**实验报告**

**实验项目： 地震震源理论初步**

**专 业： 地球物理学**

**班 级： 200112**

**姓名（学号）：**

**指导教师： 万永革**

**地球科学学院编制**

**2022年 12月 29日**

**实验项目：地震震源理论初步**

**实验目的：掌握根据断层走向、倾角、滑动角得到北东下和上南东坐标系的矩张量各元素的程序使用，会使用dsrin.m、pt2ds.m、mom2other.m进行震源机制参数的相互转换，掌握P波和S波辐射花样的空间分布规律，会使用wulff投影和Schmidt投影绘制震源机制图；掌握单侧和双侧破裂的P波和S波辐射花样的空间分布规律；掌握采用P波初动数据求解震源机制的方法，并熟练使用从globalCMT数据中读取所需震源机制的程序。**

**实验原理及步骤（按照项目编号写出）：**

**1、已知一个断层的走向、倾角、滑动角分别为30度，60度、80度，采用程序得到北东下和上南东坐标系的矩张量各元素的值。**

**2、读懂dsrin.m程序，采用第一题中的断层数据，得到震源机制的另外一个节面参数及P,T,B轴走向和倾伏角的值。**

**3、读懂pt2ds.m程序,在网站www.globalCMT.org找到最近发生的一个地震，将其中的P,T轴的走向和倾伏角输入，看输出的震源机制的各个参数与网站上是否对应。**

**4、读懂mom2other.m程序，将第一题得到的北东下坐标系的各元素值代入该程序，得到地震矩张量所对应的2个节面的走向、倾角和滑动角，P,T,B轴的走向、倾角和滑动角。**

**5、读懂P10\_1.m程序，绘制P波辐射花样，旋转空间图像，分析总结其特征。**

**6、读懂P10\_2.m、P10\_3.m、P10\_4.m程序，绘制S波辐射花样，旋转空间图像，分析总结其特征。**

**7、读懂并运行P10\_5.m程序，总结滑动角在节面上的表示规律。**

**8、采用beachball\_wf.m和beachball\_ea.m分别绘制两个网站www.globalCMT.org，绘制其投影，并与网站的投影图进行对比分析。**

**9、读懂P10\_10.m程序，总结采用P波初动求解震源机制的步骤；**

**10、采用P10\_11.m程序分析10个随机产生的P波初动数据，得到震源机制，要求分数在85分以上，截图放入实验报告。**

**11、运行程序p10\_12和P10\_15.m程序，总结单侧和双侧破裂得到的P波和S波辐射花样的规律。**

**12、读懂程序P10\_16.m程序，以自己家乡为中心，读取一定范围内的GlobalCMT数据中的地震震源机制数据，并列表。**

**第一题**

**function [m]=dis2mom(strike,dip,rake);**

**% function dis2mom(strike,dip,rake)**

**% 计算断层面表示的矩张量**

**% 输入: strike 断层走向(正北顺时针旋转的角度)**

**% dip：倾角，断层面与水平方向夹角**

**% rake：滑动角（在断层面上走向计算的逆时针旋转至滑动方向的角度）**

**% 所有角度的单位均为度**

**% 输出: Mxx=m(1)， Myy=m(2)， Mzz=m(3)，Mxy=m(4)，Mxz=m(5)，Myz=m(6)，为北东下坐标系中表示的矩张量**

**%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**con=pi/180.;**

**s=strike\*con;**

**d=dip\*con;**

**r=rake\*con;**

**%**

**m(1)=-sin(s)\*sin(s)\*sin(r)\*sin(2\*d) - sin(2\*s)\*cos(r)\*sin(d);**

**m(2)=-cos(s)\*cos(s)\*sin(r)\*sin(2\*d) + sin(2\*s)\*cos(r)\*sin(d);**

**m(3)=sin(r)\*sin(2\*d);**

**m(4)=cos(2\*s)\*cos(r)\*sin(d) + 0.5\*sin(2\*s)\*sin(r)\*sin(2\*d);**

**m(5)=-cos(s)\*cos(r)\*cos(d) - sin(s)\*sin(r)\*cos(2\*d);**

**m(6)=-sin(s)\*cos(r)\*cos(d) + cos(s)\*sin(r)\*cos(2\*d);**

**%**

**return;**

**-0.3435 -0.5094 0.8529 0.4445 0.1710 -0.4698**

**第二题**

**function [Ptrpl,Ttrpl,Btrpl,str2,dip2,rake2]=dsrin(str,dip,rake)**

**%**

**% 采用地震震源机制的一个节面的走向str、倾角dip和滑动角rake求其他参数的值**

**% 输入str为走向、dip为倾角、rake为滑动角.**

**% 输出Ptrpl为两个元素，第一个元素为P轴走向、第二个元素为P轴倾伏角**

**% 输出Ttrpl为两个元素，第一个元素为T轴走向、第二个元素为T轴倾伏角**

**% 输出Btrpl为两个元素，第一个元素为B轴走向、第二个元素为B轴倾伏角**

**% 输出str2,dip2,rake2为第二个节面的走向、倾角和滑动角**

**% 该程序中所有的角度值得单位均为度**

**SR2=sqrt(2);**

**str=deg2rad(str);dip=deg2rad(dip);rake=deg2rad(rake); %角度转化成弧度值**

**%走向、倾角和滑动角转换为滑动矢量A和法向N**

**A=[cos(rake)\*cos(str)+sin(rake)\*cos(dip)\*sin(str), cos(rake)\*sin(str)-sin(rake)\*cos(dip)\*cos(str), -sin(rake)\*sin(dip)]; %A（滑动矢量）的单位方向矢量,根据（10-2-8）式计算**

