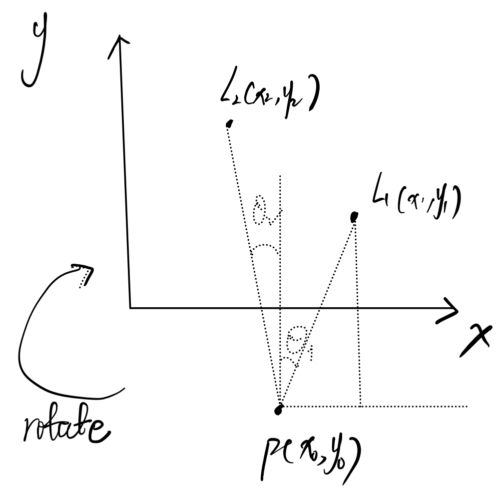
1.(a)



on map, we can observe that:

$$\frac{y_i-y_0}{x_i-x_0}=\cot(\theta_i)$$

$$\frac{y_i-y_0}{x_i-x_0}=\tan\left(\frac{\pi}{2}-\theta_i\right) \quad (x_i\neq x_0)$$

We have $\begin{cases} y_0=m_ix_0+b_i \\ y_i=m_ix_i+b_i \end{cases} \rightarrow \frac{y_i-y_0}{x_i-x_0}=m_i \quad (x_i\neq x_0)$

$$\Rightarrow \boxed{m_i=\frac{y_0-y_i}{x_0-x_i}=\tan\left(\frac{\pi}{2}-\theta_i\right)}$$

then $y_0=\frac{(y_0-y_i)x_0}{x_0-x_i}+b_i$

$$\boxed{b_i=\frac{\cancel{x_0y_0}-x_iy_0-\cancel{x_0y_0}+x_0y_i}{x_0-x_i}=\frac{x_0y_i-x_iy_0}{x_0-x_i}}$$

$$\left.\begin{aligned} m_i=\frac{y_0-y_i}{x_0-x_i} &\Rightarrow m_ix_i-y_i=m_ix_0-y_0 \\ b_i=\frac{x_0y_i-x_iy_0}{x_0-x_i} &\Rightarrow b_ix_i=b_ix_0-y_ix_0+x_iy_0 \end{aligned}\right\} \Rightarrow \begin{bmatrix} m_ix_i-y_i \\ b_ix_i \end{bmatrix}=\begin{bmatrix} m_i & -1 \\ b_i-y_i & x_i \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

(b) $m_i=T_i$

We have: $m_ix_i-y_i=m_ix_0-y_0$

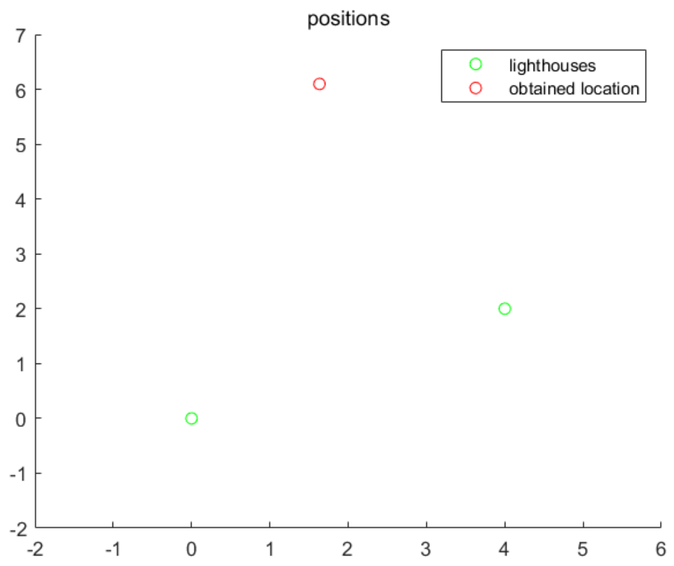
$T_ix_i-y_i=m_ix_0-y_0$

$$\begin{bmatrix} T_1x_1-y_1 \\ T_2x_2-y_2 \end{bmatrix}=\begin{bmatrix} T_1 & -1 \\ T_2 & -1 \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \underset{(T_1\neq T_2)}{\Longrightarrow} \boxed{\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}=\begin{bmatrix} T_1 & -1 \\ T_2 & -1 \end{bmatrix}^{-1}\begin{bmatrix} T_1x_1-y_1 \\ T_2x_2-y_2 \end{bmatrix}}$$

