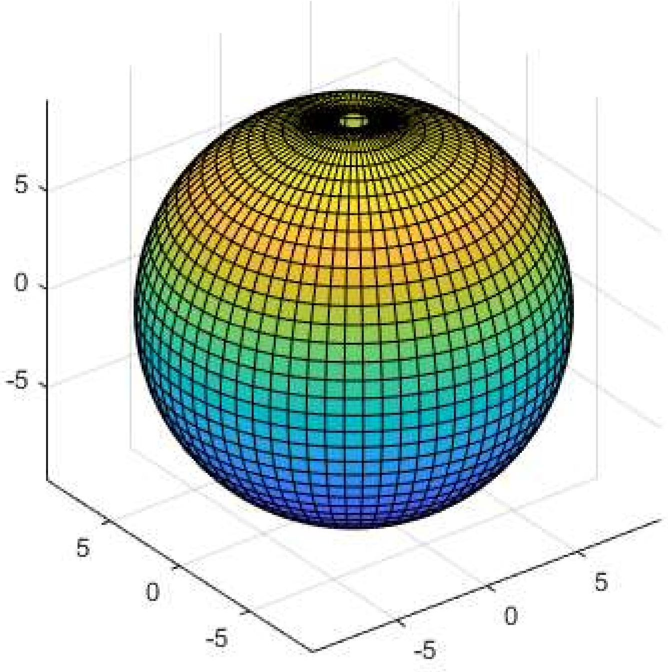
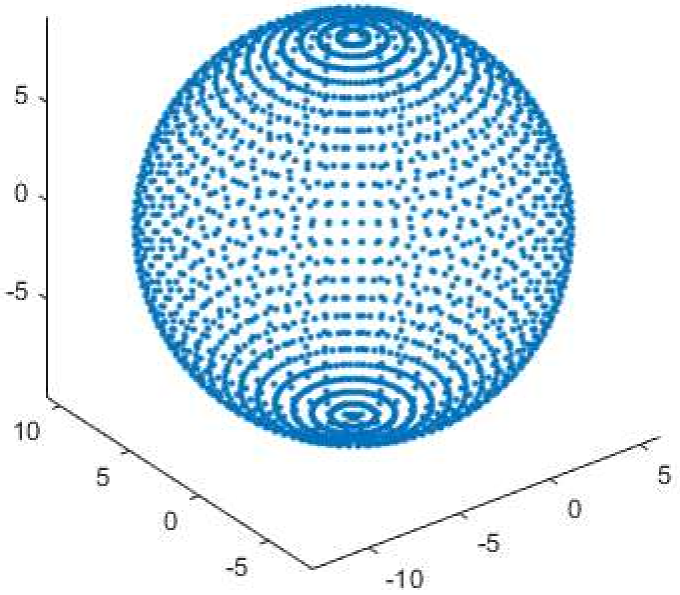
Contents

Test 1: define sphere in spherical coords and find parameters

```
close all; clear; clc

% full sphere
az = [0:5:359]; % azimuth range in degrees
el = [-89:5:89]; % elevation range in degrees
[AZ,EL] = meshgrid(az,el);

rval = 9.638; % define radius
RR = ones(size(AZ)).*rval;

zval = RR.*sind(EL);
xval = RR.*cosd(EL).*cosd(AZ);
yval = RR.*cosd(EL).*sind(AZ);

% visualize sphere
figure()
surf(xval, yval, zval)
axis equal

% create point cloud (3 x num points) from meshgrid
origin = [-3.5 1.4 -0.3]'; % define  origin
x = xval(:) + origin(1);
y = yval(:) + origin(2);
z = zval(:) + origin(3);

xyz = [
    x'
    y'
    z'
    ];

% visualize point cloud
figure()
plot3(x, y, z, '.')
axis equal

% find the origin and radius using method of least-squares
[cen, radius] = wlsqSphere(xyz);

disp('center of sphere')
disp('================')
disp('estimate    error')
[cen       cen-origin]


disp('radius of sphere')
disp('================')
disp('estimate    error')
[radius       radius-rval]
```

```
center of sphere
================
estimate    error

ans =

   -3.5000   -0.0000
    1.4000    0.0000
   -0.3000   -0.0000

radius of sphere
================
estimate    error

ans =

    9.6380         0
```



