# Optimum Design - Sheet 2 - Solution
# Graphical Optimization

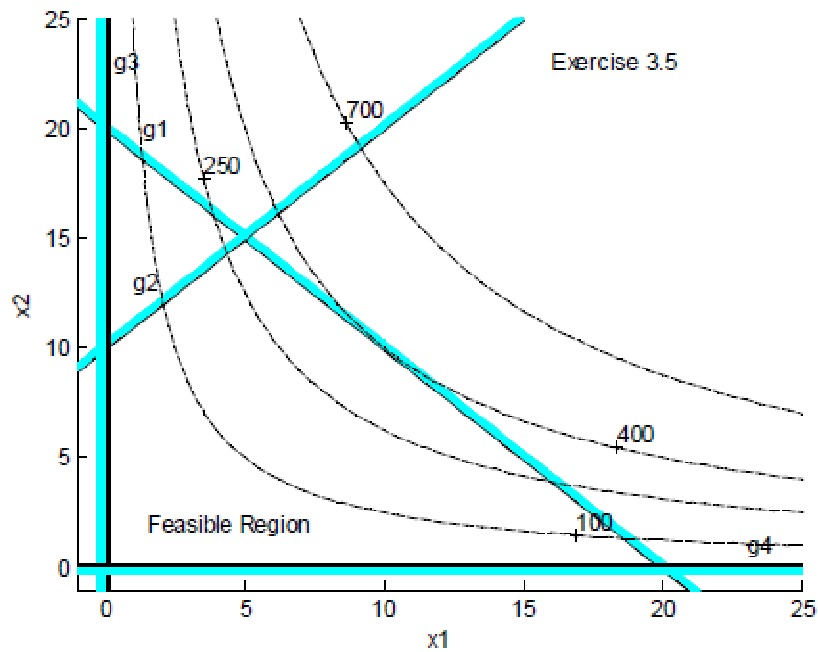
1. Solve the following problems using the graphical method by hand and a Matlab code:

   a. Maximize $f(x_1, x_2) = 4\, x_1 x_2$
      subject to $x_1 + x_2 \leq 20$
      $x_2 - x_1 \leq 10$
      $x_1, x_2 \geq 0$

**Solution:**

$$F = 4x_1x_2;$$
$$g_1 = x_1 + x_2 - 20 \leq 0;$$
$$g_2 = x_2 - x_1 - 10 \leq 0;$$
$$g_3 = -x_1 \leq 0;$$
$$g_4 = -x_2 \leq 0$$

The optimum solution is: $x_1^* = 10, x_2^* = 10, F^* = 400$
Active constraint: $g_1$.



## Matlab Code

```
%Create a grid from -1 to 7 with an increment of 0.01 for the variables x1 and x2
[x1,x2]=meshgrid(-1:0.5:25.0, -1:0.5:25.0);
        %Enter functions for the minimization problem
f=4*x1.*x2;
g1=x1+x2-20;
g2=x2-x1-10;
g3=-x1;
g4=-x2;
cla reset
axis auto            %Minimum and maximum values for axes are determined
automatically
                        %Limits for x- and y-axes may be specified with the command
                        %axis ([xmin xmax ymin ymax])
xlabel('x1'),ylabel('x2')       %Specifies labels for x- and y-axes
hold on
text(16,23,'Exercise 3.5')
cv1=[0 0];
const1=contour(x1,x2,g1,cv1,'k','LineWidth',3);
text(1.35,20,'g1')
cv11=[0.01:0.01:0.3];
cv22=[0.01:0.01:0.3];
const1=contour(x1,x2,g1,cv22,'c');
const2=contour(x1,x2,g2,cv1,'k','Linewidth',3);
const2=contour(x1,x2,g2,cv11,'c');
text(1,13,'g2')
const3=contour(x1,x2,g3,cv1,'k','Linewidth',4);
const3=contour(x1,x2,g3,cv11,'c');
text(23,1,'g4')
const4=contour(x1,x2,g4,cv1,'k','LineWidth',3);
text(0.3,23,'g3')
const4=contour(x1,x2,g4,cv22,'c');
text(1.5,2,'Feasible Region')
fv=[100 250 400 700];       %Defines contours for the minimization function
fs=contour(x1,x2,f,fv,'k--');       %'k' specifies black dashed lines for function
contours
clabel(fs)              %Automatically puts the contour value on the graph
hold off                %Indicates end of this plotting sequence
                        %Subsequent plots will appear in separate windows
```

