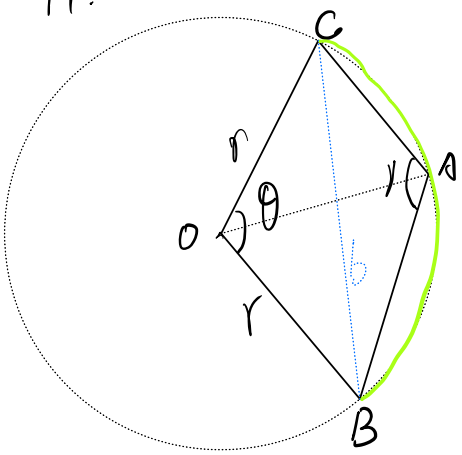
P1:



Considering B, C as fixed point, $0° < \gamma < 180°$

$OA = OC = OB = r$

$\angle OCA = \angle OAC$, $\angle OAB = \angle OBA$

$\angle COB + \angle CAB + \angle OCA + \angle OBA = 360°$

$\theta + \gamma + (\angle OAC + \angle OAB) = 360°$

$\theta + 2\gamma = 360°$

$\gamma = \frac{360° - \theta}{2} = 180° - \frac{\theta}{2}$

$\Rightarrow$ A could be any where on the arc $\overset{\frown}{BC}$

A is located on a portion of a circle that contains points A, B, C

$\theta = 360° - 2\gamma$

$$r = \frac{\frac{b}{2}}{\sin\frac{\theta}{2}} = \frac{b}{2\sin(180° - \gamma)}$$

For 2a the codes has 1 function:"fun_2a"

The function is listed as follow:

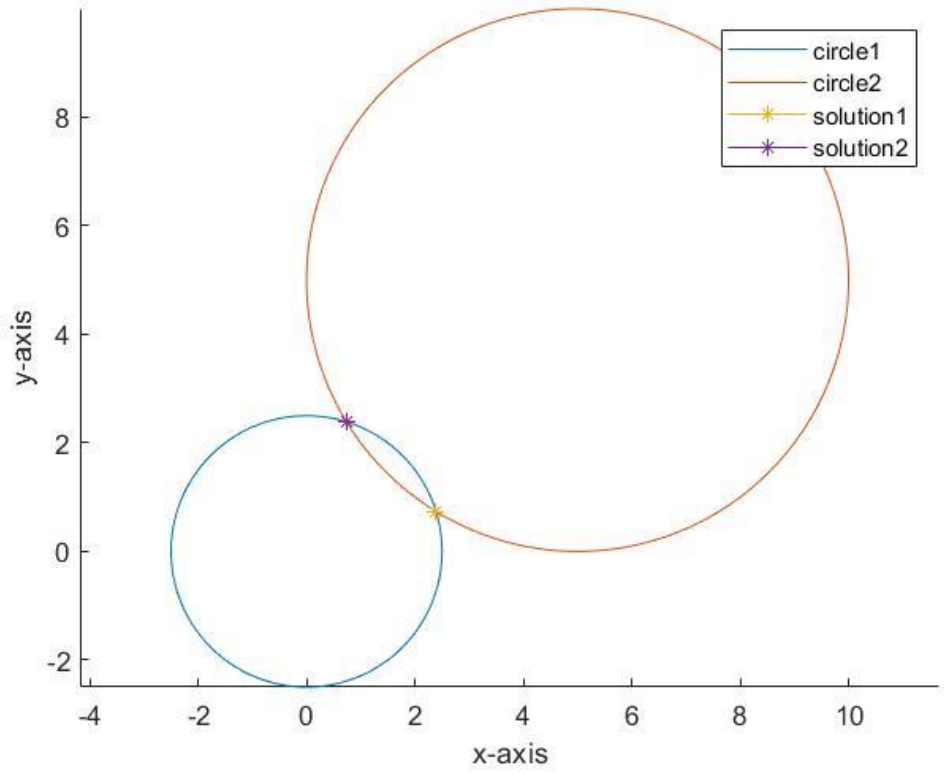
```
function fun = fun_2a(x,L1,L2,R1,R2)
fun = [(x(1)-L1(1))^2+(x(2)-L1(2))^2-R1^2;
      (x(1)-L2(1))^2+(x(2)-L2(2))^2-R2^2];
fun = fun(1)^2+fun(2)^2;
end
```

