CHAPTER 8

8.1 Ideal gas law:

$$v = \frac{RT}{p} = \frac{0.082054(400)}{2.5} = 13.12864$$

van der Waals equation:

$$f(v) = \left(p + \frac{a}{v^2}\right)(v - b) - RT$$

$$f(v) = \left(2.5 + \frac{14.09}{v^2}\right)(v - 0.0994) - 0.082054(400)$$

Any of the techniques in Chaps 5 or 6 can be used to determine the root as v = 12.7908 L/mol. The Newton-Raphson method would be a good choice because (a) the equation is relatively simple to differentiate and (b) the ideal gas law provides a good initial guess. The Newton-Raphson method can be formulated as

$$v_{i+1} = v_i - \frac{\left(p + \frac{a}{v_i^2}\right)(v_i - b) - RT}{\left(p + \frac{a}{v_i^2}\right) - (v_i - b)\frac{2a}{v_i^3}}$$

Using the ideal gas law for the initial guess results in an accurate root determination in a few iterations:

i	x_i	$f(x_i)$	$f'(x_i)$	ε_a
0	13.12864	0.816601	2.419491	
1	12.79113	0.000711	2.415221	2.6386%
2	12.79084	5.7E-10	2.415217	0.0023%
3	12.79084	0	2.415217	0.0000%

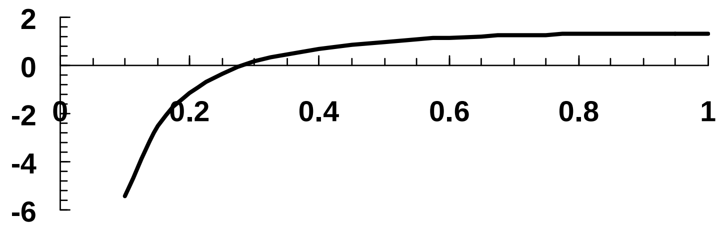
8.2 The function to be solved is

$$f(R) = \ln\frac{1 + R(1 - X_{Af})}{R(1 - X_{Af})} - \frac{R + 1}{R[1 + R(1 - X_{Af})]} = 0$$

or substituting X_{Af} = 0.96,

$$f(R) = \ln\frac{1 + 0.04R}{R(0.04)} - \frac{R + 1}{R(1 + 0.04R)} = 0$$

A plot of the function indicates a root at about R = 0.3

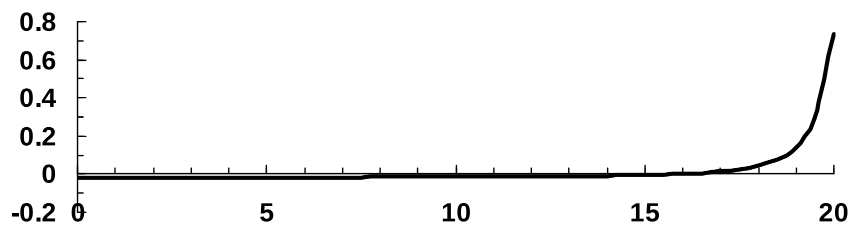


Bisection with initial guesses of 0.01 and 1 can be used to determine a root of 0.28194 after 16 iterations with ε_a = 0.005%.

8.3 The function to be solved is

$$f(x) = \frac{(4+x)}{(42-2x)^2(28-x)} - 0.016 = 0$$

(a) A plot of the function indicates a root at about x = 16.



(b) The shape of the function indicates that false position would be a poor choice (recall Fig. 5.14). Bisection with initial guesses of 0 and 20 can be used to determine a root of 15.85938 after 8 iterations with ε_a = 0.493%. Note that false position would have required 68 iterations to attain comparable accuracy.

i	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0	20	10	-0.01592	-0.01439	0.000229	100.000%
2	10	20	15	-0.01439	-0.00585	8.42E-05	33.333%
3	15	20	17.5	-0.00585	0.025788	-0.00015	14.286%
4	15	17.5	16.25	-0.00585	0.003096	-1.8E-05	7.692%
5	15	16.25	15.625	-0.00585	-0.00228	1.33E-05	4.000%
6	15.625	16.25	15.9375	-0.00228	0.000123	-2.8E-07	1.961%
7	15.625	15.9375	15.78125	-0.00228	-0.00114	2.59E-06	0.990%
8	15.78125	15.9375	15.85938	-0.00114	-0.00052	5.98E-07	0.493%