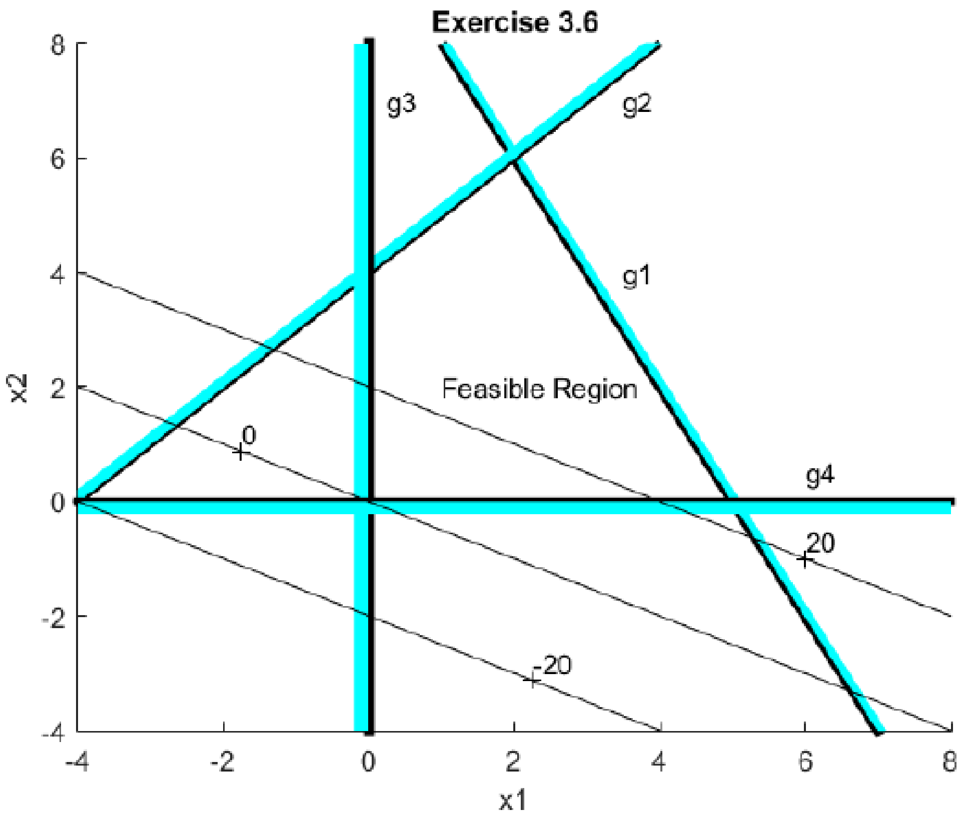
b. Minimize $f(x_1, x_2) = 5x_1 + 10x_2$
subject to $10x_1 + 5x_2 \leq 50$
$5x_1 - 5x_2 \geq -20$
$x_1, x_2 \geq 0$

**Solution:**

$$f = 5x_1 + 10x_2;$$
$$g_1 = 10x_1 + 5x_2 - 50 \leq 0;$$
$$g_2 = -5x_1 + 5x_2 - 20 \leq 0;$$
$$g_3 = -x_1 \leq 0;$$
$$g_4 = -x_2 \leq 0$$

The optimum solution is: $x_1^* = 0,\ x_2^* = 0,\ f^* = 0$

Active constraints: $g_3$ and $g_4$.



