SubModel

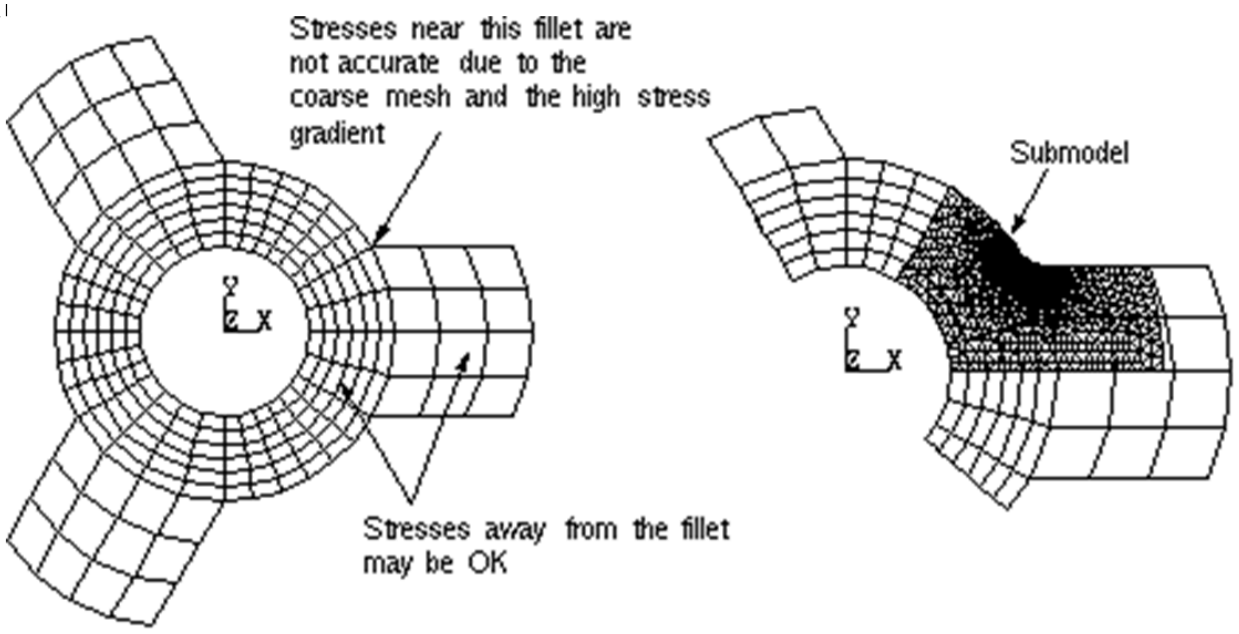
Xie Yu

1 介绍

SubModel类主要用于子模型分析。

2 原理

子模型是得到模型部分区域中更加精确解的有限单元技术。在有限元分析中往往出现这种情况，即对于用户关心的区域，如应力集中区域，网格太疏不能得到满意的结果，而对于这些区域之外的部分，网格密度已经足够了。



