```
format shortG;

number_of_users = 10;

hard_userPositions = generateUserPosition(number_of_users);

numConfigs = 4;  % Number of configurations to simulate
numSats = 4;      % Number of satellites in each configuration

disp("Generated User Positions")
```

```
Generated User Positions
```

```
disp("          x            y            z  ")
```

```
         x           y           z
```

```
disp(hard_userPositions)
```

```
        -2156      -5263.7       2869.6
       4183.4       3771.8        -2977
      -1803.5       1046.1       6020.2
      -1868.7       6025.5       889.11
        -3869         2768       4237.7
      -22.928      -732.21      -6328.7
       -146.8      -141.04       6367.7
       6042.8      -748.34      -1874.6
       5214.5       1337.2      -3407.5
      -2306.3      -5313.2       2653.4
```

```
disp("Condition values and corresponding satellite positions")
```

```
Condition values and corresponding satellite positions
```

```
userError = zeros(numConfigs,1);
correctedError = zeros(numConfigs,1);
conditionValues = zeros(numConfigs,1);
satPositionNorm = zeros(numConfigs,1);
for i = 1:numConfigs
    noise = 4*rand(numSats-1,3);
    userPositions = zeros(number_of_users,3);
    correctedPositions = zeros(number_of_users,3);
    [unused_conditionValues, satConfigurations, unused_userPositions] = simulateSatelliteConfig
    satPositionNorm(i) = norm(transpose(satConfigurations{i})*satConfigurations{i});
    conditionValues(i) = computeCondition(satConfigurations{i},numSats);
    for k =1:number_of_users
        satRadius = mygenerateUserSatelliteRadii(satConfigurations{i},hard_userPositions(k,:),n
        userPositions(k,:) = NoiseTriang(satConfigurations{i},satRadius,numSats,noise);
        correctedPositions(k,:) = correctedUserPosition(satConfigurations{i},satRadius,numSats,
        userError(i) = norm(hard_userPositions - userPositions,'fro');
        correctedError(i) = norm(hard_userPositions - correctedPositions,'fro');
    end
    disp(['Configuration ' num2str(i) ': Condition = ' num2str(conditionValues(i,1))]);
```

```
    disp('Satellite Positions:');
    disp(satConfigurations{i});
    disp("Triangulated User Positions")
    disp("          x           y           z  ");
    disp(userPositions);
    disp(['Error: ' num2str(userError(i))]);
    disp("Corrected User Positions")
    disp("          x           y           z  ");
    disp(correctedPositions);
    disp(['Error: ' num2str(correctedError(i))]);
%     project_View(satConfigurations{i}, hard_userPositions, 'Simulated User Position'); hold o
%     project_View(satConfigurations{i}, userPositions,'Triangulated User Position'); hold on;
end
```

```
Configuration 1: Condition = 974.447
Satellite Positions:
       391.01      -1488.2      -8228.4
      -3026.1      -4376.8       6462.2
       6629.4       4315.4        -2739
       -577.4      -642.77       8326.3
Triangulated User Positions
          x           y           z
      -2153.3      -5266.9       2870.1
         4181       3774.8      -2977.4
      -1809.7       1051.6       6019.3
      -1873.8       6030.5       888.35
      -3873.8       2772.2       4237.1
      -15.752      -738.95      -6327.7
      -153.17      -135.14       6366.9
       6042.2      -747.08      -1874.7
       5214.1       1338.3      -3407.6
      -2303.2      -5316.8       2653.9
Error: 19.8738
Corrected User Positions
          x           y           z
        -2156      -5263.7       2869.6
       4183.4       3771.8        -2977
      -1803.5       1046.1       6020.2
      -1868.7       6025.5       889.11
        -3869         2768       4237.7
      -22.928      -732.21      -6328.7
       -146.8      -141.04       6367.7
       6042.8      -748.34      -1874.6
       5214.5       1337.2      -3407.5
      -2306.3      -5313.2       2653.4
Error: 5.4204e-11
Configuration 2: Condition = 1420.2633
Satellite Positions:
      -3216.9         7417      -2171.1
      -9.3449       6088.5       5744.9
      -3595.3       6126.2       4429.1
       329.98       6819.5       4843.5
Triangulated User Positions
          x           y           z
      -2157.3        -5256       2871.6
       4184.2       3766.7      -2978.6
      -1805.4       1059.6         6022
      -1868.1         6023       888.06
      -3869.7       2773.7       4238.4
      -20.526      -748.75      -6330.9
      -149.28      -124.18         6370
```

```
       6042.4       -746.3      -1874.7
         5215       1333.2      -3408.6
      -2307.5      -5306.2       2655.3
Error: 31.3026