8.4 The functions to be solved are

$$K_1 = \frac{(c_{c,0} + x_1 + x_2)}{(c_{a,0} - 2x_1 - x_2)^2(c_{b,0} - x_1)}$$

$$K_2 = \frac{(c_{c,0} + x_1 + x_2)}{(c_{a,0} - 2x_1 - x_2)(c_{d,0} - x_2)}$$

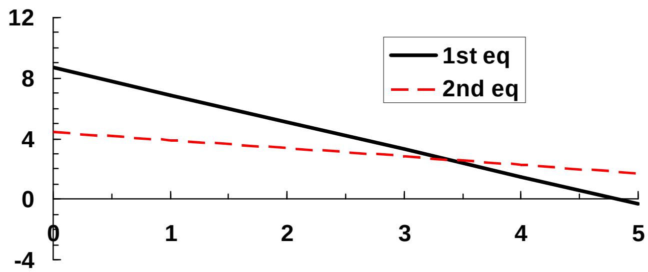
or

$$f_1(x_1, x_2) = \frac{5 + x_1 + x_2}{(50 - 2x_1 - x_2)^2 (20 - x_1)} - 4 \times 10^{-4}$$
$$f_2(x_1, x_2) = \frac{(5 + x_1 + x_2)}{(50 - 2x_1 - x_2)(10 - x_2)} - 3.7 \times 10^{-2}$$

Graphs can be generated by specifying values of x_1 and solving for x_2 using a numerical method like bisection.

first equation		second equation	
x_1	x_2	x_1	x_2
0	8.6672	0	4.4167
1	6.8618	1	3.9187
2	5.0649	2	3.4010
3	3.2769	3	2.8630
4	1.4984	4	2.3038
5	-0.2700	5	1.7227

These values can then be plotted to yield



Therefore, the root seems to be at about $x_1 = 3.3$ and $x_2 = 2.7$. Employing these values as the initial guesses for the two-variable Newton-Raphson method gives

$f_1(3.3, 2.7) = -2.36\times10^{-6}$
$f_2(3.3, 2.7) = 2.33\times10^{-5}$

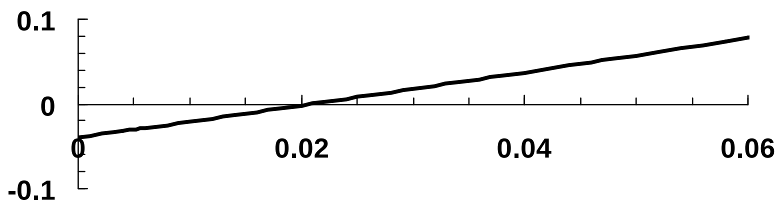
$$\frac{\partial f_1}{\partial x_1} = 9.9 \times 10^{-5} \qquad \frac{\partial f_2}{\partial x_1} = 5.185 \times 10^{-3}$$
$$\frac{\partial f_1}{\partial x_2} = 5.57 \times 10^{-5} \qquad \frac{\partial f_2}{\partial x_2} = 9.35 \times 10^{-3}$$
$$|J| = 6.37 \times 10^{-7}$$
$$x_1 = 3.3 - \frac{-2.36 \times 10^{-6}(9.35 \times 10^{-3}) - 2.33 \times 10^{-5}(5.57 \times 10^{-5})}{6.37 \times 10^{-7}} = 3.3367$$
$$x_2 = 2.7 - \frac{2.33 \times 10^{-5}(9.9 \times 10^{-5}) - (-2.36 \times 10^{-6})(5.185 \times 10^{-3})}{6.37 \times 10^{-7}} = 2.677$$

The second iteration yields $x_1 = 3.3366$ and $x_2 = 2.677$, with a maximum approximate error of 0.003%.

8.5 The function to be solved is

$$f(x) = \frac{x}{1-x}\sqrt{\frac{7}{2+x}} - 0.04 = 0$$

A plot of the function indicates a root at about $x = 0.02$.



