SPHERICAL PROJECTIONS

I Main Topics
  A What is a spherical projection?
  B Spherical projection of a line
  C Spherical projection of a plane
  D Determination of fold axes
  E Free spherical projection program for the MacIntosh:
    "Stereonet" by Rick Allmendinger at Cornell University

II What is a spherical projection?
    A A 2-D projection for describing the orientation of 3-D features. A spherical projection shows where lines or planes that intersect the surface of a (hemi)sphere, provided that the lines/planes also pass through the center of the (hemi)sphere.
    B Great circle: intersection of the surface of a sphere with a plane that passes through the center of the sphere (e.g., lines of longitude)
    C Small circle: intersection of the surface of a sphere with a plane that does not pass through the center of the sphere (e.g., lines of latitude). A line rotated about an axis traces a small circle too.
  B Types of spherical projections
    1 Equal angle projection (Wulff net)
    2 Equal area projection (Schmidt net)

III Spherical projection of a line
  A Technique
    1 A line is at the intersection of two planes: 1) a vertical plane coinciding with the trend of the line and (2) an inclined plane coinciding with the plunge of the line.
    2 Trend and plunge: The point representing a line plots away from the center of the spherical plot in the direction of the trend of the line. **The trend of a line is measured along a horizontal great circle. The plunge of the line is measured along a vertical great circle.**
    3 Rake: If the strike and dip of a plane is specified, the rake (pitch) of a line in the plane can be measured along the cyclographic trace of the

great circle representing that plane. Rake is measured from the direction of strike.

B <u>Plane containing two lines</u>: Two intersecting lines uniquely define a plane. The cyclographic trace of the great circle representing that plane will pass through the points representing the lines.

C <u>Angle between two lines:</u> This angle is measured along the cyclographic trace of the unique great circle representing the plane containing the two lines

IV Spherical projection of a plane

A A plane plots as the cyclographic trace of a great circle

B <u>Strike and dip</u>: **The strike is measured around the perimeter of the primitive circle. The dip of the line is measured along a vertical great circle perpendicular to the line of strike.**

C <u>Intersection of two planes</u>

1 Two planes intersect in a line, which projects as a point in a spherical projection. This point is at the intersection of the cyclographic traces of the two planes.

2 The intersection is also 90° from the poles to the two planes; these 90° angles are measured along the great circles representing the planes containing the poles.

D <u>Angles between planes</u>

1 The angle between two planes is the angle between the poles to the planes. This angle is measured along the cyclographic trace of the unique great circle representing the plane containing the poles to the two planes.

2 <u>For equal angle projections alone</u>, the angle between two planes is the angle between tangent lines where the cyclographic traces of two planes intersect (hence the name of the projection)

V Fold axes of cylindrical folds

A The fold axis is along the line of intersection of beds (β diagram). (See IVA)

B The fold axis is perpendicular to the plane containing the poles to beds (π diagram); this approach works better for many poles to beds (See IIIA)

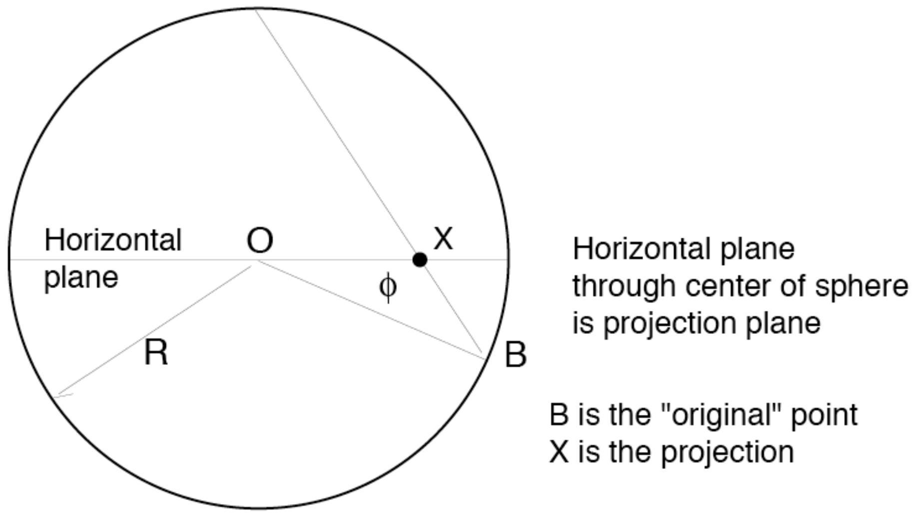
Geometrical Properites of Equal Angle and Equal Area projections

(From Hobbs, Means, and Williams, 1976, An Outline of Structural Geology)

Property	Equal angle projection	Equal area projection
Net type	Wulff net	Schmidt net
Projection does not preserve ...	Areas	Angles
Projection preserves ...	Angles	Areas
A line project as a ...	Point	Point
A great circle projects as a ...	Circle	Fourth-order quadric
A small circle projects as a ...	Circle	Fourth-order quadric
Distance from center of primitive circle to cyclographic trace measured in direction of dip	$R\tan\left(\frac{\pi}{4}-\frac{dip}{2}\right)$	$R\sqrt{2}\sin\left(\frac{\pi}{4}-\frac{dip}{2}\right)$
Distance from center of primitive circle to pole of plane measured in the direction opposite to that of the dip	$R\tan\left(\frac{dip}{2}\right)$	$R\sqrt{2}\sin\left(\frac{dip}{2}\right)$
Distance from center of primitive circle to point that represents a plunging line	$R\tan\left(\frac{\pi}{4}-\frac{plunge}{2}\right)$	$R\sqrt{2}\sin\left(\frac{\pi}{4}-\frac{plunge}{2}\right)$
Best use	Measuring angular relations	Contouring orientation data