The code is listed as follow:

```
clc;
clear all;
L1 = [0,0];
L2 = [5,5];
R1=2.5;
R2=5;
fun = @(x) fun_2a(x,L1,L2,R1,R2);
x0 = [10,0];
options = optimoptions('fminunc','OptimalityTolerance',10e-6);
[x1,fval] = fminunc(fun,x0,options);
x0_2=[0,10];
```

```
[x2,fval2]=fminunc(fun,x0_2,options);


theta = linspace(-pi,pi,1000);
circle1_x = L1(1)+R1*cos(theta);
circle1_y = L1(2)+R1*sin(theta);
circle2_x = L2(1)+R2*cos(theta);
circle2_y = L2(2)+R2*sin(theta);
figure(1)
hold on
plot(circle1_x,circle1_y);
plot(circle2_x,circle2_y);
plot(x1(1),x1(2),'Marker','*');
plot(x2(1),x2(2),'Marker','*');
xlabel('x-axis')
ylabel('y-axis')
axis equal
legend('circle1','circle2','solution1','solution2');
hold off
```

The picture looks as following:



Picture1: circles and solutions

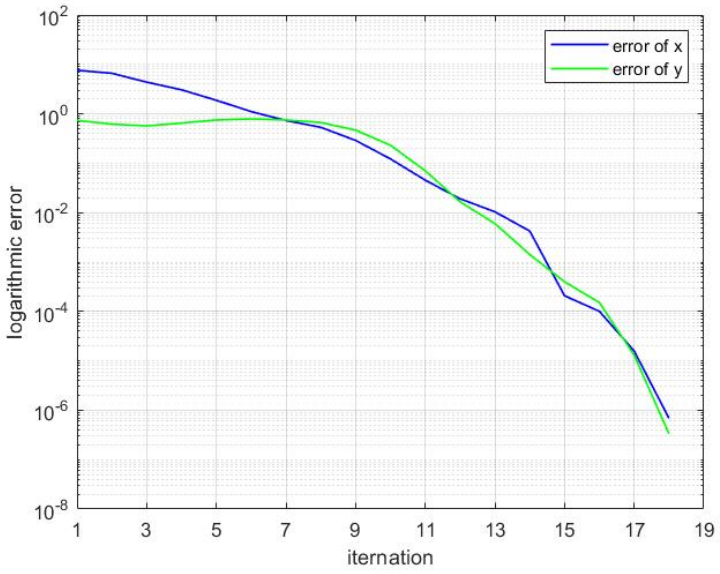
For 2b the codes are listed as follows:

```
clc;

clear all;

global h;

h.x=[];

h.fval=[];

L1 = [0,0];

L2 = [5,5];

R1=2.5;
```

```
R2=5;

fun = @(x) fun_2a(x,L1,L2,R1,R2);

x0 = [10,0];

options = optimoptions('fminunc','OptimalityTolerance',10e-

16,'OutputFcn',@outfun);

[x1,fval] = fminunc(fun,x0,options);

x0_2=[0,10];

%[x2,fval2]=fminunc(fun,x0_2,options);


figure(1)

iter = 1:length(h.x);

xxx = h.x-x1;

semilogy(iter,abs(xxx(:,1)),'b','linewidth',1);

hold all

semilogy(iter,abs(xxx(:,2)),'g','linewidth',1);

xlim([min(iter),max(iter)]);

xticks(1:2:max(iter));

xlabel('iternation');

ylabel('logarithmic error');

legend('error of x','error of y');

grid on;
```

```
function stop = outfun(x,optimValues,state)
    global h;
    stop = false;
    switch state
        case 'iter'
            h.fval = [h.fval; optimValues.fval];
            h.x=[h.x;x];
        otherwise
    end
end
```