Because the function is so linear, false position is a good choice. Using initial guesses of 0.01 and 0.03, the first iteration is

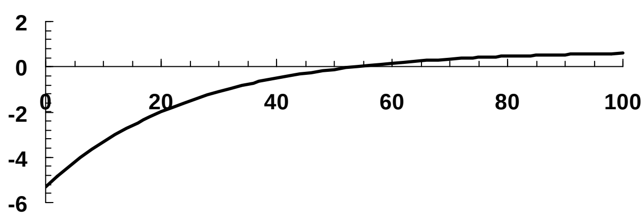
$$x_r = 0.03 - \frac{0.017432(0.01-0.03)}{-0.02115-0.017432} = 0.020964$$

After 3 iterations, the result is 0.021041 with $\varepsilon_a = 0.003\%$.

8.6 The function to be solved is

$$f(t) = 12\left(1-e^{-0.04t}\right) + 5e^{-0.04t} - 10.2 = 0$$

A plot of the function indicates a root at about $t = 55$.

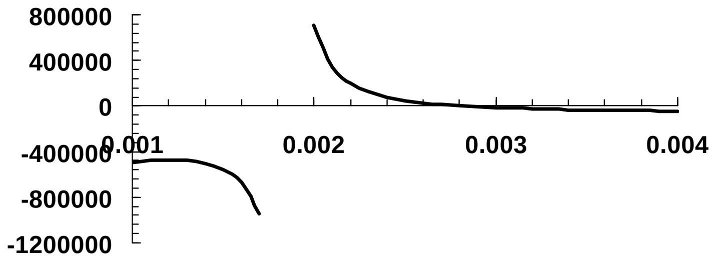


Bisection with initial guesses of 0 and 60 can be used to determine a root of 53.711 after 16 iterations with $\varepsilon_a = 0.002\%$.

8.7 Using the given values, $a = 12.5578$ and $b = 0.0018626$. Therefore, the roots problem to be solved is

$$f(v) = \frac{0.518(233)}{(v-0.0018626)} - \frac{12.5578}{v(v+0.0018626)\sqrt{233}} - 65{,}000$$

A plot indicates a root at about 0.0028.

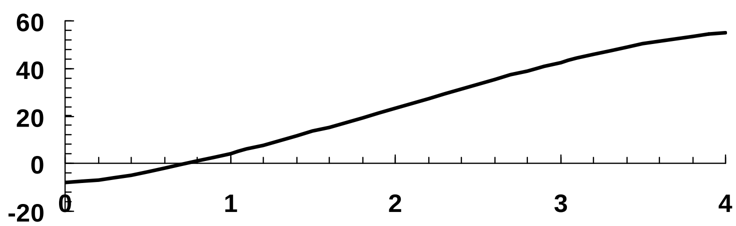


Using initial guesses of 0.002 and 0.004, bisection can be employed to determine the root as 0.002807 after 12 iterations with ε_a = 0.017%. The mass of methane contained in the tank can be computed as 3/0.00275 = 1068.6 kg.

8.8 Using the given values, the roots problem to be solved is

$$f(h) = \left[4\cos^{-1}\left(\frac{2-h}{2}\right) - (2-h)\sqrt{4h-h^2}\right]5 - 8 = 0$$

A plot indicates a root at about 0.8.

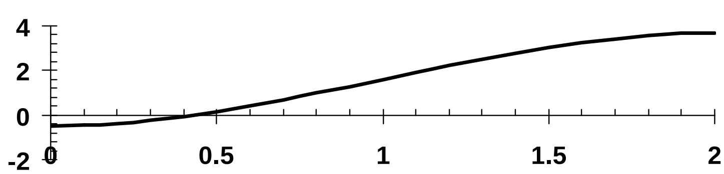


A numerical method can be used to determine that the root is 0.74002.

8.9 Using the given values, the roots problem to be solved is

$$f(h) = \frac{\pi h^2(3-h)}{3} - 0.75 = 0$$

A plot indicates a root at about 0.45.

