|  |
| --- |
| **附件清单** |
| 1. **数据预处理准备工作** 2. **问题一主程序** 3. **问题二主程序** 4. **问题二检验** 5. **问题二优化过程** 6. **问题三主程序1** 7. **问题三主程序2** 8. **程序运行所需m函数文件** |

|  |
| --- |
| **1.数据预处理准备工作（Prepare.m）** |
| %找出三角板、节点、节点坐标的调用关系，即MeshGrid、Node\_ID、NodeLoc  clear;clc;close all;  %载入节点数据，[num,txt]=xlsread('xxx')  [NodeLoc,NodeId1]=xlsread('附件1.csv');  [NodeLoc2,NodeId2]=xlsread('附件2.csv');  [~,NodeId3]=xlsread('附件3.csv');  R=abs(min(NodeLoc(:,3)));%记录离原点（球心）最远的坐标点  f=0.466;%焦距  NodeNum=size(NodeLoc,1);%记录主索节点个数，size（n,1）返回行数  MeshNum=size(NodeId3,1)-1;%记录反射面板个数  MeshGrid=zeros(MeshNum,3);%预分配理想矩阵与反射面板中节点一一对应  for ii=1:NodeNum%乱序节点遍历      NodeId1Name=NodeId1{ii+1,1};%对应节点编号      Row=NodeId1Name(1)-'A'+1;%首字母编号转化为数字编号，A为1，B为2...      Col=str2double(NodeId1Name(2:end))+1;      Node\_ID(Row,Col)=ii;%使NodeLoc中每一行坐标对应其节点编号  end  %使三角面板（MeshGrid）与节点编号（Node\_ID）一三对应  for ii=1:MeshNum%三角反射面板遍历      for jj=1:3          NodeId1Name=NodeId3{ii+1,jj};%对应三角面板中某一个节点          Row=NodeId1Name(1)-'A'+1;          Col=str2double(NodeId1Name(2:end))+1;          MeshGrid(ii,jj)=Node\_ID(Row,Col);      end  end  save NodeInfo NodeLoc  NodeLoc2 NodeId1 NodeId2 NodeId3…  Node\_ID MeshGrid NodeNum R f |

|  |
| --- |
| **2.问题一主程序（Q1.m）** |
| clear;clc;close all;  load NodeInfo  %将节点三维直角坐标转化成相应的二维极坐标  Loc\_ji=zeros(NodeNum,3);  for ii=1:NodeNum    Loc\_ji(ii,1)=atan(norm(NodeLoc(ii,1:2))/abs(NodeLoc(ii,3)));%θ，表示节点与球心间的连线和Z坐标轴的夹角    Loc\_ji(ii,2)=norm(NodeLoc(ii,:));%ρ，% 表示节点与球心间的距离  end  %更新坐标  [detall,LocNew,ff]=Update(NodeLoc);  %绘图  figure('color','w');  figure(1);clf;hold on;  PlotModel(LocNew,MeshGrid,'y');  PlotModel(NodeLoc,MeshGrid,'b');  title('基准态（蓝色）与理想抛物面（黄色）')  % 300m内伸缩范围  detall2=detall;  for ii=1:NodeNum      if Loc\_ji(ii,3)==0         detall2(ii,:)=0;      end  end  detall2(all(detall2==0,2),:)=[];%删除全0行  figure('color','w');  x=1:length(detall2);  plot(x,detall2')  ylabel('伸缩量(m)')  %输出  format short  disp('搜寻到焦距')  ff  disp('搜寻到顶点z坐标为')  peak=LocNew(1,3) |

|  |
| --- |
| **3.问题2主程序（Q2.m）** |
| clc;clear;close all;  load NodeInfo  LocOld=NodeLoc;  %旋转坐标系  NodeLoc=xz(NodeLoc);  %绘图  figure('color','w');  figure(1);  scatter3(LocOld(:,1),LocOld(:,2),LocOld(:,3),'r.')  axis([-320 320 -320 320 -320 0]);  axis equal  title('原坐标系')  figure('color','w');  figure(2)  scatter3(NodeLoc(:,1),NodeLoc(:,2),NodeLoc(:,3),'b.');  axis([-320 320 -320 320 -320 0]);  axis equal  title('旋转坐标系后')  %旋转后就可以套用问一的模型了  [detall,LocNew,ff,Loc\_ji]=Update(NodeLoc);  %进行坐标系的还原  LocNew=hy(LocNew);  peak=[0,0,-300.8446]; %抛物线顶点坐标转化  peak=hy(peak);  %剔除300m外的点  