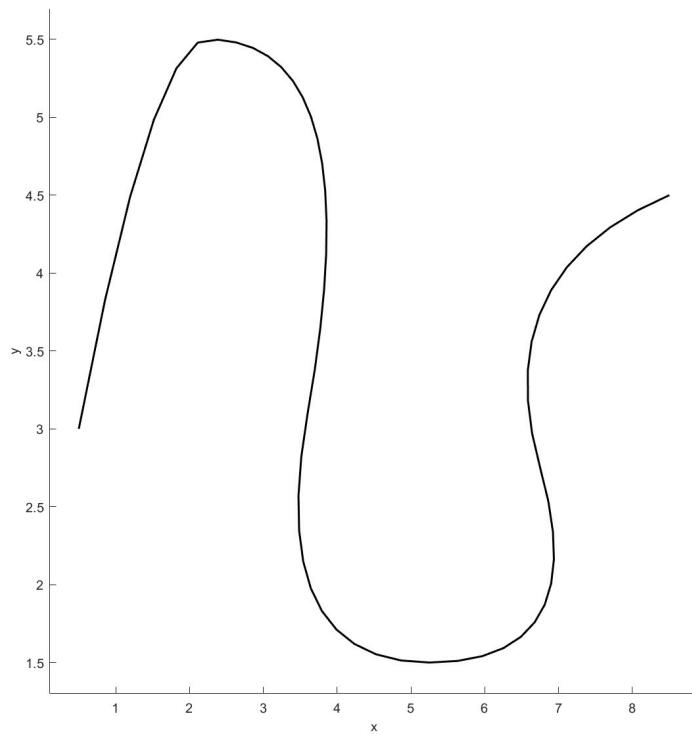
拉伸成体:



## 2.29 Rotate to shell (Flag=29)

```
1 Points = [0.5 1.5 4.5 3.0 7.5 6.0 8.5;
2 3.0 5.5 5.5 1.5 1.5 4.0 4.5]';
3 Knots=[0 0 0 1/10 2/5 3/5 4/5 1 1 1];
4 b=Line2D('Nurb Test');
5 b=AddNurb(b,Points,Knots);
6 Plot(b);
7 m1=Mesh('Mesh1');
8 m1=Rot2Shell(m1,b,'Type',2);
9 PlotFace(m1);
10 m2=Mesh('Mesh2');
11 m2=Rot2Shell(m2,b,'Type',1);
12 PlotFace(m2);
```