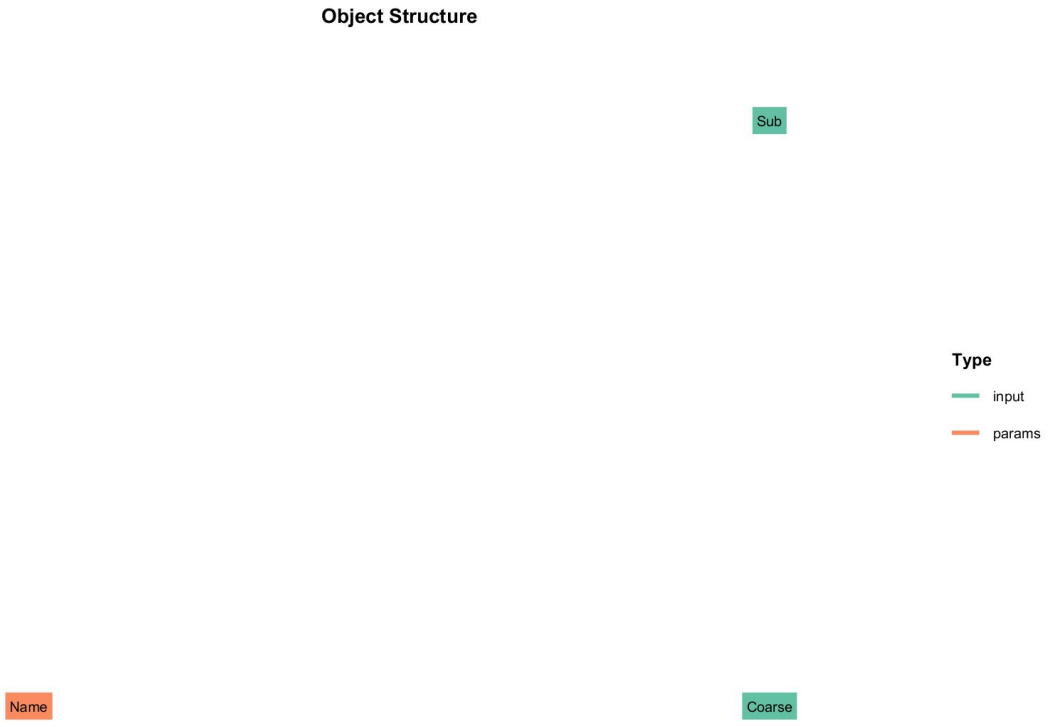
要得到这些区域的较精确的解，可以采取两种办法：(a)用较细的网格重新划分并分析整个模型，或(b)只在关心的区域细化网格并对其分析。显而易见，方法a太耗费机时，方法b即为子模型技术。

子模型方法又称为切割边界位移法或特定边界位移法。切割边界就是子模型从整个较粗糙的模型分割开的边界。整体模型切割边界的计算位移值即为子模型的边界条件。

子模型分析的过程包括以下步骤：

1. 生成并分析较粗糙的模型。
2. 生成子模型。
3. 提供切割边界插值。
4. 分析子模型。
5. 验证切割边界和应力集中区域的距离应足够远。

3 类结构



输入 input：

- Sub : 子模型网格
- Coarse : 粗网格

参数 params:

- Name : 名称

4 案例

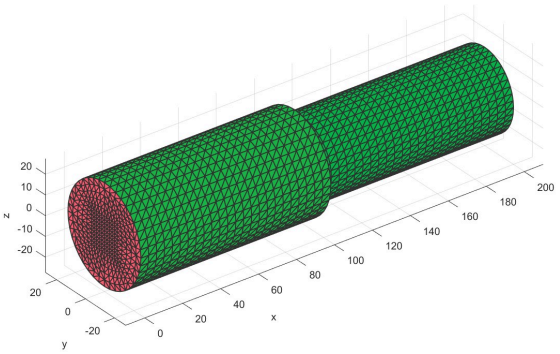
4.1 Shaft radius submodel analysis (Flag=1)