LocNew2=LocNew;  for ii=1:NodeNum      if Loc\_ji(ii,3)==0         LocNew2(ii,:)=0;      end  end  LocNew2(all(LocNew2==0,2),:)=[];%删除全0行  %绘图  figure(3)  figure('color','w');  scatter3(LocOld(:,1),LocOld(:,2),LocOld(:,3),'b.')  hold on  scatter3(LocNew2(:,1),LocNew2(:,2),LocNew2(:,3),'y')  axis([-320 320 -320 320 -320 0]);  hold on;  tu=[peak;[0,0,0]];  plot3(tu(:,1),tu(:,2),tu(:,3),'y','Linewidth',1.2);  axis equal  grid off  title('基准态（蓝色）与照明区域（黄色）')  figure('color','w');clf;hold on;  PlotModel(LocNew,MeshGrid,'y');  PlotModel(LocOld,MeshGrid,'b');  title('基准态（蓝色）与理想抛物面（黄色）')  %300m内伸缩范围  detall2=detall;  for ii=1:NodeNum      if Loc\_ji(ii,3)==0         detall2(ii,:)=0;      end  end  detall2(all(detall2==0,2),:)=[];%删除全0行  figure('color','w');  x=1:length(detall2);  plot(x,detall2');  ylabel('伸缩量(m)') |

|  |
| --- |
| **4.问题2检验（Jury.m）** |
| for ii=1:MeshNum\*3      if jury2(ii)==0          count=count+1;      end  end  rate=count/(MeshNum\*3);%所有点的达标率  %记录超标的点  trial=LocNew;  Chao=zeros(MeshNum,1);  for ii=1:MeshNum          if  jury2(ii,1)==1              Chao(MeshGrid(ii,1),1)=1;              Chao(MeshGrid(ii,2),1)=1;          end          if  jury2(ii,2)==1              Chao(MeshGrid(ii,1),1)=1;              Chao(MeshGrid(ii,3),1)=1;          end          if  jury2(ii,3)==1              Chao(MeshGrid(ii,2),1)=1;              Chao(MeshGrid(ii,3),1)=1;          end  end  %保留不达标的点，删除达标的点  for ii=1:NodeNum      if Chao(ii)==0      trial(ii,:)=0;      end  end  trial(all(trial==0,2),:)=[];  %绘制区域图  figure('color','w');clf;hold on;  PlotModel(LocNew,MeshGrid,'y');  PlotModel(LocOld,MeshGrid,'b');  scatter3(trial(:,1),trial(:,2),trial(:,3),'r.');  title('基准态（蓝色）与理想抛物面（黄色）、超标点（红色）')  %300m范围内的达标率  rate2=1-length(trial)/sum(detall~=0);  %检验伸缩量是否超标0.6m  jury1=zeros(MeshNum,3);  jury1(abs(detall)-0.6>0)=1;%超出约束  jury1(abs(detall)-0.6>0)=1;%没有超出约束 |

|  |
| --- |
| **5.问题二优化过程（Q2\_Opt.m）** |
| clc,clear;  load NodeInfo  %save MeshGrid MeshGrid  load MeshGrid  figure(1)  plot3(NodeLoc(:,1),NodeLoc(:,2),NodeLoc(:,3),'r')  a=-36.795/180\*pi;  b=-(90-78.169)/180\*pi;  Hz=[cos(a),-sin(a),0;sin(a),cos(a),0;0,0,1];  Hy=[cos(b),0,sin(b);0,1,0;-sin(b),0,cos(b)];  %Hx=[1,0,0;0,cos(b),-sin(b);0,sin(b),cos(b)];  [m,n]=size(NodeLoc);  XZ=zeros(m,n);  for i=1:m      XZ(i,:)=Hz\*NodeLoc(i,:)';      XZ(i,:)=Hy\*XZ(i,:)';%XZ旋转后基准态的坐标  end  figure(2)  plot3(XZ(:,1),XZ(:,2),XZ(:,3),'b')  xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签  hold on  ff=140.431;  figure(2)  plot3(XZ(:,1),XZ(:,2),XZ(:,3),'b')  hold on  x=-150:1:150;  y=-150:1:150;  [x,y]=meshgrid(x,y);  z=(x.