Corrected User Positions
          x            y            z
       -2156      -5263.7       2869.6
       4183.4       3771.8        -2977
      -1803.5       1046.1       6020.2
      -1868.7       6025.5       889.11
        -3869         2768       4237.7
      -22.928      -732.21      -6328.7
       -146.8      -141.04       6367.7
       6042.8      -748.34      -1874.6
       5214.5       1337.2      -3407.5
      -2306.3      -5313.2       2653.4
Error: 5.1822e-11
Configuration 3: Condition = 13.5701
Satellite Positions:
       6600.6       1816.5      -4817.3
      -516.14        985.7       8296.7
      -3593.3      -1015.2      -7492.1
      -4330.9       6059.4      -3821.2
Triangulated User Positions
          x            y            z
      -2156.5      -5263.2         2870
         4184       3771.5      -2977.4
      -1803.1       1046.6       6020.2
      -1868.3       6025.1       888.82
      -3868.9       2768.1       4237.7
      -23.563      -733.03      -6328.7
      -146.26      -140.19       6367.8
       6043.4      -747.99      -1874.8
         5215       1337.1      -3407.8
      -2306.8      -5312.8       2653.8
Error: 2.3922
Corrected User Positions
          x            y            z
       -2156      -5263.7       2869.6
       4183.4       3771.8        -2977
      -1803.5       1046.1       6020.2
      -1868.7       6025.5       889.11
        -3869         2768       4237.7
      -22.928      -732.21      -6328.7
       -146.8      -141.04       6367.7
       6042.8      -748.34      -1874.6
       5214.5       1337.2      -3407.5
      -2306.3      -5313.2       2653.4
Error: 7.8301e-12
Configuration 4: Condition = 237.3607
Satellite Positions:
         6516      -4048.8        -3350
       4355.9       5421.9      -4658.7
       781.55      -224.21       8331.4
       2527.5       890.79       7930.4
Triangulated User Positions
          x            y            z
      -2157.7      -5263.3       2868.8
       4184.7         3771      -2976.6
      -1803.1       1045.9         6020
      -1866.8       6025.6       889.94
      -3868.1       2768.3         4238
      -23.304      -731.64      -6328.3
      -146.71      -141.55       6367.2
```

```
        6042.7       -749.34        -1875
          5215        1336.4      -3407.4
         -2308       -5312.8       2652.6
Error: 4.3089
Corrected User Positions
             x             y            z
         -2156       -5263.7       2869.6
        4183.4        3771.8        -2977
       -1803.5        1046.1       6020.2
       -1868.7        6025.5       889.11
         -3869          2768       4237.7
       -22.928       -732.21      -6328.7
        -146.8       -141.04       6367.7
        6042.8       -748.34      -1874.6
        5214.5        1337.2      -3407.5
       -2306.3       -5313.2       2653.4
Error: 2.4577e-11
```

```
function SatelliteRadii = mygenerateUserSatelliteRadii(P,U,numSatellites)
    SatelliteRadii = zeros(numSatellites,1);
    for i=1:1:numSatellites
        for j=1:1:3
            SatelliteRadii(i,1) = SatelliteRadii(i,1) + (P(i,j)-U(j))^2;
        end
        SatelliteRadii(i,1) = sqrt(SatelliteRadii(i,1));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [xx, new_cond] = iterativeRefinement(N, TOL, precision, A, b, x, n)
    % INPUT: N - number of iterations
    %        TOL - tolerance (approx precision)
    %        precision - digit precision
    %        A, b, x from Ax = b
    %        n - number of equations
    % OUTPUT:xx - new approx
    %        new condition number
    k = 1;
    r = zeros(n, 1);
    xx = zeros(3,1);
    while k <= N
        for i = 1:1:n
            r(i) = b(i) - sum(A(i,:).*transpose(x));
        end
        y = linsolve(A,r);

%          disp(x);
%          disp(y);
        xx = x + y;


        new_cond = norm(y)/norm(xx)*10^precision;
```

```
        if  norm(x-xx) < TOL
%             disp("new condition number: ")
%             disp(new_cond)
            break
        end

        k = k + 1;
        x = xx;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function userPosition = generateUserPosition(number_of_users)
    userPosition = zeros(number_of_users, 3);
    for k = 1:number_of_users
        r = 6371;
        azimuth = rand() * 2 * pi;
        elevation = rand() * pi - pi/2;
        x = r * cos(elevation) * cos(azimuth);
        y = r * cos(elevation) * sin(azimuth);
        z = r * sin(elevation);
        userPosition(k, :) = [x,y,z];
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function satPositions = generateSatellitePositions(numSats, earthRadius, satAltitude)