## Matlab Code

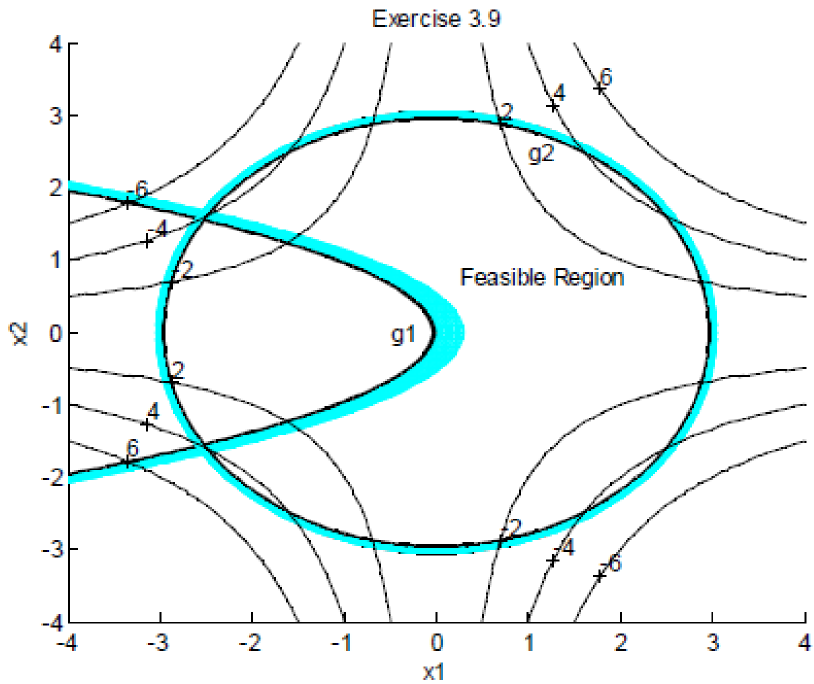
```
                %Create a grid from -1 to 7 with an increment of 0.01 for the variables x1 and x2
[x1,x2]=meshgrid(-4:0.5:8.0, -4:0.5:8.0);
                %Enter functions for the minimization problem
f=5*x1+10*x2;
g1=10*x1+5*x2-50;
g2=-5*x1+5*x2-20;
g3=-x1;
g4=-x2;
cla reset
axis auto                   %Minimum and maximum values for axes are determined automatically
                            %Limits for x- and y-axes may be specified with the command
                            %axis ([xmin xmax ymin ymax])
xlabel('x1'),ylabel('x2')   %Specifies labels for x- and y-axes
Title ('Exercise 3.6')
hold on                     %retains the current plot and axes properties for all subsequent plots
                            %Use the "contour" command to plot constraint/minimization functions
cv1=[0 0];                  %Specifies two contour values
cv12=[0.01:0.01:1];
const1=contour(x1,x2,g1,cv1,'k','LineWidth',4);
text(3.5,4,'g1')
const1=contour(x1,x2,g1,cv12,'c');
const2=contour(x1,x2,g2,cv1,'k','Linewidth',3);
const2=contour(x1,x2,g2,cv12,'c');
text(3.5,7,'g2')
cv34=[0.01:0.01:0.2];
const3=contour(x1,x2,g3,cv1,'k','Linewidth',4);
const3=contour(x1,x2,g3,cv34,'c');
text(0.25,6,'g3')
const4=contour(x1,x2,g4,cv1,'k','LineWidth',3);
text(7,0.25,'g4')
const4=contour(x1,x2,g4,cv34,'c');
text(1,2,'Feasible Region')
fv=[-20 0 20];              %Defines contours for the minimization function
fs=contour(x1,x2,f,fv,'k'); %'k' specifies black dashed lines for function contours
clabel(fs)                  %Automatically puts the contour value on the graph
hold off                    %Indicates end of this plotting sequence
                            %Subsequent plots will appear in separate windows
```

c. Minimize $f(x_1, x_2) = x_1x_2$
subject to $x_1 + x_2^2 \le 0$
$x_1^2 + x_2^2 \le 9$

**Solution:**

$f = x_1 x_2;$
$g_1 = x_1 + x_2^2 \le 0;$
$g_2 = x_1^2 + x_2^2 - 9 \le 0$

The optimum solution is: $x_1^* \doteq -2.5,\ x_2^* \doteq 1.58,\ f^* \doteq -3.95$
Active constraints: $g_1$ and $g_2$.