For 1(c) we use following codes.

```
clc;
clear all;
L1 = [0,0];
L2 = [4,2];
theta1 = -165/180*pi;
theta2 = 150/180*pi;
pos = get_pos(theta1,theta2,L1,L2);
hold on
plot([L1(1),L2(1)],[L1(2),L2(2)],'go')
plot(pos(1),pos(2),'ro')
axis([-2,6,-2,7])
title('positions')
legend("lighthouses","obtained location")
```

And the picture look like below:

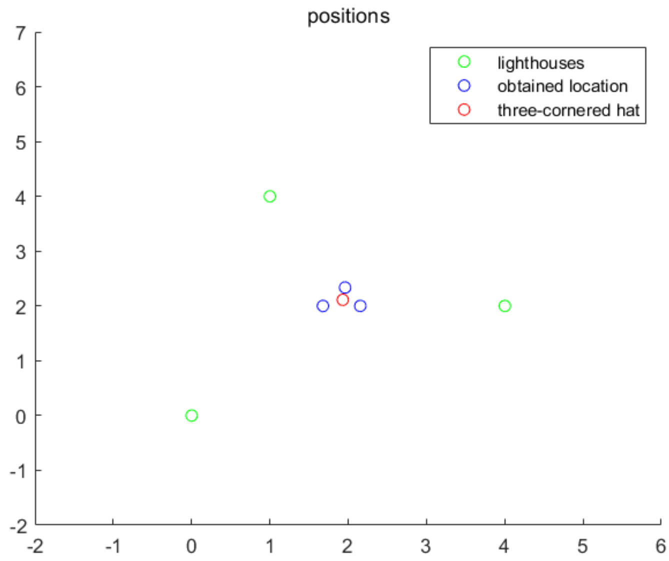


P1: position of light house and obtained location

For 1(d) we use following codes.

```
clc;
clear all;
L1 = [0,0];
L2 = [4,2];
L3 = [1,4];
theta1 = -140/180*pi;
theta2 = 90/180*pi;
theta3 = -30/180*pi;
pos1 = get_pos(theta1,theta2,L1,L2);
pos2 = get_pos(theta2,theta3,L2,L3);
pos3 = get_pos(theta1,theta3,L1,L3);
pos = [pos1, pos2,pos3];

hold on
plot([L1(1),L2(1),L3(1)],[L1(2),L2(2),L3(2)],'go')
plot(pos(1,:),pos(2,:),'bo')
plot(sum(pos(1,:))/3,sum(pos(2,:))/3,'ro')
axis([-2,6,-2,7])
title('positions')
legend("lighthouses","obtained location","three-cornered hat");
```

And the picture look like below:



P2: position of light house and obtained location

For 2(a) we use following codes. It include a function “fun_2a” .

The function code is shown below:

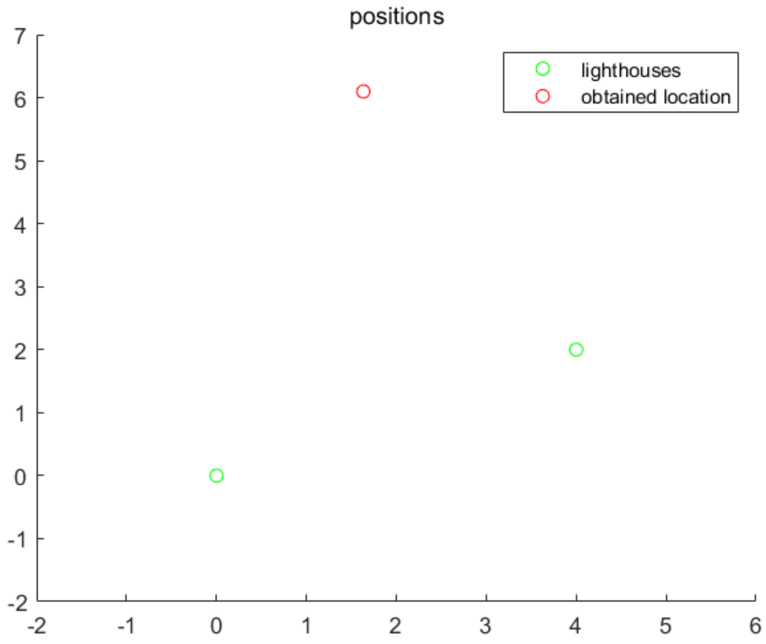
```
function fun = fun_2a(L1,L2,x,theta,phi)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...
    *norm(x-L2)*cos(theta)-norm(x-L1)^2;
eq2 = dot(L2-x,[0;1])-cos(phi)*norm(L2-x);
fun = eq1^2+eq2^2;
if(norm(x-L1)<0.1 || norm(x-L2)<0.1)
    fun = fun+10001/(norm(x-L1)*norm(x-L2)+0.01);
end
end
```

The other code is shown as follows:

```
clc;
clear all;
```

```
L1 = [0,0];
L2 = [4,2];
theta1 = -165/180*pi;
theta2 = 150/180*pi;
theta = 2*pi+theta1-theta2;
phi = -theta1+theta;
%pos = get_pos(theta1,theta2,L1,L2);
tt = fun_2a(L1,L2,[4,2],theta,phi);
fun = @(x) fun_2a(L1,L2,x,theta,phi);
fval_list = [];
x_all = [];
for i = -2:2:10
    for j = -2:2:10
x0 = [i,j];
options = optimoptions('fminunc','OptimalityTolerance',10e-6);
[x_ans,fval] = fminunc(fun,x0,options);
fval_list = [fval_list,fval];
x_all = [x_all;x_ans];
    end
end
fval_list = fval_list';
ind = find(fval_list==min(min(fval_list)));
x_final = x_all(ind,:);
hold on
plot([L1(1),L2(1)],[L1(2),L2(2)],'go')
plot(x_final(1),x_final(2),'ro')
axis([-2,6,-2,7])
title('positions')
legend("lighthouses","obtained location")
```

And the picture look like below:



P3: position of light house and obtained location

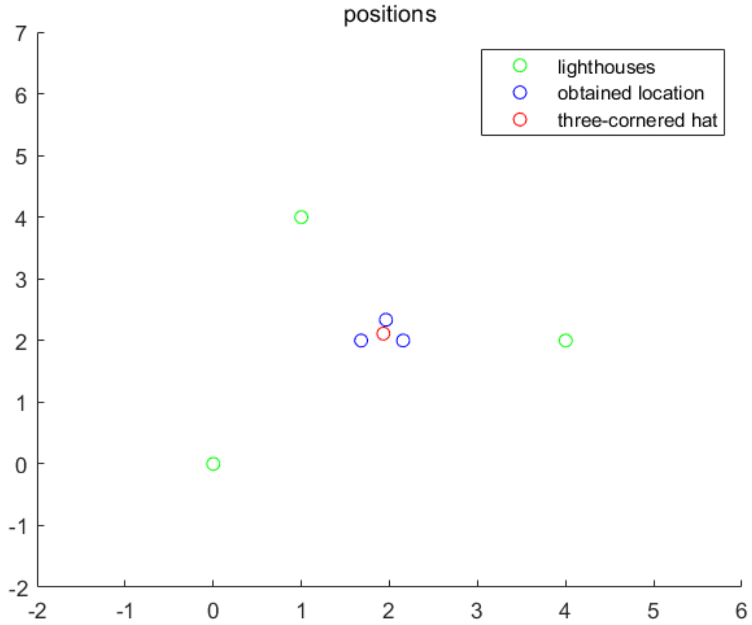
For 2(b) we use following codes.

```
clc;

clear all;

L1 = [0,0];

L2 = [4,2];

L3 = [1,4];

theta1 = -140/180*pi;

theta2 = 90/180*pi;

theta3 = -30/180*pi;


theta = 2*pi+theta1-theta2;

theta_2 = 2*pi+theta3-theta2; % 3 and 2

theta_3 = -theta1+theta3; %1 and 3


phi = -theta1+theta; %1 and 2

phi_1 = theta2; %3 and 2
```

```
phi_2 = theta3;%1 and 3
tt = fun_2a(L1,L2,[4,2],theta,phi);
fun = @(x) fun_2a(L1,L2,x,theta,phi);
fun2 = @(x) fun_2a(L3,L2,x,theta_2,phi_1);
fun3 = @(x) fun_2a(L1,L3,x,theta_3,phi_2);
fval_list = [];
x_all = [];
fval_list2= [];
x_all2 = [];
fval_list3 = [];
x_all3 = [];
for i = -2:2:10
    for j = -2:2:10
x0 = [i,j];
options = optimoptions('fminunc','OptimalityTolerance',10e-6);
[x_ans,fval] = fminunc(fun,x0,options);
fval_list = [fval_list,fval];
x_all = [x_all;x_ans];

%3 and 2
[x_ans2,fval2] = fminunc(fun2,x0,options);
fval_list2 = [fval_list2,fval2];
x_all2 = [x_all2;x_ans2];

%1 and 3
[x_ans3,fval3] = fminunc(fun3,x0,options);
fval_list3 = [fval_list3,fval3];
x_all3 = [x_all3;x_ans3];

    end
end
```

```
fval_list = fval_list';
ind = find(fval_list==min(min(fval_list)));
x_final_1 = x_all(ind,:);

fval_list2 = fval_list2';
ind = find(fval_list2==min(min(fval_list2)));
x_final_2 = x_all2(ind,:);

fval_list3 = fval_list3';
ind = find(fval_list3==min(min(fval_list3)));
x_final_3 = x_all3(ind,:);

x_lists_all = [x_final_1;x_final_2;x_final_3];

hold on
plot([L1(1),L2(1),L3(1)],[L1(2),L2(2),L3(2)],'go')
plot(x_lists_all(:,1),x_lists_all(:,2),'bo')
plot(sum(x_lists_all(:,1))/3,sum(x_lists_all(:,2))/3,'ro')
axis([-2,6,-2,7])
title('positions')
legend("lighthouses","obtained location","three-cornered hat")
```

And the picture look like below:



P4: position of light house and obtained location

For 3(a) we use following codes. And it includes a function "fun_3". The code of the function is shown below:

```
function fun = fun_3(L1,L2,x,theta,phi)


eq1 = dot(x-L1,L2-L1)+norm(x-L1)...
    *norm(x-L2)*cos(theta)-norm(x-L1)^2;
eq2 = dot(L2-x,[0,1,0])-cos(phi)*norm(L2-x);
fun = eq1^2+eq2^2;
%if(norm(x-L1)<0.1 || norm(x-L2)<0.1)
    %fun = fun+10001/(norm(x-L1)*norm(x-L2)+0.01);
%end
end
```

The other code is shown as follows:

