

```

% gda12_11
% Supports Figure 12.15
% example of estimating central vector of Fisher distribution
% using principle of Maximum Likelihood. The data are P-axes of
% moment tensors of deep earthquakes on the Kurile-Kamchatka
% subduction zone. The bootstrap method is used to compute
% confidence limits of the azimuth and plunge of the central vector

clear all;

% plot set-up
figure(1);
clf;
set( gca, 'LineWidth', 1 );
set( gca, 'Xtick', [] );
set( gca, 'Ytick', [] );
axis( [-1.1 1.1 -1.1 1.1] );
hold on;

% draw stereonet
% stereonet angles:
%   theta, angle E of N, so theta=0 is north
%   phi, vertical angle, 0 phi=0 when down
theta = (2*pi/360)*[0:360];
plot( sin(theta), cos(theta), 'k', 'LineWidth', 3 );
theta = (pi/180.0)*0.0;
text( 1.05*sin(theta), 1.05*cos(theta), 'N' );
theta = (pi/180.0)*90.0;
text( 1.05*sin(theta), 1.05*cos(theta), 'E' );
plotgrid=1;
if( plotgrid == 1 )
    phi = (pi/180.0)*[30, 60];
    theta = (2*pi/360)*[0:360];
    for p = phi
        plot( stereo(p)*sin(theta), stereo(p)*cos(theta), 'k--', 'LineWidth', 2 );
    end
    theta = (pi/180.0)*linspace(0,330,12);
    r = linspace(0,1,2);
    for t = theta
        plot( r*sin(t), r*cos(t), 'k--', 'LineWidth', 2 );
    end
end

% load data from file
X=load(' ../data/moments.txt');
% file: lon lat depth mrr mtt mpp mrt mrp mtp iexp
%       1  2  3    4  5  6  7  8  9  10
% moment tensor elements as read from file:
RR=X(:,4); TT=X(:,5); PP=X(:,6); RT=X(:,7); RP=X(:,8); TP=X(:,9);

% restrict calculation to data in a specific depth range
dmin=300; dmax=600; % depth range
j=find( (X(:,3)>=dmin) & (X(:,3)<=dmax) );
N = length(j);
Pgs=zeros(N,3);

% plot all vectors
k=1;
for i = j % loop over earthquakes