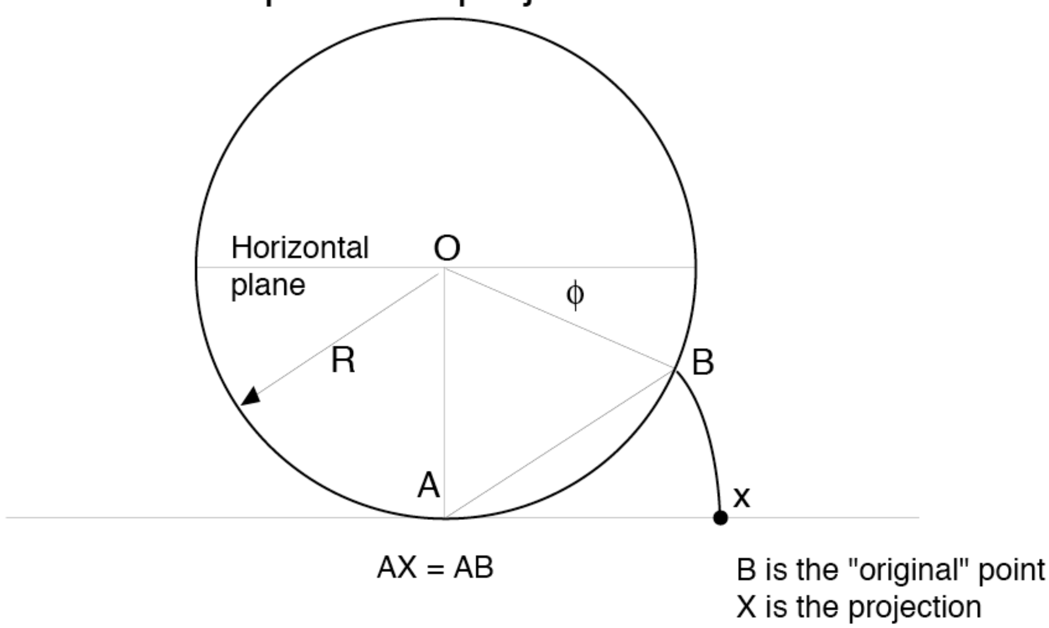
Spherical Projections

Equal-angle projection (Sterographic Projection)



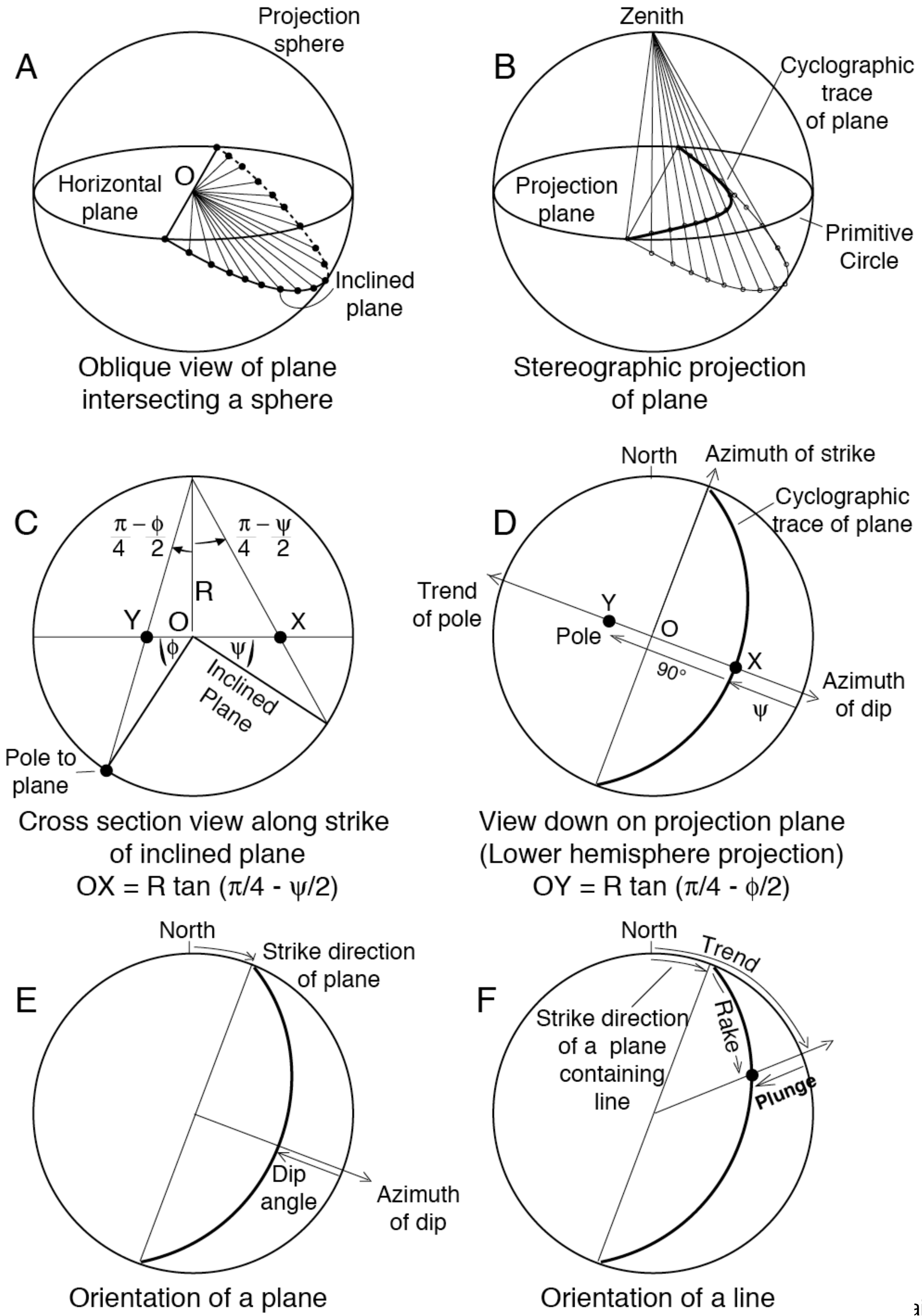
The **shapes** of plane shapes on the surface of the sphere **are preserved** in this projection, but the relative **areas are altered**. Good for measuring the angles between the cyclographic traces of planes.