**N=[-sin(str)\*sin(dip), cos(str)\*sin(dip), -cos(dip)]; %N（节面法线）的单位方向矢量，根据（10-2-9）式计算**

**T=SR2\*(A+N); %向量相加，并且单位化，得主张应力轴单位矢量，(10-3-10)第一式**

**P=SR2\*(A-N); %向量相减，并且单位化，得主压应力轴单位矢量，（10-3-10）第二式**

**B=cross(P,T); %向量叉乘，得到中间轴单位矢量，（10-3-10）第三式**

**[Ptrpl]=v2trpl(P); %求出P轴走向和滑动角**

**[Ttrpl]=v2trpl(T); %求出T轴走向和滑动角**

**[Btrpl]=v2trpl(B); %求出B轴走向和滑动角**

**[str2,dip2,rake2]=an2dsr\_wan(N,A); %根据滑动矢量和法线单位矢量（两个矢量互换）求出另一个节面的参数**

**Return**

**127.2846 14.4489**

**第三题**

**function [str1,dip1,rake1,str2,dip2,rake2,Btrpl,PTangle]=pt2ds(Ptrpl,Ttrpl)**

**% 采用震源机制的PT轴的走向和倾伏角求解震源机制的其他表示形式**

**%输入：Ptrpl为P轴的走向和倾伏角，Ttrpl为T轴的走向和倾伏角，单位均为度**

**%输出：str1,dip1,rake1为震源机制第一个节面的走向、倾角和滑动角；**

**%str2,dip2,rake2为震源机制第二个节面的走向、倾角和滑动角；**

**%Btrpl为B轴的走向和倾伏角；**

**%PTangle为P,T轴之间的夹角，应该接近90度,如果不接近90度,程序会给出警告**

**%所有输出变量的单位均为度**

**SR2=0.707107; %2开平方**

**Ptrpl=deg2rad(Ptrpl);Ttrpl=deg2rad(Ttrpl); %将角度转化成弧度值**

**P=[cos(Ptrpl(1))\*cos(Ptrpl(2)),sin(Ptrpl(1))\*cos(Ptrpl(2)),sin(Ptrpl(2))]; %P轴单位方向矢量，根据（10-3-1）式给出**

**T=[cos(Ttrpl(1))\*cos(Ttrpl(2)),sin(Ttrpl(1))\*cos(Ttrpl(2)),sin(Ttrpl(2))]; %T轴单位方向矢量，根据（10-3-1）式给出**

**PTangle=rad2deg(acos(dot(P,T))); %p与T进行点乘，因为模数为1，得到P轴与T轴夹角余弦值，再进行反余弦**

**if(abs(PTangle-90)>10)**

**warning('两个节面不垂直') %判断两个节面是否垂直**

**end**

**B=cross(T,P); %T轴与P轴进行叉乘，得到B轴单位方向矢量**

**A=SR2\*(T+P); %根据（10-3-11）第一式得到滑动方向**

**N=SR2\*(T-P); %根据（10-3-11）第一式得到断层面法向**

**Btrpl=v2trpl(B); %求出B轴走向和倾伏角**

**[str1,dip1,rake1]=an2dsr\_wan(A,N); %得到第一个节面的走向、倾角和滑动角**

**[str2,dip2,rake2]=an2dsr\_wan(N,A); %得到第一个节面断层的走向、倾角和滑动角**

**Return**

**是一样的**

**第四题**

**function [str1,dip1,rake1,str2,dip2,rake2,Ptrpl,Ttrpl,Btrpl]=mom2other(m)**

**%将地震矩张量表示为剪切位错的其他表示方式**

**%输入：m为矩张量的各个元素，分别对应于Mxx=m(1)， Myy=m(2)，**

**%Mzz=m(3)，Mxy=m(4)，Mxz=m(5)，Myz=m(6)，此处定义的坐标系为北东下坐标系**

**%如果采用矩心矩张量定义的坐标系必须进行转换**

**%使得m(1)=Mtt,m(2)=Mpp,m(3)=Mrr;m(4)=-Mtp,m(5)=Mrt;m(6)=-Mrp**

**c(1,1)=m(1);c(1,2)=m(4);c(1,3)=m(5);c(2,1)=m(4);c(2,2)=m(2);**

**c(2,3)=m(6);c(3,1)=m(5);c(3,2)=m(6);c(3,3)=m(3);**

**[W,D]=eig(c);%求本征值和本征向量**

**[triD,id]=sort(diag(D));%对本征值由小到大进行排列**

**W\_opt=W(:,id'); %对本征向量对于于本征值由小到大进行排列**

**[Ptrpl]=v2trpl([W\_opt(:,1)]'); %将最优的压轴单位向量转换为走向和倾伏角**

**[Ttrpl]=v2trpl([W\_opt(:,3)]'); %将最优的张轴单位向量转换为走向和倾伏角**

**[str1,dip1,rake1,str2,dip2,rake2,Btrpl,PTangle]=pt2ds(Ptrpl,Ttrpl); %采用pt2ds程序求得其他参数**

**Return**

**30.0024**

**第五题**

**%P10\_1.m**

**% 0 <= theta <= pi为列矢量**

**n=50; %用的点数**

**fai = (0:2:2\*n)/n\*pi; %走向从0~2pi**

**theta =(0:2:2\*n)'/n\*pi/2; %倾角从0~pi**

**costheta = cos(theta); costheta(1) = 1; costheta(n+1) = -1;**

**sintheta=sin(theta); %sin(theta)**

**sintheta(1)=0;sintheta(n+1)=0;**

**sin2theta=sin(2\*theta); %sin(2\*theta)**

**sin2theta(1)=0;sin2theta(n+1)=0;**

**sinfai = sin(fai); sinfai(1) = 0; sinfai(n+1) = 0;**

**cosfai = cos(fai); cosfai(1) = 1; cosfai(n+1) = 1;**

**r=sin2theta\*cosfai; %按（10-4-6）给出P波的径向位移的辐射花样**

**x =r.\*(sintheta\*cosfai); %转换为直角坐标系中的x**

**y = r.\*(sintheta\*sinfai); %转换为直角坐标系中的y**

**z = r.\*(costheta\*ones(1,n+1)); %转换为直角坐标系中的z**

**H=surf(x,y,z,r); %在直角坐标系下绘制P波径向位移的辐射花样**

**set(H,'FaceLighting','phong','FaceColor','interp','AmbientStrength',0.5)**

**light('Position',[1,0,0],'Style','infinite');**

**hold on**

**plot3([-2,2],[0,0],[0,0]); %画x轴**

**text(2,0,0,'x','FontSize',20) %给出x轴的标记**

**plot3([0,0],[-2,2],[0,0]) %绘制y轴**

**text(0,2,0,'y','FontSize',20) %给出y轴的标记**

**plot3([0,0],[0,0],[-1,1]) %绘制z轴**

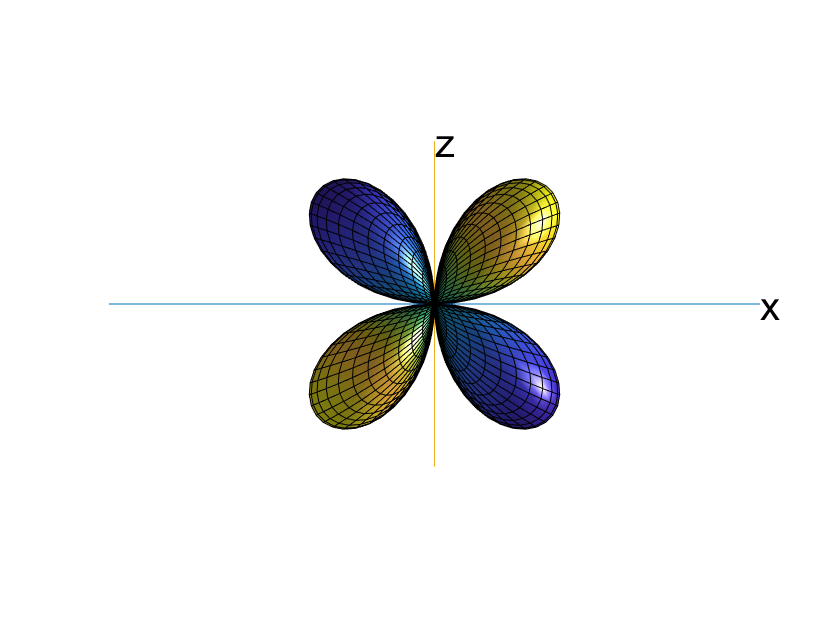
**text(0,0,1,'z','FontSize',20) %给出z轴的标记**

