CHAPTER 7

7.1 Equation (E7.1.1) can be differentiated to give

$$\frac{d^2z}{dt^2} = -\left(g + \frac{c}{m}v_0\right)e^{-(c/m)t}$$

Therefore, the Newton-Raphson method can be written as

$$t_{i+1} = t_i + \frac{v_0 e^{-(c/m)t} - \frac{mg}{c}\left(1 - e^{-(c/m)t}\right)}{\left(g + \frac{c}{m}v_0\right)e^{-(c/m)t}}$$

or simplifying

$$t_{i+1} = t_i + \frac{v_0 + \frac{mg}{c} - \frac{mg}{c}e^{(c/m)t}}{g + \frac{c}{m}v_0}$$

and substituting parameters

$$t_{i+1} = t_i + \frac{107.32 - 52.32e^{0.1875t_i}}{20.1225}$$

The first iteration gives

$$t_1 = 3 + \frac{107.32 - 52.32e^{0.1875(3)}}{20.1225} = 3.7706$$

After three iterations, the process converges on the correct result.

i	*t*	*f*	*f'*
0	3	8.829093	-11.4655
1	3.77006	0.607799	-9.92396
2	3.831306	0.003477	-9.81065
3	3.83166	1.15E-07	-9.81

Also notice how the derivative of the velocity (acceleration) converges on *g* as the velocity is driven to zero

7.2 (a) The function can be differentiated to give

$$f'(x) = -2x + 8$$

This function can be set equal to zero and solved for $x = 8/2 = 4$. The derivative can be differentiated to give the second derivative

$$f''(x) = -2$$

Because this is negative, it indicates that the function has a maximum at $x = 4$.

(b) Using Eq. 7.10

$x_1 = 0 \quad f(x_1) = -12$
$x_2 = 2 \quad f(x_2) = 0$
$x_3 = 6 \quad f(x_3) = 0$

$$x_4 = 2 - \frac{1}{2}\frac{(2-0)^2[0-0]-(2-6)^2[0+12]}{(2-0)[0-0]-(2-6)[0+12]} = 4$$

7.3 Differentiating the function and setting the result equal to zero results in the following roots problem to locate the minimum

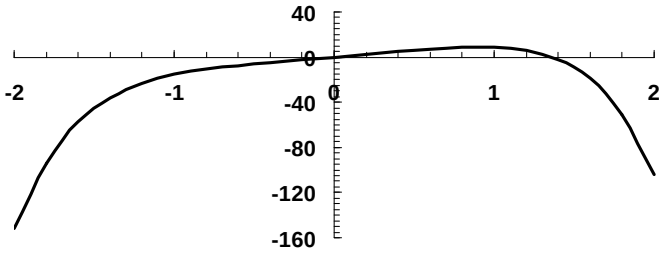
$$f'(x) = 6 + 10x + 9x^2 + 16x^3$$

Bisection can be employed to determine the root. Here are the first few iterations:

iteration	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	-2.0000	1.0000	-0.5000	-106.000	1.2500	-132.500	300.00%
2	-2.0000	-0.5000	-1.2500	-106.000	-23.6875	2510.875	60.00%
3	-1.2500	-0.5000	-0.8750	-23.6875	-6.5781	155.819	42.86%
4	-0.8750	-0.5000	-0.6875	-6.5781	-1.8203	11.974	27.27%
5	-0.6875	-0.5000	-0.5938	-1.8203	-0.1138	0.207	15.79%
6	-0.5938	-0.5000	-0.5469	-0.1138	0.6060	-0.0689	8.57%
7	-0.5938	-0.5469	-0.5703	-0.1138	0.2562	-0.0291	4.11%
8	-0.5938	-0.5703	-0.5820	-0.1138	0.0738	-0.0084	2.01%
9	-0.5938	-0.5820	-0.5879	-0.1138	-0.0193	0.0022	1.00%
10	-0.5879	-0.5820	-0.5850	-0.0193	0.0274	-0.0005	0.50%

The approach can be continued to yield a result of $x = -0.5867$.

7.4 (a) The function can be plotted



(b) The function can be differentiated twice to give

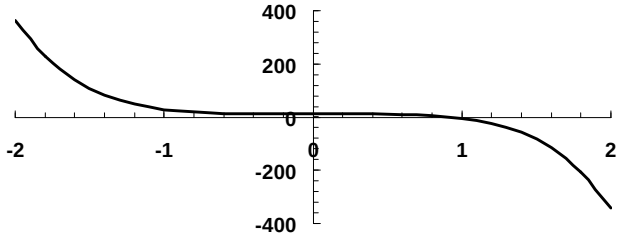
$$f''(x) = -45x^4 - 24x^2$$

Thus, the second derivative will always be negative and hence the function is concave for all values of x.

(c) Differentiating the function and setting the result equal to zero results in the following roots problem to locate the maximum

$$f'(x) = 0 = -9x^5 - 8x^3 + 12$$

A plot of this function can be developed



A technique such as bisection can be employed to determine the root. Here are the first few iterations:

iteration	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0.00000	2.00000	1.00000	12	-5	-60.0000	
2	0.00000	1.00000	0.50000	12	10.71875	128.6250	100.00%
3	0.50000	1.00000	0.75000	10.71875	6.489258	69.5567	33.33%
4	0.75000	1.00000	0.87500	6.489258	2.024445	13.1371	14.29%
5	0.87500	1.00000	0.93750	2.024445	-1.10956	-2.2463	6.67%

The approach can be continued to yield a result of $x = 0.91692$.

7.5 First, the golden ratio can be used to create the interior points,

$$d = \frac{\sqrt{5}-1}{2}(2-0) = 1.2361$$

$$x_1 = 0 + 1.2361 = 1.2361 \qquad x_2 = 2 - 1.2361 = 0.7639$$

The function can be evaluated at the interior points

$$f(x_2) = f(0.7639) = 8.1879 \qquad f(x_1) = f(1.2361) = 4.8142$$

Because $f(x_2) > f(x_1)$, the maximum is in the interval defined by x_l, x_2, and x_1.where x_2 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1 - 0.61803)\left|\frac{2-0}{0.7639}\right| \times 100\% = 100\%$$

For the second iteration, $x_l = 0$ and $x_u = 1.2361$. The former x_2 value becomes the new x_1, that is, $x_1 = 0.7639$ and $f(x_1) = 8.1879$. The new values of d and x_2 can be computed as

$$d = \frac{\sqrt{5}-1}{2}(1.2361-0) = 0.7639 \qquad x_2 = 1.2361 - 0.7639 = 0.4721$$

The function evaluation at $f(x_2) = 5.5496$. Since this value is less than the function value at x_1, the maximum is in the interval prescribed by x_2, x_1 and x_u. The process can be repeated and all three iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	0.0000	0.0000	0.7639	8.1879	1.2361	4.8142	2.0000	-104.0000	1.2361	0.7639	100.00%
2	0.0000	0.0000	0.4721	5.5496	0.7639	8.1879	1.2361	4.8142	0.7639	0.7639	61.80%
3	0.4721	5.5496	0.7639	8.1879	0.9443	8.6778	1.2361	4.8142	0.4721	0.9443	30.90%

The approach can be continued to yield a result of $x = 0.91692$.

7.6 First, the function values at the initial values can be evaluated

$f(x_1) = f(0) = 0$
$f(x_2) = f(1) = 8.5$
$f(x_3) = f(2) = -104$

and substituted into Eq. (7.10) to give,

$$x_4 = 1 - \frac{1}{2}\frac{(1-0)^2[8.5+104] - (1-2)^2[8.5-0)]}{(1-0)[8.5+104] - (1-2)[8.5-0)]} = 0.570248$$

which has a function value of $f(0.570248) = 6.5799$. Because the function value for the new point is lower than for the intermediate point (x_2) and the new x value is to the left of the intermediate point, the lower guess (x_1) is discarded. Therefore, for the next iteration,

$f(x_1) = f(0.570248) = 6.6799$
$f(x_2) = f(1) = 8.5$
$f(x_3) = f(2) = -104$

which can be substituted into Eq. (7.10) to give $x_4 = 0.812431$, which has a function value of $f(0.812431) = 8.446523$. At this point, an approximate error can be computed as

$$\varepsilon_a = \left|\frac{0.81243 - 0.570248}{0.81243}\right| \times 100\% = 29.81\%$$

The process can be repeated, with the results tabulated below:

i	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$	x_4	$f(x_4)$	ε_a
1	0.00000	0.00000	1.00000	8.50000	2.0000	-104	0.57025	6.57991	
2	0.57025	6.57991	1.00000	8.50000	2.0000	-104	0.81243	8.44652	29.81%
3	0.81243	8.44652	1.00000	8.50000	2.0000	-104	0.90772	8.69575	10.50%

Thus, after 3 iterations, the result is converging on the true value of $f(x) = 8.69793$ at $x = 0.91692$.