旋转成壳



## 2.30 DrillThroughHole (Flag=30)

```

1 % Derive patch data for a cylinder
2 mm=Mesh('Cylinder Mesh');
3 mm=MeshCylinder(mm,3,30,50);
4 PlotFace(mm);
5 % The cylinder is meshed using tetrahedral elements using tetgen
6 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
7 sec.pos=[0,0,0];
8 sec.vec=[1,0,0];
9 PlotElement(mm,'section',sec);
10 mm=DrillThroughHole(mm,[10;5],[20;50],[0,0],[3;2],'type',5,'slice',8);
11 mm=Mesh3D(mm); % Mesh model using tetrahedral elements using tetGen
12 PlotFace(mm);

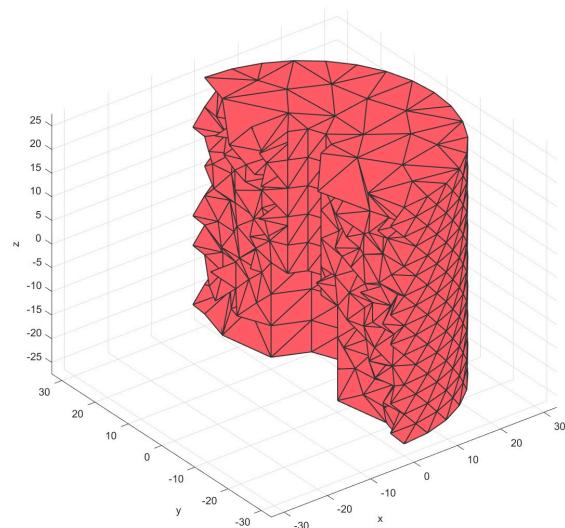
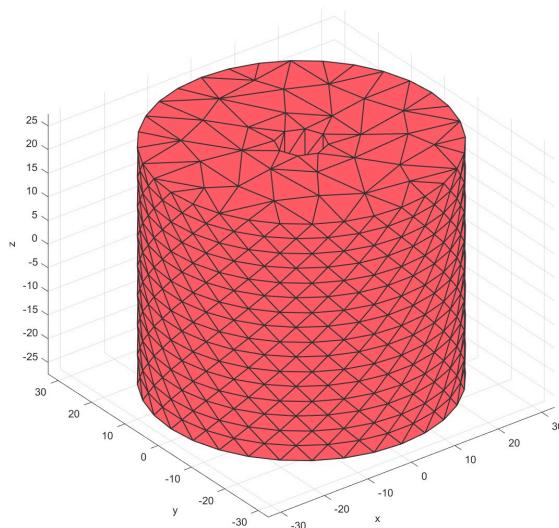
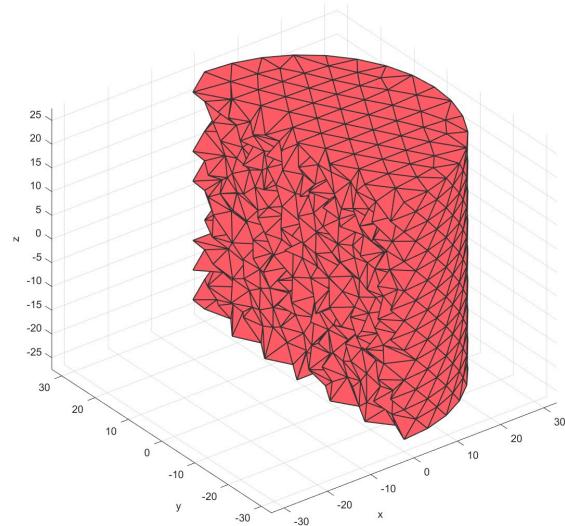
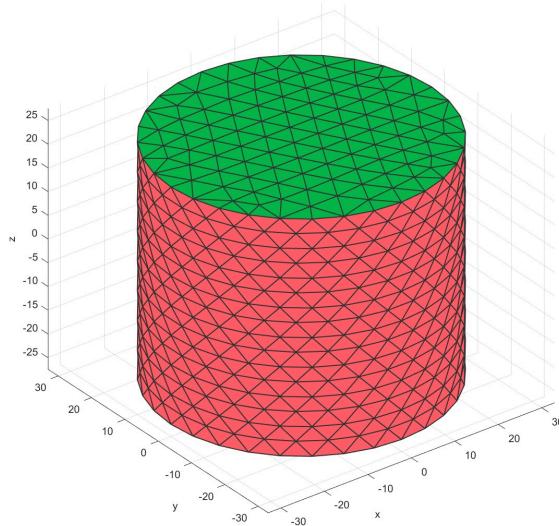
```

```

13 | sec.pos=[0,0,0];
14 | sec.vec=[1,0,0];
15 PlotElement(mm,'section',sec);