**axis off**

**axis equal**

**view(0,0)**

**hold off**

****

**第六题**

**%P10\_2.m**

**% Fai向的波的辐射花样**

**n=30; %采用的数据点数**

**fai = (0:2:2\*n)/n\*pi; %Fai的取值范围**

**theta =(0:2:2\*n)'/n\*pi/2; %theta的取值范围**

**costheta = cos(theta); costheta(1) = 1; costheta(n+1) = -1;**

**sintheta=sin(theta);sintheta(1)=0;sintheta(n+1)=0;**

**sinfai = sin(fai); sinfai(1) = 0; sinfai(n+1) = 0;**

**cosfai = cos(fai); cosfai(1) = 1; cosfai(n+1) = 1;**

**r=-costheta\*sinfai; %根据（10-4-8）的 分量振幅**

**x =r.\*(sintheta\*cosfai); %转换直角坐标系的x坐标**

**y = r.\*(sintheta\*sinfai); %转换直角坐标系的y坐标**

**z = r.\*(costheta\*ones(1,n+1)); %转换直角坐标系的z坐标**

**H=surf(x,y,z,r); %绘图**

**set(H,'FaceLighting','phong','FaceColor','interp','AmbientStrength',0.5) %设置图的显示方式**

**light('Position',[1,0,0],'Style','infinite'); %设置灯光**

**hold on%使得后面的绘图基于原来绘图基础上**

**plot3([-2,2],[0,0],[0,0]); %绘制x轴**

**text(2,0,0,'x','FontSize',20) %给出x轴的标记**

**plot3([0,0],[-2,2],[0,0]) %设置y轴**

**text(0,2,0,'y','FontSize',20) %给出y轴的标记**

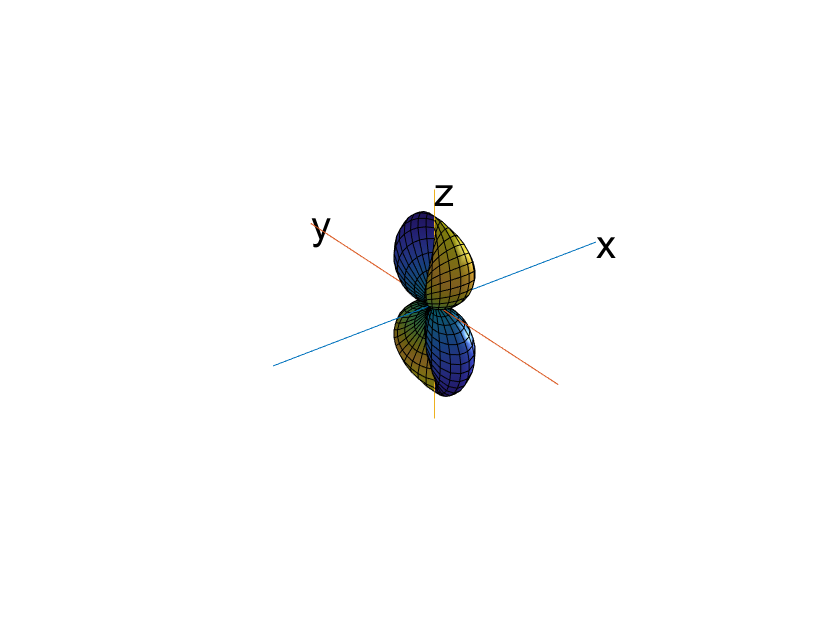
**plot3([0,0],[0,0],[-1.3,1.3]) %设置z轴**

**text(0,0,1.3,'z','FontSize',20) %给出z轴的标记**

**axis off%去掉坐标轴**

**axis equal%使坐标轴的尺度相同**

**hold off**

****

**%P10\_3.m**

**% Theta方向S波的位移**

**n=30; %所用的网格数**

**fai = (0:2:2\*n)/n\*pi; %Fai的范围从0到2\*pi**

**theta =(0:2:2\*n)'/n\*pi/2; %Theta范围从0到pi**

**costheta = cos(theta); costheta(1) = 1; costheta(n+1) = -1; %cos(theta)**

**sintheta=sin(theta);sintheta(1)=0;sintheta(n+1)=0; %sin(theta)**

**cos2theta=cos(2\*theta);cos2theta(1)=1;cos2theta(n+1)=1; %cos(2\*theta)**

**sinfai = sin(fai); sinfai(1) = 0; sinfai(n+1) = 0; %sin(fai)**

**cosfai = cos(fai); cosfai(1) = 1; cosfai(n+1) = 1; %cos(fai)**

**r=abs(cos2theta\*cosfai); %根据（10-4-8）的 分量表达，这里加上了绝对值，避免缠绕**

**x =r.\*(sintheta\*cosfai); %转换为直角坐标系的x坐标**

**y = r.\*(sintheta\*sinfai); %转换为直角坐标系的y坐标**

**z = r.\*(costheta\*ones(1,n+1)); %转换为直角坐标系的z坐标**

**H=surf(x,y,z,r); %绘制表面图**

**set(H,'FaceLighting','phong','FaceColor','interp','AmbientStrength',0.5) %设置绘图属性**

**light('Position',[1,0,0],'Style','infinite'); %设置灯光属性**

**hold on**

**plot3([-2,2],[0,0],[0,0]); %绘制x轴**

**text(2,0,0,'x','FontSize',20) %给出x轴标记**

**plot3([0,0],[-2,2],[0,0]) %绘制y轴**

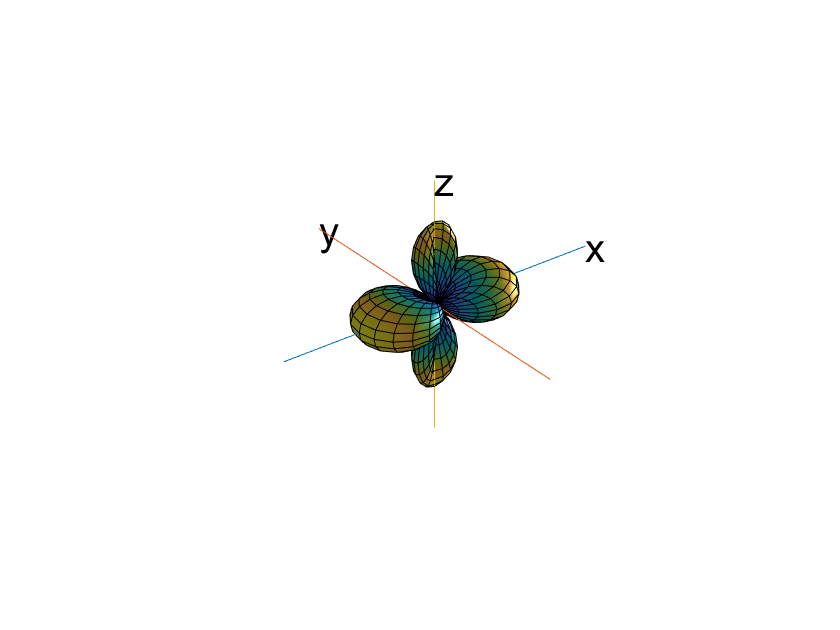
**text(0,2,0,'y','FontSize',20) %给出y轴标记**