本案例是一个轴倒角应力集中的计算，在得到粗网格的计算结果，可直接通过插值边界的方式得到局部细网格的应力结果。

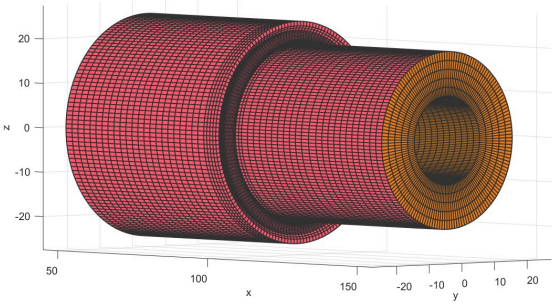
```
% Shaft
inputshaft1.Length = [100;200];
inputshaft1.ID = [[0,0];[0,0]];
inputshaft1.OD = [[50,50];[40,40]];
paramsshaft1 = struct();
Shaft1= shaft.Commonshaft(paramsshaft1, inputshaft1);
Shaft1 = Shaft1.solve();
Plot3D(Shaft1);
%% Assembly
Ass1=Assembly('Assembly1');
Ass1=AddAssembly(Ass1,Shaft1.output.Assembly);
%% Load
Load=[0,0,-1e4,0,0,0];
```

```
Ass1=AddLoad(Ass1,1,'No',303);
Ass1=SetLoad(Ass1,1,Load);
%% Boundary
Bound1=[1,1,1,0,0,0];
Ass1=AddBoundary(Ass1,1,'No',301);
Ass1=SetBoundaryType(Ass1,1,Bound1);
%% Solution
opt.ANTYPE=0;
Ass1=AddSolu(Ass1,opt);
% Housing
a=Point2D('Point Ass1');
a=AddPoint(a,[50;50],[10;25]);
a=AddPoint(a,[50;100],[25;25]);
a=AddPoint(a,[100;100],[25;24]);
a=AddPoint(a,[100;100-4/sqrt(3)],[24;20]);
a=AddPoint(a,[100-4/sqrt(3);150],[20;20]);
a=AddPoint(a,[150;150],[20;10]);
a=AddPoint(a,[150;50],[10;10]);
b=Line2D('Line Ass1');
for i=1:7
  b=AddCurve(b,a,i);
end
b=CreateRadius(b,4,2);
inputSub.Outline= b;
inputSub.Meshsize=2;
paramsSub.Degree = 360;
paramsSub.N_Slice=160;
Sub=housing.Housing(paramsSub, inputSub);
Sub=Sub.solve();
mm=Sub.output.SolidMesh;
Vm=PatchCenter(mm);
rm=sqrt(Vm(:,2).^2+Vm(:,3).^2);
Cb=mm.Cb;
Cb(or(Vm(:,1)==50,Vm(:,1)==150),:)=11;
Cb(rm<=10,:)=12;
mm.Cb=Cb;
mm.Meshoutput.boundaryMarker=Cb;
Sub.output.SolidMesh=mm;
Sub=OutputAss(Sub);
Plot3D(Sub);
%% Assembly
Ass2=Assembly('Assembly2');
Ass2=AddAssembly(Ass2,Sub.output.Assembly);
%% CutBoundary
Ass2=AddCutBoundary(Ass2,[1;1],[11;12]);
%% Solution
opt.ANTYPE=0;
Ass2=AddSolu(Ass2,opt);
inputStruct.Coarse=Ass1;
inputStruct.Sub=Ass2;
paramsStruct=struct();
obj= solve.SubModel(paramsStruct, inputStruct);
obj= obj.solve();
Plot3D(obj)
```