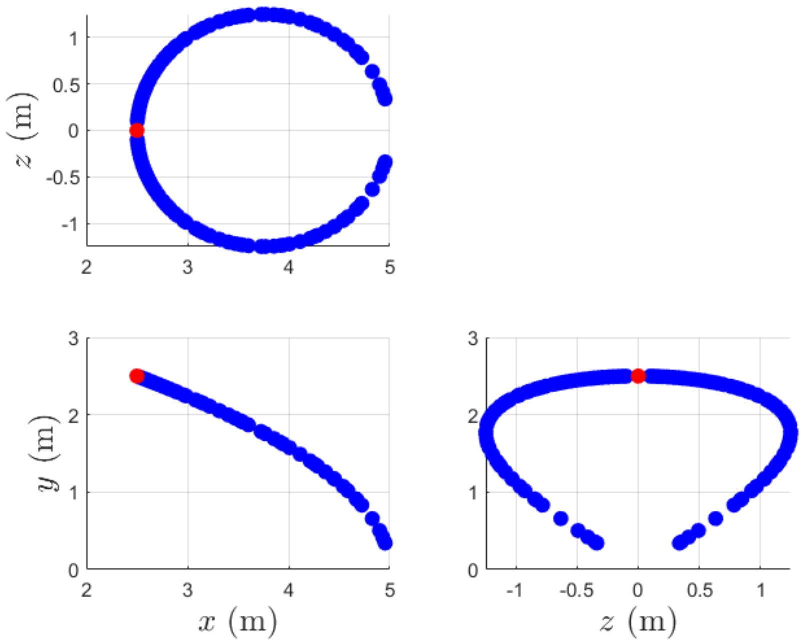
```
clc;
clear all;
```

```
L1 = [0,0,0];
L2 = [5,0,0];
theta = 0.5*pi;
phi = 135/180*pi;
%pos = get_pos(theta1,theta2,L1,L2);
fun = @(x) fun_3(L1,L2,x,theta,phi);
fval_list = [];
x_all = [];
for i = -5:2:5
    for j = -5:2:5
        for q = -5:2:5
x0 = [i,j,q];
options = optimoptions('fminunc','OptimalityTolerance',10e-16);
[x_ans,fval] = fminunc(fun,x0,options);
fval_list = [fval_list,fval];
x_all = [x_all;x_ans];
        end
    end
end
fval_list = fval_list';
ind = find(fval_list==min(min(fval_list)));
x_final = x_all(ind,:);
%3D plot
center = [2.5,2.5,0];
x = x_all(:,1);
y = x_all(:,2);
z = x_all(:,3);
figure(1)
scatter3(x,y,z,50,'b','fill')
hold on
scatter3(center(1), center(2), center(3),50,'r','fill')
```

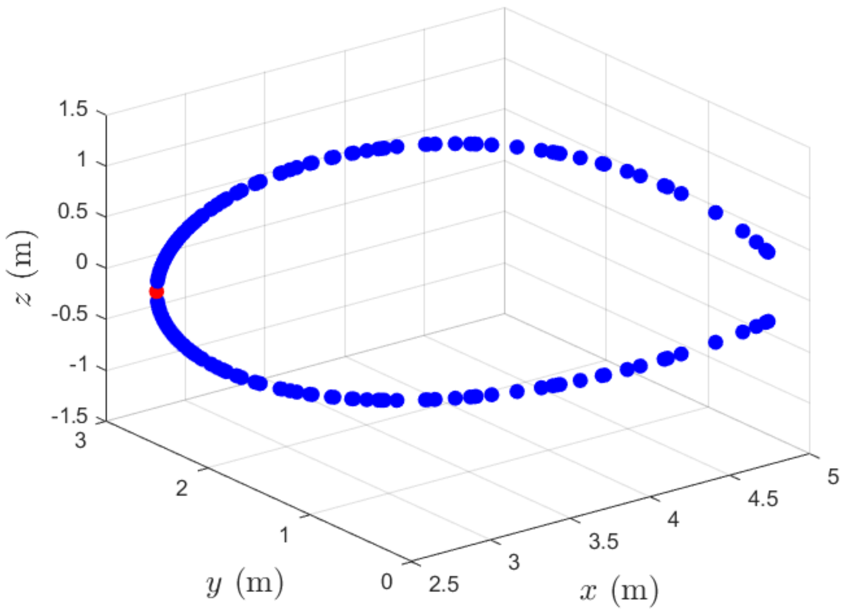
```
hold off
zlabel('$z$ (m)','interpreter','latex','fontsize',15)
ylabel('$y$ (m)','interpreter','latex','fontsize',15)
xlabel('$x$ (m)','interpreter','latex','fontsize',15)

%2D projections

figure(2)
subplot(2,2,1)
scatter(x,z,50,'b','fill')
hold on
scatter(center(1), center(3),50,'r','fill')
hold off
grid on
ylabel('$z$ (m)','interpreter','latex','fontsize',15)
%xlabel('$x$ (m)','interpreter','latex','fontsize',15)
subplot(2,2,3)
scatter(x,y,50,'b','fill')
hold on
scatter(center(1), center(2),50,'r','fill')
hold off
grid on
ylabel('$y$ (m)','interpreter','latex','fontsize',15)
xlabel('$x$ (m)','interpreter','latex','fontsize',15)
subplot(2,2,4)
scatter(z,y,50,'b','fill')
hold on
scatter(center(3), center(2),50,'r','fill')
hold off
%ylabel('$y$ (m)','interpreter','latex','fontsize',15)
xlabel('$z$ (m)','interpreter','latex','fontsize',15)
```

```
grid on

figure(3)
[Xsp,Ysp,Zsp] = sphere;
surf(2.5.*Xsp+ 2.5,2.5.*Ysp,2.5.*Zsp,'FaceAlpha',0.25);
hold on
scatter3(x,y,z,50,'b','fill')
grid on
hold off
axis equal
```