7.7 (a) First, the golden ratio can be used to create the interior points,

$$d = \frac{\sqrt{5}-1}{2}(4-(-2)) = 3.7082$$
$$x_1 = -2 + 3.7082 = 1.7082$$
$$x_2 = 4 - 3.7082 = 0.2918$$

The function can be evaluated at the interior points

$$f(x_2) = f(0.2918) = 1.04156$$
$$f(x_1) = f(1.7082) = 5.00750$$

Because $f(x_1) > f(x_2)$, the maximum is in the interval defined by x_2, x_1 and x_u where x_1 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1 - 0.61803)\left|\frac{4-(-2)}{1.7082}\right| \times 100\% = 134.16\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	-2.0000	-29.6000	0.2918	1.0416	1.7082	5.0075	4.0000	-12.8000	3.7082	1.7082	134.16%
2	0.2918	1.0416	1.7082	5.0075	2.5836	5.6474	4.0000	-12.8000	2.2918	2.5836	54.82%
3	1.7082	5.0075	2.5836	5.6474	3.1246	2.9361	4.0000	-12.8000	1.4164	2.5836	33.88%
4	1.7082	5.0075	2.2492	5.8672	2.5836	5.6474	3.1246	2.9361	0.8754	2.2492	24.05%
5	1.7082	5.0075	2.0426	5.6648	2.2492	5.8672	2.5836	5.6474	0.5410	2.2492	14.87%
6	2.0426	5.6648	2.2492	5.8672	2.3769	5.8770	2.5836	5.6474	0.3344	2.3769	8.69%
7	2.2492	5.8672	2.3769	5.8770	2.4559	5.8287	2.5836	5.6474	0.2067	2.3769	5.37%
8	2.2492	5.8672	2.3282	5.8853	2.3769	5.8770	2.4559	5.8287	0.1277	2.3282	3.39%
9	2.2492	5.8672	2.2980	5.8828	2.3282	5.8853	2.3769	5.8770	0.0789	2.3282	2.10%
10	2.2980	5.8828	2.3282	5.8853	2.3468	5.8840	2.3769	5.8770	0.0488	2.3282	1.30%
11	2.2980	5.8828	2.3166	5.8850	2.3282	5.8853	2.3468	5.8840	0.0301	2.3282	0.80%

(b) First, the function values at the initial values can be evaluated

$f(x_1) = f(1.75) = 5.1051$
$f(x_2) = f(2) = 5.6$
$f(x_3) = f(2.5) = 5.7813$

and substituted into Eq. (7.10) to give $x_4 = 2.3341$, which has a function value of $f(2.3341) = 5.8852$. Because the function value for the new point is higher than for the intermediate point (x_2) and the new x value is to the right of the intermediate point, the lower guess (x_1) is discarded. Therefore, for the next iteration,

$f(x_1) = f(2) = 5.6$
$f(x_2) = f(2.3341) = 5.8852$
$f(x_3) = f(2.5) = 5.7813$

which can be substituted into Eq. (7.10) to give $x_4 = 2.3112$, which has a function value of $f(2.3112) = 5.8846$. At this point, an approximate error can be computed as

$$\varepsilon_a = \left|\frac{2.3112 - 2.3341}{2.3112}\right| \times 100\% = 0.99\%$$

The process can be repeated, with the results tabulated below:

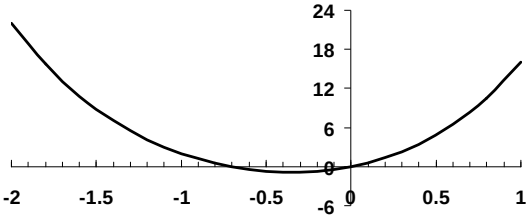
i	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$	x_4	$f(x_4)$	ε_a
1	1.7500	5.1051	2.0000	5.6000	2.5000	5.7813	2.3341	5.8852	
2	2.0000	5.6000	2.3341	5.8852	2.5000	5.7813	2.3112	5.8846	0.99%
3	2.0000	5.6000	2.3112	5.8846	2.3341	5.8852	2.3270	5.8853	0.68%
4	2.3112	5.8846	2.3270	5.8853	2.3341	5.8852	2.3263	5.8853	0.03%
5	2.3112	5.8846	2.3263	5.8853	2.3270	5.8853	2.3264	5.8853	0.00%

Thus, after 5 iterations, the result is converging rapidly on the true value of $f(x) = 5.8853$ at $x = 2.3264$.

7.8 The function can be differentiated twice to give

$$f'(x) = 4x^3 + 6x^2 + 16x = 5$$
$$f''(x) = 12x^2 + 12x + 16$$

which is positive for $-2 \le x \le 1$. This suggests that an optimum in the interval would be a minimum. A graph of the original function shows a maximum at about $x = -0.35$.



7.9 (a) First, the golden ratio can be used to create the interior points,

$$d = \frac{\sqrt{5}-1}{2}(1-(-2)) = 1.8541$$
$$x_1 = -2 + 1.8541 = -0.1459$$
$$x_2 = 1 - 1.8541 = -0.8541$$

The function can be evaluated at the interior points

$$f(x_2) = f(-0.8541) = 0.8514$$
$$f(x_1) = f(-0.1459) = -0.5650$$

Because $f(x_1) < f(x_2)$, the minimum is in the interval defined by x_2, x_1 and x_u where x_2 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1-0.61803)\left|\frac{1-(-2)}{-0.1459}\right| \times 100\% = 785.41\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	-2	22	-0.8541	0.851	-0.1459	-0.565	1	16.000	1.8541	-0.1459	785.41%
2	-0.8541	0.851	-0.1459	-0.565	0.2918	2.197	1	16.000	1.1459	-0.1459	485.41%
3	-0.8541	0.851	-0.4164	-0.809	-0.1459	-0.565	0	2.197	0.7082	-0.4164	105.11%
4	-0.8541	0.851	-0.5836	-0.475	-0.4164	-0.809	0	-0.565	0.4377	-0.4164	64.96%
5	-0.5836	-0.475	-0.4164	-0.809	-0.3131	-0.833	0	-0.565	0.2705	-0.3131	53.40%
6	-0.4164	-0.809	-0.3131	-0.833	-0.2492	-0.776	0	-0.565	0.1672	-0.3131	33.00%
7	-0.4164	-0.809	-0.3525	-0.841	-0.3131	-0.833	0	-0.776	0.1033	-0.3525	18.11%
8	-0.4164	-0.809	-0.3769	-0.835	-0.3525	-0.841	0	-0.833	0.0639	-0.3525	11.19%
9	-0.3769	-0.835	-0.3525	-0.841	-0.3375	-0.840	0	-0.833	0.0395	-0.3525	6.92%
10	-0.3769	-0.835	-0.3619	-0.839	-0.3525	-0.841	0	-0.840	0.0244	-0.3525	4.28%
11	-0.3619	-0.839	-0.3525	-0.841	-0.3468	-0.841	0	-0.840	0.0151	-0.3468	2.69%
12	-0.3525	-0.841	-0.3468	-0.841	-0.3432	-0.841	0	-0.840	0.0093	-0.3468	1.66%
13	-0.3525	-0.841	-0.3490	-0.841	-0.3468	-0.841	0	-0.841	0.0058	-0.3468	1.03%
14	-0.3490	-0.841	-0.3468	-0.841	-0.3454	-0.841	0	-0.841	0.0036	-0.3468	0.63%

(b) First, the function values at the initial values can be evaluated

$f(x_1) = f(-2) = 22$
$f(x_2) = f(-1) = 2$
$f(x_3) = f(1) = 16$

and substituted into Eq. (7.10) to give $x_4 = -0.38889$, which has a function value of $f(-0.38889) = -0.829323$. Because the function value for the new point is lower than for the intermediate point (x_2) and the new x value is to the right of the intermediate point, the lower guess (x_1) is discarded. Therefore, for the next iteration,

$f(x_1) = f(-1) = 2$
$f(x_2) = f(-0.38889) = -0.829323$
$f(x_3) = f(1) = 16$

which can be substituted into Eq. (7.10) to give $x_4 = -0.41799$, which has a function value equal to $f(-0.41799) = -0.80776$. At this point, an approximate error can be computed as

$$\varepsilon_a = \left|\frac{-0.41799 - (-0.38889)}{-0.41799}\right| \times 100\% = 6.96\%$$

The process can be repeated, with the results tabulated below:

i	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$	x_4	$f(x_4)$	ε_a
1	-2	22	-1.0000	2	1.0000	16	-0.38889	-0.82932	
2	-1.0000	2	-0.3889	-0.82932	1.0000	16	-0.41799	-0.80776	6.96%
3	-0.4180	-0.80776	-0.3889	-0.82932	1.0000	16	-0.36258	-0.83924	15.28%
4	-0.3889	-0.82932	-0.3626	-0.839236	1.0000	16	-0.35519	-0.84038	2.08%
5	-0.3626	-0.83924	-0.3552	-0.840376	1.0000	16	-0.35053	-0.84072	1.33%

After 5 iterations, the result is converging on the true value of $f(x) = -0.8408$ at $x = -0.34725$.