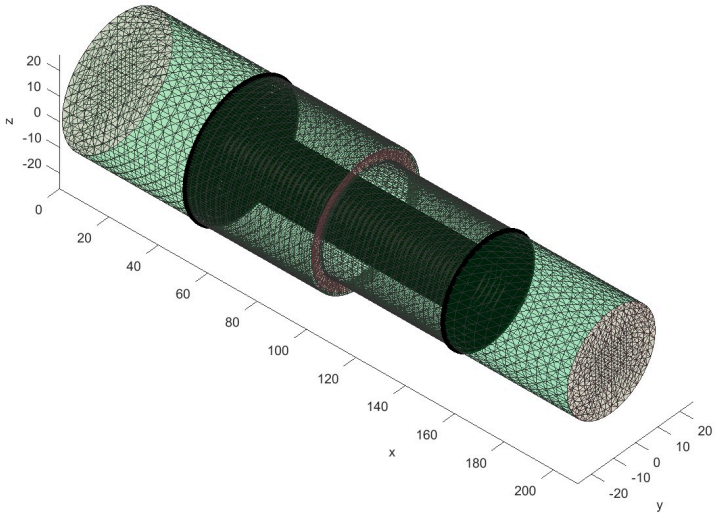
粗网格和细网格示例和插值边界：



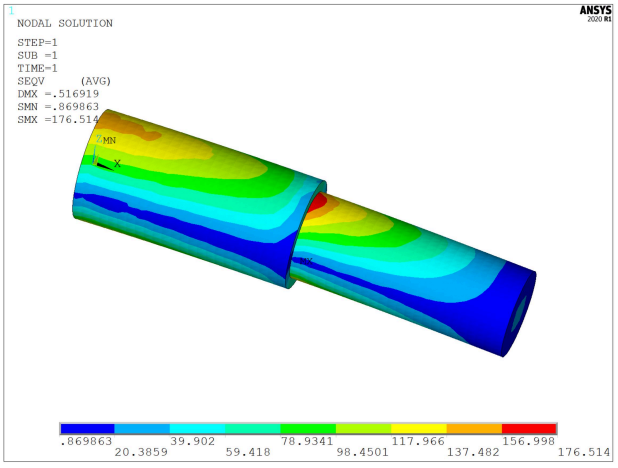
粗网格

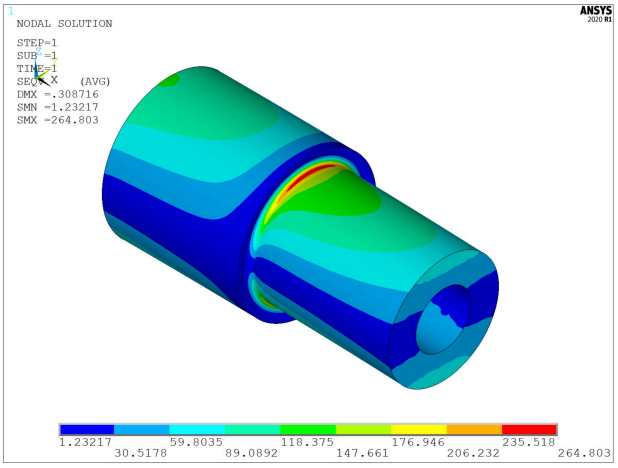


细网格



计算结果如下，可以看到倒角处的应力为265Mpa。





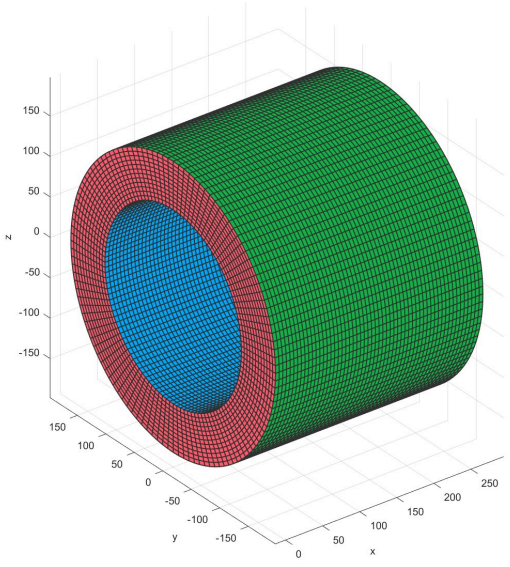
4.2 Interference fit submodel analysis (Flag=2)