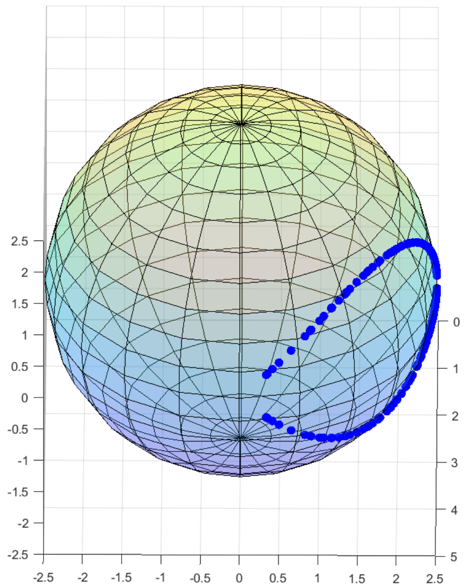
And the pictures look like below:



P5: 2D projection of solutions and the point



P6: solutions in actual 3D space



P7: sphere with solutions

We can observe that actually solutions are on the surface of a sphere. We get the solutions based on a subtended angle and a bearing angle. In 3D space, the same subtended angle within 2 lighthouses represents a sphere and the same bearing angle within a lighthouse represents a conical. Thus, these two constrictions form the solutions we have: a curve on the surface of a 3D sphere.

For 4 we use following codes. The codes includes a function call "fun_4". The function shows as follows:

```
function fun = fun_4(L1,L2,x,theta,phi1,star1,phi2,star2)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...
    *norm(x-L2)*cos(theta)-norm(x-L1)^2;
eq2 = dot(L2-x,star1)-cos(phi1)*norm(L2-x);
eq3 = dot(L2-x,star2)-cos(phi2)*norm(L2-x);
fun = eq1^2+eq2^2+eq3^2;
if(norm(x-L1)<0.1 || norm(x-L2)<0.1)
    fun = fun+10001/(norm(x-L1)*norm(x-L2)+0.01);
end
end
```

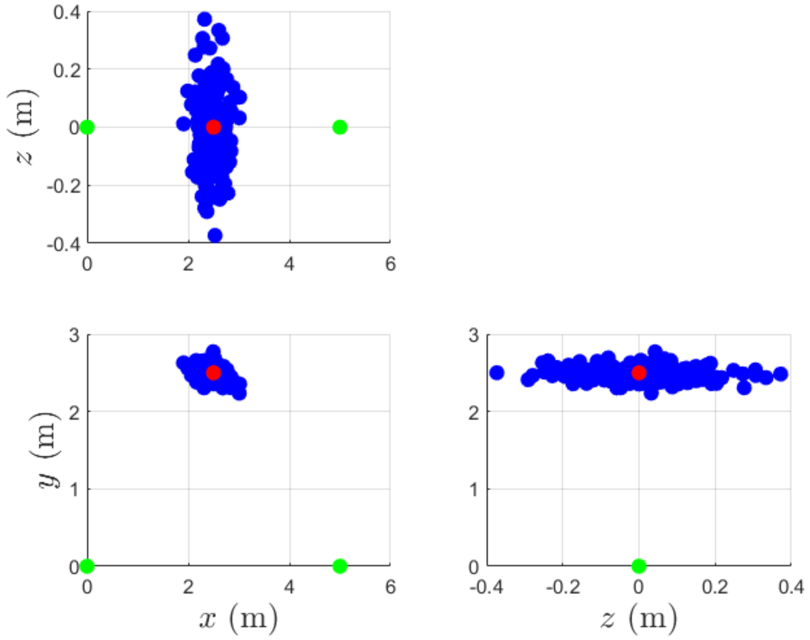
The other codes shows as follows:

```
clc;
clear all;
L1 = [0,0,0];
L2 = [5,0,0];
theta = 0.5*pi;
phi1 = 135/180*pi;
phi2 = 90/180*pi;
star1 = [0,1,0];
```