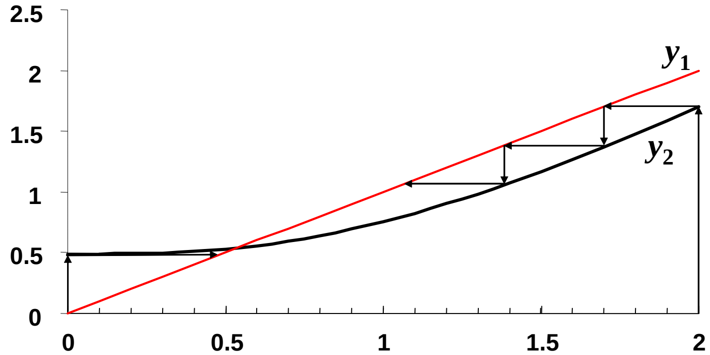


A numerical method can be used to determine that the root is 0.43112.

8.10 The best way to approach this problem is to use the graphical method displayed in Fig. 6.3. For the first version, we plot

$$y_1 = h \quad \text{and} \quad y_2 = \sqrt{\frac{h^3 + 0.7162}{3}}$$

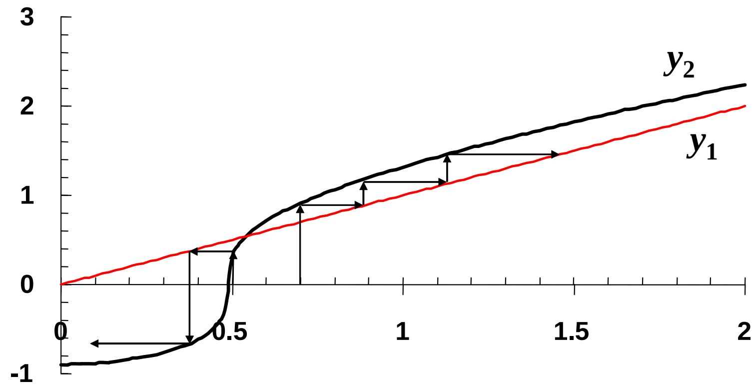
versus the range of h. Note that for the sphere, h ranges from 0 to $2r$. As displayed below, this version will always converge.



For the second version, we plot

$y_1 = h$ and $y_2 = \sqrt[3]{3h^2 - 0.7162}$

versus the range of h. As displayed below, this version is not convergent.



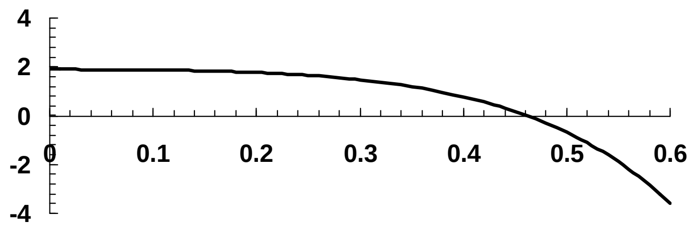
8.11 Substituting the parameter values yields

$$10\frac{\varepsilon^3}{1-\varepsilon} = 150\frac{1-\varepsilon}{1000} + 1.75$$

This can be rearranged and expressed as a roots problem

$$f(\varepsilon) = 0.15(1-\varepsilon) + 1.75 - 10\frac{\varepsilon^3}{1-\varepsilon} = 0$$

A plot of the function suggests a root at about 0.46.



But suppose that we do not have a plot. How do we come up with a good initial guess? The void fraction (the fraction of the volume that is not solid; i.e. consists of voids) varies between 0 and 1. As can be seen, a value of 1 (which is physically unrealistic) causes a division by zero. Therefore, two physically-based initial guesses can be chosen as 0 and 0.99. Note that the zero is not physically realistic either, but since it does not cause any mathematical difficulties, it is OK. Applying bisection yields a result of $\varepsilon = 0.461857$ in 15 iterations with an absolute approximate relative error of 6.54×10^{-3}%.