**plot3([0,0],[0,0],[-1.5,1.5]) %绘制z轴**

**text(0,0,1.5,'z','FontSize',20) %给出z轴标记**

**axis off%不显示坐标轴**

**axis equal%使得坐标轴尺度相等**

****

**%P10\_4.m**

**% S波辐射花样，包括SH和SV**

**n=30;**

**fai = (0:2:2\*n)/n\*pi; %Fai的范围:0~2\*pi**

**theta =(0:2:2\*n)'/n\*pi/2; %Theta的范围0~pi**

**costheta = cos(theta); costheta(1) = 1; costheta(n+1) = -1; %cos(theta)**

**sintheta=sin(theta);sintheta(1)=0;sintheta(n+1)=0; %sin(theta)**

**cos2theta=cos(2\*theta);cos2theta(1)=1;cos2theta(n+1)=1; %cos(2\*theta)**

**sinfai = sin(fai); sinfai(1) = 0; sinfai(n+1) = 0; %sin(fai)**

**cosfai = cos(fai); cosfai(1) = 1; cosfai(n+1) = 1; %cos(fai)**

**r=sqrt((cos2theta\*cosfai).^2+(costheta\*sinfai).^2); %S波合成的位移**

**x =r.\*(sintheta\*cosfai); %转换为直角坐标系的x坐标**

**y = r.\*(sintheta\*sinfai); %转换为直角坐标系的y坐标**

**z = r.\*(costheta\*ones(1,n+1)); %转换为直角坐标系的z坐标**

**H=surf(x,y,z,r); %绘制位移在不同方向的大小**

**set(H,'FaceLighting','phong','FaceColor','interp','AmbientStrength',0.5) %设置绘图属性**

**light('Position',[1,0,0],'Style','infinite'); %设置光源的特性**

**hold on**

**plot3([-1.5,1.5],[0,0],[0,0]); %绘制x轴**

**text(1.5,0,0,'x','FontSize',20) %给出x轴的标示**

**plot3([0,0],[-2,2],[0,0]) %绘制y轴**

**text(0,2,0,'y','FontSize',20) %绘出y轴的标示**

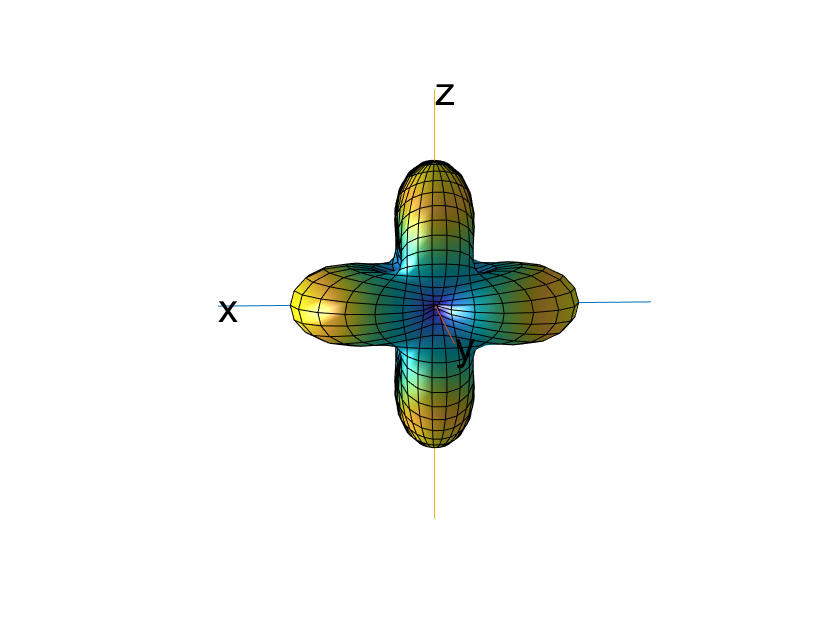
**plot3([0,0],[0,0],[-1.5,1.5]) %绘制z轴**

**text(0,0,1.5,'z','FontSize',20) %给出z轴的标示**

**axis off%不显示坐标轴**

**axis equal%使得坐标轴的尺度相等**

**view(176,8) %给出图的视角**

****

**第七题**

**%P10\_5.m**

**close all**

**str=50;dip=30; rake=60; %断层面的走向、倾角和滑动角**

**a=2;%绘制的Wulff网的大圆半径**

**n=500;%用500个点绘制断层面在Wulff网上的投影**

**x0=0;y0=0; %坐标原点**

**str=deg2rad(str);dip=deg2rad(dip); rake=deg2rad(rake); %改为弧度表示**

**xp=x0+a\*(tan(pi/4-dip/2)\*sin(str+pi/2)+tan(pi/4+dip/2)\*sin(str-pi/2))/2; %公式（10-5-5,6,7）**

**yp=y0+a\*(tan(pi/4-dip/2)\*cos(str+pi/2)+tan(pi/4+dip/2)\*cos(str-pi/2))/2; %公式（10-4-5,6,7）**

**rp = a/cos(dip); %公式（10-5-8）**

**PO = sqrt((xp - x0) \* (xp - x0) + (yp - y0) \* (yp - y0)); %公式（10-5-9）**

**if (PO < 0.001) TP = pi; else TP = 2\*atan(a/ PO); end%断层所用弧线的角度，根据公式（10-5-10）得到**

**T1= str+pi/2-TP/2; %起始方位角（10-5-11）**

**T=T1:TP/n:T1+TP; %绘制断层圆弧所用的方位角（10-5-11）**

**xs=xp+ rp \* sin(T); %圆弧x坐标**

**ys=yp+ rp \* cos(T); %圆弧y坐标**

**x=-a:2\*a/n:a;y=sqrt(a\*a-x.^2); %大圆的一半**

**plot([x,x(n+1:-1:1)],[y,-y(n+1:-1:1)],'b') %绘制投影大圆**

**hold on**

**plot(xs,ys,'r','LineWidth',3); %绘制断层面投影**

**if(rake>0) rake=pi-rake; end%判断是正断层性质还是逆断层性质**

**cosih=-sin(dip)\*sin(rake); %滑动方向与z轴的方向余弦,按照前面的公式给出**

**cosdet=sqrt(1-cosih.^2); %得到滑动方向与水平面夹角的方向余弦**

**det=acos(abs(cosdet)); %求得滑动方向与水平面的夹角**

**fai=acos(cos(rake)./cosdet); %根据球面直角三角形的公式得到在Wulff网圆内的与走向直径的夹角（10-5-14）**

**str1=str+fai; %在走向角的基础上增加fai**

**x=a\*tan(pi/4-det/2).\*sin(str1); %公式（10-5-5）**

**y=a\*tan(pi/4-det/2).\*cos(str1); %公式（10-5-5）**

**scale=1;**

**arw\_scale=0.03;**

**dx=-a/5\*sign(rake)\*sin(str1); %采用半径的1/5给出箭头长短**

**dy=-a/5\*sign(rake)\*cos(str1);**

**[arw, vec,xend,yend]=vecplot(x, y, dx, dy, scale,arw\_scale); %给出箭头的参数**

**plot(vec(1:2,:),vec(3:4,:),'b','linewidth',1); %绘制箭头矢量长度**

**fill(arw(1:3,:),arw(4:6,:),'b','linewidth',1,'edgecolor','b') %绘制箭头**

**line([0,0],[a,a\*1.1]); %绘制北向**

**text(0,a\*(1.1),'N') %标示北向**

**line([-a,-a\*1.1],[0,0]); %绘制西向**

**text(-a\*(1.2),0,'W') %标示北向**

**line([a,a\*1.1],[0,0]); %绘制东向**

**text(a\*1.1,0,'E') %标示东向**

**line([0,0],[-a,-a\*1.1]); %绘制南向**

**text(0,-a\*1.1,'S') %标示南向**

**%绘制断层的走向**

**line(a\*[cos(pi/2-(str-pi)),cos(pi/2-str)],a\*[sin(pi/2-(str-pi)),sin(pi/2-str)]);**

**%下面绘制不同的倾角对应的断层弧线的位置**

**line([0, a\*cos(pi/2-(str+pi/2))],[0, a\*sin(pi/2-(str+pi/2))]);**

**%rak=0:10:180;**

**rake=deg2rad(0:-10:-180);**

**cosih=-sin(dip)\*sin(rake); %滑动方向与z轴的方向余弦**

**cosdet=sqrt(1-cosih.^2); %得到滑动方向与水平面夹角的方向余弦**

**det=acos(cosdet); %求得滑动方向与水平面的夹角**

**fai=acos(cos(rake)./cosdet); %根据球面直角三角形的公式得到在Wulff网圆内的与走向直径的夹角**

**str1=str+fai; %在走向角的基础上增加fai**

**x=a\*tan(pi/4-det/2).\*sin(str1); %公式（10-5-5）**

**y=a\*tan(pi/4-det/2).\*cos(str1); %公式（10-5-5）**

**plot(x,y,'o') %在滑动角的位置画一个圆圈**

**for ii=1:length(x)**

**text(x(ii),y(ii),num2str(rad2deg(rake(ii))),'Verticalalignment','top','Horizontalalignment','center','rotation',rad2deg(pi/2-str));**

**text(x(ii),y(ii),num2str(rad2deg(-rake(length(x)-ii+1))),'Verticalalignment','bottom','Horizontalalignment','center','rotation',rad2deg(pi/2-str));**

**end**

**dip=0:pi/18:pi/2;**

**x=a\*tan(pi/4-dip/2)\*sin(str+pi/2); %公式（10-5-5）**

**y=a\*tan(pi/4-dip/2)\*cos(str+pi/2); %公式（10-5-5）**

**plot(x,y,'o') %在倾角点的位置画一个圆圈**

**dx=a\*0.02\*sin(pi/2-str);**

**dy=a\*0.02\*cos(pi/2-str);**

**for ii=1:length(dip)**

**text(x(ii)+dx,y(ii)+dy,num2str(rad2deg(dip(ii))),'rotation',rad2deg(pi/2-str)) %在断层倾角的位置给出倾角的标志**

**end**

**text(x(6)-10\*dx,y(6)-10\*dy,'\delta','FontSize',20,'rotation',rad2deg(pi/2-str))**

**axis square%坐标轴尺度一致，使得绘制的圆形看起来为正圆**

**axis off%去掉坐标轴**

****

**由此可见，断层在wulff网上的表现为面对断层弧的凸向，断层的右侧方向为断层走向。断层倾角表现为断层倾角的弯曲程度，断层弧越靠近wulff网投影圆，断层越缓（倾角越小），在投影圆上的断层倾角为零；反之，越靠近断层走向的直径，断层越陡（倾角越大），与断层走向的直径重合表示垂直断层。**