Equal-area projection

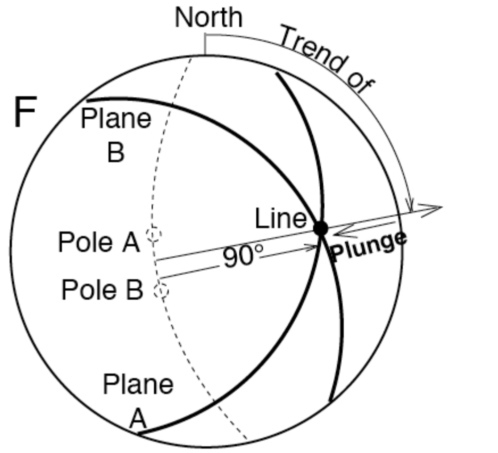


The relative **areas** of plane shapes on the surface of the sphere **are preserved** in this projection, but the **shapes are altered**. Good for representing the density of poles.

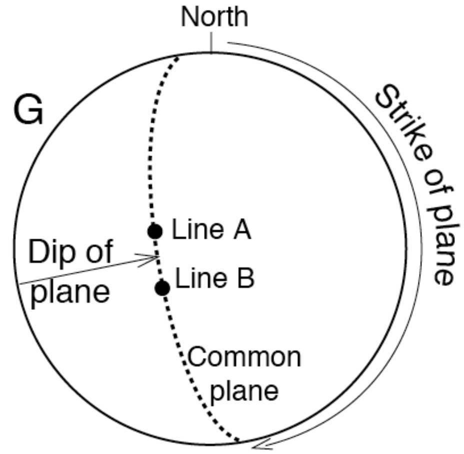
Stereographic (Equal-angle) Projections (I)



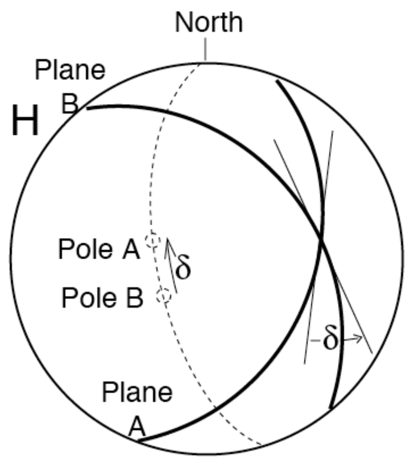
Stereographic (Equal-angle) Projections (II)