    % Randomly generate positions for satellites in orbit
    satPositions = zeros(numSats, 3);
    for i = 1:numSats
        % Random azimuth and elevation angles
        azimuth = rand() * 2 * pi;
        elevation = rand() * pi - pi/2;

        % Convert spherical coordinates to Cartesian coordinates
        r = earthRadius + satAltitude;
        x = r * cos(elevation) * cos(azimuth);
        y = r * cos(elevation) * sin(azimuth);
        z = r * sin(elevation);

        satPositions(i, :) = [x, y, z];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function  improvedUserPosition = correctedUserPosition(R,W,n,noise)
    A = zeros(n-1,3);
```

```
    y = zeros(n-1,1);
    for i=1:1:n-1
        for j=1:1:3
            A(i,j) = R(i+1,j)-R(1,j);
            y(i,1) = y(i,1)+(R(i+1,j)^2-R(1,j)^2);
        end
        y(i,1) = y(i,1)-(W(i+1)^2-W(1)^2);
    end
    A = 2*A;
    A = A + noise;
    At = transpose(A);
    userPositionn = (At*A)^(-1)*At*y;
    improvedUserPosition = iterativeRefinement(100, .1, 4, A, y, userPositionn, n-1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function condition = computeCondition(R,n)
    A = zeros(n-1,3);
    for i=1:1:n-1
        for j=1:1:3
            A(i,j) = R(i+1,j)-R(1,j);
        end
    end
    A = 2*A;
    condition = cond(transpose(A)*A,'fro');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function  [userPosition,condition] = NoiseTriang(R,W,n,noise)
    A = zeros(n-1,3);
    y = zeros(n-1,1);
    for i=1:1:n-1
        for j=1:1:3
            A(i,j) = R(i+1,j)-R(1,j);
            y(i,1) = y(i,1)+(R(i+1,j)^2-R(1,j)^2);
        end
        y(i,1) = y(i,1)-(W(i+1)^2-W(1)^2);
    end
    A = 2*A;
    condition = cond(transpose(A)*A,'fro');
    A = A + noise;
    userPositionn = (transpose(A)*A)^(-1)*transpose(A)*y;
    userPosition = transpose(userPositionn);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function project_View(satPositions, userPosition, titleInput)
```

```
earthRadius = 6371; % Earth's radius in kilometers
% Generate satellite positions


% Define the radius of the Earth's sphere
sphere_radius = earthRadius;

% Plotting the Earth's sphere
figure;
[xs, ys, zs] = sphere(50); % create a sphere
surf(sphere_radius * xs, sphere_radius * ys, sphere_radius * zs, 'FaceAlpha', 0.1, 'EdgeColor',
hold on;

% Plot the satellite positions on the sphere
plot3(satPositions(:, 1), satPositions(:, 2), satPositions(:, 3), 'ro', 'MarkerSize', 10, 'Mark

plot3(userPosition(:, 1), userPosition(:, 2), userPosition(:, 3), 'bo', 'MarkerSize', 10, 'Mark

% Set axis properties
axis equal; % equal aspect ratio
xlabel('X');
ylabel('Y');
zlabel('Z');
grid on;
title(titleInput);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [conditionValues, satConfigurations, userPositions] = simulateSatelliteConfigurations(
    % simulateSatelliteConfigurations - Simulates various satellite configurations
    % and calculates GDOP for each, also returns satellite positions for each configuration.
    %
    % Inputs:
    %   userPosition - A 1x3 vector representing the user's position (X, Y, Z)
    %   numConfigs - Number of satellite configurations to simulate
    %   numSats - Number of satellites in each configuration
    %
    % Output:
    %   gdopValues - Array of GDOP values for each configuration
    %   satConfigurations - Cell array containing satellite positions for each configuration

    % Initialize array to hold GDOP values and cell array for satellite positions
    conditionValues = zeros(numConfigs, 1);
    satConfigurations = cell(numConfigs, 1);
    userPositions = zeros(size(hard_userPositions,1),3);
    % Constants for the Earth (assuming satellites are in LEO)
    earthRadius = 6371; % in kilometers
    satAltitude = 2000; % Satellite altitude from the Earth's surface in kilometers
```

```
    % Iterate through each configuration
    for k = 1:numConfigs
        % Randomly generate satellite positions for this configuration
        satPositions = generateSatellitePositions(numSats, earthRadius, satAltitude);
        for i = 1:size(hard_userPositions,1)
            hard_userPosition = hard_userPositions(i,:);
            num_sats_used = numSats;
            [sat_radius, Satellite_pos] = generateUserSatelliteRadii(satPositions, hard_userPos
            [userPositions(i,:),conditionValues(k)] = NoiseTriang(Satellite_pos, sat_radius, nu

        end
        % Store the satellite positions
        satConfigurations{k} = satPositions;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```