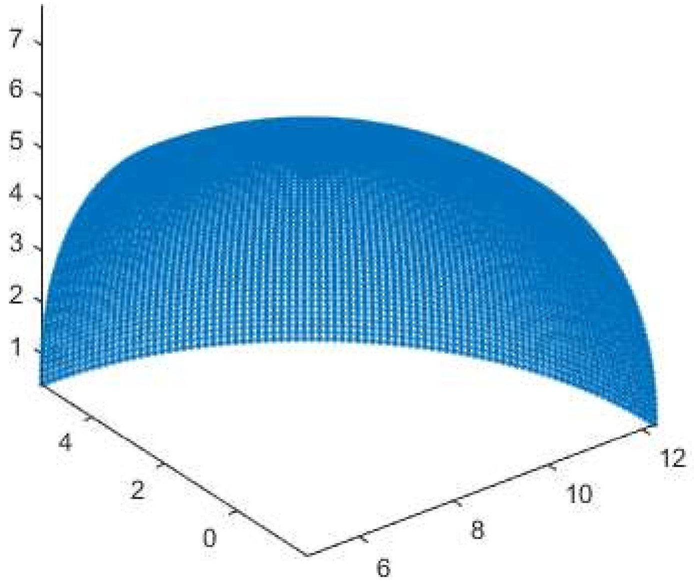


Test 2: use quarter sphere to test sampling skewed toward a direction

```
close all; clear; clc

% skew measurements to the one quadrant of sphere
```

```
az = [0:1:90]; % azimuth range in degrees
el = [0:1:90]; % elevation range in degrees
[AZ,EL] = meshgrid(az,el);

rval = 7.375; % define radius
RR = ones(size(AZ)).*rval;

zval = RR.*sind(EL);
xval = RR.*cosd(EL).*cosd(AZ);
yval = RR.*cosd(EL).*sind(AZ);

% create point cloud (3 x num points) from meshgrid
origin = [4.9 -1.96 0.42]'; % define  origin
x = xval(:) + origin(1);
y = yval(:) + origin(2);
z = zval(:) + origin(3);

xyz = [
    x'
    y'
    z'
    ];

% visualize point cloud
figure()
plot3(x, y, z, '.')
axis equal

% find the origin and radius using method of least-squares
[cen, radius] = wlsqSphere(xyz);

disp('center of sphere')
disp('================')
disp('estimate    error')
[cen       cen-origin]


disp('radius of sphere')
disp('================')
disp('estimate    error')
[radius       radius-rval]
```

```
center of sphere
================
estimate    error

ans =

    4.9000    0.0000
   -1.9600    0.0000
    0.4200    0.0000

radius of sphere
================
estimate    error

ans =
    7.3750   -0.0000
```

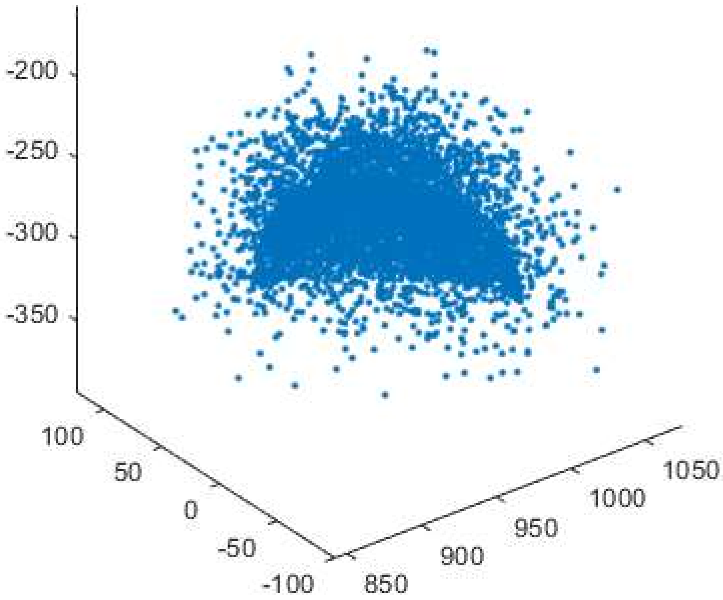


Test 3: add measurement noise to quarter sphere and weight the samples

```
close all; clear; clc

% skew measurements to the one quadrant of sphere
az = [5:1:85]; % azimuth range in degrees
el = [5:1:85]; % elevation range in degrees
[AZ,EL] = meshgrid(az,el);
npts = length(AZ(:));

rval = 112.5253; % define  radius
RR = ones(size(AZ)).*rval;

zval = RR.*sind(EL);
xval = RR.*cosd(EL).*cosd(AZ);
yval = RR.*cosd(EL).*sind(AZ);

% define size of noise
A = rval/5; % define the size at one standard-deviation (20% of radius)

% randomly "pollute" the samples with noise
headstails = coinflip(npts);
N = A .* randn(npts,3);
N(~headstails,:) = 0;

% create point cloud (3 x num points) from meshgrid
origin = [907.453 -42.345 -340.0234]'; % define  origin

x = xval(:) + origin(1) + N(:,1);
```

```
y = yval(:) + origin(2) + N(:,2);
z = zval(:) + origin(3) + N(:,3);

% compute weights of each sample
W = repmat(0.9,npts,1); % assign initial high-confidence weight
W(headstails) = 0.1; % assign low-confidence weights to polluted samples

xyz = [
    x'
    y'
    z'
    ];

% visualize point cloud
figure()
plot3(x, y, z, '.')
axis equal

% find the origin and radius using method of least-squares
[cen, radius] = wlsqSphere(xyz); % does not weight measurements
[cen_w, radius_w] = wlsqSphere(xyz,W); % apply weights to measurements

disp('center of sphere')
disp('================')
disp('estimate    weighted  actual')
[cen       cen_w       origin]

disp('radius of sphere')
disp('================')
disp('estimate    weighted  actual')
[radius       radius_w               rval]



function headsOrTails = coinflip(npts)
%COINFLIP Summary of this function goes here
%   Detailed explanation goes here
headsOrTails = rand(npts,1) >= 0.5;

end
```

```
center of sphere
================
estimate    weighted  actual

ans =

  947.1562  911.3483  907.4530
   -2.1234  -38.4123  -42.3450
 -288.3052 -334.8291 -340.0234

radius of sphere
================
estimate    weighted  actual

ans =

   64.2496  105.9949  112.5253
```



Published with MATLAB® R2022a