## Matlab Code

```
%Exercise 3.9
%Create a grid from -1 to 7 with an increment of 0.01 for the variables x1 and x2
[x1,x2]=meshgrid(-4:0.1:4.0, -4:0.1:4.0);
%Enter functions for the minimization problem
f=x1.*x2;
g1=x1+x2.^2;
g2=x1.^2+x2.^2-9;
cla reset
axis auto                %Minimum and maximum values for axes are determined
automatically
                         %Limits for x- and y-axes may be specified with the command
                         %axis ([xmin xmax ymin ymax])
xlabel('x1'),ylabel('x2')        %Specifies labels for x- and y-axes
title('Exercise 3.9')
hold on
cv1=[0 0];
cv12=[0.01:0.01:0.3];
const1=contour(x1,x2,g1,cv1,'k','LineWidth',4);
text(-0.5,0,'g1')
const1=contour(x1,x2,g1,cv12,'c');
const2=contour(x1,x2,g2,cv1,'k','Linewidth',4);
const2=contour(x1,x2,g2,cv12,'c');
text(1,2.5,'g2')
text(0.25,0.75,'Feasible Region')
fv=[2 -2 4 -4 6 -6];             %Defines contours for the minimization function
fs=contour(x1,x2,f,fv,'k');      %'k' specifies black dashed lines for function
contours
clabel(fs)               %Automatically puts the contour value on the graph
hold off                 %Indicates end of this plotting sequence
                         %Subsequent plots will appear in separate windows
```

2. Solve the rectangular beam problem of Sheet 2 Problem 4 graphically by hand and a Matlab code for the following data: $M$ = 80 kN·m, $V$ = 150 kN, $\sigma_a$ = 8 MPa, and $\tau_a$ = 3MPa.

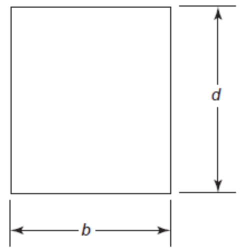



Fig.1 Cross section of a rectangular beam.

**Solution:**

Rewrite the formulation of Problem 2 in sheet 4

$M = 80$ kN.m $= 8.0\times10^6$ N.cm; $V = 150$ kN $= 1.5\times10^5$ N; $\sigma_a = 8$ MPa $= 800$ N/cm$^2$; $\tau_a = 300$ N/cm$^2$

Using units of Newtons and centimeters, we have: minimize $f = bd$; subject to

$g_1: \dfrac{6M}{bd^2} - \sigma_a \le 0;\quad g_1 = \dfrac{6(8.0 \text{ x } 10^6)}{bd^2} - 800 \le 0;$