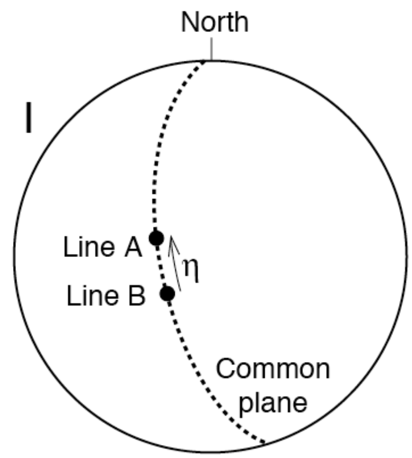
Line of intersection of two planes



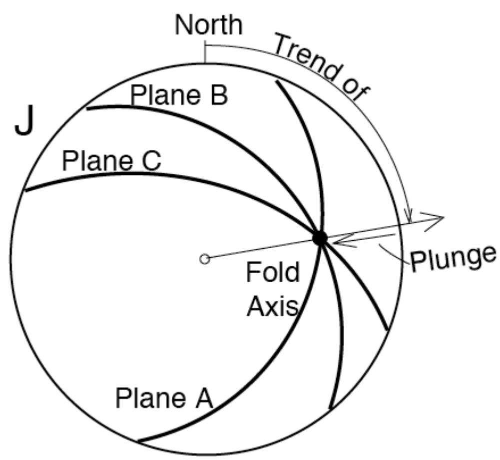
Plane containing two lines



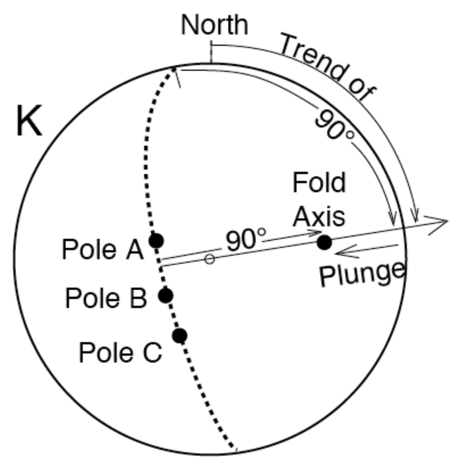
Angle δ between two planes



Angle η between two lines

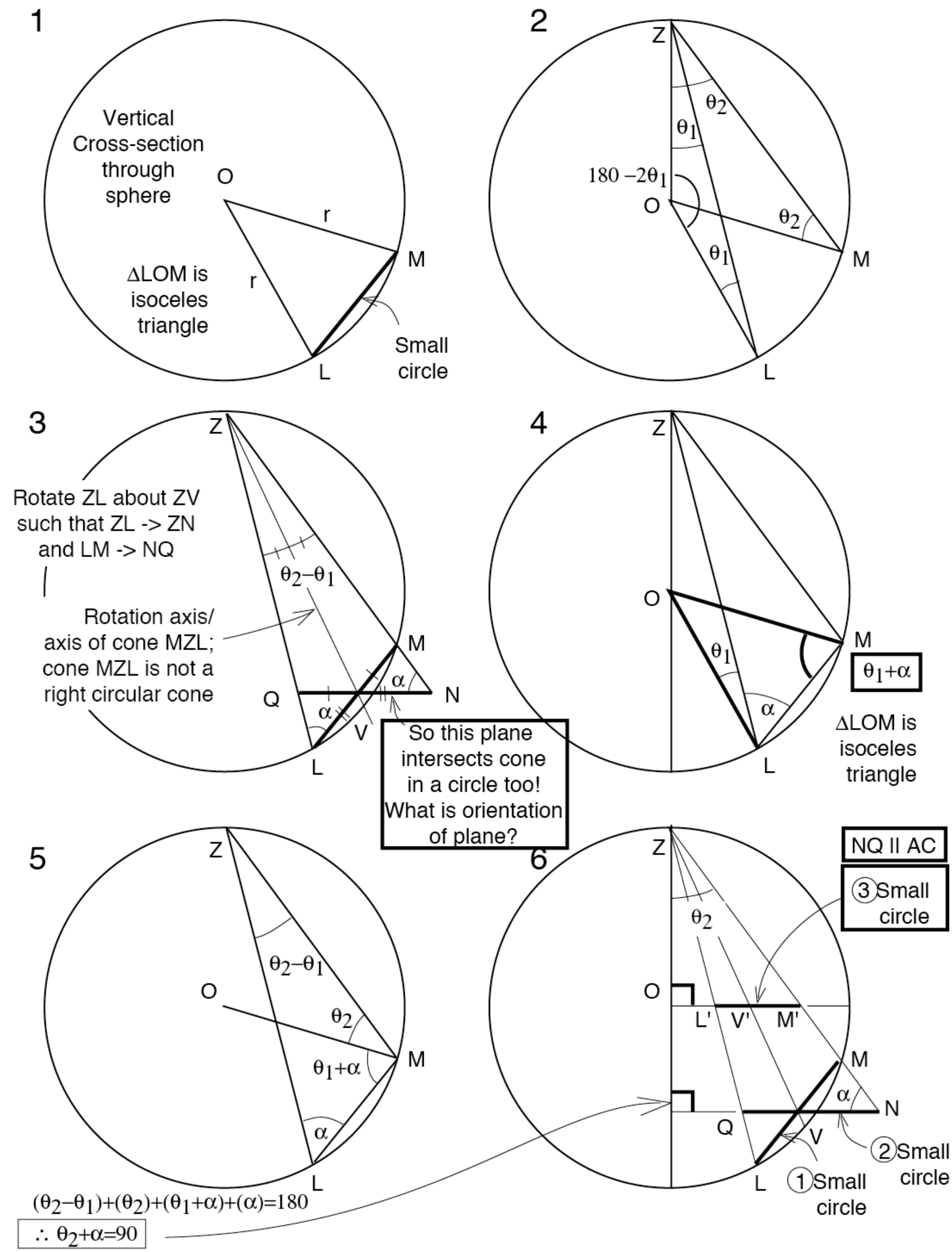


Cylindrical fold axis
by intersecting bedding planes
β diagram

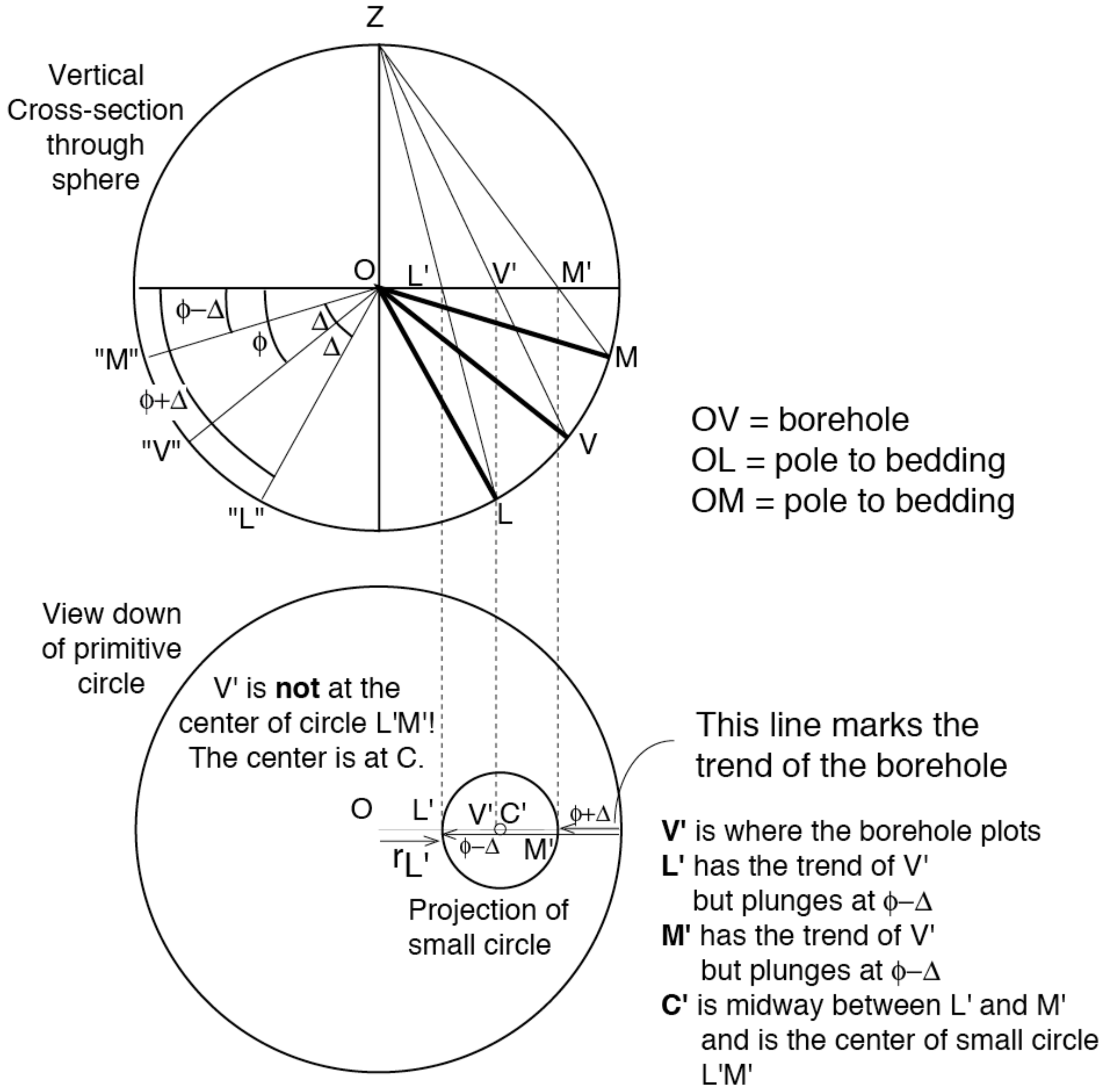


Cylindrical fold axis
by normal to poles
π diagram

Equal-Angle Projection of a Small Circle



Equal-Angle Projection of a Small Circle (II)



$$\frac{r_{L'}}{R} = \tan\left[\frac{90^o - (\phi+\Delta)}{2}\right] \qquad r_{L'} = R\tan\left[\frac{90^o - (\phi+\Delta)}{2}\right] \qquad \phi+\Delta = 90^o - 2(\tan^{-1}[r_{L'}/R])$$

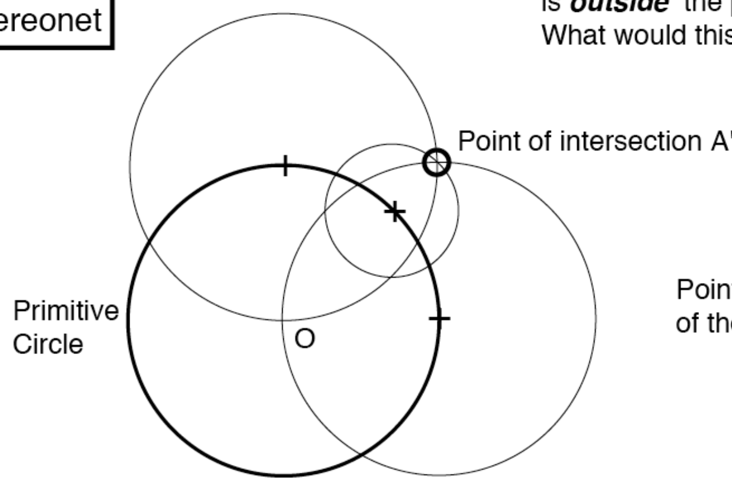
$$\frac{r_{M'}}{R} = \tan\left[\frac{90^o - (\phi-\Delta)}{2}\right] \qquad r_{M'} = R\tan\left[\frac{90^o - (\phi-\Delta)}{2}\right] \qquad \phi-\Delta = 90^o - 2(\tan^{-1}[r_{M'}/R])$$

$$r_C = \frac{r_{L'}+r_{M'}}{2} \qquad \text{radius of proj. small circle} = \frac{r_{L'} - r_{M'}}{2}$$

Points Outside a Primitive Circle in Equal-angle Projections