The picture looks as following:



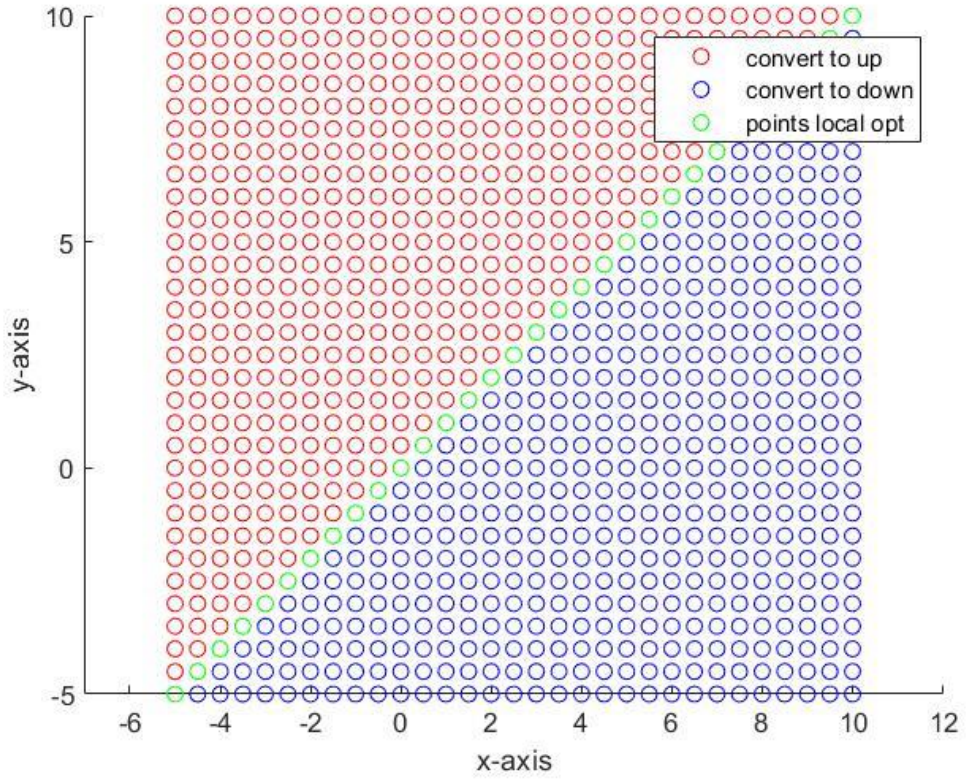
Picture2: error versus iteration plot

For 2c the codes are listed as follows:

```
clc;

clear all;

L1 = [0,0];

L2 = [5,5];

R1=2.5;

R2=5;

fun = @(x) fun_2a(x,L1,L2,R1,R2);

x0 = [10,0];

options = optimoptions('fminunc','OptimalityTolerance',10e-11);

[x1,fval] = fminunc(fun,x0,options);

x0_2=[0,10];

[x2,fval2]=fminunc(fun,x0_2,options);

p = -5:0.5:10;

q=10:-0.5:-5;


up_one = [];

down_one = [];

local_opt= [];

cc = [];

for i=1:length(p)

    for j=1:length(q)
```

```
            dot = [p(i),q(j)];
            [x,fval] = fminunc(fun,dot,options);
            if(norm(x2-x)<10e-6)
                up_one = [up_one;dot];
            else if(norm(x1-x)<10e-6)
                down_one = [down_one;dot];
                else
                    local_opt = [local_opt;dot];
                end
            end
        end
    end
    hold all
    scatter(up_one(:,1),up_one(:,2),[],'r');
    scatter(down_one(:,1),down_one(:,2),[],'b');
    scatter(local_opt(:,1),local_opt(:,2),[],'g');
    xlabel('x-axis')
    ylabel('y-axis')
    axis equal
    legend('convert to up','convert to down','points local opt');
```