7.10 First, the function values at the initial values can be evaluated

$f(x_1) = f(0.1) = 30.2$
$f(x_2) = f(0.5) = 7$
$f(x_3) = f(5) = 10.6$

and substituted into Eq. (7.10) to give $x_4 = 2.7167$, which has a function value of $f(2.7167) = 6.5376$. Because the function value for the new point is lower than for the intermediate point (x_2) and the new x value is to the right of the intermediate point, the lower guess (x_1) is discarded. Therefore, for the next iteration,

$f(x_1) = f(0.5) = 7$
$f(x_2) = f(2.7167) = 6.5376$
$f(x_3) = f(5) = 10.6$

which can be substituted into Eq. (7.10) to give $x_4 = 1.8444$, which has a function value of $f(1.8444) = 5.3154$. At this point, an approximate error can be computed as

$$\varepsilon_a = \left|\frac{1.8444 - 2.7167}{1.8444}\right| \times 100\% = 47.29\%$$

The process can be repeated, with the results tabulated below:

i	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$	x_4	$f(x_4)$	ε_a
1	0.1	30.2	0.5000	7.0000	5.0000	10.6000	2.7167	6.5376	
2	0.5000	7	2.7167	6.5376	5.0000	10.6000	1.8444	5.3154	47.29%
3	0.5000	7	1.8444	5.3154	2.7167	6.5376	1.6954	5.1603	8.79%
4	0.5000	7	1.6954	5.1603	1.8444	5.3154	1.4987	4.9992	13.12%

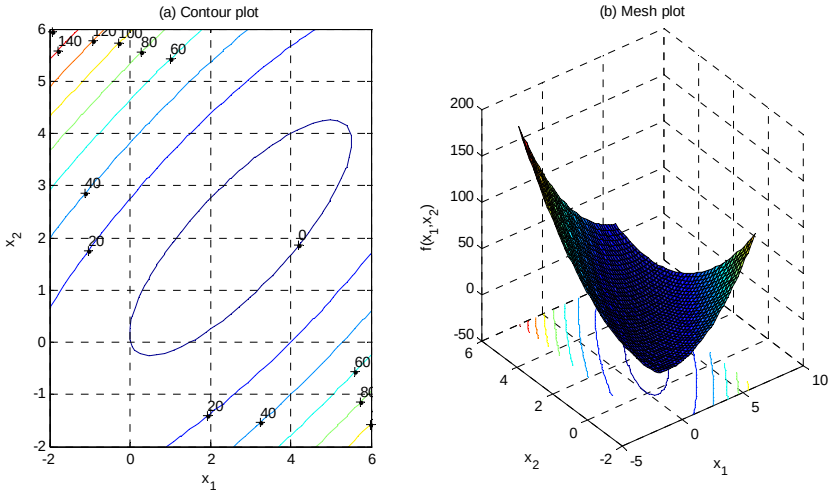
5	0.5000	7	1.4987	4.9992	1.6954	5.1603	1.4236	4.9545	5.28%
6	0.5000	7	1.4236	4.9545	1.4987	4.9992	1.3556	4.9242	5.02%
7	0.5000	7	1.3556	4.9242	1.4236	4.9545	1.3179	4.9122	2.85%
8	0.5000	7	1.3179	4.9122	1.3556	4.9242	1.2890	4.9054	2.25%
9	0.5000	7	1.2890	4.9054	1.3179	4.9122	1.2703	4.9023	1.47%
10	0.5000	7	1.2703	4.9023	1.2890	4.9054	1.2568	4.9006	1.08%

Thus, after 10 iterations, the result is converging very slowly on the true value of $f(x) = 4.8990$ at $x = 1.2247$.

7.11 The script can be written as

```
x=linspace(-2,6,40);y=linspace(-2,6,40);
[X,Y] = meshgrid(x,y);
Z=2*X.^2+3*Y.^2-4*X.*Y-Y-3*X;
subplot(1,2,1);
cs=contour(X,Y,Z);clabel(cs);
xlabel('x_1');ylabel('x_2');
title('(a) Contour plot');grid;
subplot(1,2,2);
cs=surfc(X,Y,Z);
zmin=floor(min(Z));
zmax=ceil(max(Z));
xlabel('x_1');ylabel('x_2');zlabel('f(x_1,x_2)');
title('(b) Mesh plot');
pause
f=@(x) 2*x(1)^2+3*x(2)^2-4*x(1)*x(2)-x(2)-3*x(1);
[x,fval]=fminsearch(f,[4,4])
```

When it is run, the following plot and solution are generated:



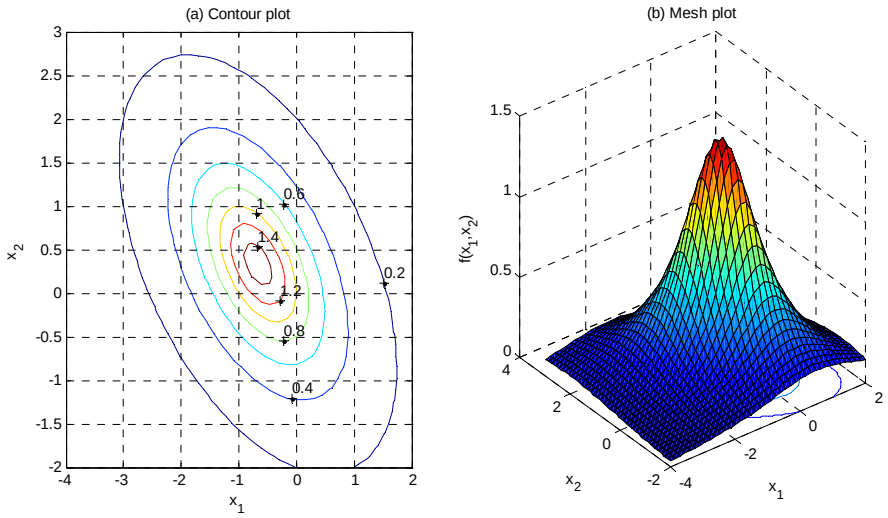

```
x =
    2.7500    2.0000
fval =
   -5.1250
```

7.12 The script can be written as

```
x=linspace(-4,2,40);y=linspace(-2,3,40);
[X,Y] = meshgrid(x,y);
Z=1./(1+X.^2+Y.^2+X+X.*Y);
```

```
subplot(1,2,1);
cs=contour(X,Y,Z);clabel(cs);
xlabel('x_1');ylabel('x_2');
title('(a) Contour plot');grid;
subplot(1,2,2);
cs=surfc(X,Y,Z);
zmin=floor(min(Z));
zmax=ceil(max(Z));
xlabel('x_1');ylabel('x_2');zlabel('f(x_1,x_2)');
title('(b) Mesh plot');
pause
f=@(x) -1/(1+x(1).^2+x(2).^2+x(1)+x(1)*x(2));
[x,fval]=fminsearch(f,[4,4])
```

Notice that because we are looking for a maximum, we enter the negative of the function for `fminsearch`. When it is run, the following plot and solution are generated:




```
x =
   -0.6667    0.3333
fval =
   -1.5000
```

Thus, the solution is 1.5 at location (–0.6667, 0.3333).