View down onto stereonet

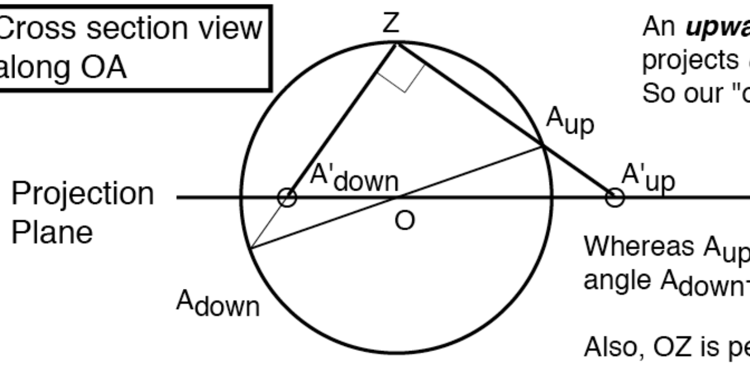
The point of intersection of three circles is ***outside*** the primitive circle. What would this mean?



Point O is the center of the primitive circle

Let's return to how the projection is done to answer the question

Cross section view along OA

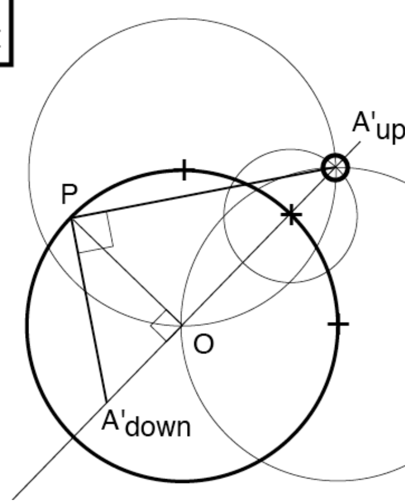


An ***upward*** pointing line projects ***outside*** the primitive circle! So our "outside" point A' is really A'_{up}.

Whereas A_{up}-A_{down} is a diameter, then angle A_{down}-Z-A_{up} must be a right angle.

Also, OZ is perpendicular to A'_{up}.

View down onto stereonet



To plot the downward-pointing pole corresponding to A'_{up}, we turn the equal-angle projection method "on its side":

1 Draw a line from A'_{up} through O

2 Draw line OP perpendicular to line OA'_{up}

3 Draw line OA'_{down} perpendicular to OA'up. Points A'down, O , and A'up lie on one line

Lab 5 Spherical Projections

Use a separate piece of paper for each exercise, and include printouts of your Matlab work. 125 points total.