The picture looks as following:



Picture3: converge positions

For 3a we use a function:"fun_3a", codes are listed as following:

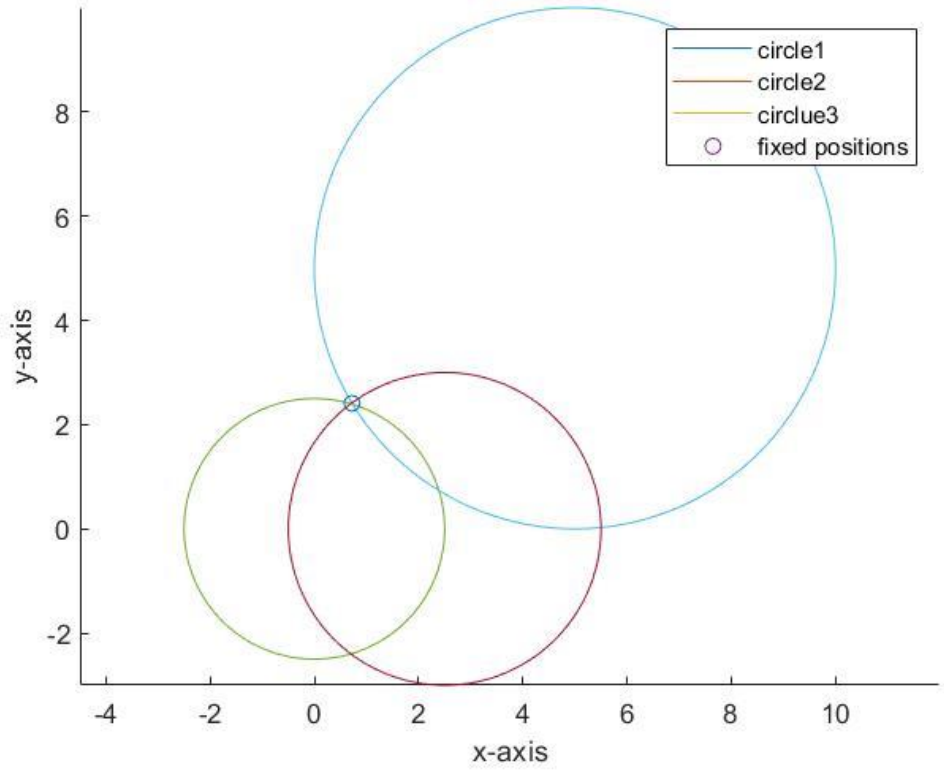
```
function fun = fun_3a(x,L1,L2,L3,R1,R2,R3)

    fun = [(x(1)-L1(1))^2+(x(2)-L1(2))^2-R1^2;

        (x(1)-L2(1))^2+(x(2)-L2(2))^2-R2^2;

        (x(1)-L3(1))^2+(x(2)-L3(2))^2-R3^2];

    fun = fun(1)^2+fun(2)^2+fun(3)^2;

end
```

the main codes are listed as follows:

```
clc;
clear all;
L1 = [0,0];
L2 = [5,5];
L3=[2.5,0];
R1=2.5;
R2=5;
R3=3;
fun = @(x) fun_3a(x,L1,L2,L3,R1,R2,R3);
x0 = [10,0];
options = optimoptions('fminunc','OptimalityTolerance',10e-16);
[x1,fval] = fminunc(fun,x0,options);
x0_2=[0,10];
p = -5:1:10;
q=10:-1:-5;
solutions = [];
for i=1:length(p)
    for j=1:length(q)
        dot = [p(i),q(j)];
        [sol,fval] = fminunc(fun,dot,options);
        solutions=[solutions;sol];
    end
```

```
end


theta = linspace(-pi,pi,1000);
circle1_x = L1(1)+R1*cos(theta);
circle1_y = L1(2)+R1*sin(theta);
circle2_x = L2(1)+R2*cos(theta);
circle2_y = L2(2)+R2*sin(theta);
c3_x = L3(1)+R3*cos(theta);
c3_y = L3(2)+R3*sin(theta);
figure(1)
hold all
plot(circle1_x,circle1_y);
plot(circle2_x,circle2_y);
plot(c3_x,c3_y);
scatter(solutions(:,1),solutions(:,2));


xlabel('x-axis')
ylabel('y-axis')
axis equal
legend('circle1','circle2','circlue3','fixed positions');
hold off
```