7.13 This problem can be approached by developing the following equation for the potential energy of the bracketing system,

$$V(x,y) = \frac{EA}{\ell}\left(\frac{w}{2\ell}\right)^2 x^2 + \frac{EA}{\ell}\left(\frac{h}{\ell}\right)^2 y^2 - Fx\cos\theta - Fy\sin\theta$$

Solving for an individual angle is straightforward. For $\theta = 30^o$, the given parameter values can be substituted to yield

$$V(x, y) = 5{,}512{,}026x^2 + 28{,}471{,}210y^2 - 8{,}600x - 5{,}000y$$

The minimum of this function can be determined in a number of ways. For example, a function can be set up to compute the potential energy

```
function Vxy = prob0713f(x,th)
E=2e11;A=0.0001;w=0.44;l=0.56;h=0.5;F=10000;
Vxy=E*A/l*(w/2/l)^2*x(1)^2+E*A/l*(h/l)^2*x(2)^2-F*x(1)*cos(th)-F*x(2)*sin(th);
```

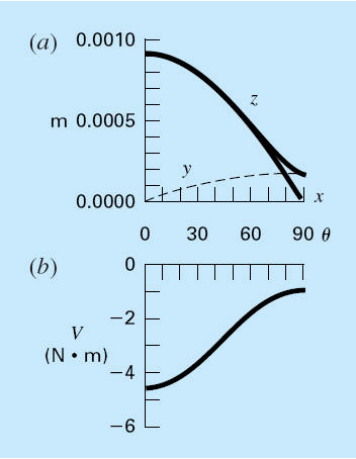
Then, the `fminsearch` function can be used to evaluate the minimum potential energy at a particular angle. For the following case, we use $\theta = 30^o$,

```
>> format short g
>> th=30*pi/180;
>> options = optimset('TolFun',1e-6,'TolX',1e-6);
>> [x fxy] = fminsearch(@prob0713f,[0 0],options,th)

x =
   0.00078585  8.7771e-005
fxy =
      -3.6212
```

Thus, the minimum potential energy is −3.6212 with deflections of $x = 0.00078585$ and $y = 0.000087771$ m.

Of course, this calculation could be implemented repeatedly for different values of θ to assess how the solution changes as the angle changes. If this is done, the results are displayed in the following figure:



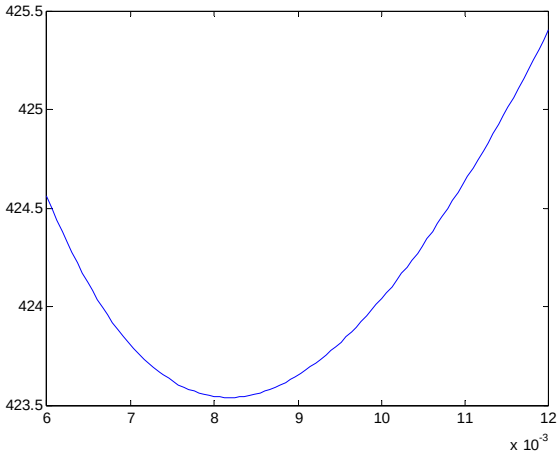
(*a*) The impact of different angles on the deflections (note that *z* is the resultant of the *x* and *y* components) and (*b*) potential energy.

As might be expected, the x deflection is at a maximum when the load is pointed in the x direction ($\theta = 0^o$) and the y deflection is at a maximum when the load is pointed in the y direction ($\theta = 90^o$). However, notice that the x deflection is much more pronounced than that in the y direction. This is also manifest in part (b), where the potential energy is higher at low angles. Both results are due to the geometry of the strut. If w were made bigger, the deflections would be more uniform.

7.14 The following script generates a plot of wire temperature versus insulation thickness. Then, it employs `fminbnd` to locate the thickness of insulation that minimizes the wire's temperature.

```
clear, clc, clf, format short g
q=75;rw=6e-3;k=0.17;h=12;Tair=293;
rplot=linspace(rw,rw*2);
T=@(ri) Tair+q/(2*pi)*(1/k*log((rw+ri)/rw)+1/h*1./(rw+ri));
Tplot=T(rplot);
plot(rplot,Tplot)
```

```
[rimin Tmin]=fminbnd(T,rw,2*rw)
rtotal=rimin+rw
```




```
rimin =
    0.0081725
Tmin =
       423.54
rtotal =
     0.014173
```

Thus, an insulation thickness of 8.1725 mm yields a minimum wire temperature of 423.54 K. The total radius of the wire and insulation is 14.1725 mm.

7.15 The algorithm from Fig. 7.7 can be converted to locate a maximum by merely changing the sense of the `if` statement highlighted below:

```
function [x,fx,ea,iter]=goldmax(f,xl,xu,es,maxit,varargin)
% goldmax: maximization golden section search
%   [xopt,fopt,ea,iter]=goldmax(f,xl,xu,es,maxit,p1,p2,...):
%      uses golden section search to find the maximum of f
% input:
%   f = name of function
%   xl, xu = lower and upper guesses
%   es = desired relative error (default = 0.0001%)
%   maxit = maximum allowable iterations (default = 50)
%   p1,p2,... = additional parameters used by f
% output:
%   x = location of maximum
%   fx = maximum function value
%   ea = approximate relative error (%)
%   iter = number of iterations

if nargin<3,error('at least 3 input arguments required'),end
if nargin<4|isempty(es), es=0.0001;end
if nargin<5|isempty(maxit), maxit=50;end
phi=(1+sqrt(5))/2;
iter=0;
while(1)
  d = (phi-1)*(xu - xl);
  x1 = xl + d;
  x2 = xu - d;
  if f(x1,varargin{:}) > f(x2,varargin{:})
    xopt = x1;
    xl = x2;
```

```
  else
    xopt = x2;
    xu = x1;
  end
  iter=iter+1;
  if xopt~=0, ea = (2 - phi) * abs((xu - xl) / xopt) * 100;end
  if ea <= es | iter >= maxit,break,end
end
x=xopt;fx=f(xopt,varargin{:});
```

An application to solve Example 7.1 can be developed as

```
>> g=9.81;z0=100;v0=55;m=80;c=15;
>> z=@(t) z0+m/c*(v0+m*g/c)*(1-exp(-c/m*t))-m*g/c*t;
>> [xmax,fmax,ea,iter]=goldmax(z,0,8)

xmax =
    3.8317
fmax =
  192.8609
ea =
  6.9356e-005
iter =
    29
```

7.16 Each iteration of the golden-section search reduces the interval by a factor of $1/\phi$. Hence, if we start with an interval, $E_a^0 = \Delta x_0 = x_u - x_l$, the interval after the nth iteration will be

$$E_a^n = \frac{\Delta x^0}{\phi^n}$$

Each iteration of the golden-section search reduces the interval by a factor of $1/\phi$. Hence, if we start with an interval, $\Delta x^0 = x_u - x_l$, the interval after the nth iteration will be

$$\Delta x^n = \frac{\Delta x^0}{\phi^n}$$

In addition, on p. 191, we illustrate that the maximum error is $(2 - \phi)$ times the interval width. If we define the desired error as $E_{a,d}$,

$$E_{a,d} = (2-\phi)\frac{\Delta x^0}{\phi^n}$$

This equation can be solved for

$$n = \frac{\log_2\left[(2-\varphi)\dfrac{\Delta x^0}{E_{a,d}}\right]}{\log_2 \varphi}$$

or through further simplification,

$$n = \frac{\log_2\left(\frac{\Delta x^0}{E_{a,d}}\right)}{\log_2 \varphi} - 2$$

The following function implements n iterations of the golden section search:

```
function [x,fx]=prob0716(f,xl,xu,Ead,varargin)
% prob0716: minimization golden section search
%   [x,fx]=prob0716(f,xl,xu,Ead,p1,p2,...):
%      uses golden section search to find the minimum of f
%      within a prescribed tolerance
% input:
%   f = name of function
%   xl, xu = lower and upper guesses
%   Ead = desired absolute error (default = 0.000001)
%   p1,p2,... = additional parameters used by f
% output:
%   x = location of minimum
%   fx = minimum function value

if nargin<3,error('at least 3 input arguments required'),end
if nargin<4|isempty(Ead), Ead=0.000001;end
phi=(1+sqrt(5))/2;
n=ceil(log2((xu-xl)/Ead)/log2(phi)-2);
if n<1,n=1;end
for i = 1:n
  d = (phi-1)*(xu - xl);
  x1 = xl + d; x2 = xu - d;
  if f(x1,varargin{:}) < f(x2,varargin{:})
    xopt = x1;
    xl = x2;
  else
    xopt = x2;
    xu = x1;
  end
end
x=xopt;fx=f(xopt,varargin{:});
```

