```
%Three-Point Method to Determine Strike and Dip
%Avery Welker
%01/09/2017
%Calculation of the strike, dip, and dip direction were found by adapting
%the steps in: Hasbargen, L.E., 2012, A test of the three-point vector method
to determine strike
%and dip utilizing digital aerial imagery and topography, in Whitmeyer,
%S.J., Bailey, J.E., De Paor, D.G., and Ornduff, T., eds., Google Earth and
%Virtual Visualizations in Geoscience Education and Research: Geological
%Society of America Special Paper 492, p. 199-208,
%doi:10.1130/2012.2492(14).

%NOTE: x, y, z are easting, northing, and elevation respectively
%Input file name for Excel file: must be in 'filename.xlsx' form
prompt = 'Excel file name: ';
filename = input(prompt)

%Read Excel spreadsheet into matrices, user selected range
data = xlsread(filename, -1);

%Count number of rows to make zeros array for S vector
numRows = size(data, 1);
zeroCol = zeros(numRows, 1);

%Split data into x1, y1, z1, x2, y2, z2, x3, y3, z3
P1 = [data(:,1) data(:,2) data(:,3)];
P2 = [data(:,4) data(:,5) data(:,6)];
P3 = [data(:,7) data(:,8) data(:,9)];

%Define two vectors V1 and V2
%V1 is from P2 to P1
%V2 is from P2 to P3
V1 = [P1(:,1)-P2(:,1) P1(:,2)-P2(:,2) P1(:,3)-P2(:,3)];
V2 = [P3(:,1)-P2(:,1) P3(:,2)-P2(:,2) P3(:,3)-P2(:,3)];

%Calculate cross product V1 X V2
%Results in normal vector U = U1i - U2j + U3k
U = [V1(:,2).*V2(:,3)-V2(:,2).*V1(:,3) -(V1(:,1).*V2(:,3)-V2(:,1).*V1(:,3))
V1(:,1).*V2(:,2)-V2(:,1).*V1(:,2)];

%Compute Strike vector
%Cross product U X <0,0,1> (Pole to arbitrary horizontal plane)
%S will become (Easting, Northing, 0)
%Check for sign of U(:,3), if negative, change [E,N,0] to [-E,-N,0]
%S = [U(:,2) -U(:,1) 0];
S = zeros(numRows,3);
for h = 1:numRows
    if U(h,3) > 0
        S(h,1) = -U(h,2);
        S(h,2) = U(h,1);
        S(h,3) = 0;
    else
        S(h,1) = U(h,2);
        S(h,2) = -U(h,1);
        S(h,3) = 0;
    end
end
```

```
%Must check if U2 == U1 == 0
%This will make an infinite number of strike lines possible
%If true, this will set strike line to 0 degrees
%All other strike lines are calculated within the 'else' statement
%Pre-allocate matrix bearing for speed
bearing = zeros(numRows, 1);
for i = 1:numRows
    if U(i,2) && U(i,1) == 0
        bearing(i,1) = 0;
    else
        bearing(i,1) = rad2deg(acos(S(i,2) / (sqrt(((S(i,1))^2 +
(S(i,2))^2)))));
    end
end


%Correct bearing to azimuthal form
%Pre-allocate matrix azi for speed
azi = zeros(numRows, 1);
for j = 1:numRows
    if bearing(j,1) == 0
        azi(j,1) = bearing(j,1);
    elseif S(j,1) > 0
        if S(j,2) > 0
            azi(j,1) = bearing(j,1);
        elseif S(j,2) < 0
            azi(j,1) = bearing(j,1);
        end
    elseif S(j,1) < 0
        if S(j,2) < 0
            azi(j,1) = 360 - bearing(j,1);
        elseif S(j,2) > 0
            azi(j,1) = 360 - bearing(j,1);
        end
    else
        if S(j,1) == 0
            if S(j,2) == 0
                azi(j,1) = bearing(j,1);
            elseif S(j,2) > 0
                azi(j,1) = 0;
            elseif S(j,2) < 0
                azi(j,1) = 180;
            end
        elseif S(j,2) == 0
            if S(j,1) > 0
                azi(j,1) = 90;
            elseif S(j,1) < 0
                azi(j,1) = 270;
            end
        end
    end
end

%Calculate dip magnitude
dip = zeros(numRows, 1);
for k = 1:numRows
```

```
    dip(k,1) = rad2deg(asin(sqrt((U(k,1))^2 + (U(k,2))^2) / sqrt((U(k,1))^2 +
(U(k,2))^2 + (U(k,3))^2)));
end

%Calculate Dip Direction
dip_direction = zeros(numRows, 1);
for q = 1:numRows
    dip_direction(q,1) = azi(q,1) + 90;
    if dip_direction(q,1) > 360
        dip_direction(q,1) = dip_direction(q,1) - 360;
    end
end

%Output resulting matrices into new worksheet on new worksheet in input
%Excel file
warning('off','MATLAB:xlswrite:AddSheet');
CombinedResults = cat(2, azi, dip, dip_direction);
xlswrite(filename, CombinedResults, 'Strike, Dip, and Dip Direction');

%Create .txt file of new matricies
dlmwrite('output.txt',CombinedResults,'precision','%10f');

%The collection of data you now have is the strike azimuth
%array (azi), the dip magnitude array (dip), and the dip direction array
%(dip_direction)
```

View StrikeAndDipThreePoints.m