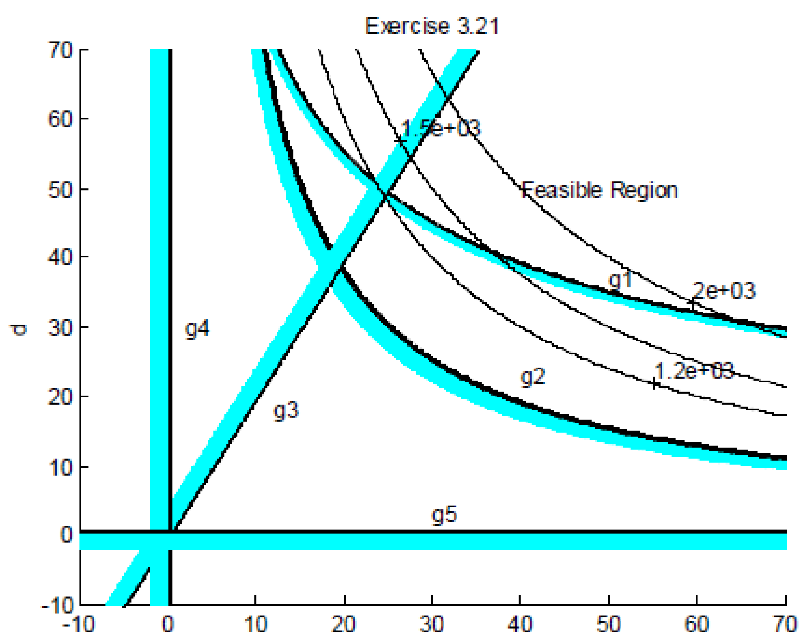
$g_2: \dfrac{3V}{2bd} - \tau_a \le 0;\quad g_2 = \dfrac{3(1.5\times10^5)}{2bd} - 300 \le 0;$

$g_3 = d - 2b \le 0;$

$g_4 = -b \le 0;$

$g_5 = -d \le 0$

Optimum solution: $b^* \doteq 24.66$ cm, $d^* \doteq 49.32$ cm, $f^* \doteq 1216$ cm$^2$; $g_1$ (bending stress) and $g_3$ (depth- ratio) constraints are active.



## Matlab Code

```
%Create a grid from -10 to 70 with an increment of 0.1 for the variables x1 and x2
[b,d]=meshgrid(-10:0.1:70.0, -10:0.1:70.0);
%Enter functions for the minimization problem
f=b.*d;
g1=(48*10.^6)./(b.*(d.^2))-800;
g2=(2.25*10^5)./(b.*d)-300;
g3=0.5.*d-b;
g4=-b;
g5=-d;
cla reset
axis auto
%Minimum and maximum values for axes are determined automatically
xlabel('b'),ylabel('d')    %Specifies labels for x- and y-axes
title('Exercise 3.21')
hold on      %retains the current plot and axes properties for all subsequent plots
%Use the "contour" command to plot constraint and minimization functions
cv1=[0 0]; %Specifies two contour values
const1=contour(b,d,g1,cv1,'k','LineWidth',4);
text(50,37,'g1')
cv11=[0.01:0.05:2];
cv22=[5:2:40];
const1=contour(b,d,g1,cv22,'c');
const2=contour(b,d,g2,cv1,'k','Linewidth',4);
const2=contour(b,d,g2,cv22,'c');
text(40,23,'g2')
const3=contour(b,d,g3,cv1,'k','Linewidth',4);
const3=contour(b,d,g3,cv11,'c');
text(12,18,'g3')
const4=contour(b,d,g4,cv1,'k','Linewidth',4);
const4=contour(b,d,g4,cv11,'c');
text(30,3,'g5')
const5=contour(b,d,g5,cv1,'k','Linewidth',4);
const5=contour(b,d,g5,cv11,'c');
text(2,30,'g4')
text(40,50,'Feasible Region')
fv=[1200 1500 2000];     %Defines contours for the minimization function
fs=contour(b,d,f,fv,'k');    %'k' specifies black dashed lines for function contours
clabel(fs)          %Automatically puts the contour value on the graph
hold off     %Indicates end of this plotting sequence
             %Subsequent plots will appear in separate windows
```

3. Solve the cantilever beam problem of Sheet 2 Problem 7 graphically by hand and a Matlab code for the following data: $P$ = 10 kN; $L$ = 5.0 m; modulus of elasticity, $E$ = 210 Gpa; allowable bending stress, $\sigma_b$ = 250 MPa; allowable shear stress, $\tau_a$ = 90 MPa; mass density, $\rho$= 7850 kg/m3; $R_o \le 20.0$ cm; $R_i \le 20.0$ cm.