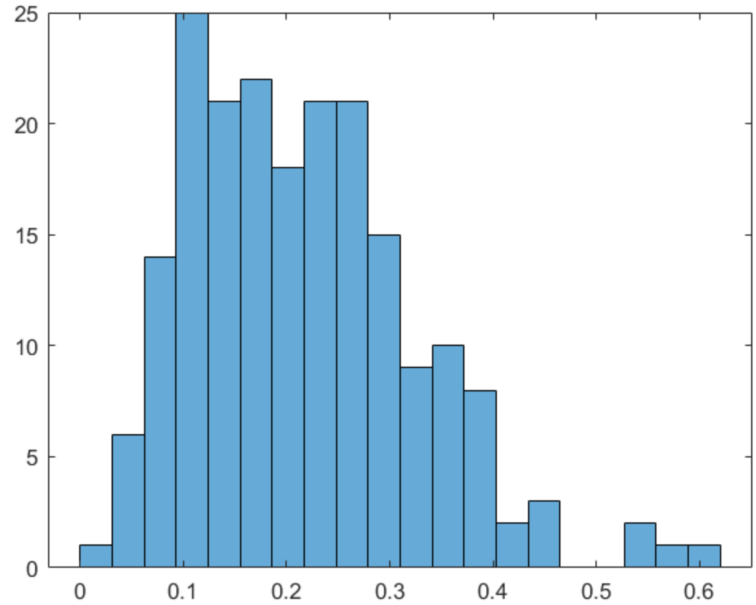
```
star2 = [0,0,1];

fun = @(x) fun_4(L1,L2,x,theta,phi1,star1,phi2,star2);
fval_list = [];
x_all = [];
for i = -5:2:5
    for j = -5:2:5
        for q = -5:2:5
x0 = [i,j,q];
options = optimoptions('fminunc','OptimalityTolerance',10e-16);
[x_ans,fval] = fminunc(fun,x0,options);
fval_list = [fval_list,fval];
x_all = [x_all;x_ans];
        end
    end
end
fval_list = fval_list';
ind = find(fval_list==min(min(fval_list)));
x_final = x_all(ind,:);
center = x_final;
house = [L1;L2];
x = house(:,1);
y = house(:,2);
z = house(:,3);
figure(1)
subplot(2,2,1)
scatter(x,z,50,'g','fill')
hold on
scatter(center(1), center(3),50,'r','fill')
hold off
grid on
```

```
ylabel('$z$ (m)','interpreter','latex','fontsize',15)
%xlabel('$x$ (m)','interpreter','latex','fontsize',15)
subplot(2,2,3)
scatter(x,y,50,'g','fill')
hold on
scatter(center(1), center(2),50,'r','fill')
hold off
grid on
ylabel('$y$ (m)','interpreter','latex','fontsize',15)
xlabel('$x$ (m)','interpreter','latex','fontsize',15)
subplot(2,2,4)
scatter(z,y,50,'g','fill')
hold on
scatter(center(3), center(2),50,'r','fill')
hold off
%ylabel('$y$ (m)','interpreter','latex','fontsize',15)
xlabel('$z$ (m)','interpreter','latex','fontsize',15)
grid on
```

And the picture looks like below:



P8: projection of obtained locations



P9: histogram of position fix errors

We get:

deviation of error

0.1089

mean of error

0.2152