The picture looks as follows:



Picture4: circles and fixed points

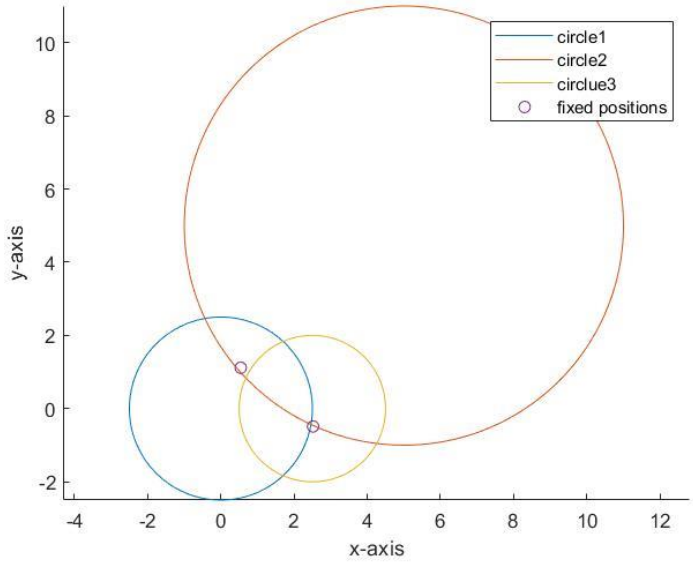
For 3b the codes are listed as follows:

```
clc;

clear all;

L1 = [0,0];

L2 = [5,5];

L3=[2.5,0];

R1=2.5;

R2=6;
```

```
R3=2;

fun = @(x) fun_3a(x,L1,L2,L3,R1,R2,R3);

x0 = [10,0];

options = optimoptions('fminunc','OptimalityTolerance',10e-16);

[x1,fval] = fminunc(fun,x0,options);

x0_2=[0,10];

[x2,fval] = fminunc(fun,x0_2,options);

p = -5:5:10;

q=10:-5:-5;

solutions = [];

for i=1:length(p)

    for j=1:length(q)

        dot = [p(i),q(j)];

        [sol,fval] = fminunc(fun,dot,options);

        solutions=[solutions;sol];

    end

end

theta = linspace(-pi,pi,1000);

circle1_x = L1(1)+R1*cos(theta);

circle1_y = L1(2)+R1*sin(theta);

circle2_x = L2(1)+R2*cos(theta);

circle2_y = L2(2)+R2*sin(theta);
```

```
c3_x = L3(1)+R3*cos(theta);

c3_y = L3(2)+R3*sin(theta);

figure(1)

hold all

plot(circle1_x,circle1_y);

plot(circle2_x,circle2_y);

plot(c3_x,c3_y);

scatter(solutions(:,1),solutions(:,2));

xlabel('x-axis')

ylabel('y-axis')

axis equal

legend('circle1','circle2','circlue3','fixed positions');

hold off
```