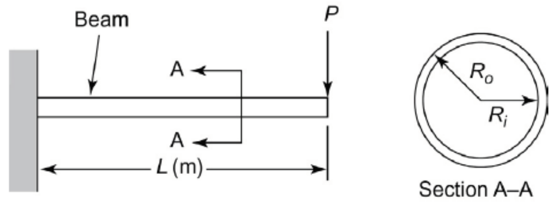



Fig.2 Cantilever beam.

**Solution:**

Using kg, N and cm as units

**Given Data**: (this data will change if different units are used)

$P = 10\text{ kN} = 10^4\text{ N}$

$L = 5\text{ m} = 500\text{ cm}$

$\sigma_a = 250\text{ MPa} = 2.5\times10^4\text{ N/cm}^2$;

$\tau_a = 90\text{ MPa} = 9000\text{ N/cm}^2$

$\rho = 7850\text{ kg/m}^3 = 7.85\times10^{-3}\text{ kg/cm}^3$;

Cross-sectional area of hollow tubes: $A = \pi(R_o^2 - R_i^2)$

Moment of inertia of a hollow tube is $I = \pi\left(R_o^4 - R_i^4\right)/4$

**Maximum bending stress:**

$$\sigma = \frac{PL}{I}R_0$$

**Maximum shearing stress:**

$$\tau = \frac{VQ}{Ib};\quad V = P;\quad Q = \frac{2}{3}(R_o^3 - R_i^3);\quad b = 2(R_o - R_i)$$

Substituting various quantities and simplifying the expression, we get

$$\tau = \frac{P}{3I}(R_o^2 + R_0R_i + R_i^2)$$

In addition, it must be ensured that $R_o > R_i$ which can be imposed as a constraint on the wall thickness as $t \ge t_{min}$ with $t_{min}$ as, say 0.1 cm.

Thickness: $t = R_o - R_i$

Referring to Exercise 2.23 and the given data, the problem is formulated in terms of the design variables only as follows:

$$f = (7.85\times10^{-3})(500)\ \pi\left(R_o^2 - R_i^2\right) = 12.331\left(R_o^2 - R_i^2\right)$$

$$g_1 = \frac{4PlR_o}{\pi\left(R_o^4 - R_i^4\right)} \le \sigma_a;\ \text{or}$$

$$g_1 = \frac{R_o\left(4.0\times10^4\right)(500)}{\pi\left(R_o^4 - R_i^4\right)} \le \left(2.5\times10^4\right);\ \text{or}$$

$$g_1 = \frac{R_o 6.3662\times10^6}{\left(R_o^4 - R_i^4\right)} - 2.5\times10^4 \le 0$$

$$g_2 = \frac{4P\left(R_o^2 + R_oR_i + R_i^2\right)}{3\pi\left(R_o^4 - R_i^4\right)} \le \tau_a; \text{ or}$$

$$g_2 = \frac{\left(4.0\times10^4\right)\left(R_o^2 + R_oR_i + R_i^2\right)}{3\pi\left(R_o^4 - R_i^4\right)} \le 9000; \text{ or}$$

$$g_2 = \frac{4244.13\left(R_o^2 + R_oR_i + R_i^2\right)}{\left(R_o^4 - R_i^4\right)} - 9000 \le 0;$$

$$g_3 = R_o - 20 \le 0;$$

$$g_4 = R_i - 20 \le 0;$$

$$g_5 = -R_o \le 0;$$

$$g_6 = -R_i \le 0$$

**FORMULATION 2: Using Intermediate Variables**

**Step 4: Optimization Criterion**

Optimization criterion is to minimize mass of hollow tube, and the cost function is defined as $f = \rho AL$