```

网格开通孔。



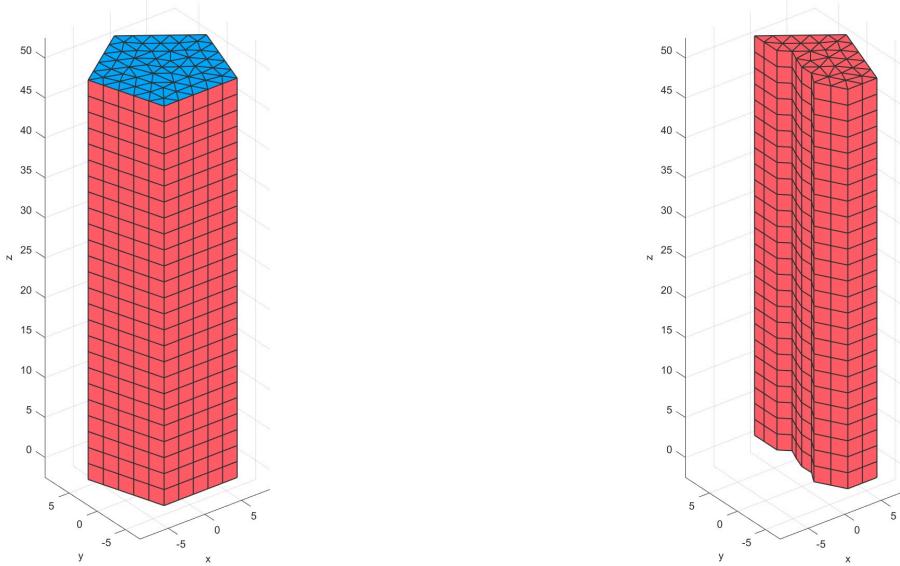
## 2.31 Mesh prism (Flag=31)

```

1 mm=Mesh('Prism Mesh');
2 mm=MeshPrism(mm,5,10,50);
3 PlotFace(mm);
4 sec.pos=[0,0,0];
5 sec.vec=[1,0,0];
6 PlotElement(mm,'section',sec);

```

棱柱网格。

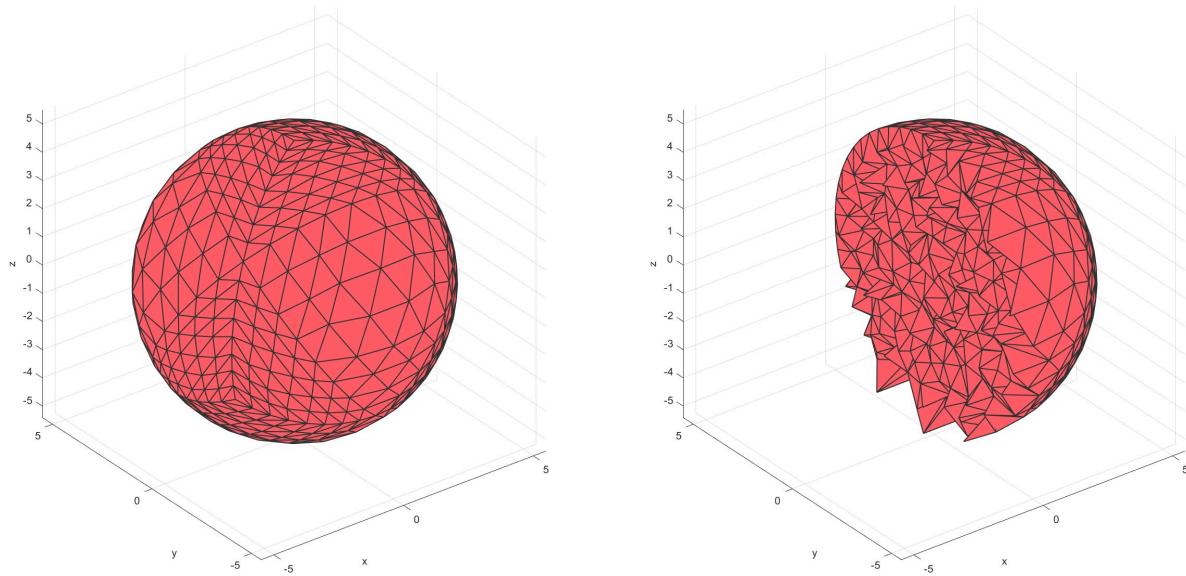


### 2.32 Mesh Geosphere (Flag=32)

```

1 mm=Mesh('Geosphere Mesh');
2 mm=MeshGeoSphere(mm,4,5,'Type',1);
3 PlotFace(mm);
4 mm=Mesh3D(mm);
5 sec.pos=[0,0,0];
6 sec.vec=[1,0,0];
7 PlotElement(mm,'section',sec);