**第八题**

**function beachball\_wf(str,dip,rak,siz,n,clr,x0,y0,iptb,insew,fillcode)**

**%..........................................................**

**% beachball\_wf.m - 采用极射赤面（Wulff）投影绘制震源机制解的程序**

**% 输入: str - 走向, dip - 倾角, rak - 滑动角,单位均为度**

**% 这里的滑动角大于180，必须转换为角度为负**

**% siz - 绘制投影的大小**

**% n - 绘制投影的分辨率，越大分辨率越高，通常去做50**

**% clr - 膨胀部分所用的颜色**

**% y0 - 事件的纬度，y坐标**

**% x0 - 事件的经度，x坐标**

**% fillcode-是否填充拉张区，填充为1，不填充为零**

**% iptb-是否绘制PTB轴，如果未非零，则绘制PTB轴**

**% insew-是否绘制外边的NSEW方向标志，如果未非零，则绘制这种标志**

**%..........................................................**

**if n<5,n=5;elseif n>50,n=50;end,d2r=pi/180;pi2=pi\*2;**

**str=str\*d2r;dip=dip\*d2r;rak=rak\*d2r; %将度表示为弧度**

**if(iptb)**

**[Ptrpl,Ttrpl,Btrpl,str2,dip2,rake2]=dsrin(rad2deg(str),rad2deg(dip),rad2deg(rak))**

**else**

**A=[cos(rak)\*cos(str)+sin(rak)\*cos(dip)\*sin(str), cos(rak)\*sin(str)-sin(rak)\*cos(dip)\*cos(str), -sin(rak)\*sin(dip)]; %A（滑动矢量）的单位方向矢量，根据（10-2-8）式计算**

**N=[-sin(str)\*sin(dip), cos(str)\*sin(dip), -cos(dip)]; %N（节面法线）的单位方向矢量，根据（10-2-9）式计算**

**[str2,dip2,rake2]=an2dsr\_wan(N,A); %得到另外一个节面的走向、倾角和滑动角**

**end**

**%**

**xp=x0+siz\*(tan(pi/4-dip/2)\*sin(str+pi/2)+tan(pi/4+dip/2)\*sin(str-pi/2))/2; %公式（10-5-5,6,7）**

**yp=y0+siz\*(tan(pi/4-dip/2)\*cos(str+pi/2)+tan(pi/4+dip/2)\*cos(str-pi/2))/2; %公式（10-4-5,6,7）**

**rp = siz/cos(dip); %公式（10-5-8）**

**PO = sqrt((xp - x0) \* (xp - x0) + (yp - y0) \* (yp - y0)); %公式（10-5-9）**

**if (PO < 0.001) TP = pi; else TP = 2\*atan(siz/ PO); end%断层所用弧线的角度，根据公式（10-5-10）得到**

**T1= str+pi/2-TP/2; %起始方位角（10-5-11）**

**T=T1:TP/n:T1+TP; %绘制断层圆弧所用的方位角（10-5-11）**

**x1=xp+ rp \* sin(T); %圆弧x坐标**

**y1=yp+ rp \* cos(T); %圆弧y坐标**

**str2=deg2rad(str2);dip2=deg2rad(dip2);rake2=deg2rad(rake2);**

**xp=x0+siz\*(tan(pi/4-dip2/2)\*sin(str2+pi/2)+tan(pi/4+dip2/2)\*sin(str2-pi/2))/2; %公式（10-5-5,6,7）**

**yp=y0+siz\*(tan(pi/4-dip2/2)\*cos(str2+pi/2)+tan(pi/4+dip2/2)\*cos(str2-pi/2))/2; %公式（10-4-5,6,7）**

**rp = siz/cos(dip2); %公式（10-5-8）**

**PO = sqrt((xp - x0) \* (xp - x0) + (yp - y0) \* (yp - y0)); %公式（10-5-9）**

**if (PO < 0.001) TP = pi; else TP = 2\*atan(siz/ PO); end%断层所用弧线的角度，根据公式（10-5-10）得到**

**T1= str2+pi/2-TP/2; %起始方位角（10-5-11）**

**T=T1:TP/n:T1+TP; %绘制断层圆弧所用的方位角（10-5-11）**

**x2=xp+ rp \* sin(T); %圆弧x坐标**

**y2=yp+ rp \* cos(T); %圆弧y坐标**

**str1=str+pi;if str1 > pi2,str1=str1-pi2;end %断层面走向的反方向**

**d=str2; %第二个节面的走向**

**d1=d+pi; %第二个节面走向的反方向**

**if str1-d>pi,d=d-pi2;elseif str1-d>=pi,str1=str1-pi2;end**

**if d1-str>pi,d1=d1-pi2;elseif d1-str>=pi,str=str-pi2;end**

**st1=str1:(d-str1)/n:d; %两个节面所加的小弧，自第一个节面走向的反方向到第二个节面的走向**

**st2=d1:(str-d1)/n:str; %两个节面所加的小弧，自第二个节面走向的反方向到第一个节面的走向**

**p=[x1,sin(st1)\*siz,x2,sin(st2)\*siz]+x0;**

**q=[y1,cos(st1)\*siz,y2,cos(st2)\*siz]+y0;**

**% 绘制Wulff网投影**

**n=n+n;x=-siz:2\*siz/n:siz;y=sqrt(siz\*siz-x.^2); %绘制大圆所用的坐标序列**

**if(fillcode)**

**if(rak>0) %如果第一个节面的滑动角为正，则有向上逆冲分量，对应的小弧所加为膨胀区**

**fill([x,x(n+1:-1:1)]+x0,[y,-y(n+1:-1:1)]+y0,'w',p,q,clr)**

**else%如果第一个节面的滑动角为负，则有向下的正断层分量，对应的小弧所加为压缩区，则相反的区域为膨胀区，对相反的区域进行填充**

**fill([x,x(n+1:-1:1)]+x0,[y,-y(n+1:-1:1)]+y0,clr,p,q,'w')**

**end**

**else**

**plot(siz\*cos(0:2\*pi/n:2\*pi),siz\*sin(0:2\*pi/n:2\*pi),'k'); %绘制大圆**

**hold on**

**plot(x1,y1,clr,x2,y2,clr); %绘制两个节面**

**end**

**if(insew)**

**hold on**

**line([0,0],[siz,siz\*1.1]); %绘制北向**

**text(0,siz\*(1.1),'N') %标示北向**

**line([-siz,-siz\*1.1],[0,0]); %绘制西向**

**text(-siz\*(1.1),0,'W') %标示北向**

**line([siz,siz\*1.1],[0,0]); %绘制东向**

**text(siz\*1.1,0,'E') %标示东向**

**line([0,0],[-siz,-siz\*1.1]); %绘制南向**

**text(0,-siz\*1.1,'S') %标示南向**

**end**

**if(iptb)**

**hold on**

**%绘制P轴**

**rp = siz\*tand((90-Ptrpl(2))/2); %根据（10-5-3）计算Wulff投影的长短**

**xp=x0+rp\*sind(Ptrpl(1)); yp=y0+rp\*cosd(Ptrpl(1));**

**plot(xp,yp,'.')**

**text(xp+0.1,yp,'P');**

**%绘制T轴**

**rp = siz\*tand((90-Ttrpl(2))/2); %根据（10-5-3）计算Wulff投影的长短**

**xp=x0+rp\*sind(Ttrpl(1)); yp=y0+rp\*cosd(Ttrpl(1));**

**plot(xp,yp,'o')**

**text(xp+0.1,yp,'T');**

**%绘制B轴**

**rp = siz\*tand((90-Btrpl(2))/2); %根据（10-5-3）计算Wulff投影的长短**

**xp=x0+rp\*sind(Btrpl(1)); yp=y0+rp\*cosd(Btrpl(1));**

**plot(xp,yp,'+')**

**text(xp+0.1,yp,'B');**

**end**

**return**

**function beachball\_ea(str,dip,rake,siz,n,clr,lat,lon,iptb,insew,fillcode)**

**%..........................................................**

**% beachball\_ea.m - 采用等积（Schmidt）投影绘制震源机制解的程序**

**%**

**% 输入: str - 走向, dip - 倾角, rake - 滑动角,单位均为度,**

**% 这里的滑动角大于180，必须转换为角度为负**

**% siz - 绘制投影的大小**

**% n - 绘制投影的分辨率,范围为5-50.**

**% clr - 膨胀部分所用的颜色**

**% lat - 事件的纬度，y坐标**

**% lon - 事件的经度，x坐标**

**% iptb-是否绘制PTB轴，如果未非零，则绘制PTB轴**

**% insew-是否绘制外边的NSEW方向标志，如果未非零，则绘制这种标志**

**% fillcode-是否填充拉张区，填充为1，不填充为零**

**%..........................................................**

**if n<5,n=5;elseif n>50,n=50;end,d2r=pi/180;pi2=pi\*2;**

**str=str\*d2r;dip=dip\*d2r;rake=rake\*d2r;**

**if(iptb)**

**[Ptrpl,Ttrpl,Btrpl,str2,dip2,rake2]=dsrin(rad2deg(str),rad2deg(dip),rad2deg(rake));**

**else**

**A=[cos(rake)\*cos(str)+sin(rake)\*cos(dip)\*sin(str), cos(rake)\*sin(str)-sin(rake)\*cos(dip)\*cos(str), -sin(rake)\*sin(dip)]; %A（滑动矢量）的单位方向矢量**