Exercise 1: Plots of lines **(30 points total)**

Plot and neatly label the following lines on an equal angle projection:

Line	Trend **(1 point each)**	Plunge **(1 point each)**
A	N40°W	4°
B	S30°W	10°
C	N85°E	30°

Draw with a light line the cyclographic traces of the three planes containing the three pairs of lines (1), determine the angles between the lines (1), and label the angles on the stereographic plot (1).

Lines	Angle in degrees **(3 points each)**
A & B	
B & C	
C & A	

Now check your results using dot products of unit vectors along the lines. First find the direction cosines for each line using Matlab.

Line	α **(1 point each)**	β **(1 point each)**	γ **(1 point each)**
A			
B			
C			

Now take the dot products and use them to find the angles between the lines (remember to convert to degrees)

Lines	Dot product **(1 point each)**	Angle (°) **(1 point each)**
A & B		
B & C		
C & A		

<u>Exercise 2: Plots of planes</u> **(36 points total)**

Plot and neatly label the following planes (strike and dip follow right-hand rule convention) and the poles to those planes on an equal angle projection. Use a fairly heavy line to designate the planes.

Plane	Strike **1 point each**	Dip **1 point each**	Trend of pole **1 point each**	Plunge of pole **1 point each**
F	256°	22°		
G	68°	72°		
H	145°	44°		

Draw with a light line the cyclographic traces of the three planes containing the three pairs of poles (1), determine the angles between the lines (1), and label the angles on the stereographic plot (1).

Planes	Angle in degrees **(3 points each)**
F & G	
G & H	
H & F	

Now check your results using dot products of unit vectors along the lines. First find the direction cosines for each pole using Matlab

Line	α **(1 point each)**	β **(1 point each)**	γ **(1 point each)**
Pole to plane F			
Pole to plane G			
Pole to plane H			

Now take the dot products of the unit normals, and use them with Matlab's acos function to find the angles between the lines (remember to convert to degrees)

Poles to planes...	Dot product **(1 point each)**	Angle (°) **(1 point each)**
F & G		
G & H		
H & F		

Exercise 3: Intersection of planes problem (fold axes) **(18 points total)**

Using a β-plot (direct intersection of planes), determine the trend and plunge of the fold axis for a cylindrical fold by plotting the bedding attitudes listed below and finding the trend and plunge of the line of intersection.

Bed	Strike **(1 point)**	Dip **(1 point)**
E1	84°	60°S
E2	117°	90°

Fold axis trend **(1 point)**	Fold axis plunge **(1 point)**

Now check your results using vector algebra. First find the direction cosines for each pole using Matlab

Line	α **(1 point each)**	β **(1 point each)**	γ **(1 point each)**
Pole to E1 **(n1)**			
Pole to E2 **(n2)**			

Now take the cross products of the unit normals, and find the trend and plunge of the vector that is produced. Do not give an answer with a negative plunge, and give the angles in degrees, not radians.

n1x n2 **(1 point)**	**|n1x n2|** **(1 point)**	α **(1 point)**	β **(1 point)**	γ **(1 point)**
, ,				

Cross product trend (°) **(1 point)**	Cross product plunge (°) **(1 point)**

Exercise 4 (24 points total)

First find the orientations of the poles to bedding, plot the poles, and then use a π-plot (poles to bedding) to determine the trend and plunge of the fold axis for a cylindrical fold. Show the cyclographic trace of the plane containing the poles in a light line

Plane	Strike	Dip	Trend of pole 2 point each	Plunge of pole 2 point each
F1	345°	40°E		
F2	213°	68°W		

Fold axis trend (2 point)	Fold axis plunge (2 point)

Now check your results using vector algebra. First find the direction cosines for each pole using Matlab

Line	α (1 point each)	β (1 point each)	γ (1 point each)
Pole to F1 (n3)			
Pole to F2 (n4)			

Now take the cross products of the unit normals, and find the trend and plunge of the vector that is produced. Do not give an answer with a negative plunge, and give the angles in degrees, not radians.

Cross product (1 point)	$\lvert n3 \times n4 \rvert$ (1 point)	α (1 point)	β (1 point)	γ (1 point)
, ,				

Cross product trend (°) (1 point)	Cross product plunge (°) (1 point)

Exercise 5 (**17 points total**)

Slope stability sliding block problem (orientation-of-intersection problem)

Three sets of fractures are present in the bedrock along the shores of a reservoir. You are to evaluate whether fracture-bounded blocks might pose a hazard to the reservoir by being able to slide into the reservoir. The attitudes of the fractures are:

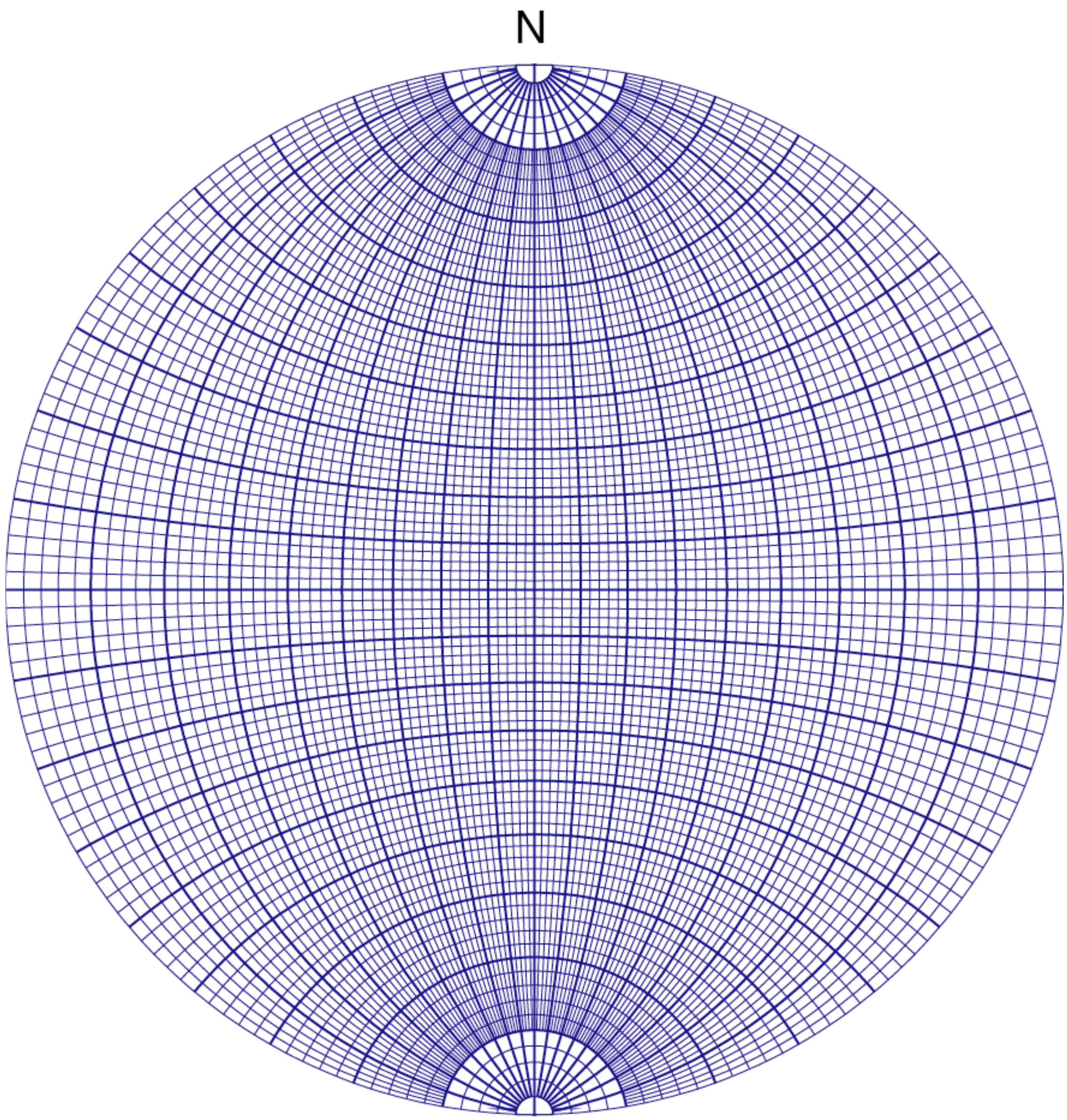
Set	Strike	Dip
1	0°	40°E
2	96°	30°S
3	264°	22°N

On the north side of the reservoir the ground surface slopes due south at 30°. On the south side of the reservoir the ground surface slopes due north at 45°.

Noting that (a) a fracture-bounded block can only slide parallel to the intersection of two fractures, and (b) a block can slide only if the slide direction has a component in the downhill direction, determine the trend and plunge of possible sliding directions. After considering the sliding directions and the geometries of the slopes, do any of these directions seem like they might pose a hazard to the reservoir? Why? Drawing a north-south cartoon cross section may help you here.

Scoring: **2 points for each of the three planes = 6 points total**
2 points for each of the three intersections = 6 points total
5 points for the discussion

