2. Consider the problem: $min\ f(\underline{x})\ s.t.\ h(\underline{x}) \geq 0$ where $f(\underline{x}) = (x_1-1)^2 + 2(x_2-2)^2$ and $h(\underline{x}) = [1 - x_1{}^2 - x_2^2, x_1 + x_2]^T$.
(a) Plot the contour of f(x) and the feasible set on one single figure, i.e., overlay the feasible set on the contour plot of f(x);
(b) Find a solution to the problem using the natural logarithmic barrier function, i.e., the barrier function is -log($h_1(\underline{x})$) - log($h_2(\underline{x})$). Use initialization vector $[0.5\ 0.5]^T$ and the initial penalty parameter equal to 1 and reduce it by ½ in each iteration. Use a stopping threshold of 0.002;
(c) In a 2-D figure, plot the trajectory (i.e., the values connected by lines with arrows) of the computed solution vector as the number of iteration progresses.

```
clf
syms x1 x2;
[X,Y] = meshgrid(0:0.1:10,0:0.1:10);
%fcontour(@(x1,x2) (x1-1)^2+2*(x2-2)^2)

figure(1)
F =(x1-1).^2 +2*(x2-2).^2;
fcontour(F,'linewidth',1);
hold on
H1x=x1^2+x2^2>=1;
H2x=x1+x2>=0;

s1 = solve(H1x,[x1])     %Solving the equations
```

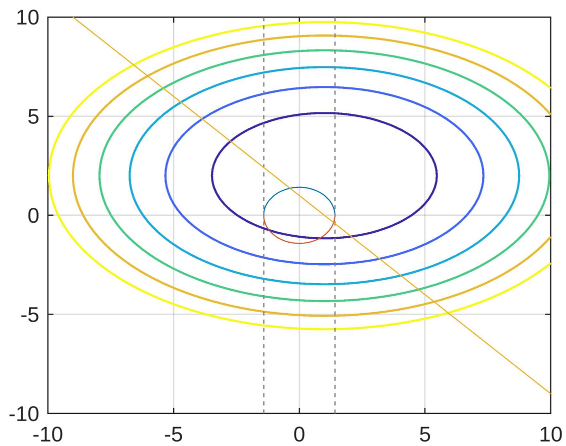
s1 =

$$\begin{pmatrix} \sqrt{2 - x_2{}^2} \\ -\sqrt{2 - x_2{}^2} \end{pmatrix}$$

```
s2 = solve(H2x,[x2])
```

s2 = $1 - x_1$

```
fplot(s1)                %Plottig values of solved H1x and H2x equations
fplot(s2)
xlim([-10 10])
ylim([-10 10])
hold on
grid on
hold off
```



10
5
0
-5
-10
-10
-5
0
5
10

(b) Find a solution to the problem using the natural logarithmic barrier function, i.e., the barrier function is -log(h1()) - log(h2()). Use initialization vector [0.5 0.5]T and the initial penalty parameter equal to 1 and reduce it by ½ in each iteration. Use a stopping threshold of 0.002;

```
clf
format short;
syms x1 x2;
F=(x1-1)^2+2*(x2-2)^2; %Given function
H1x=1-x1^2-x2^2;
H2x=x1+x2;
x=[0.5 0.5];
e=0.002;
c=1;
true=1;
epoch=0;
while(true)
    epoch=epoch+1;

    B=(F+c*(-log(H1x)-log(H2x))^2); % Barrier function
    DB=[diff(B,x1);diff(B,x2)];

    Hessien=[diff(DB(1),x1) diff(DB(1),x2); %Finding Hessien values
      diff(DB(2),x1) diff(DB(2),x2)];

      DB_val=subs(DB,x1,x(1));
      DB_val=subs(DB_val,x2,x(2));
      Hessien_val=subs(Hessien,x1,x(1));
      Hessien_val=subs(Hessien_val,x2,x(2));


alpha=vpa((DB_val'*DB_val)/(DB_val'*Hessien_val*DB_val));

 eucledian=norm(alpha*DB_val);
 if(e<=eucledian);
  F_val=subs(F,x1,x(1));
  F_val=subs(F_val,x2,x(2));

  stepsize1=alpha*DB_val(1);
  stepsize2=alpha*DB_val(2);

  x1_new=vpa(x(1)-stepsize1);
  x2_new=vpa(x(2)-stepsize2);
  x=[x1_new x2_new];

  c=c*0.5;
  result_matrix= [double(epoch) double(x(1)) double(x(2)) double(F_val)];

 else

 true=0;
```

```
 end
 end
%%
ResultTable(epoch,:)=result_matrix
```

```
ResultTable = 11×4
    1.0000    0.4772    0.7718    4.7500
    2.0000    0.4545    0.7445    3.2904
    3.0000    0.4251    0.8083    3.4498
    4.0000    0.4098    0.8418    3.1711
    5.0000    0.4070    0.8671    3.0310
    6.0000    0.4059    0.8857    2.9185
    7.0000    0.4052    0.8969    2.8364
    8.0000    0.4049    0.9044    2.7873
    9.0000    0.4046    0.9086    2.7548
   10.0000    0.4045    0.9113    2.7367
      ⋮
```

```
X1v = ResultTable(:,2)
```

```
X1v = 11×1
    0.4772
    0.4545
    0.4251
    0.4098
    0.4070
    0.4059
    0.4052
    0.4049
    0.4046
    0.4045
      ⋮
```

```
X2v = ResultTable(:,3)
```

```
X2v = 11×1
    0.7718
    0.7445
    0.8083
    0.8418
    0.8671
    0.8857
    0.8969
    0.9044
    0.9086
    0.9113
      ⋮
```

(c) In a 2-D figure, plot the trajectory (i.e., the values connected by lines with arrows) of the computed solution vector as the number of iteration progresses.

```
dx1v = gradient(X1v) ;
dx2v = gradient(X2v) ;

figure(2) %Figure to show values connected by lines with arrows
hold on
fcontour(B,'LineWidth',1)
```

```
hold on

scatter(X1v,X2v,'filled')
quiver(X1v, X2v, dx1v, dx2v)
hold on
xlim([0.35 0.5])
ylim([0.73 0.92])
grid on
hold on
```