Here is a session that uses this function to solve the minimization from Example 7.2.

```
>> format long
>> f=@(x) x^2/10-2*sin(x);
>> [x,fx]=prob0716(f,0,4,0.0001)

x =
   1.42752749558362
fx =
  -1.77572565250482
```

7.17

```
function [xopt,fopt]=prob0713(func,xlow,xhigh,es,maxit,varargin)
% prob0717: minimization parabolic interpolation
%   [xopt,fopt]=prob0717(func,xlow,xhigh,es,maxit,p1,p2,...):
%      uses parabolic interpolation to find the minimum of f
% input:
%   f = name of function
%   xl, xu = lower and upper guesses
%   es = desired relative error (default = 0.0001%)
%   maxit = maximum allowable iterations (default = 50)
```

```
%   p1,p2,... = additional parameters used by f
% output:
%   xopt = location of minimum
%   fopt = minimum function value

if nargin<3,error('at least 3 input arguments required'),end
if nargin<4|isempty(es), es=0.0001;end
if nargin<5|isempty(maxit), maxit=50;end
iter = 0;
x1 = xlow;   x3 = xhigh;
x2 = (x1 + x3) / 2;
f1 = func(x1,varargin{:});
f2 = func(x2,varargin{:});
f3 = func(x3,varargin{:});
if f2<f1 & f2<f3
  xoptold = x2;
  while(1)
    xopt=x2-0.5*((x2-x1)^2*(f2-f3)-(x2-x3)^2*(f2-f1))/((x2-x1)...
                                              *(f2-f3)-(x2-x3)*(f2-f1));
    fopt = func(xopt,varargin{:});
    iter = iter + 1;
    if xopt > x2
      x1 = x2;
      f1 = f2;
    else
      x3 = x2;
      f3 = f2;
    end
    x2 = xopt; f2 = fopt;
    if xopt~=0,ea=abs((xopt - xoptold) / xopt) * 100;end
    xoptold = xopt;
    if ea<=es | iter>=maxit,break,end
  end
else
  error('bracket does not contain minimum')
end

>> f=@(x) x^2/10-2*sin(x);
>> [x,fx]=prob0717(f,0,4)
x =
    1.4276
fx =
   -1.7757
```

7.18 The first iteration of the golden-section search can be implemented as

$$d = \frac{\sqrt{5}-1}{2}(4-2) = 1.2361$$

$$x_1 = 2 + 1.2361 = 3.2361$$

$$x_2 = 4 - 1.2361 = 2.7639$$

$$f(x_2) = f(2.7639) = -6.1303$$

$$f(x_1) = f(3.2361) = -5.8317$$

Because $f(x_2) < f(x_1)$, the minimum is in the interval defined by x_l, x_2, and x_1 where x_2 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1 - 0.61803)\left|\frac{4-2}{2.7639}\right| \times 100\% = 27.64\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	2	-3.8608	2.7639	-6.1303	3.2361	-5.8317	4	-2.7867	1.2361	2.7639	27.64%
2	2	-3.8608	2.4721	-5.6358	2.7639	-6.1303	3.2361	-5.8317	0.7639	2.7639	17.08%
3	2.4721	-5.6358	2.7639	-6.1303	2.9443	-6.1776	3.2361	-5.8317	0.4721	2.9443	9.91%
4	2.7639	-6.1303	2.9443	-6.1776	3.0557	-6.1065	3.2361	-5.8317	0.2918	2.9443	6.13%

After four iterations, the process is converging on the true minimum at $x = 2.8966$ where the function has a value of $f(x) = -6.1847$.

7.19 The first iteration of the golden-section search can be implemented as

$$d = \frac{\sqrt{5}-1}{2}(60 - 0) = 37.0820$$
$$x_1 = 0 + 37.0820 = 37.0820$$
$$x_2 = 60 - 37.0820 = 22.9180$$
$$f(x_2) = f(22.9180) = 18.336$$
$$f(x_1) = f(37.0820) = 19.074$$

Because $f(x_1) > f(x_2)$, the maximum is in the interval defined by x_2, x_1, and x_u where x_1 is the optimum. The error at this point can be computed as

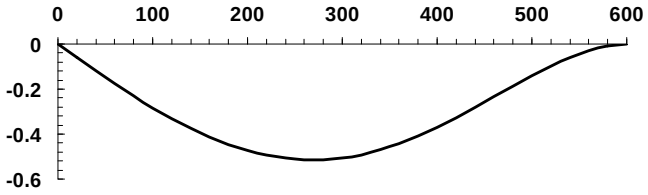
$$\varepsilon_a = (1 - 0.61803)\left|\frac{60-0}{37.0820}\right| \times 100\% = 61.80\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	0	1	22.9180	18.336	37.0820	19.074	60	4.126	37.0820	37.0820	61.80%
2	22.9180	18.336	37.0820	19.074	45.8359	15.719	60	4.126	22.9180	37.0820	38.20%
3	22.9180	18.336	31.6718	19.692	37.0820	19.074	45.8359	15.719	14.1641	31.6718	27.64%
4	22.9180	18.336	28.3282	19.518	31.6718	19.692	37.0820	19.074	8.7539	31.6718	17.08%
5	28.3282	19.518	31.6718	19.692	33.7384	19.587	37.0820	19.074	5.4102	31.6718	10.56%
6	28.3282	19.518	30.3947	19.675	31.6718	19.692	33.7384	19.587	3.3437	31.6718	6.52%
7	30.3947	19.675	31.6718	19.692	32.4612	19.671	33.7384	19.587	2.0665	31.6718	4.03%
8	30.3947	19.675	31.1840	19.693	31.6718	19.692	32.4612	19.671	1.2772	31.1840	2.53%
9	30.3947	19.675	30.8825	19.689	31.1840	19.693	31.6718	19.692	0.7893	31.1840	1.56%
10	30.8825	19.689	31.1840	19.693	31.3703	19.693	31.6718	19.692	0.4878	31.3703	0.96%

After ten iterations, the process falls below the stopping criterion and the result is converging on the true maximum at $x = 31.3713$ where the function has a value of $y = 19.6934$.

7.20 (a) A graph indicates the minimum at about $x = 270$.



(b) Golden section search,

$$d = \frac{\sqrt{5}-1}{2}(600-0) = 370.820$$
$$x_1 = 0 + 370.820 = 370.820$$
$$x_2 = 600 - 370.820 = 229.180$$
$$f(x_2) = f(229.180) = -0.5016$$
$$f(x_1) = f(370.820) = -0.4249$$

Because $f(x_2) < f(x_1)$, the minimum is in the interval defined by x_l, x_2, and x_1 where x_2 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1-0.61803)\left|\frac{600-0}{370.820}\right| \times 100\% = 100\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	0	0	229.180	-0.5016	370.820	-0.4249	600	0	370.820	229.1796	100.00%
2	0	0	141.641	-0.3789	229.180	-0.5016	370.820	-0.4249	229.180	229.1796	61.80%
3	141.641	-0.3789	229.180	-0.5016	283.282	-0.5132	370.820	-0.4249	141.641	283.2816	30.90%
4	229.180	-0.5016	283.282	-0.5132	316.718	-0.4944	370.820	-0.4249	87.539	283.2816	19.10%
5	229.180	-0.5016	262.616	-0.5149	283.282	-0.5132	316.718	-0.4944	54.102	262.6165	12.73%
6	229.180	-0.5016	249.845	-0.5121	262.616	-0.5149	283.282	-0.5132	33.437	262.6165	7.87%
7	249.845	-0.5121	262.616	-0.5149	270.510	-0.5151	283.282	-0.5132	20.665	270.5098	4.72%
8	262.616	-0.5149	270.510	-0.5151	275.388	-0.5147	283.282	-0.5132	12.772	270.5098	2.92%
9	262.616	-0.5149	267.495	-0.5152	270.510	-0.5151	275.388	-0.5147	7.893	267.4948	1.82%
10	262.616	-0.5149	265.631	-0.5151	267.495	-0.5152	270.510	-0.5151	4.878	267.4948	1.13%
11	265.631	-0.5151	267.495	-0.5152	268.646	-0.5152	270.510	-0.5151	3.015	268.6465	0.69%

After eleven iterations, the process falls below the stopping criterion and the result is converging on the true minimum at $x = 268.3281$ where the function has a value of $y = -0.51519$.