**N=[-sin(str)\*sin(dip), cos(str)\*sin(dip), -cos(dip)]; %N（节面法线）的单位方向矢量**

**[str2,dip2,rake2]=an2dsr\_wan(N,A); %得到另外一个节面的走向、倾角和滑动角**

**end**

**%第一个断层面**

**rak=0:-pi/n:-pi;**

**cosih=-sin(dip)\*sin(rak); %滑动方向与z轴的方向余弦,按照前面的公式给出**

**ih=acos(cosih); %只取ih为锐角的情况**

**cosdet=sqrt(1-cosih.^2); %得到滑动方向与水平面夹角的方向余弦**

**fai=acos(cos(rak)./cosdet); %根据球面直角三角形的公式得到在水平大圆内的与走向直径的夹角**

**str1=str+fai; %在走向角的基础上增加fai**

**xs1=siz\*sqrt(2)\*sin(ih/2).\*sin(str1);**

**ys1=siz\*sqrt(2)\*sin(ih/2).\*cos(str1); %Schmidt投影的坐标**

**%第二个断层面**

**str2=deg2rad(str2);dip2=deg2rad(dip2);**

**cosih=-sin(dip2)\*sin(rak); %滑动方向与z轴的方向余弦,按照前面的公式给出**

**ih=acos(cosih); %只取ih为锐角的情况**

**cosdet=sqrt(1-cosih.^2); %得到滑动方向与水平面夹角的方向余弦**

**fai=acos(cos(rak)./cosdet); %根据球面直角三角形的公式得到在水平大圆内的与走向直径的夹角**

**str21=str2+fai; %在走向角的基础上增加fai**

**xs2=siz\*sqrt(2)\*sin(ih/2).\*sin(str21);**

**ys2=siz\*sqrt(2)\*sin(ih/2).\*cos(str21); %Schmidt投影的坐标**

**str1=str+pi;if str1 > pi2,str1=str1-pi2;end%断层面走向的反方向**

**d=str2; %第二个节面的走向**

**d1=d+pi; %第二个节面走向的反方向**

**if str1-d>pi,d=d-pi2;elseif str1-d>=pi,str1=str1-pi2;end**

**if d1-str>pi,d1=d1-pi2;elseif d1-str>=pi,str=str-pi2;end**

**st1=str1:(d-str1)/n:d; %两个节面所加的小弧，自第一个节面走向的反方向到第二个节面的走向**

**st2=d1:(str-d1)/n:str; %两个节面所加的小弧，自第二个节面走向的反方向到第一个节面的走向**

**p=[xs1,sin(st1)\*siz,xs2,sin(st2)\*siz]+lon;**

**q=[ys1,cos(st1)\*siz,ys2,cos(st2)\*siz]+lat;**

**% 绘制等面积投影**

**n=n+n;x=-siz:2\*siz/n:siz;y=sqrt(siz\*siz-x.^2); %绘制大圆所用的坐标序列**

**if(fillcode)**

**if(rake>0) %如果第一个节面的滑动角为正，则有向上逆冲分量，对应的小弧所加为膨胀区**

**fill([x,x(n+1:-1:1)]+lon,[y,-y(n+1:-1:1)]+lat,'w',p,q,clr)**

**else%如果第一个节面的滑动角为负，则有向下的正断层分量，对应的小弧所加为压缩区，则相反的区域为膨胀区，对相反的区域进行填充**

**fill([x,x(n+1:-1:1)]+lon,[y,-y(n+1:-1:1)]+lat,clr,p,q,'w')**

**end**

**else**

**plot(siz\*cos(0:2\*pi/n:2\*pi),siz\*sin(0:2\*pi/n:2\*pi),'k'); %绘制大圆**

**hold on**

**plot(xs1,ys1,clr,xs2,ys2,clr); %绘制第一个节面**

**end**

**if(insew)**

**hold on**

**line([0,0],[siz,siz\*1.1]); %绘制北向**

**text(0,siz\*(1.1),'N') %标示北向**

**line([-siz,-siz\*1.1],[0,0]); %绘制西向**

**text(-siz\*(1.1),0,'W') %标示北向**

**line([siz,siz\*1.1],[0,0]); %绘制东向**

**text(siz\*1.1,0,'E') %标示东向**

**line([0,0],[-siz,-siz\*1.1]); %绘制南向**

**text(0,-siz\*1.1,'S') %标示南向**

**end**

**if(iptb)**

**hold on**

**%绘制P轴**

**rp = siz\*sqrt(2)\*sind((90-Ptrpl(2))/2); %根据（10-5-13）计算Schmidt投影的距中心点距离**

**xp=lon+rp\*sind(Ptrpl(1)); yp=lat+rp\*cosd(Ptrpl(1));**

**plot(xp,yp,'.')**

**text(xp+0.1,yp,'P');**

**%绘制T轴**

**rp = siz\*sqrt(2)\*sind((90-Ttrpl(2))/2); %根据（10-5-13）计算Schmidt投影的距中心点距离**

**xp=lon+rp\*sind(Ttrpl(1)); yp=lat+rp\*cosd(Ttrpl(1));**

**plot(xp,yp,'o')**

**text(xp+0.1,yp,'T');**

**%绘制B轴**

**rp = siz\*sqrt(2)\*sind((90-Btrpl(2))/2); %根据（10-5-13）计算Schmidt投影的距中心点距离**

**xp=lon+rp\*sind(Btrpl(1)); yp=lat+rp\*cosd(Btrpl(1));**

**plot(xp,yp,'+')**

**text(xp+0.1,yp,'B');**

**end**

**return**

**%P10\_6.m**

**close all**

**str=24;dip=62; rake=91; %断层面的走向和倾角和滑动角**

**a=2;%绘制的Wulff网的大圆半径**

**n=500;%用500个点绘制断层面在Wulff网上的投影**

**x0=0;y0=0; %坐标原点**

**beachball\_wf(str,dip,rake,a,n,'r',0,0,1,1,1); %调用绘图程序**

**axis equal**

**axis off %不显示坐标轴**

**%P10\_6.m**

**close all**

**str=24;dip=62; rake=91; %断层面的走向和倾角和滑动角**

**a=2;%绘制的Wulff网的大圆半径**

**n=500;%用500个点绘制断层面在Wulff网上的投影**

**x0=0;y0=0; %坐标原点**

**beachball\_ea(str,dip,rake,a,n,'r',0,0,1,1,1); %调用绘图程序**

**axis equal**

**axis off %不显示坐标轴**

****

****



**一样**

**第九题**

**%P10\_10.m**

**%根据观测的P波初动符号求取震源机制的程序**

**x=[168.2924,86.92303,1**

**326.7913,87.72113,1**

**58.21029,85.43469,-1**

**185.5229,85.20844,1**

**60.82632,82.18145,1**

**348.8161,86.96339,1**

**204.1394,87.28638,-1**

**275.4883,84.50092,-1**

**357.0545,82.98753,1**

**16.11048,86.80354,1**

**346.7924,88.00838,1**

**98.66737,83.21033,-1**

**46.43449,87.10820,-1**

**129.3074,86.72790,1**

**233.6746,86.81039,-1**

**168.0691,88.02249,1**

**104.6950,87.11818,1**

**]; %给出的观测P波初动的走向、偏垂角及符号**

**Indxp=find(x(:,3)==1); %找到初动为正的数据序号**

**Indxn=find(x(:,3)==-1); %找到初动为负的数据序号**

**azi=x(:,1);tak=x(:,2);psign=x(:,3); %将数据的走向、俯角和初动符号分开**

**nlength=length(azi); %得到数据的总个数**

**ddelt=5;daz=5;dr=5; %搜索间隔**

**vect=[cosd(azi).