本案例用来分析过盈边缘处开应力释放槽处的应力。

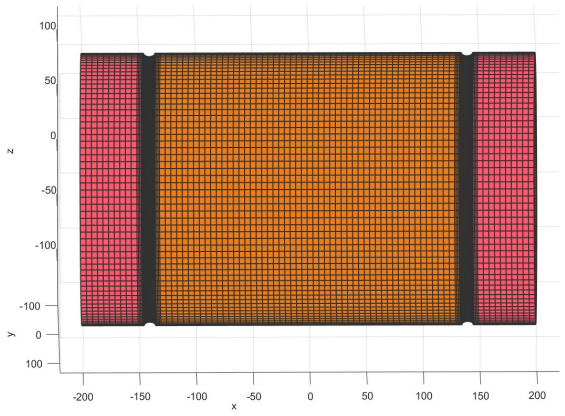
```
S=RMaterial('Basic');
Mat=GetMat(S,22);
inputStruct.Hub_Mat=Mat{1,1};
inputStruct.Shaft_Mat=Mat{1,1};
inputStruct.DaA=440;
inputStruct.DF=240;
inputStruct.Dil=0;
inputStruct.LF=280;
inputStruct.Umin=0.094;
inputStruct.Umax=0.169;
paramsStruct.RzA=32;
paramsStruct.Rzl=32;
Con1=connection.InterferenceFit(paramsStruct, inputStruct);
Con1=Con1.solve;
Con1=OutputAss(Con1);
Ass1=Con1.output.Assembly;
% Shaft
a=Point2D('Point Ass1');
a=AddPoint(a,[-200;-145],[120;120]);
a=AddPoint(a,-140,120);
a=AddPoint(a,[-135;135],[120;120]);
a=AddPoint(a,140,120);
a=AddPoint(a,[145;200],[120;120]);
a=AddPoint(a,[200;200],[120;60]);
a=AddPoint(a,[200;-200],[60;60]);
a=AddPoint(a,[-200;-200],[60;120]);
b=Line2D('Line Ass1');
b=AddCurve(b,a,1);
b=AddEllipse(b,5,3,a,2,'sang',180,'ang',180);
b=AddCurve(b,a,3);
b=AddEllipse(b,5,3,a,4,'sang',180,'ang',180);
for i=5:8
  b=AddCurve(b,a,i);
end
inputShaft.Outline= b;
inputShaft.Meshsize=5;
paramsShaft.Degree = 360;
paramsShaft.N_Slice=120;
```

```
Shaft=housing.Housing(paramsShaft, inputShaft);
Shaft = Shaft.solve();
mm=Shaft.output.SolidMesh;
Vm=PatchCenter(mm);
rm=sqrt(Vm(:,2).^2+Vm(:,3).^2);
Cb=mm.Cb;
tol=0.2;
Cb(and(abs(Vm(:,1))<135,rm(:,1)>=120-tol),:)=11;
Cb(or(Vm(:,1)==-200,Vm(:,1)==200),:)=12;
Cb(rm(:,1)<=60,:)=13;
mm.Cb=Cb;
mm.Meshoutput.boundaryMarker=Cb;
Shaft.output.SolidMesh=mm;
Shaft=OutputAss(Shaft);
Plot3D(Shaft)
% hub
inputHub.Length = 280;
inputHub.ID = [240,240];
inputHub.OD = [360,360];
inputHub.Meshsize=5;
paramsHub.E_Revolve = 120;
Hub = shaft.Commonshaft(paramsHub, inputHub);
Hub = Hub.solve();
Plot3D(Hub);
%% Assembly
Ass2=Assembly('Assembly2');
Ass2=AddAssembly(Ass2,Shaft.output.Assembly);
position=[-140,0,0,0,0,0];
Ass2=AddAssembly(Ass2,Hub.output.Assembly,'position',position);
%% ET
ET1.name='173';ET1.opt=[10,2];ET1.R=[]; % Standard contact
Ass2=AddET(Ass2,ET1);
ET2.name='170';ET2.opt=[];ET2.R=[];
Ass2=AddET(Ass2,ET2);
Acc_ET=GetNET(Ass2);
mat1.table=["MU",Con1.params.uf];
Ass2=AddMaterial(Ass2,mat1);
Acc_Mat=GetNMaterial(Ass2);
%% Contact
Ass2=AddCon(Ass2,1,11);
Ass2=AddTar(Ass2,1,2,201);
Ass2=SetConMaterial(Ass2,1,Acc_Mat);
Ass2=SetConET(Ass2,1,Acc_ET-1);
Ass2=SetTarET(Ass2,1,Acc_ET);
option=[10,Con1.output.Uwmin/2];
Ass2=SetConRealConstants(Ass2,1,option);
%% CutBoundary
Ass2=AddCutBoundary(Ass2,[1;1],[12;13]);
Ass2=AddCutBoundary(Ass2,2,101);
%% Solution
opt.ANTYPE=0;
Ass2=AddSolu(Ass2,opt);
inputStruct.Coarse=Ass1;
inputStruct.Sub=Ass2;
paramsStruct=struct();
obj= solve.SubModel(paramsStruct, inputStruct);
```

```
Plot3D(obj)
obj= obj.solve();
```