7.21 The velocity of a falling object with an initial velocity and first-order drag can be computed as

$$v = v_0 e^{-(c/m)t} + \frac{mg}{c}\left(1 - e^{-(c/m)t}\right)$$

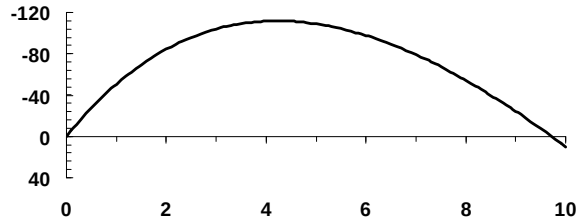
The vertical distance traveled can be determined by integration

$$z = z_0 + \int_0^t v_0 e^{-(c/m)t} + \frac{mg}{c}\left(1 - e^{-(c/m)t}\right) dt$$

where negative z is distance upwards. Assuming that $z_0 = 0$, evaluating the integral yields

$$z = \frac{mg}{c}t + \frac{m}{c}\left(v_0 - \frac{mg}{c}\right)\left(1 - e^{-(c/m)t}\right)$$

Therefore the solution to this problem amounts to determining the minimum of this function (since the most negative value of z corresponds to the maximum height). This function can be plotted using the given parameter values. As in the following graph (note that the ordinate values are plotted in reverse), the maximum height occurs after about 4.2 s and appears to be about 110 m.



Here is the result of using the golden-section search to determine the maximum height

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	0	0	1.9098	-82.008	3.0902	-105.171	5	-108.922	3.0902	3.0902	61.80%
2	1.9098	-82.008	3.0902	-105.171	3.8197	-111.014	5	-108.922	1.9098	3.8197	30.90%
3	3.0902	-105.171	3.8197	-111.014	4.2705	-111.790	5.0000	-108.922	1.1803	4.2705	17.08%
4	3.8197	-111.014	4.2705	-111.790	4.5492	-111.272	5.0000	-108.922	0.7295	4.2705	10.56%
5	3.8197	-111.014	4.0983	-111.735	4.2705	-111.790	4.5492	-111.272	0.4508	4.2705	6.52%
6	4.0983	-111.735	4.2705	-111.790	4.3769	-111.680	4.5492	-111.272	0.2786	4.2705	4.03%
7	4.0983	-111.735	4.2047	-111.804	4.2705	-111.790	4.3769	-111.680	0.1722	4.2047	2.53%
8	4.0983	-111.735	4.1641	-111.791	4.2047	-111.804	4.2705	-111.790	0.1064	4.2047	1.56%
9	4.1641	-111.791	4.2047	-111.804	4.2299	-111.804	4.2705	-111.790	0.0658	4.2047	0.97%
10	4.1641	-111.791	4.1892	-111.801	4.2047	-111.804	4.2299	-111.804	0.0407	4.2047	0.60%

After ten iterations, the process falls below a stopping of 1% criterion and the result is converging on the true minimum at $x = 4.2167$ where the function has a value of $y = -111.805$.

7.22 The inflection point corresponds to the point at which the derivative of the normal distribution is a minimum. The derivative can be evaluated as

$$\frac{dy}{dx} = -2xe^{-x^2}$$

Starting with initial guesses of $x_l = 0$ and $x_u = 2$, the golden-section search method can be implemented as

$$d = \frac{\sqrt{5}-1}{2}(2-0) = 1.2361$$

$$x_1 = 0 + 1.2361 = 1.2361 \qquad x_2 = 2 - 1.2361 = 0.7639$$

$$f(x_2) = f(0.7639) = -2(0.7639)e^{-(0.7639)^2} = -0.8524$$

$$f(x_1) = f(1.2361) = -2(1.2361)e^{-(1.2361)^2} = -0.5365$$

Because $f(x_2) < f(x_1)$, the minimum is in the interval defined by x_l, x_2, and x_1 where x_2 is the optimum. The error at this point can be computed as

$$\varepsilon_a = (1 - 0.61803)\left|\frac{2-0}{0.7639}\right| \times 100\% = 100\%$$

The process can be repeated and all the iterations summarized as

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d	x_{opt}	ε_a
1	0	0.0000	0.7639	-0.8524	1.2361	-0.5365	2	-0.0733	1.2361	0.7639	100.00%
2	0	0.0000	0.4721	-0.7556	0.7639	-0.8524	1.2361	-0.5365	0.7639	0.7639	61.80%
3	0.4721	-0.7556	0.7639	-0.8524	0.9443	-0.7743	1.2361	-0.5365	0.4721	0.7639	38.20%
4	0.4721	-0.7556	0.6525	-0.8525	0.7639	-0.8524	0.9443	-0.7743	0.2918	0.6525	27.64%
5	0.4721	-0.7556	0.5836	-0.8303	0.6525	-0.8525	0.7639	-0.8524	0.1803	0.6525	17.08%
6	0.5836	-0.8303	0.6525	-0.8525	0.6950	-0.8575	0.7639	-0.8524	0.1115	0.6950	9.91%
7	0.6525	-0.8525	0.6950	-0.8575	0.7214	-0.8574	0.7639	-0.8524	0.0689	0.6950	6.13%
8	0.6525	-0.8525	0.6788	-0.8564	0.6950	-0.8575	0.7214	-0.8574	0.0426	0.6950	3.79%
9	0.6788	-0.8564	0.6950	-0.8575	0.7051	-0.8578	0.7214	-0.8574	0.0263	0.7051	2.31%
10	0.6950	-0.8575	0.7051	-0.8578	0.7113	-0.8577	0.7214	-0.8574	0.0163	0.7051	1.43%
11	0.6950	-0.8575	0.7013	-0.8577	0.7051	-0.8578	0.7113	-0.8577	0.0100	0.7051	0.88%

After eleven iterations, the process falls below a stopping of 1% criterion and the result is converging on the true minimum at $x = 0.707107$ where the function has a value of $y = -0.85776$.

7.23

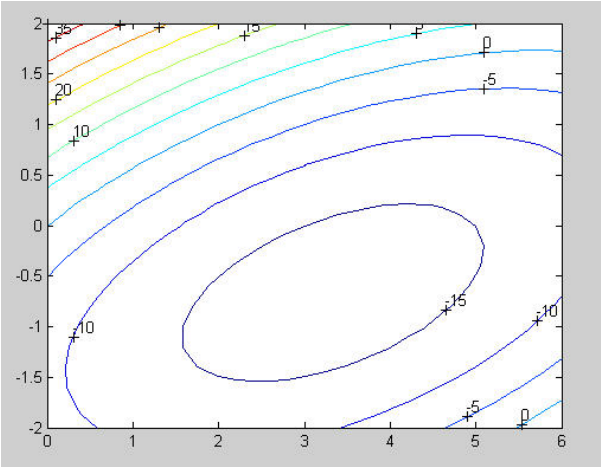
```
>> fxy=@(x) 2*x(2)^2-2.25*x(1)*x(2)-1.75*x(2)+1.5*x(1)^2;
>> [x,fval]= fminsearch(fxy,[0,0])
x =
    0.5676    0.7568
fval =
   -0.6622
```

7.24

```
>> fxy=@(x) -(4*x(1)+2*x(2)+x(1)^2-2*x(1)^4+2*x(1)*x(2)-3*x(2)^2);
>> [x,fval]= fminsearch(fxy,[0,0])
x =
    0.9676    0.6559
fval =
   -4.3440
```

7.25 (a)

```
>> [x,y] = meshgrid(0:.2:6,-2:.2:2);
>> z=-8*x+x.^2+12*y+4*y.^2-2*x.*y;
>> cs=contour(x,y,z);clabel(cs)
```



(b) and (c)

```
>> fxy=@(x) -8*x(1)+x(1)^2+12*x(2)+4*x(2)^2-2*x(1)*x(2);
>> [x,fval]= fminsearch(fxy,[0,0])
x =
    3.3333   -0.6666
fval =
  -17.3333
```

7.26 This problem can be solved in a number of different ways. For example, using the golden section search, the result is