^2+y.^2)/(4\*ff)-300.8446;  plot3(x,y,z,'r')  xlabel('x轴'); ylabel('y轴'); zlabel('z轴'); % 加上坐标轴的标签  %此时顶点坐标为(0,0,-3000.8446)  t5=[0,0,-300.8446]';  a=36.795/180\*pi;%还原  b=(90-78.169)/180\*pi;  Hz=[cos(a),-sin(a),0;sin(a),cos(a),0;0,0,1];  Hy=[cos(b),0,sin(b);0,1,0;-sin(b),0,cos(b)];  tr=Hy\*t5;  tr=Hz\*tr ;%还原后的顶点坐标  JZ=XZ;%XZ旋转后坐标赋值给基准态  %load Loc\_ji  %ID=[1:1:2226]';  %ID(Loc\_ji(:,3)==0,:)=[];%保留点对应的编号  %JZ(Loc\_ji(:,3)==0,:)=[];%将不在300米内的点删去  [m,n]=size(JZ);  %XZ旋转后基准态的坐标转化为球坐标,先求第5步  JZ\_Q=zeros(m,n);%基准态球坐标  for i=1:m      x=XZ(i,1);y=XZ(i,2);z=XZ(i,3);      cita=atan(y/x);      rou=sqrt(x^2+y^2+z^2);      fai=acos(z/rou);      JZ\_Q(i,:)=[cita,fai,rou];  end  LX=zeros(m,3);%理想抛物面坐标  LX\_Q=zeros(m,3);%理想抛物面坐标  %在抛物面z=(x.^2+y.^2)/(4\*ff)-300.8446;找到对应节点  for i=1:m%筛选出300内的点后，把2226改成点的个数      cita=JZ\_Q(i,1);fai=JZ\_Q(i,2);      rou=Lrou(fai);%旋转抛物面rou值的确定      x=rou\*sin(fai)\*cos(cita);      y=rou\*sin(fai)\*sin(cita);      z=rou\*cos(fai);      %z1=(x^2+y^2)/561.724-300.8446%检验      LX(i,:)=[x,y,z];      LX\_Q(i,:)=[cita,fai,rou];  end  format short g  %伸缩相差量6、7向上为负  cha=zeros(m,1);  load Net  B=zeros(4300,3);  load bc  load MeshGrid  for t=1:100  B=zeros(4300,3);  for i=1:m%伸缩相差量6、7向上为负      cha(i)=JZ\_Q(i,3)-LX\_Q(i,3);      if cha(i)>0.6%伸          cha(i)=0.6;          LX\_Q(i,3)=JZ\_Q(i,3)-cha(i);%更新理想球坐标          LX(i,:)=qiu2zhi(LX\_Q(i,1),LX\_Q(i,2),LX\_Q(i,3));      end      if cha(i)<-0.6%缩          cha(i)=-0.6;          LX\_Q(i,3)=JZ\_Q(i,3)-cha(i);%更新理想球坐标          LX(i,:)=qiu2zhi(LX\_Q(i,1),LX\_Q(i,2),LX\_Q(i,3));      end  end  %现更新所有板的理想坐标  %LX\_totle=XZ;  %for j=1:m  %    LX\_totle(ID(j),:)=LX(j,:);  %end  %MeshGrid\_2=MeshGrid;  %ID\_max=max(ID);  %MeshGrid\_2(MeshGrid\_2(:,1)>ID\_max,:)=[];  bc\_i=solvebc(LX);  num=zeros(2226,1);%该点是不符合约束，1不符合  num\_c=zeros(2226,1);%该点是符合约束，1符合  B=zeros(4300,3);  for i=1:4300      for j=1:3          temp1=abs(bc\_i(i,j)-bc(i,j));%距离差的绝对值          temp2=bc(i,j)\*0.0007;          if temp1<temp2              B(i,j)=0;%满足条件等于0          else              B(i,j)=1;%不满足条件等于1              %找出是哪两个点的边不符合条件              temp\_h=i;%MeshGrid中的第i行              if j==1                  temp\_y=[1,2];%MeshGrid中的第1,2列              end              if j==2                  temp\_y=[1,3];%MeshGrid中的第1,3列              end              if j==3                  temp\_y=[2,3];%MeshGrid中的第2,3列              end              temp\_num=MeshGrid(temp\_h,temp\_y(1));              num(temp\_num)=1;              temp\_num=MeshGrid(temp\_h,temp\_y(2));              num(temp\_num)=1;          end      end  end  SU\_b=sum(B(:));  SU\_d=sum(num);  num(num==0)=2;  num\_c(num==2)=1;%该点是符合约束，1符合  num(num==2)=0;%将num还原  trial=zeros(2226,3);%初始化不满足条件点的坐标   %迭代过程。