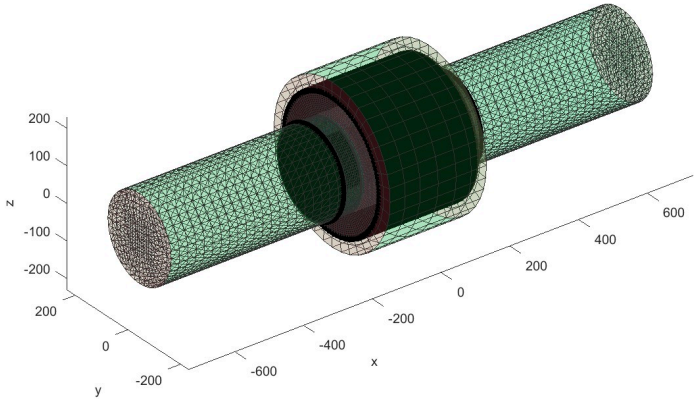


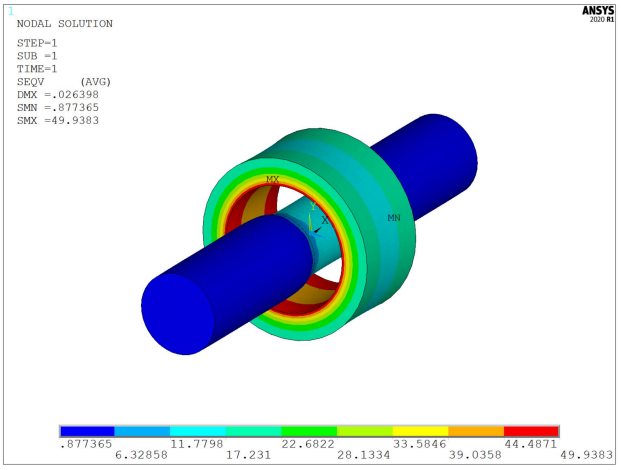
轴套

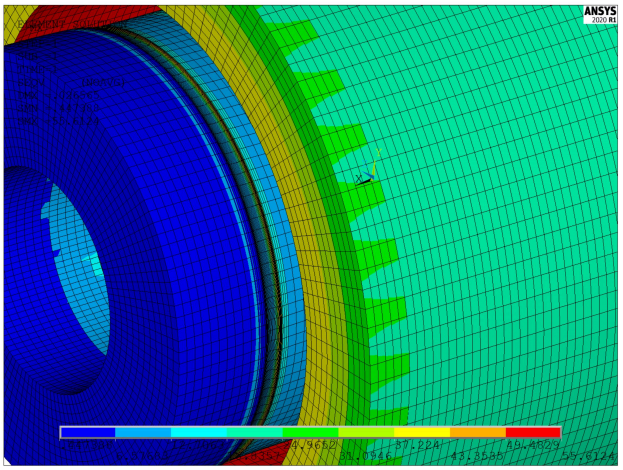


轴

插值边界如下：







5 参考文献

[1] ANSYS帮助文件