**Step 5: Formulation of Constraints**

$g_1$ : bending stress should be smaller than the allowable bending stress; $\sigma \le \sigma_a$

$$g_1 = \sigma - \sigma_a \le 0$$

$g_2$ : shear stress smaller than allowable shear stress: $\tau \le \tau_a$

$$g_2 = \tau - \tau_a \le 0$$

$$g_3 = R_o - 20 \le 0$$

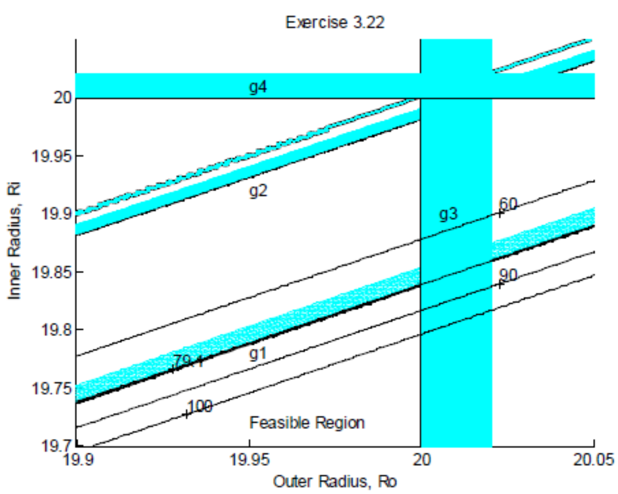
$$g_4 = R_i - 20 \le 0$$

$$g_5 = -R_o \le 0$$

$$g_6 = -R_i \le 0$$

$$g_7 = t_{min} - t \le 0$$

Optimum solution: $R_o^* = 20$ cm, $R_i^* \doteq 19.84$ cm, $f^* \doteq 79.1$ kg, $g_1$ (bending stress) and $g_3$ (max. outer radius) constraints are active.