\*sind(tak),sind(azi).\*sind(tak),cosd(tak)]'; %给出数据在北东下坐标系中的投影向量**

**chisqmin=1.0e20; %记录最大平方和**

**posqmax=0; %初动为正的总权重的最大值**

**MDBmin=[]; %记录得到结果的矛盾比**

**sol=[];**

**start=[];**

**for delt=0:ddelt:90**

**delt %显示搜索的进程**

**csdelt=cosd(delt);sndelt=sind(delt);**

**for az=0:daz:360**

**csaz=cosd(az); snaz=sind(az);**

**for r=0:dr:90**

**csR=cosd(r); snR=sind(r);**

**T=[csaz\*csR-snaz\*csdelt\*snR,snaz\*csR+csaz\*csdelt\*snR,sndelt\*snR]; %张轴在北东下坐标系中的表示**

**P=[-csaz\*snR-snaz\*csdelt\*csR,-snaz\*snR+csaz\*csdelt\*csR,sndelt\*csR]; %压轴在北东下坐标系中的表示**

**B=cross(T,P); %中间轴在北东下坐标系中的表示**

**RTPB=[T;P;B]; %转换矩阵**

**PTPB=RTPB\*vect; %给出地震射线在TPB轴下的表示**

**trpl=[];**

**for ii=1:nlength %求出与T轴夹角fai和与B轴的夹角theta**

**trpl=[trpl;v2trpl([PTPB(:,ii)]')]; %trpl的第一个数为该向量和T轴的夹角，第二个数为该向量与TP面的夹角**

**end**

**weit=cosd(2\*trpl(:,1)).\*sind(90-trpl(:,2));**

**mul=weit.\*psign;**

**Indxfit=find(mul>0); %符合初动符号的序号**

**Indxunfit=find(mul<0); %符合初动符号的序号**

**posq=sum(weit(Indxfit)); %符合初动符号的权重**

**chisq=sum(4\*weit(Indxunfit).^2); %不符合初动符号的权重的平方和**

**MDB=length((Indxunfit))/(length(Indxunfit)+length(Indxfit)); %矛盾比**

**N=sum(abs(weit));**

**if(chisq==chisqmin)**

**if(posq==posqmax)**

**PTchange=[PTchange;0];**

**start=[start;az,delt,r];**

**sol=[sol;T;P;B];**

**MDBmin=[MDBmin,MDB];**

**else**

**posqmax=posq;**

**end**

**elseif(chisq<chisqmin)**

**PTchange=[0];**

**start=[az,delt,r];**

**sol=[T;P;B];**

**chisqmin=chisq;**

**posqmax=posq;**

**MDBmin=[MDB];**

**end**

**%交换P,T轴的位置就覆盖了另外的一半空间，将计算的权重符号完全相反，求这种情况下的拟合情况**

**weit=-weit; %将权重符号反号，相当于原来的正号区变为负号区，原来的负号区变为正号区,虽然符号改变，但权重的绝对值不改变**

**mul=weit.\*psign;**

**Indxfit=find(mul>0);**

**Indxunfit=find(mul<0);**

**MDB=length(Indxunfit)/(length(Indxunfit)+length(Indxfit));**

**chisq=sum(4\*weit(Indxunfit).^2);**

**N=sum(abs(weit));**

**if(chisq==chisqmin)**

**if(posq==posqmax)**

**PTchange=[PTchange;1];**

**start=[start;az,delt,r];**

**sol=[sol;T;P;B];**

**MDBmin=[MDBmin,MDB];**

**else**

**posqmax=posq;**

**end**

**elseif(chisq<chisqmin)**

**PTchange=[1];**

**start=[az,delt,r];**

**sol=[T;P;B];**

**chisqmin=chisq;**

**posqmax=posq;**

**MDBmin=[MDB];**

**end**

**end**

**end**

**end**

**%绘图**

**siz=5; %绘制等面积投影的半径**

**alpha=0:0.1:2\*pi+0.1;**

**plot(siz\*cos(alpha),siz\*sin(alpha));**

**rp = siz\*sqrt(2)\*sin(deg2rad(x(Indxp,2))/2); %根据（10-5-13）计算Schmidt投影的距中心点距离**

**xp=rp.\*sin(deg2rad(x(Indxp,1))); yp=rp.\*cos(deg2rad(x(Indxp,1))); %得到初动为正的数据在大圆上的Schmidt投影点坐标**

**rn = siz\*sqrt(2)\*sin(deg2rad(x(Indxn,2))/2); %根据（10-5-13）计算Schmidt投影的距中心点距离**

**xn=rn.\*sin(deg2rad(x(Indxn,1))); yn=rn.\*cos(deg2rad(x(Indxn,1))); %得到初动为负的数据在大圆上的Schmidt投影点坐标**

**hold on**

**for ii=1:length(PTchange)**

**Ttrpl=v2trpl(sol((ii-1)\*3+1,:));**

**Ptrpl=v2trpl(sol((ii-1)\*3+2,:));**

**if(PTchange(ii))**

**temp=Ttrpl;Ttrpl=Ptrpl; Ptrpl=temp;**

**end**

**Btrpl=v2trpl(sol((ii-1)\*3+3,:));**

**[str1,dip1,rake1,str2,dip2,rake2,Btrpl,PTangle]=pt2ds(Ptrpl,Ttrpl);**

**beachball\_ea(str1,dip1,rake1,siz,50,'r',0,0,1,1,1)**

**plot(xp,yp,'.') %绘制初动为正的数据点**

**plot(xn,yn,'o') %绘制初动为负的数据点**

**axis equal %使坐标轴相等，看上去为正圆**

**end**

**axis off %不显示坐标轴**

**disp(sprintf('所求震源机制节面1走向%5.1f，倾角%4.1f，滑动角%6.1f\n',str1,dip1,rake1))**

**disp(sprintf('所求震源机制节面2走向%5.1f，倾角%4.1f，滑动角%6.1f\n',str2,dip2,rake2))**

**disp(sprintf('所求震源机制P轴走向%5.1f，倾伏角%4.1f\n',Ptrpl))**

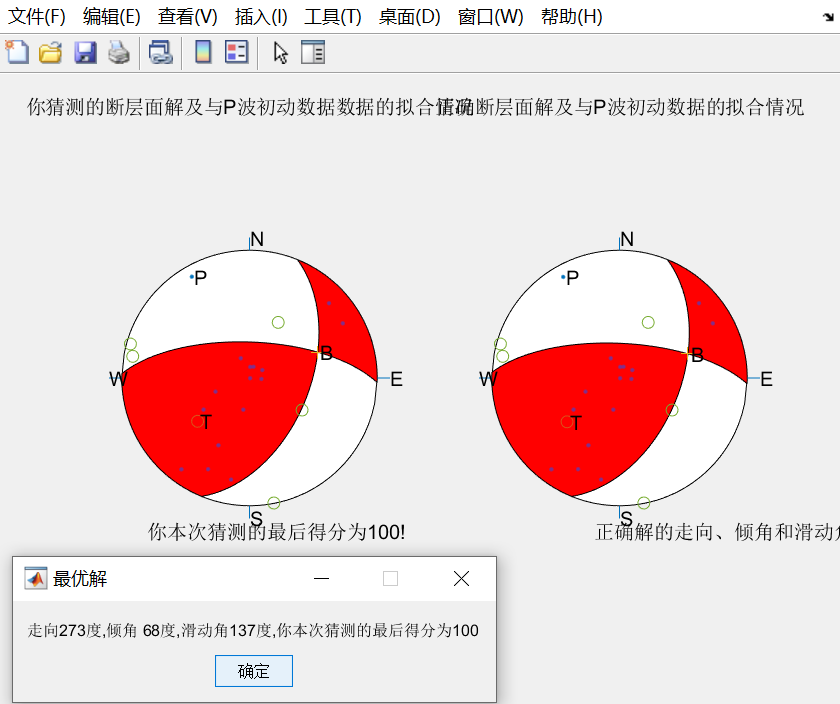
**disp(sprintf('所求震源机制T轴走向%5.1f，倾伏角%4.1f\n',Ttrpl))**