更新LX矩阵。LX与JZ的中间点存入LX矩阵中   for i=1:2226%更新不满足的点       if num(i)~=0 %不满足条件更新LX矩阵。LX与JZ的中间点存入LX矩阵中           trial(i,:)=LX(i,:);%记录不满足条件点的坐标，后面画图           LX(i,:)=0.5\*(LX(i,:)+JZ(i,:));%直角坐标           LX\_Q(i,3)=0.5\*(LX\_Q(i,3)+JZ\_Q(i,3));       end   end   for i=1:2226           if num\_c(i)==1           %先更新满足约束点的坐标。对应JZ中的点           JZ(i,:)=LX(i,:);           JZ\_Q(i,3)=LX\_Q(i,3);           %JZ(i,:)=0.9\*LX(i,:)+0.1\*JZ(i,:);           %JZ\_Q(i,3)=0.9\*LX\_Q(i,3)+0.1\*JZ\_Q(i,3);           %JZ\_Q(i,:)=zhi2qiu(JZ(i,1),JZ(i,2),JZ(i,3)); %球坐标       end   end  end  PlotModel(XZ,MeshGrid,'b');  hold on  scatter3(trial(:,1),trial(:,2),trial(:,3),'y.');  title('基准态（蓝色）、超标点（黄色）') |

|  |
| --- |
| **6.问题三主程序1（Q3\_1.m）** |
| clear all;clc;close all;  load JZ  load XZ  load NodeInfo  %NodeLoc=JZ;%求工作状态的  %NodeLoc=XZ;%求刚旋转后的  %save MeshGrid MeshGrid  load MeshGrid  %备注将坐标系旋转后，300口径范围内的点代入  for ii=1:MeshNum     if (Loc\_ji(MeshGrid(ii,1),3)==0)||(Loc\_ji(MeshGrid(ii,2),3)==0)…  ||(Loc\_ji(MeshGrid(ii,3),3)==0)         MeshGrid(ii,:)=[0,0,0];     end  end   MeshGrid(all(MeshGrid==0,2),:)=[];%删除全零行   MeshNum=length(MeshGrid);  format long g  R=300;  F=0.466\*R;  Z=-(R-F);-300.4-Z+0.6;  P=[0,0,-(R-F)];  n=10000;%每块板找点数  triangle=zeros(3,3);%三角板顶点坐标，一行为一个坐标  sum1=0;sum2=0;%1有效面积，2总面积  for i=1:MeshNum%三角板个数4300块      %此次循环的triangle,%相连接的三个点      %测试用的一个三角板的坐标，每行表示一个点的坐标      triangle(1,:)=LocNew1(MeshGrid(i,1),:);%三角板第一个点坐标      triangle(2,:)=LocNew1(MeshGrid(i,2),:);      triangle(3,:)=LocNew1(MeshGrid(i,3),:);      %triangle=[0,0,-300.4;6.1078,8.407,-300.2202;9.8827,-3.211,-300.2202];      count\_m=0;      p1=triangle(1,:);      p2=triangle(2,:);      p3=triangle(3,:);      fdir=cross((p1-p2),(p3-p2));%三角板的法向量      if fdir(3)<0   %保证法向量指向圆心          fdir=-fdir;      end      mj=area(p1,p2,p3);  %[f,mj,mdloc]=meshrefcal(triangle(1,:),…  triangle(2,:),triangle(3,:));      for i=1:n          r1=rand(1);r2=rand(1);          E=zeros(3,1);%E点          F=zeros(3,1);%F点          F=[r1,1-r1]\*triangle(2:3,:);          E=[r2,1-r2]\*[triangle(1,:);F];          Out=E-2\*dot(E,fdir)\*fdir;%fdir法向量，Out反射光          xp=P(1);yp=P(2);zp=P(3);          xe=E(1);ye=E(2);ze=E(3);          xr=Out(1);yr=Out(2);zr=Out(3);          %t=((xp-xe)+(yp-ye)+(zp-ze))/xr+yr+zr;公式错误          t=(Z-ze)/zr;          M=E+t\*Out;%焦点平面焦点求解公式错误          xm=xe+t\*xr;ym=ye+t\*yr;zm=ze+t\*zr;          temp=(xm-xp)^2+(ym-yp)^2+(zm-zp)^2;          if temp<=0.5^2              count\_m=count\_m+1;          end      end      sum1=sum1+count\_m/n\*mj;      sum2=sum2+mj;  end  y=sum1/sum2 %接收比 |