## Matlab Code

```
%Create a grid
[Ro,Ri]=meshgrid(19.9:0.001:20.05, 19.70:0.001:20.05);
%Enter functions for the minimization problem; use Newton and cm as units
P=10000;  L=500;  sigma_a=25000;  tau_a=9000;  ro=7.85/1000;
A=pi.*(Ro.*Ro-Ri.*Ri)
I=0.25.*pi.*(Ro.^4-Ri.^4)
sigma=P.*L.*Ro./I
tau=P.*(Ro.*Ro+Ro.*Ri+Ri.*Ri)./(3.*I)
t=Ro-Ri
f=ro.*A.*L
g1=sigma./sigma_a - 1
g2= tau./tau_a - 1
%f=12.331*((Ro.^2)-(Ri.^2));
%g1=Ro*(6.3662*10.^6)-(2.5*10^4)*((Ro.^4)-(Ri.^4));
%g2=(4244.13)*(Ro.*Ro+Ri.*Ro+Ri.*Ri)-9000*(Ro.^4).*(Ri.^4);
g3=Ro./20 - 1;
g4=Ri./20 - 1;
g5=-Ro;
g6=-Ri;
cla reset
axis auto
%Minimum and maximum values for axes are determined automatically

title('Exercise 3.22')
xlabel('Outer Radius, Ro'),ylabel('Inner Radius, Ri')   %Specifies labels for x-
and y-axes
hold on     %retains the current plot and axes properties for all subsequent plots
cv1=[0 0];  %Specifies two contour values
            %Use the "contour" command to plot constraint and minimization
functions
const1=contour(Ro,Ri,g1,cv1,'k','LineWidth',2);
cv11=[0.0:0.00001:0.001];
cv22=[0.01:0.01:0.1];
cv33=[0.01:0.01:0.8];
const1=contour(Ro,Ri,g1,cv22,'c');
const2=contour(Ro,Ri,g2,cv1,'k','Linewidth',3);
const2=contour(Ro,Ri,g2,cv33,'c');
const3=contour(Ro,Ri,g3,cv1,'k','Linewidth',2);
const3=contour(Ro,Ri,g3,cv11,'c');
const4=contour(Ro,Ri,g4,cv1,'k','Linewidth',3);
const4=contour(Ro,Ri,g4,cv11,'c');
const4=contour(Ro,Ri,g5,cv1,'k','Linewidth',3);
const4=contour(Ro,Ri,g5,cv11,'c');
const5=contour(Ro,Ri,g6,cv1,'k','Linewidth',3);
const5=contour(Ro,Ri,g6,cv11,'c');
%Label constraints
text(19.95,19.78,'g1')
text(19.95,19.92,'g2')
text(20.005,19.9,'g3')
text(19.95,20.01,'g4')
text(6,0.5,'g5')
text(19.95,19.72,'Feasible Region')
fv=[60 79.1 90 100];    %Defines contours for the minimization function
fs=contour(Ro,Ri,f,fv,'k--');   %'k' specifies black dashed lines for function
contours
clabel(fs)          %Automatically puts the contour value on the graph
hold off    %Indicates end of this plotting sequence
            %Subsequent plots will appear in separate windows
```

```
%Exercise 3.22; L=10m
%Create a grid
[Ro,Ri]=meshgrid(19.8:0.001:20.05, 19.30:0.001:20.05);
%Enter functions for the minimization problem; use Newton and cm as units
P=10000;  L=1000;  sigma_a=25000;  tau_a=9000;  ro=7.85/1000;
A=pi.*(Ro.*Ro-Ri.*Ri)
I=0.25.*pi.*(Ro.^4-Ri.^4)
sigma=P.*L.*Ro./I
tau=P.*(Ro.*Ro+Ro.*Ri+Ri.*Ri)./(3.*I)
t=Ro-Ri
f=ro.*A.*L
g1=sigma./sigma_a - 1
g2= tau./tau_a - 1
%f=12.331*((Ro.^2)-(Ri.^2));
%g1=Ro*(6.3662*10.^6)-(2.5*10^4)*((Ro.^4)-(Ri.^4));
%g2=(4244.13)*(Ro.*Ro+Ri.*Ro+Ri.*Ri)-9000*(Ro.^4).*(Ri.^4);
g3=Ro./20 - 1;
g4=Ri./20 - 1;
g5=-Ro;

g6=-Ri;
cla reset
axis auto
%Minimum and maximum values for axes are determined automatically
title('Exercise 3.22 with L=1000cm')
xlabel('Outer Radius, Ro'),ylabel('Inner Radius, Ri')   %Specifies labels for x-
and y-axes
hold on     %retains the current plot and axes properties for all subsequent plots
cvl=[0 0];  %Specifies two contour values
            %Use the "contour" command to plot constraint and minimization
functions
const1=contour(Ro,Ri,g1,cvl,'k','LineWidth',2);
cv11=[0.0:0.00001:0.001];
cv22=[0.01:0.01:0.1];
cv33=[0.01:0.01:0.8];
const1=contour(Ro,Ri,g1,cv22,'c');
const2=contour(Ro,Ri,g2,cvl,'k','Linewidth',3);
const2=contour(Ro,Ri,g2,cv33,'c');
const3=contour(Ro,Ri,g3,cvl,'k','Linewidth',2);
const3=contour(Ro,Ri,g3,cv11,'c');
const4=contour(Ro,Ri,g4,cvl,'k','Linewidth',3);
const4=contour(Ro,Ri,g4,cv11,'c');
const4=contour(Ro,Ri,g5,cvl,'k','Linewidth',3);
const4=contour(Ro,Ri,g5,cv11,'c');
const5=contour(Ro,Ri,g6,cvl,'k','Linewidth',3);
const5=contour(Ro,Ri,g6,cv11,'c');
%Label constraints
text(19.9,19.6,'g1')
text(19.95,19.92,'g2')
text(20.005,19.9,'g3')
text(19.95,20.01,'g4')
text(6,0.5,'g5')
text(19.9,19.4,'Feasible Region')
fv=[250 319.185 350 400 450];    %Defines contours for the minimization function
fs=contour(Ro,Ri,f,fv,'k');   %'k' specifies black dashed lines for function
contours
clabel(fs)          %Automatically puts the contour value on the graph
hold off    %Indicates end of this plotting sequence
            %Subsequent plots will appear in separate windows
```