**disp(sprintf('所求震源机制B轴走向%5.1f，倾伏角%4.1f\n',Btrpl))**

**disp(sprintf('所求震源机制的矛盾比为%f\n',MDBmin))**

****

**第十题**



**第十一题**

**%P10\_12.m**

**fai=-pi:0.01:pi; %fai旋转一周**

**alpha=0.4; %破裂速度和传播速度之比**

**r=sin(2\*fai) ./(1-alpha\*cos(fai)); %P波辐射花样在Oxy平面的表示**

**x=abs(r).\*cos(fai); %转换为平面坐标**

**y=abs(r).\*sin(fai);**

**subplot(1,2,1),plot(x,y) %绘图**

**axis equal**

**title('P波辐射花样')**

**xlabel('x');ylabel('y')**

**r=cos(2\*fai) ./(1-alpha\*cos(fai)); %SH波辐射花样在Oxy平面的表示**

**x=abs(r).\*cos(fai); %转换为平面坐标**

**y=abs(r).\*sin(fai);**

**subplot(1,2,2),plot(x,y) %绘图**

**axis equal**

**title('SH波辐射花样')**

**xlabel('x');ylabel('y')**

**%P10\_15.m**

**fai=-pi:0.01:pi; %fai旋转一周**

**alpha=0.4; %破裂速度和传播速度之比**

**r=sin(2\*fai) .\*(1./(1-alpha\*cos(fai))+1./(1+alpha\*cos(fai))); %P波辐射花样在Oxy平面的表示**

**x=abs(r).\*cos(fai); %转换为平面坐标**

**y=abs(r).\*sin(fai);**

**subplot(1,2,1),plot(x,y) %绘图**

**axis equal**

**title('P波辐射花样')**

**xlabel('x');ylabel('y')**

**r=cos(2\*fai) .\*(1./(1-alpha\*cos(fai))+1./(1+alpha\*cos(fai))); %SH波辐射花样在Oxy平面的表示**

**x=abs(r).\*cos(fai); %转换为平面坐标**

**y=abs(r).\*sin(fai);**

**subplot(1,2,2),plot(x,y) %绘图**

**axis equal**

**title('SH波辐射花样')**

**xlabel('x');ylabel('y')**

****

****

**第十二题**

**%P10\_16.m**

**clear all;**

**infname='jan76\_mar15.txt'; %所给出的ndk格式的自76年1月至2013年9月的矩心矩张量目录文件**

**outname='hubei.dat'; %给出的存储找到的新疆地区的震源机制解的文件名**

**% 根据一定的条件读取哈佛CMT 地震目录的程序**

**lonlim=[108,116];%经度范围**

**latlim=[29,33]; %纬度范围**

**deplim=[0,30]; %深度范围**

**Syear=1967;Smon=1;Sday=1;Shour=0;Sminute=0;Ssec=0; %搜索目录的开始时间**

**StartTime=datenum(Syear,Smon,Sday,Shour,Sminute,Ssec); %将搜索目录的开始时间转换为统一时间**

**Eyear=2015;Emon=3;Eday=31;Ehour=23;Eminute=59;Esec=59; %搜索目录的结束时间**

**EndTime=datenum(Eyear,Emon,Eday,Ehour,Eminute,Esec); %搜索目录的结束时间**

**myData = textread(infname, '%s', 'delimiter', '\n', 'whitespace', ''); %读取ndk格式的文件**

**outData =[]; %储存所需要输出的CMT参数值**

**for ii = 1:length(myData)/5 %对目录中的所有地震进行搜索**

**lat=str2num(myData{(ii-1)\*5+1}(28:33)); %纬度**

**lon=str2num(myData{(ii-1)\*5+1}(35:41)); %经度**

**dep=str2num(myData{(ii-1)\*5+1}(43:47)); %深度**

**year = str2num(myData{(ii-1)\*5+1}(6:9)); % 年**

**mon = str2num(myData{(ii-1)\*5+1}(11:12)); % 月**

**day = str2num(myData{(ii-1)\*5+1}(14:15)); % 日**

**hour = str2num(myData{(ii-1)\*5+1}(17:18)); % 时**

**minute = str2num(myData{(ii-1)\*5+1}(20:21)); %分**

**sec=str2num(myData{(ii-1)\*5+1}(23:26)); % 秒**

**eqtime=datenum(year,mon,day,hour,minute,sec); %转换为统一的时间**

**%判断是否在所给的范围之内**

**if(lat>latlim(1)&lat<latlim(2)&lon>lonlim(1)&lon<lonlim(2)&dep>deplim(1)&dep<deplim(2)&eqtime>StartTime&eqtime<EndTime)**

**%[year,mon,day,hour,minute,sec]**

**mo1 = str2num(myData{(ii-1)\*5+4}(1:2)); % Mo (part1)**

**mo2 = str2num(myData{(ii-1)\*5+5}(52:56)); % Mo (part2)**

**mo = mo2\*10.0^mo1; %地震矩，单位为dyn.cm**

**mw = log10(mo)/1.5 - 10.73; %转化为矩震级的统计公式**

**str1 = str2num(myData{(ii-1)\*5+5}(58:60)); % 走向-1**

**dip1 = str2num(myData{(ii-1)\*5+5}(62:63)); % 倾角-1**

**rak1 = str2num(myData{(ii-1)\*5+5}(65:68)); % 滑动角-1**

**str2 = str2num(myData{(ii-1)\*5+5}(70:72)); % 走向-2**

**dip2 = str2num(myData{(ii-1)\*5+5}(74:75)); % 倾角-2**

**rak2 = str2num(myData{(ii-1)\*5+5}(77:80)); % 滑动角-2**

**Tpl=str2num(myData{(ii-1)\*5+5}(13:14)); %T轴俯角**

**Taz=str2num(myData{(ii-1)\*5+5}(16:18)); %T轴走向**

**Bpl=str2num(myData{(ii-1)\*5+5}(28:29)); %B轴俯角**

**Baz=str2num(myData{(ii-1)\*5+5}(31:33)); %B轴走向**

**Ppl=str2num(myData{(ii-1)\*5+5}(43:44)); %P轴俯角**

**Paz=str2num(myData{(ii-1)\*5+5}(46:48)); %P轴走向**

**outData=[outData;eqtime,lat,lon,dep,mo,mw,str1,dip1,rak1,str2,dip2,rak2,Taz,Tpl,Paz,Ppl];**

**%将所需要的参数列到输出数据中，读者可以根据某项研究的需要进行修改**

**end**

**end**

**fid=fopen(outname,'w'); %打开要写入的文件**

**fprintf(fid,'%15.2e %8.2f %8.2f %8.2f %15.2e %8.2f %4d %4d %4d %4d %4d %4d %4d %4d %4d %4d \n',outData'); %将筛选的数据写入的文件中**

**%注意输出格式的位数要能够容纳索要存储的数据**

**fclose(fid); %关闭打开的文件**

**%save outname outData -ascii %也可以采用该语句将输出数据写到指定文件名的文件中**

**7.33e+05 29.69 115.69 10.00 7.18e+23 5.17 324 57 23 221 71 145 179 38 275 9**

**7.36e+05 31.07 110.41 17.80 3.20e+23 4.94 71 65 165 167 77 26 32 28 297 8**

**实验总结及感想**

**学会了通过matlab看震源机制球和描述断层**