```



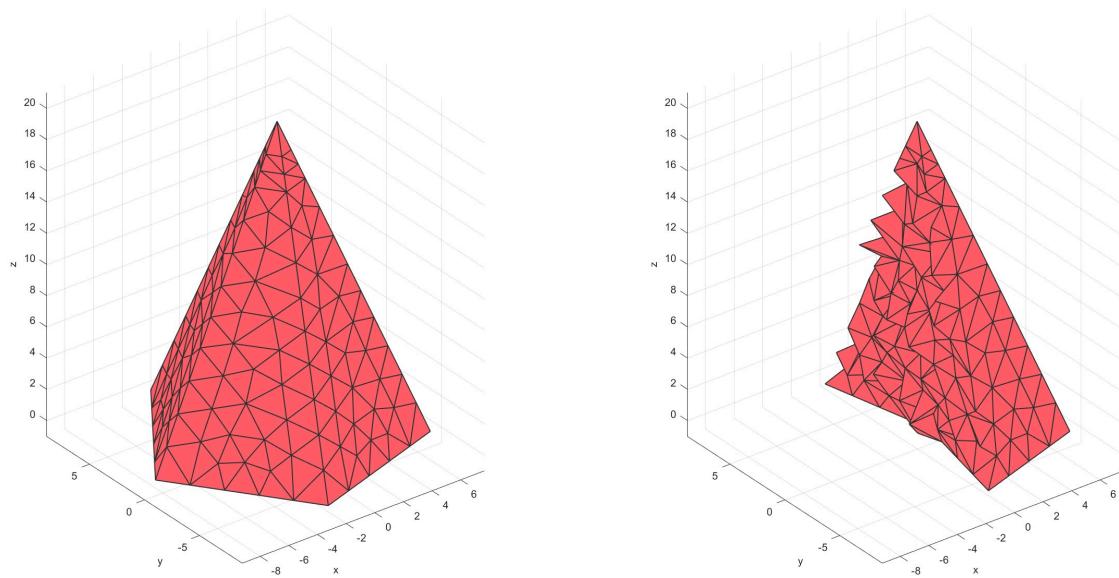
### 2.33 Mesh pyramid (Flag=33)

```

1 mm=Mesh('Pyramid Mesh');
2 mm=MeshPyramid(mm,5,10,20);
3 PlotFace(mm);
4 mm=Mesh3D(mm);
5 sec.pos=[0,0,0];
6 sec.vec=[1,0,0];
7 PlotElement(mm,'section',sec);