|  |
| --- |
| **7.问题三主程序2（Q3\_2.m）** |
| %进行坐标系的还原,找焦点坐标  clear,clc;  load NodeInfo  load NodeInfo2   load LocNew1  MeshNum=4300;  number=0;  for ii=1:MeshNum     if (Loc\_ji(MeshGrid(ii,1),3)==0)||(Loc\_ji(MeshGrid(ii,2),3)==0)…  ||(Loc\_ji(MeshGrid(ii,3),3)==0)         MeshGrid(ii,:)=[0,0,0];     end  end   MeshGrid(all(MeshGrid==0,2),:)=[];%删除全零行   MeshNum=length(MeshGrid);  F=0.466\*300.4;  focus=[0,0,-300.4+F];  focus=hy(focus);  for ii=1:MeshNum      n=1e4;      nmd1=unifrnd(0,1,[1,n]);      nmd2=unifrnd(0,1,[1,n]);      p1=LocOld(MeshGrid(ii,1),:);      p2=LocOld(MeshGrid(ii,2),:);      p3=LocOld(MeshGrid(ii,3),:);     for jj=1:1e4      [XX,YY,ZZ]=Carol(p1,p2,p3,nmd1(jj),nmd2(jj));%随机点e的坐标,也是定点      V=[XX,YY,ZZ];      t=(norm(focus)^2-sum([XX,YY,ZZ].\*focus))/sum(V.\*focus);          x=XX+V(1)\*t;      y=YY+V(2)\*t;      z=ZZ+V(3)\*t;      D=norm([focus(1)-x,focus(2)-y,focus(3)-z]);%交点与焦点距离      if D<0.5          number=number+1;      end     end  end  rate=number/(n\*MeshNum) |

|  |
| --- |
| **8.程序运行所需m函数文件** |
| **Update.m** |
| function [detall,LocNew,ff,Loc\_ji]=Update(NodeLoc)  NodeNum=size(NodeLoc,1);%记录主索节点个数，size（n,1）返回行数  R=abs(min(NodeLoc(:,3)));%记录基准抛物面最低点的z值  F=0.466\*R;  min\_all=9e10;%预赋值所有伸缩量的平方和的开根，用于比小  LocNew=zeros(NodeNum,3);  rou=zeros(NodeNum,1);    %将节点三维直角坐标转化成相应的二维极坐标  Loc\_ji=zeros(NodeNum,3);  for ii=1:NodeNum    Loc\_ji(ii,1)=atan(norm(NodeLoc(ii,1:2))/abs(NodeLoc(ii,3)));%θ，表示节点与球心间的连线和Z坐标轴的夹角    Loc\_ji(ii,2)=norm(NodeLoc(ii,:));%ρ，% 表示节点与球心间的距离  end  %焦距遍历  for f=135:0.001:145  %找出300m口径内的点  z=150^2/(4\*f)-f-(R-f);%边界点的z坐标  xita=asin(150/sqrt(150^2+z^2));%推导公式  %通过θ与抛物线方程找到新的节点坐标（θ，ρ），再还原成三维直角坐标  detal=zeros(NodeNum,1);%伸缩量  for ii=1:NodeNum    if  Loc\_ji(ii,1)>xita        Loc\_ji(ii,3)=0;%判断是否在300m内，0为不在    else        Loc\_ji(ii,3)=1;    end      a=cos(pi/2-Loc\_ji(ii,1))^2/(4\*f);%推导公式的求解      b=sin(pi/2-Loc\_ji(ii,1));      c=-f-(R-F);      rou(ii)=(-b+sqrt(b^2-4\*a\*c))/(2\*a);%只保留正解      rou(1)=-c;%A0节点在正下方，x、y坐标为0，不能用二次方程公式求解      detal(ii)=Loc\_ji(ii,2)-rou(ii);  end  detal(Loc\_ji(:,3)==0)=0;%不考虑不在300m内的点  if norm(detal)<min\_all      min\_all=norm(detal);%判断条件      for ii=1:NodeNum         LocNew(ii,:)=(Loc\_ji(ii,2)-detal(ii))/Loc\_ji(ii,2).\*NodeLoc(ii,:);%更新坐标，伸缩方向各坐标成比例         detall=detal;         ff=f;      end  end  end  end |
| **xz.m** |