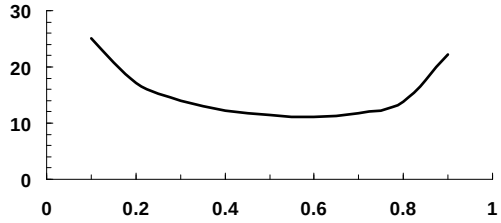
i	c_l	$g(c_l)$	c_2	$g(c_2)$	c_1	$g(c_1)$	c_u	$g(c_u)$	d	c_{opt}	ε_a
1	0.0000	0.0000	3.8197	0.2330	6.1803	0.1310	10.0000	0.0641	6.1803	3.8197	100.00%
2	0.0000	0.0000	2.3607	0.3350	3.8197	0.2330	6.1803	0.1310	3.8197	2.3607	100.00%
3	0.0000	0.0000	1.4590	0.3686	2.3607	0.3350	3.8197	0.2330	2.3607	1.4590	100.00%
4	0.0000	0.0000	0.9017	0.3174	1.4590	0.3686	2.3607	0.3350	1.4590	1.4590	61.80%
5	0.9017	0.3174	1.4590	0.3686	1.8034	0.3655	2.3607	0.3350	0.9017	1.4590	38.20%
6	0.9017	0.3174	1.2461	0.3593	1.4590	0.3686	1.8034	0.3655	0.5573	1.4590	23.61%
7	1.2461	0.3593	1.4590	0.3686	1.5905	0.3696	1.8034	0.3655	0.3444	1.5905	13.38%
8	1.4590	0.3686	1.5905	0.3696	1.6718	0.3688	1.8034	0.3655	0.2129	1.5905	8.27%
9	1.4590	0.3686	1.5403	0.3696	1.5905	0.3696	1.6718	0.3688	0.1316	1.5905	5.11%
10	1.5403	0.3696	1.5905	0.3696	1.6216	0.3694	1.6718	0.3688	0.0813	1.5905	3.16%
11	1.5403	0.3696	1.5713	0.3696	1.5905	0.3696	1.6216	0.3694	0.0502	1.5713	1.98%
12	1.5403	0.3696	1.5595	0.3696	1.5713	0.3696	1.5905	0.3696	0.0311	1.5713	1.22%
13	1.5595	0.3696	1.5713	0.3696	1.5787	0.3696	1.5905	0.3696	0.0192	1.5713	0.75%

Thus, after 13 iterations, the method is converging on the true value of $c = 1.5679$ which corresponds to a maximum specific growth rate of $g = 0.36963$.

7.27 This is a straightforward problem of varying x_A in order to minimize

$$f(x_A) = \left(\frac{1}{(1-x_A)^2}\right)^{0.6} + 6\left(\frac{1}{x_A}\right)^{0.6}$$

First, the function can be plotted versus x_A



The result indicates a minimum between 0.5 and 0.6. Using golden section search or MATLAB yields a minimum of 0.587683.

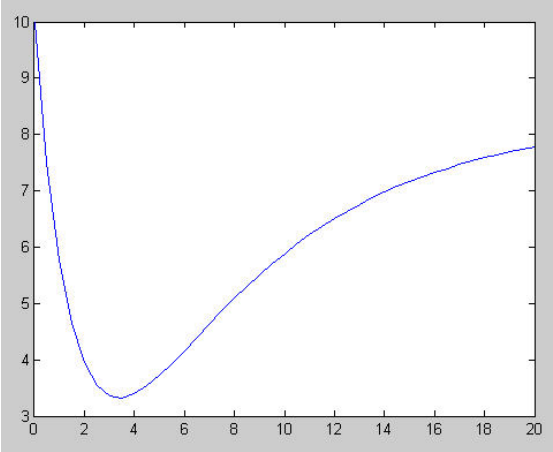
```
>> format long
>> fxa=@(xa) (1/(1-xa)^2)^0.6+6*(1/xa)^0.6;
>> [x fx]=fminbnd(fxa,0.2,1)
x =
   0.58767394491187
fx =
  11.14951022048255
```

7.28 As shown below, MATLAB gives: $x = 0.5$, $y = 0.8$ and $f_{\min} = -0.85$.

```
>> fxy=@(x) 5*x(1)^2-5*x(1)*x(2)+2.5*x(2)^2-x(1)-1.5*x(2);
>> [x,fval]=fminsearch(fxy,[0,0])
x =
    0.5000    0.8000
fval =
   -0.8500
```

7.29 A plot of the function indicates a minimum at about $t = 3.3$.

```
>> os=10;kd=0.1;ka=0.6;
>> ks=0.05;Lo=50;Sb=1;
>> ft=@(t) os-kd*Lo/(kd+ks-ka)*(exp(-ka*t)-exp(-(kd+ks)*t)) ...
                                                    -Sb/ka*(1-exp(-ka*t));
>> t=0:0.5:20;
>> o=ft(t);
>> plot(t,o)
```

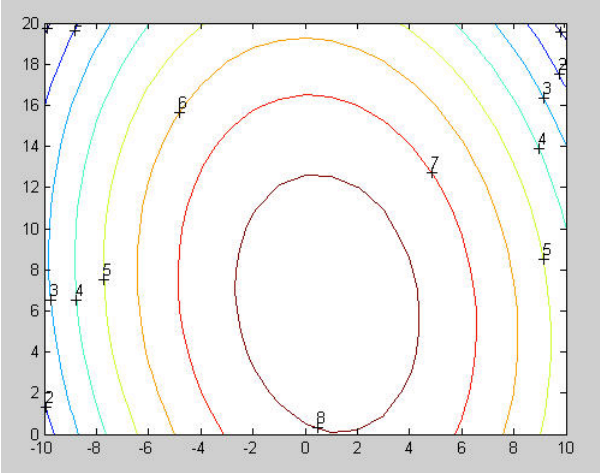


MATLAB can be used to determine that a minimum of $o = 3.3226$ occurs at a value of $t = 3.3912$.

```
>> [tmin omin]=fminbnd(ft,0,10)
tmin =
    3.3912
omin =
    3.3226
```

7.30 This problem can be solved graphically by using MATLAB to generate a contour plot of the function. As can be seen, a minimum occurs at approximately $x = 1$ and $y = 7$.

```
>> [x,y] = meshgrid(-10:10,0:20);
>> c=7.9+0.13*x+0.21*y-0.05*x.^2-0.016*y.^2-0.007*x.*y;
>> cs=contour(x,y,c);clabel(cs)
```



We can use MATLAB to determine a maximum of 8.6249 at $x = 0.853689$ and $y = 6.375787$.

```
>> c=@(x) -(7.9+0.13*x(1)+0.21*x(2)-0.05*x(1).^2...
                                  -0.016*x(2).^2-0.007*x(1).*x(2));
>> [x,fval]= fminsearch(c,[0,0])
x =
   0.85368944970842   6.37578748109051
fval =
  -8.62494446205611
```

7.31 The solution can be developed as a MATLAB script yielding $F = 1.709$ at $x = 0.6364$,

```
clear, clc, format long
e0=8.85e-12;q=2e-5;Q=q;a=0.9;
F=@(x) -(1/(4*pi*e0)*q*Q*x/(x^2+a^2)^(3/2));
[x,Fx]=fminsearch(F,0.5)

x =
   0.63642578125000
Fx =
  -1.70910974594861
```

7.32 This is a trick question. Because of the presence of $(1 - s)$ in the denominator, the function will experience a division by zero at the maximum. This can be rectified by merely canceling the $(1 - s)$ terms in the numerator and denominator to give

$$T = \frac{15s}{4s^2 - 3s + 4}$$

The following script uses `fminbnd` to determine that the maximum of $T = 3$ occurs at $s = 1$.

```
clear, clc, format long
T=@(s) -(15*s/(4*s^2-3*s+4));
[smin,Tmin]=fminbnd(T,0,4)
smin =
   0.99998407348199
Tmin =
  -2.99999999939122
```

7.33 (a) The `fminbnd` function can be used to determine the solution as in the following script:

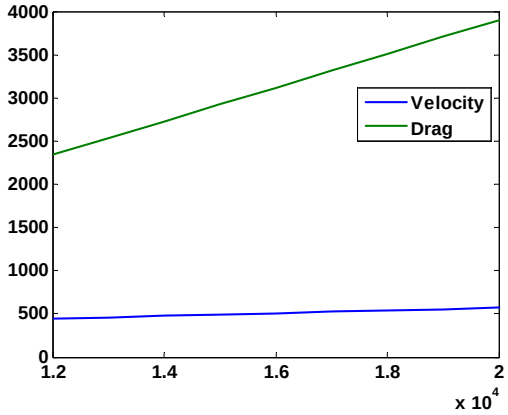
```
clear,clc,format short g
D=@(V) 0.01*(0.6)*V^2+0.95/0.6*(16000/V)^2;
[Vmin,Dmin]=fminbnd(D,0,1200)
Vmin =
       509.82
Dmin =
         3119
```