The picture looks as follows



Picture5: circles and fixed positions

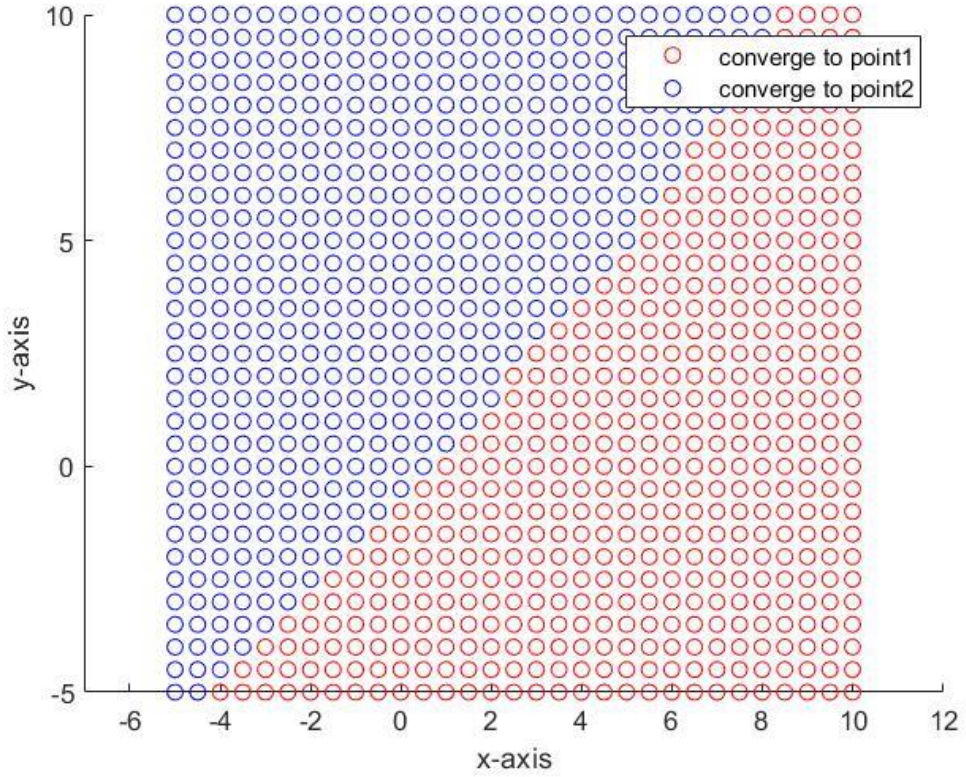
For 3c the codes are listed as follows:

```
clc;

clear all;

L1 = [0,0];

L2 = [5,5];

L3=[2.5,0];

R1=2.5;

R2=6;

R3=2;

fun = @(x) fun_3a(x,L1,L2,L3,R1,R2,R3);

x0 = [10,0];

options = optimoptions('fminunc','OptimalityTolerance',10e-12);

[x1,fval_1] = fminunc(fun,x0,options);
```

```
x0_2=[0,10];
[x2,fval_2] = fminunc(fun,x0_2,options);
x0_2=[0,10];
p = -5:0.5:10;
q=10:-0.5:-5;
sols1 = [];
sols2 = [];
for i=1:length(p)
    for j=1:length(q)
        dot = [p(i),q(j)];
        [sol,fval] = fminunc(fun,dot,options);
        if(norm(sol-x1)<10e-6)
            sols1 = [sols1;dot];
        else
            sols2 = [sols2;dot];
        end
    end
end
hold all
scatter(sols1(:,1),sols1(:,2),[],'r');
scatter(sols2(:,1),sols2(:,2),[],'b');
xlabel('x-axis')
```

```
ylabel('y-axis')

axis equal

legend('converge to point1','converge to point2');
```

The picture looks as following:



Picture6: converge positions

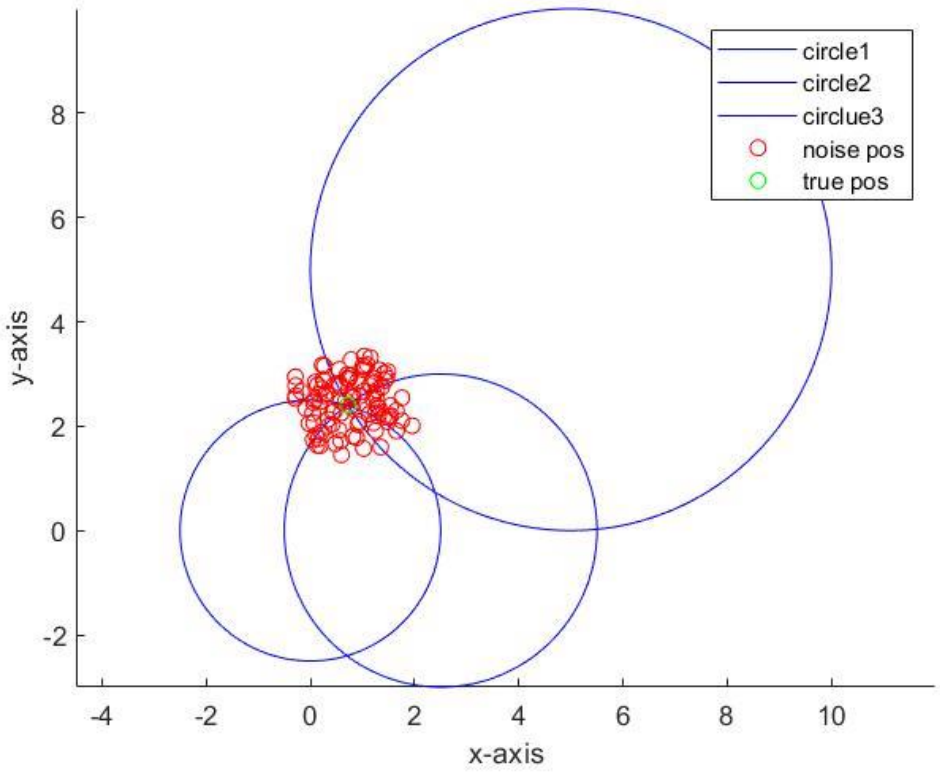
For 4a the codes are listed as follows:

```
clc;

clear all;

L1 = [0,0];
```

```
L2 = [5,5];

L3=[2.5,0];

R1=2.5;

R2=5;

R3=3;

fun = @(x) fun_3a(x,L1,L2,L3,R1,R2,R3);

x0 = [0,4];

options = optimoptions('fminunc','OptimalityTolerance',10e-6);

[x1,fval] = fminunc(fun,x0,options);


theta = linspace(-pi,pi,1000);

circle1_x = L1(1)+R1*cos(theta);

circle1_y = L1(2)+R1*sin(theta);

circle2_x = L2(1)+R2*cos(theta);

circle2_y = L2(2)+R2*sin(theta);

c3_x = L3(1)+R3*cos(theta);

c3_y = L3(2)+R3*sin(theta);

figure(1)

hold all

plot(circle1_x,circle1_y,'b');

plot(circle2_x,circle2_y,'b');

plot(c3_x,c3_y,'b');
```

```
xlabel('x-axis')

ylabel('y-axis')

axis equal


nosie_x = zeros(100,2);

a=2;

for i=1:1:100

    R1_new = R1+a*(rand(1)-0.5);

    R2_new = R2+a*(rand(1)-0.5);

    R3_new = R3+a*(rand(1)-0.5);

    fun = @(x) fun_3a(x,L1,L2,L3,R1_new,R2_new,R3_new);

    options = optimoptions('fminunc','OptimalityTolerance',10e-6);

    [x_new,fval] = fminunc(fun,x0,options);

    noise_x(i,:) = x_new;

end

scatter(noise_x(:,1),noise_x(:,2),[],'r');

scatter(x1(1),x1(2),'g');

legend('circle1','circle2','circlue3','noise pos','true pos');
```

The picture looks as follows:



Picture7: true and noised pos

For 4b the codes are listed as follows:

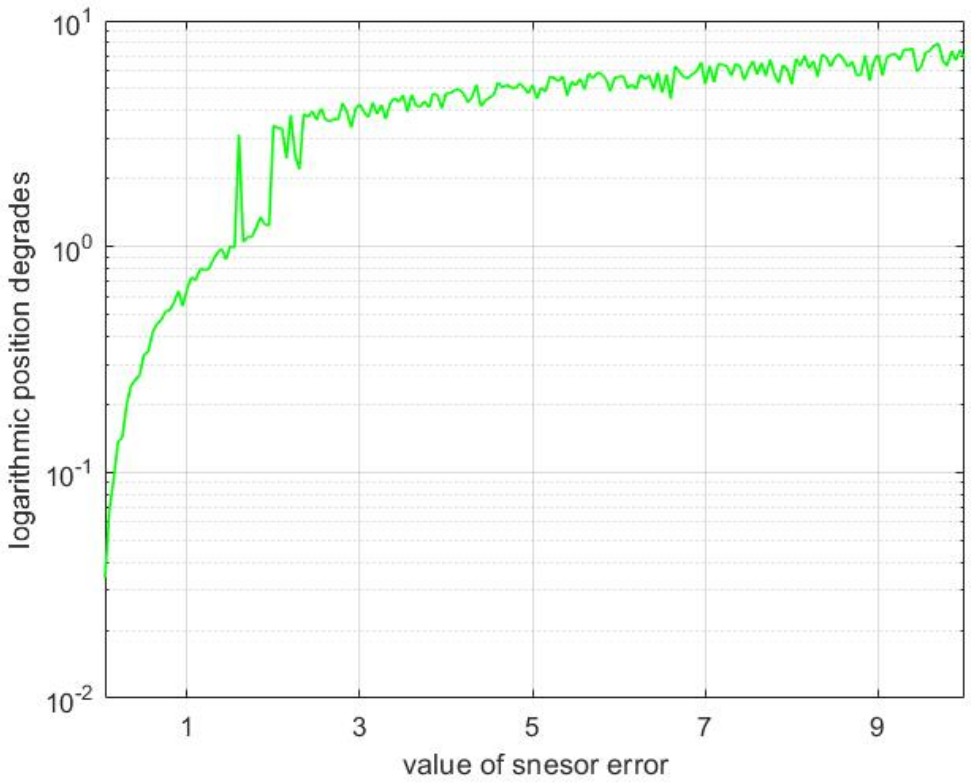
```
clc;

clear all;

L1 = [0,0];

L2 = [5,5];

L3=[2.5,0];

R1=2.5;

R2=5;

R3=3;
```

```
fun = @(x) fun_3a(x,L1,L2,L3,R1,R2,R3);

x0 = [0,4];

options = optimoptions('fminunc','OptimalityTolerance',10e-6);

[x1,fval] = fminunc(fun,x0,options);


%{

theta = linspace(-pi,pi,1000);

circle1_x = L1(1)+R1*cos(theta);

circle1_y = L1(2)+R1*sin(theta);

circle2_x = L2(1)+R2*cos(theta);

circle2_y = L2(2)+R2*sin(theta);

c3_x = L3(1)+R3*cos(theta);

c3_y = L3(2)+R3*sin(theta);

figure(1)

hold all

plot(circle1_x,circle1_y,'b');

plot(circle2_x,circle2_y,'b');

plot(c3_x,c3_y,'b');

xlabel('x-axis')

ylabel('y-axis')

axis equal

%}
```

```
x_a=[];
far = [];
for a = 0.05:0.05:10
    most_dis = 0;
    x_a = [x_a,a];
    for i=1:1:100
        R1_new = R1+a*(rand(1)-0.5);
        R2_new = R2+a*(rand(1)-0.5);
        R3_new = R3+a*(rand(1)-0.5);
        fun = @(x) fun_3a(x,L1,L2,L3,R1_new,R2_new,R3_new);
        options = optimoptions('fminunc','OptimalityTolerance',10e-6);
        [x_new,fval] = fminunc(fun,x0,options);
        if(norm(x_new-x1)>most_dis)
            most_dis = norm(x_new-x1);
        end
    end
    far = [far;most_dis];
end

figure(1)
```

```
semilogy(x_a,far,'g','linewidth',1);

xlim([min(x_a),max(x_a)]);

xticks(1:2:max(x_a));

xlabel('value of snesor error');

ylabel('logarithmic position degrades');

grid on;
```

The picture looks as follows:



Picture8: degrades versus error