| %坐标系旋转  %先绕z轴顺旋转alpha=36.795°(右手螺旋)  %再绕y轴顺旋转（90-bata）度  function [XZ]=xz(NodeLoc)    alpha=36.795;       %单位为角度    bata=78.169;    [m,n]=size(NodeLoc);%返回行数、列数   XZ=zeros(m,n);    a=-alpha/180\*pi;   b=-(90-bata)/180\*pi;   Hz=[cos(a),-sin(a),0;sin(a),cos(a),0;0,0,1];%旋转矩阵   Hy=[cos(b),0,sin(b);0,1,0;-sin(b),0,cos(b)];  %Hx=[1,0,0;0,cos(b),-sin(b);0,sin(b),cos(b)];   for i=1:m      XZ(i,:)=Hz\*NodeLoc(i,:)';      XZ(i,:)=Hy\*XZ(i,:)';   end  end |
| **hy.m** |
| %坐标系还原  %再绕y轴逆旋转（90-bata）度  %先绕z轴逆旋转alpha=36.795°(右手螺旋)  function [HY]=hy(LocNew)    alpha=36.795;       %单位为角度    bata=78.169;     [m,n]=size(LocNew);%返回行数、列数    HY=zeros(m,n);    a=alpha/180\*pi;%还原  b=(90-bata)/180\*pi;  Hz=[cos(a),-sin(a),0;sin(a),cos(a),0;0,0,1];  Hy=[cos(b),0,sin(b);0,1,0;-sin(b),0,cos(b)];  for i=1:m      HY(i,:)=Hy\*LocNew(i,:)';      HY(i,:)=Hz\*HY(i,:)';  end  end |
| **PlotModel** |
| function PlotModel(coord,facet,c)  %顶点和切面的数量  nvert=size(coord,1);  ntria=size(facet,1);  %所有顶点的坐标  xpts = coord(:,1);  ypts = coord(:,2);  zpts = coord(:,3);  %所有切面的坐标  node1 = facet(:,1);  node2 = facet(:,2);  node3 = facet(:,3);  %储存vind数组中每个顶点的节点  for i  = 1:ntria      pts = [node1(i) node2(i) node3(i)];      vind(i,:) = pts;  end  x = xpts;  y = ypts;  z = zpts;  % define X,Y,Z arrays and plot them  %定义X,Y,Z数组并绘画他们  for i = 1:ntria      X = [x(vind(i,1)) x(vind(i,2)) x(vind(i,3)) x(vind(i,1))];      Y = [y(vind(i,1)) y(vind(i,2)) y(vind(i,3)) y(vind(i,1))];      Z = [z(vind(i,1)) z(vind(i,2)) z(vind(i,3)) z(vind(i,1))];      fill3(X,Y,Z,c)      if i == 1          hold on      end  end  axis square  xlabel('x');  ylabel('y');    zlabel('z');  %这是为了避免任意一维上的最大值和最小值都为0  xmax = max(xpts); xmin = min(xpts);  ymax = max(ypts); ymin = min(ypts);  zmax = max(zpts); zmin = min(zpts);  dmax = max([xmax ymax zmax]);  dmin = min([xmin ymin zmin]);  %这是为了避免任意一维上的最大值和最小值都为0  xmax = dmax;  ymax = dmax;  zmax = dmax;  xmin = dmin;  ymin = dmin;  zmin = dmin;  axis([xmin, xmax, ymin, ymax, zmin, zmax]); |
| **Neighbor.m** |
| %节点的相邻矩阵  %load MeshGrid  Nei\_t=zeros(NodeNum,18);  Nei=zeros(NodeNum,7);  Nei(:,1)=1:NodeNum;%自己的节点编号  for ii=1:NodeNum%排节点编号      m=1;      for jj=1:MeshNum%找节点编号          for t=1:3             if MeshGrid(jj,t)==ii                 Nei\_t(ii,m:m+2)=MeshGrid(jj,:);                 m=m+3;             end           end      end  end  for ii=1:NodeNum      k=unique(Nei\_t(ii,:));%删除重复项      k(k==0)=[];               %删除多余的0      k(k==ii)=[];                %删除自己的节点编号      a=zeros(1,6-length(k));%用于补全空缺，方便合并      Nei(ii,2:end)=[k,a];  end |