```

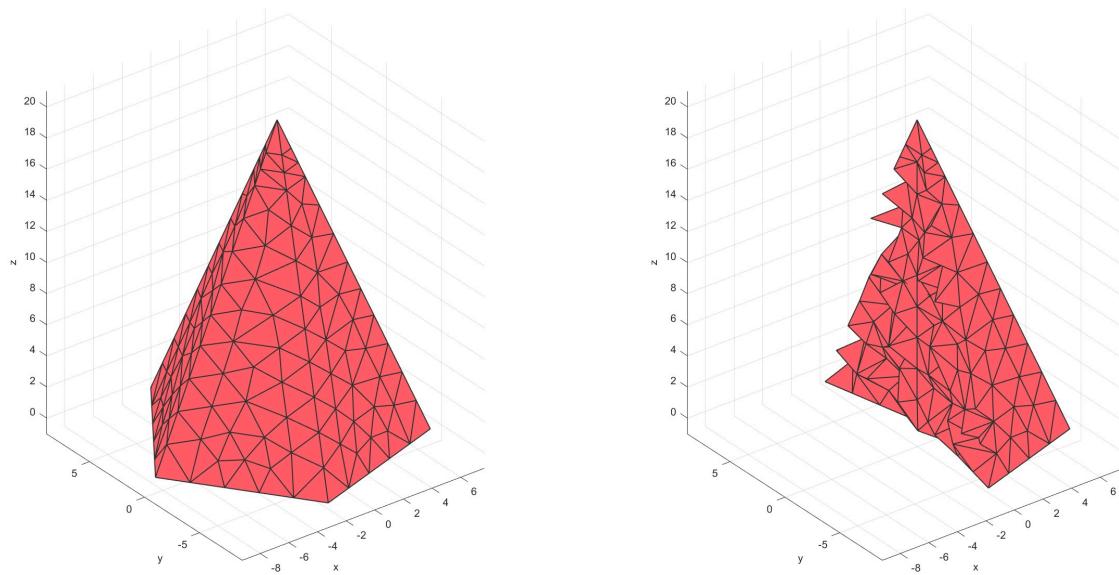
棱锥网格。



### 2.34 Add internal points (Flag=34)

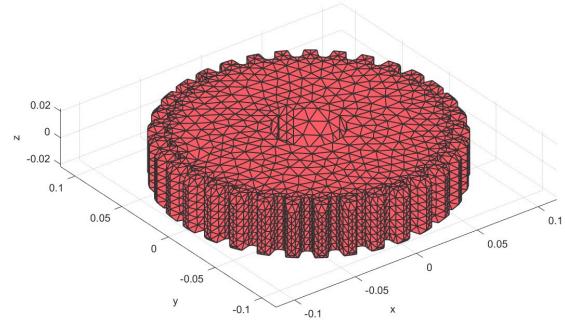
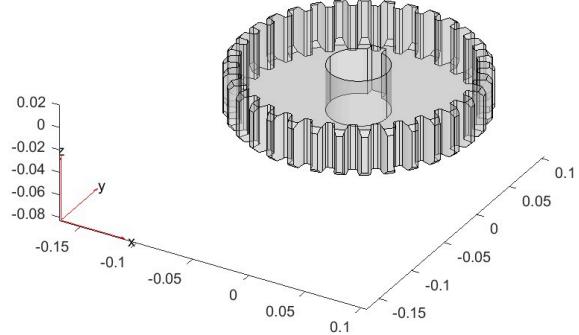
```
1 mm=Mesh('Pyramid Mesh');
2 mm=MeshPyramid(mm,5,10,20);
3 PlotFace(mm);
4 mm=Mesh3D(mm,'AddedNodes',[0,0,2;0,0,4;0,0,6;0,0,8;0,0,10]);
5 sec.pos=[0,0,0];
6 sec.vec=[1,0,0];
7 PlotElement(mm,'section',sec);
```

再网格内部加约束点，划分网格时会以内部点为约束生成网格。



## 2.35 Input STP file (Flag=35)

```
1 mm=Mesh('Gear');
2 mm=InputSTP(mm, 'Gear.step', 'Size', 0.01);
3 PlotFace(mm)
```