(b) The following script can be used to generate the sensitivity analysis:

```
clear,clc,format short g
W=[12000:1000:20000];
sigma=0.6;
for i=1:length(W)
  [Vmin(i),Dmin(i)]=fminbnd(@(V) 0.01*sigma*V^2+0.95/sigma*(W(i)/V)^2,0,1200);
end
z=[W;Vmin;Dmin];
fprintf('     Weight   Velocity   Drag\n')
fprintf('%10.0f %10.3f %10.3f\n',z)
clf
plot(W,Vmin,'--',W,Dmin)
legend('Velocity','Drag','location','best')
```

The results indicate that the minimum drag is fairly linear for the specified range.

```
     Weight   Velocity   Drag
     12000    441.515   2339.231
     13000    459.544   2534.167
     14000    476.891   2729.102
     15000    493.629   2924.038
     16000    509.818   3118.974
     17000    525.508   3313.910
     18000    540.744   3508.846
     19000    555.561   3703.782
     20000    569.994   3898.718
```



7.34 MATLAB can be used to determine the solution as

```
>> format long
>> fx=@(x) -(0.4/sqrt(1+x^2)-sqrt(1+x^2)*(1-0.4/(1+x^2))+x);
>> [x,f]=fminbnd(fx,0,2)
x =
   1.05185912272736
```

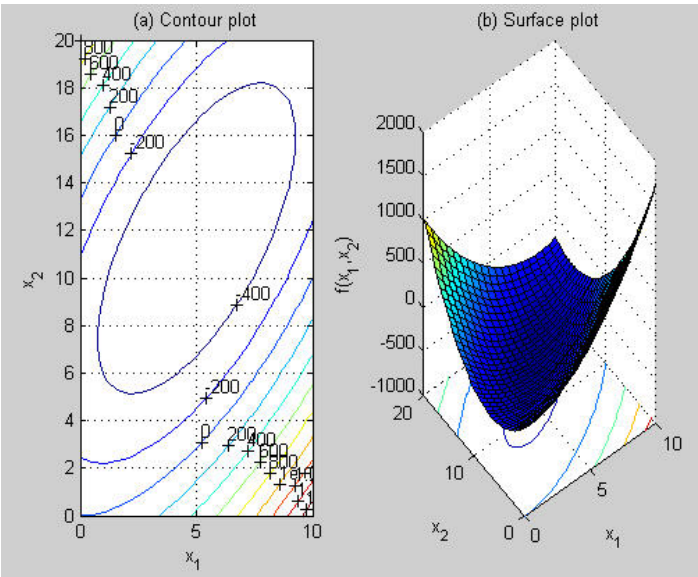
```
f =
  -0.15172444148082
```

7.35 The potential energy function can be written as

$$PE = 0.5k_a x_1^2 + 0.5k_b(x_1 - x_2)^2 - Fx_2$$

Contour and surface plots can be generated with the following script,

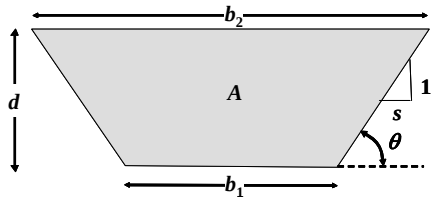
```
ka=20;kb=15;F=100;
x=linspace(0,10,20);y=linspace(0,20,40);
[x1,x2] = meshgrid(x,y);
Z=0.5*ka*x1.^2+0.5*kb*(x2-x1).^2-F*x2;
subplot(1,2,1);
cs=contour(x1,x2,Z);clabel(cs);
xlabel('x_1');ylabel('x_2');
title('(a) Contour plot');grid;
subplot(1,2,2);
cs=surfc(x1,x2,Z);
zmin=floor(min(Z));
zmax=ceil(max(Z));
xlabel('x_1');ylabel('x_2');zlabel('f(x_1,x_2)');
title('(b) Mesh plot');
```



The values of x_1 and x_2 that minimize the PE function can be determined as

```
>> PE=@(x) 0.5*ka*x(1).^2+0.5*kb*(x(2)-x(1)).^2-F*x(2);
>> [xmin,PEmin]=fminsearch(PE,[5,5])
xmin =
   4.99998098312971  11.66668727728067
PEmin =
   -5.833333333179395e+002
```

7.36 Using the diagram shown below, a number of formulas can be developed:



$$\theta = \tan^{-1}\frac{1}{s} \tag{1}$$

$$P = b_1 + 2d\sqrt{1+s^2} \tag{2}$$

$$A = (b_1 + sd)d \tag{3}$$

$$b_2 = b_1 + 2sd \tag{4}$$

We can solve Eq. 3 for b_1,

$$b_1 = \frac{A}{d} - sd \tag{5}$$

and substitute the result into Eq. 2 to give,

$$P = \frac{A}{d} + d\left(2\sqrt{1+s^2} - s\right) \tag{6}$$

Given a value for A, we can determine the values of d and s for this two-dimensional function,

```
>> P=@(x) 50/x(1)+x(1)*(2*sqrt(1+x(2).^2)-x(2));
>> [xmin,fmin]=fminsearch(P,[5,1])
xmin =
    5.3729    0.5773
fmin =
   18.6121
```

The optimal parameters are $d = 5.3729$ and $s = 0.5773$. Using Eq. 1 gives

$$\theta = \tan^{-1}\frac{1}{0.5773} = 1.0472 \text{ radians} \times \frac{180°}{\pi} = 60°$$

Thus, this specific application indicates that a 60^o angle yields the minimum wetted perimeter. The other dimensions can be computed as (Eqs. 5 and 4),

$$b_1 = \frac{50}{5.3729} - 0.5773(5.3729) = 6.2041$$

$$b_2 = 6.2041 + 2(0.5773)(5.3729) = 12.40807$$

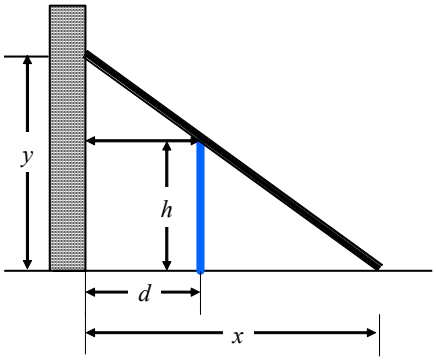
The verification of whether this result is universal can be attained inductively or deductively. The inductive approach involves trying several different desired areas in conjunction with our MATLAB solution. As long as the desired area is greater than 0, the result for the optimal design will be 60^o.

The deductive verification involves calculus. If both A and d are constants and s is a variable, the condition for the minimum perimeter is $dP/ds = 0$. Differentiating Eq. 6 with respect to s and setting the resulting equation to zero,

$$\frac{dP}{ds} = d\left(\frac{2s}{\sqrt{1+s^2}} - 1\right) = 0$$

The term in parentheses can be solved for $s = 1/\sqrt{3}$. Using Eq. 1, this corresponds to $\theta = 60^o$.

7.37 We have to first define some additional variables,



The problem amounts to minimizing the length of the ladder,

$$L = \sqrt{x^2 + y^2} \tag{1}$$

Using similar triangles

$$\frac{y}{x} = \frac{h}{x-d}$$

which can be solved for

$$y = \frac{hx}{x-d}$$

This value can be substituted into Eq. 1 along with the values of $h = d = 4$ to give

$$L = \sqrt{x^2 + \left(\frac{4x}{x-4}\right)^2}$$

MATLAB can then be used to determine the value of x that minimizes the length,

```
>> fx=@(x) sqrt(x^2+(4*x/(x-4))^2);
>> [xmin,fmin]=fminbnd(fx,1,10)
xmin =
    8.0000
fmin =
   11.3137
```

Therefore, the shortest ladder is L = 11.3137 m, x = 8, and y = 4(8)/(8 – 4) = 8.

7.38 The following script generates the plot of L versus a range of α's:

```
alphad=[45:5:135];
alpha=alphad*pi/180;
for i=1:length(alpha)
  [t,Lmin(i)]=fminsearch(@(x) 2/sin(x)+2/sin(pi-alpha(i)-x),0.3);
end
plot(alphad,Lmin)
```