```

```

% I'm using an [up, east, north] coordinate system
% the CMT is in [R=up, T=south, P=east] coordinate system
M = zeros(3,3);
M(1,1)=RR(i); M(2,2)=PP(i); M(3,3)=TT(i);
M(1,2)=RP(i); M(2,1)=RP(i);
M(1,3)=-RT(i); M(3,1)=-RT(i);
M(2,3)=-TP(i); M(3,2)=-TP(i);

% extract eigenvectors
[V,D] = eig(M);

Pg = V(:,1); % P-axis in [up, east, north] coordinate system, is third eigenvector
if( Pg(1) > 0 ) % want axis to point down
    Pg=-Pg;
end

% save in array
Pgs(k,:) = Pg';
k = k+1;

% angle east of north, north=0
theta = atan2( Pg(2), Pg(3) );
% vertical angle from down, down=0
phi = atan( sqrt(Pg(2)^2 + Pg(3)^2)/(-Pg(1)) );
r = stereo(phi);
plot( r*sin(theta), r*cos(theta), 'ko', 'LineWidth', 4 );

end

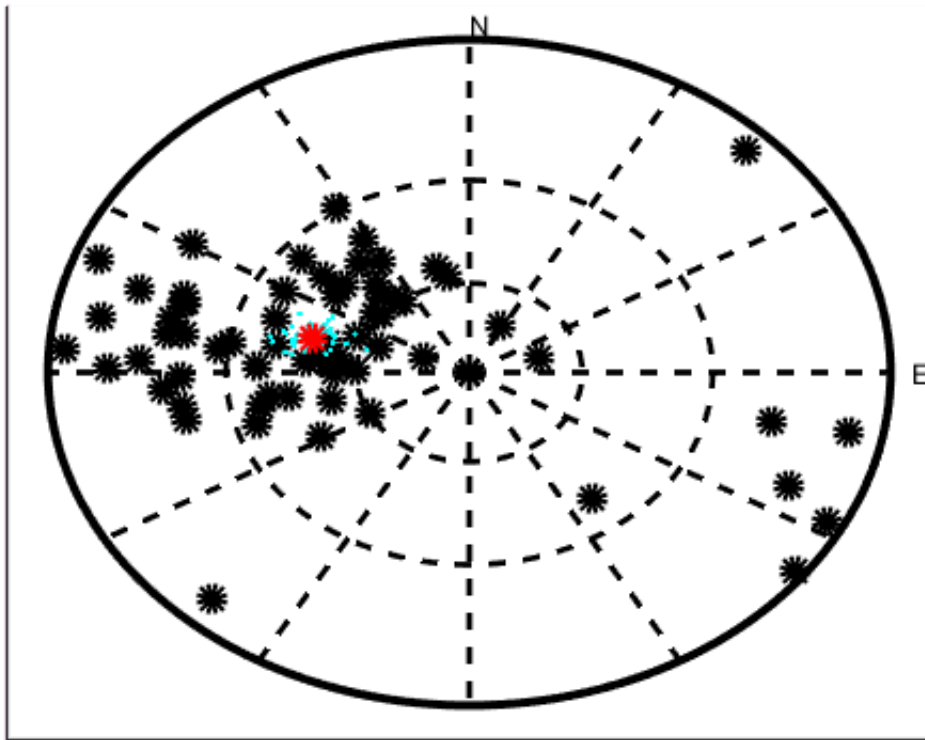
% estimate central vector
Pgsum = sum(Pgs)';
norm = sqrt(Pgsum'*Pgsum);
Pgsum = Pgsum/norm;
theta = atan2( Pgsum(2), Pgsum(3) );
phi = atan( sqrt(Pgsum(2)^2 + Pgsum(3)^2)/(-Pgsum(1)) );
r = stereo(phi);
plot( r*sin(theta), r*cos(theta), 'ro', 'LineWidth', 4 );

% bootstrap estimates saved in these arrays
Nboot=10000;
thetaboot=zeros(N,1);
phiboot=zeros(N,1);

% bootstrap calculation: resample data with duplications,
% and compute central vector estimate of each
for i=1:N
    % resample
    Pboot = Pgs(unidrnd(N,N,1),:);
    % central vector calculation
    Pbootsum = sum(Pboot)';
    norm = sqrt(Pbootsum'*Pbootsum);
    Pbootsum = Pbootsum/norm;
    % convert to angles
    thetaboot(i) = atan2( Pbootsum(2), Pbootsum(3) );
    phiboot(i) = atan( sqrt(Pbootsum(2)^2 + Pbootsum(3)^2)/(-Pbootsum(1)) );
    % save, plot as cyan dot
    r = stereo(phiboot(i));
    plot( r*sin(thetaboot(i)), r*cos(thetaboot(i)), 'c.', 'LineWidth', 2 );
end

```

```
% replot estimate, because dots from bootstrap obscure its symbol
r = stereo(phi);
plot( r*sin(theta), r*cos(theta), 'ro', 'LineWidth', 4 );
```



```
% Figure 12.15 Lower hemisphere stereonet showing P-axes of deep (300–600 km) earthquakes
% in the Kurile-Kamchatka subduction zone. The axes mostly dip to the west, parallel to the
% direction of subduction, indicating that the subducting slab is in down-dip compression.
% (Black circles) Axes of individual earthquakes. (Red circle) central axis, computed by
% maximum-likelihood technique. (Blue dots) Scatter of central axis determined by bootstrapping
% Data courtesy of the Global CMT Project. MatLab script gda12_11.
```

```
% compute standard deviation as a quick estimate of confidence intervals
sigmatheta = std(thetaboot);
sigmaphi = std(phiboot);

% display results
fprintf('Maximum Likelihood Estimate\n');
```

Maximum Likelihood Estimate

```
fprintf('azimuthal angle (North=0), theta = %.2f +/- %.2f (2 sigma)\n', 180*theta/pi, 2*180*sigmatheta/pi);
```

azimuthal angle (North=0), theta = -74.37 +/- 8.48 (2 sigma)

```
fprintf('vertical angle (Down=0), phi = %.2f +/- %.2f (2 sigma)\n', 180*phi/pi, 2*180*sigmaphi/pi);
```

vertical angle (Down=0), phi = 42.12 +/- 8.23 (2 sigma)