Equal-Angle Net
(Wulff Net)



```
% Matlab script wullf4 to generate Wulff nets
% This generates a wullf (equal angle) net by typing "wulff4"

% Definition of variables
% x,y: center of arc
% r: radius of arc
% thetaa: lower limt of arc range
% thetab: upper limit of arc range

% Clear screen
clf;
figure(1)
clf
%       Set radius of Wulff net primitive circle
bigr = 1.2;
phid = [2:2:88];   % Angular range for great circles
phir = phid*pi/180;
omegad = 90 - phid;
omegar = pi/2-phir;

% Set up for plotting great circles with centers along
% positive x-axis
x1 = bigr.*tan(phir);
y1 = zeros(size(x1));
r1 =  bigr./cos(phir);
theta1ad = (180-80)*ones(size(x1));
theta1ar = theta1ad*pi/180;
theta1bd = (180+80)*ones(size(x1));
theta1br = theta1bd*pi/180;

% Set up for plotting great circles
% with centers along the negative x-axis
x2 = -1*x1;
y2 = y1;
r2 = r1;
theta2ad = -80*ones(size(x2));
theta2ar = theta2ad*pi/180;
theta2bd =  80*ones(size(x2));
theta2br = theta2bd*pi/180;


% Set up for plotting small circles
% with centers along the positive y-axis
y3 = bigr./sin(omegar);
x3 = zeros(size(y3));
r3 =  bigr./tan(omegar);
theta3ad = 3*90-omegad;
theta3ar = 3*pi/2-omegar;
theta3bd = 3*90+omegad;
theta3br = 3*pi/2+omegar;

% Set up for plotting small circles
% with centers along the negative y-axis
y4 = -1*y3;
x4 = x3;
r4 =  r3;
theta4ad = 90-omegad;
```

```
theta4ar = pi/2-omegar;
theta4bd = 90+omegad;
theta4br = pi/2+omegar;

% Group all x, y, r, and theta information for great cricles
phi = [phid, phid];
x = [x1, x2];
y = [y1, y2];
r = [r1, r2];
thetaad = [theta1ad, theta2ad];
thetaar = [theta1ar, theta2ar];
thetabd = [theta1bd, theta2bd];
thetabr = [theta1br, theta2br];

% Plot portions of all great circles that lie inside the
% primitive circle, with thick lines (1 pt.) at 10 degree increments
for i=1:length(x)
        thd = thetaad(i):1:thetabd(i);
        thr = thetaar(i):pi/180:thetabr(i);
        xunit = x(i) + r(i).*cos(thr);
        yunit = y(i) + r(i).*sin(thr);
        p = plot(xunit,yunit,'LineWidth',0.5);
        hold on
end

% Now "blank out" the portions of the great circle cyclographic traces
% within 10 degrees of the poles of the primitive circle.
rr =  bigr./tan(80*pi/180);
ang1 = 0:pi/180:pi;
xx = zeros(size(ang1)) + rr.*cos(ang1);
yy = bigr./cos(10*pi/180).*ones(size(ang1)) - rr.*sin(ang1);
p = fill(xx,yy,'w')
yy = -bigr./cos(10*pi/180).*ones(size(ang1)) + rr.*sin(ang1);
p = fill(xx,yy,'w')

for i=1:length(x)
        thd = thetaad(i):1:thetabd(i);
        thr = thetaar(i):pi/180:thetabr(i);
        xunit = x(i) + r(i).*cos(thr);
        yunit = y(i) + r(i).*sin(thr);
        if mod(phi(i),10) == 0
                p = plot(xunit,yunit,'LineWidth',1);
                angg = thetaad(i)
        end
        hold on
end

% Now "blank out" the portions of the great circle cyclographic traces
% within 2 degrees of the poles of the primitive circle.
rr =  bigr./tan(88*pi/180);
ang1 = 0:pi/180:pi;
xx = zeros(size(ang1)) + rr.*cos(ang1);
yy = bigr./cos(2*pi/180).*ones(size(ang1)) - rr.*sin(ang1);
p = fill(xx,yy,'w')
yy = -bigr./cos(2*pi/180).*ones(size(ang1)) + rr.*sin(ang1);
p = fill(xx,yy,'w')
```

```
% Group all x, y, r, and theta information for small circles
phi = [phid, phid];
x = [x3, x4];
y = [y3, y4];
r = [r3, r4];
thetaad = [theta3ad, theta4ad];
thetaar = [theta3ar, theta4ar];
thetabd = [theta3bd, theta4bd];
thetabr = [theta3br, theta4br];

% Plot primitive circle
thd = 0:1:360;
thr = 0:pi/180:2*pi;
xunit = bigr.*cos(thr);
yunit = bigr.*sin(thr);
p = plot(xunit,yunit);
hold on

% Plot portions of all small circles that lie inside the
% primitive circle, with thick lines (1 pt.) at 10 degree increments
for i=1:length(x)
        thd = thetaad(i):1:thetabd(i);
        thr = thetaar(i):pi/180:thetabr(i);
        xunit = x(i) + r(i).*cos(thr);
        yunit = y(i) + r(i).*sin(thr);
%       blug = mod(thetaad(i),10)
        if mod(phi(i),10) == 0
                p = plot(xunit,yunit,'LineWidth',1);
                angg = thetaad(i)
        else
                p = plot(xunit,yunit,'LineWidth',0.5);
        end
        hold on
end

% Draw thick north-south and east-west diameters
xunit = [-bigr,bigr];
yunit = [0,0];
p = plot(xunit,yunit,'LineWidth',1);
hold on
xunit = [0,0];
yunit = [-bigr,bigr];
p = plot(xunit,yunit,'LineWidth',1);
hold on

% Parameters to control appearance of plot
% THESE COME AFTER THE PLOT COMMANDS!!!
axis([-bigr bigr -bigr bigr])
% axis ('square').  BAD way to get aspect ratio of plot.  It
% also considers titles and axis labels when scaling the figure!
set(gca,'DataAspectRatio',[bigr,bigr,bigr])
%axes('Position',[0,0,1,1]);
%axes('AspectRatio',[1,1]);
set(gca,'Visible','off');              % This turns off the visibility of the axes
% figure('PaperPosition',[1,3,6,6]);
print -dill wulffnet.ill
```

```
print -deps wulffnet.eps
% end
```

```
% Matlab script stereonet
% To plot lines and planes in stereographic
% (equal-angle) projections
clf
% Read input data on planes
load planes.dat
% Data in column 1 are strikes, and data in column 2 are dips
% of planes, with angles given in degrees
strike = planes(:,1)*pi/180;
dip = planes(:,2)*pi/180;
num = length(strike);
% find cyclographic traces of planes and plot them
R = 1;
rake = 0:pi/180:pi;
for i=1:num;
        plunge = asin(sin(dip(i)).*sin(rake));
        trend = strike(i) + atan2(cos(dip(i)).*sin(rake), cos(rake));
        rho = R.*tan(pi/4 - (plunge/2));
        % polarb plots ccl from 3:00, so convert to cl from 12:00
        polarb(pi/2-trend,rho,'-')
        hold on
end

load lines1.dat
% Data in column 1 are trends, data in column 2 are plunges
% of lines, with angles given in degrees
trend1 = lines1(:,1);
plunge1 = lines1(:,2);
num = length(lines1(:,1));
R = 1;
trendr1 = trend1*pi/180;
plunger1 = plunge1(:,1)*pi/180;
rho1 = R.*tan(pi/4 - ((plunger1)/2));
for i=1:num;
        % polarb plots ccl from 3:00, so convert to cl from 12:00
        polarb(pi/2-trendr1(i),rho1(i),'o')
        hold on
end

load lines2.dat
% Data in column 1 are trends, data in column 2 are plunges
% of lines, with angles given in degrees
trend2 = lines2(:,1);
plunge2 = lines2(:,2);
num = length(lines2(:,1));
R = 1;
trendr2 = trend2*pi/180;
plunger2 = plunge2*pi/180;
rho2 = R.*tan(pi/4 - ((plunger2)/2));
for i=1:num;
        % polarb plots ccl from 3:00, so convert to cl from 12:00
        polarb(pi/2-trendr2(i),rho2(i),'*')
        hold on
end
```

The following file, called lines1.dat, provides an example of an input file for the stereonet plotting program

```
19	02
43	16
52	03
51	08
110	18
190	02
232 04
235	10
242	30
000	65
340	22
270	34
```

The file lines2.dat has the same format.

The following file, called planes.dat, provides an example of an input file for the stereonet plotting program

```
20	20
230	72.5048
```