### 3 CheatTable

Name	Varargin	Description
AddCellData(obj,Data)		Add cell data to elements
AddPointData(obj,Data)		Add point data to nodes
AddPointVector(obj,Vec)		Add point vector to nodes
CenterCal(obj)		Calculate center of elements
Convert2Order2(obj)		Convert elements order
DelNullElement(obj)		Delete null elements
DrillHole(obj, radius,depth,position,faceno)	'type', 'slice','holetype'	Drill holes on elements face
DrillThroughHole(obj, radius,depth,position,faceno)	'type', 'slice','holetype'	Drill through holes on elements face
Extrude2Solid(obj,mesh2D,Height,Levels)	'Cb'	Extrude 2D elements to 3D elements
FindBoundary(obj)		Find boundary of elements
GetFaceNormal(obj)		Get elements face normal
GroupTest(obj)		Test face group
IsInMesh(obj,mesh)		Check geometry inner mesh face
InputSTP		Input STP file
KeepGroup(obj,GroupNum)		Keep group and remove unused nodes
MergeFaceNode(obj)		Merge face node
Mesh3D(obj)	'stringOpt', 'minRegionMarker'	Create 3D meshes
MeshCube(obj, cubeDimensions,cubeElementNumbers)	'ElementType', , 'outputStructType',	Mesh cube
MeshCylinder(obj,esize,Radius,Height)	'ElementType','close'  'coreRadius',	Mesh cylinder
MeshHemiSphere(obj,Radius,MantelNum)	'outputStructType', 'hollow','smoothstep'	Create Hemi Sphere mesh
MeshOctahedron(obj,dimension)		Mesh octahedron
MeshQuadSphere(obj,n,r)	'hollow','coreRadius'	Mesh QuadSphere
MeshSphere(obj,n,r)		Mesh sphere
MeshSpinodoid(obj,Length,paticleLength)	'Nw','WaveNum','relID', 'thetas','cap'	Mesh Spinodoid
MeshStochasticMicroStructure(obj, Length,paticleLength)	'Nw','q0','relID', 'anisotropy','cap'	Mesh Stochastic MicroStructure
MeshTensorGrid(obj,x,y,z)		Mesh tensor grid
MeshTriplyPeriodicMinimalSurface(obj, length)	'cap','surfaceCase', 'numPeriods',levelset',	Mesh Triply Periodic Minimal Surface
Nurb4Surf(obj,Points)	'subd'	4 points nurb surface
NurbRuled(obj,Points,crv1,crv2)	'subd'	Nurb ruled
NurbSurf(obj,pnts,knots)	'subd'	Nurb surface
PatchCentre(obj)		Get Center of Face boundary
PlotElement(obj)	'grid','axe','equal','alpha', 'face_alpha','section', 'xlim','ylim','zlim','view'	Plot element
PlotElement2(obj)		Plot elements in Paraview

Name	Varargin	Description
PlotFace(obj)	'grid','axe','equal', 'facecolor','alpha', 'face_alpha','face_normal', 'marker', 'xlim','ylim','zlim', 'view'	Plot Face of element
PlotFace2(obj)		Plot faces in Paraview
PlotG(obj)	'grid','name','axe','equal', 'alpha','face_alpha', 'section','volume'	Plot mesh MRST G
PlotG2(obj)		Plot mesh MRST G in Paraview
PlotVoronoi(obj)	'alpha','face_alpha','grid', 'equal','axe','marker'	Plot Voronoi
PlotVoronoi2(obj)		Plot voronoi mesh in Paraview
Quad2tri(obj)		Quad element to tri
Remesh(obj,length)	'anisotropy', 'pre_max_hole_area', 'pre_max_hole_edges', 'post_max_hole_area', 'post_max_hole_edges',	Remesh the elements
RemoveCells(obj,cells)		Remove cells
ReverseNormals(obj)		Reverse face normal
Revolve2Solid(obj,mesh2D)	'Slice','Type','Degree'	Revolve 2D elements to 3D elements
Rot2Shell(obj,Line2D)	'Slice','Type','Degree'	Revolve 1D elements to shell elements
Smoothface(obj,N,method)		Smooth face
STLWrite(obj)	'mode'	Write STL file
Tri2Voronoi(obj)		Tri to Voronoi mesh
VolumeCal(obj)		Calculate column of mesh
VTKWriteElement(obj)		Write VTK file of elements
VTKWriteFace(obj)	'facecolor'	Write VTK file of face
VTKWriteG(obj)		Write VTK file of G
VTKWriteVoronoi(obj)		Write VTK file of voronoi

## 4 参考文献

- [1] [https://www.mathworks.com/matlabcentral/fileexchange/48208-gibboncode-gibbon?s\\_tid=srchtitle](https://www.mathworks.com/matlabcentral/fileexchange/48208-gibboncode-gibbon?s_tid=srchtitle)
- [2] <https://www.gibboncode.org>
- [3] <https://www.sintef.no/projectweb/mrst/>
- [4] <https://wias-berlin.de/software/index.jsp?id=TetGen&lang=1>
- [5] [https://www.mathworks.com/matlabcentral/fileexchange/94955-3d-voronoi-via-fem-mesh-the-voronoi3d-class-visualizatio\\_n?s\\_tid=srchtitle](https://www.mathworks.com/matlabcentral/fileexchange/94955-3d-voronoi-via-fem-mesh-the-voronoi3d-class-visualizatio_n?s_tid=srchtitle)