8.12 (a) The Reynolds number can be computed as

$$\text{Re} = \frac{\rho VD}{\mu} = \frac{1.23(40)0.005}{1.79\times10^{-5}} = 13743$$

In order to find f, we must determine the root of the function $g(f)$

$$g(f) = -2.0\log\left(\frac{0.0000015}{3.7(0.005)} + \frac{2.51}{13743\sqrt{f}}\right) - \frac{1}{\sqrt{f}} = 0$$

As mentioned in the problem a good initial guess can be obtained from the Blasius formula

$$f = \frac{0.316}{13743^{0.25}} = 0.029185$$

Using this guess, a root of 0.028968 can be obtained with an approach like the modified secant method. This result can then be used to compute the pressure drop as

$$\Delta p = 0.028968\frac{0.2(1.23)(40)^2}{2(0.005)} = 1140.17\text{ Pa}$$

(b) For the rougher steel pipe, we must determine the root of

$$g(f) = -2.0\log\left(\frac{0.000045}{3.7(0.005)} + \frac{2.51}{13743\sqrt{f}}\right) - \frac{1}{\sqrt{f}} = 0$$

Using the same initial guess as in **(a)**, a root of 0.04076 can be obtained. This result can then be used to compute the pressure drop as

$$\Delta p = 0.04076\frac{0.2(1.23)(40)^2}{2(0.005)} = 1604.25\text{ Pa}$$

Thus, as would be expected, the pressure drop is higher for the rougher pipe.

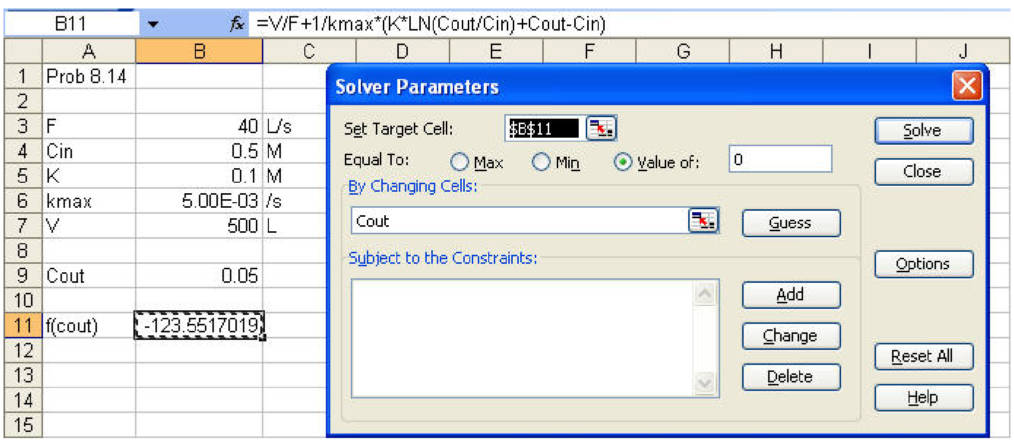
8.13 The integral can be evaluated as

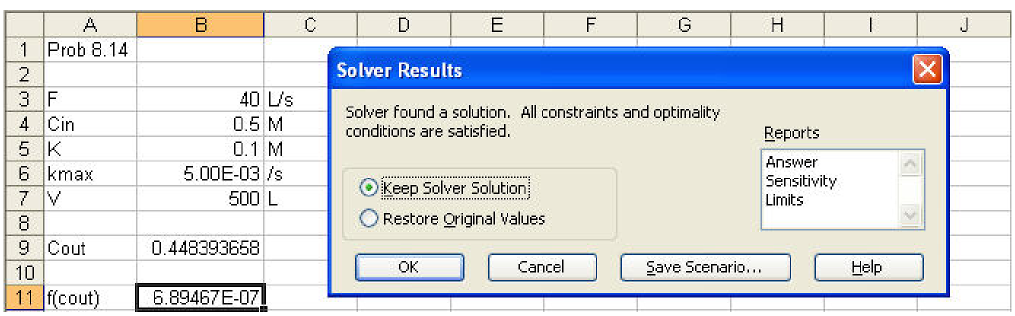
$$-\int_{C_{\text{in}}}^{C_{\text{out}}} \frac{K}{k_{\max} C} + \frac{1}{k_{\max}} dC = -\frac{1}{k_{\max}} \left[K \ln\left(\frac{C_{\text{out}}}{C_{\text{in}}} \right) + C_{\text{out}} - C_{\text{in}} \right]$$

Therefore, the problem amounts to finding the root of

$$f(C_{\text{out}}) = \frac{V}{F} + \frac{1}{k_{\max}} \left[K \ln\left(\frac{C_{\text{out}}}{C_{\text{in}}} \right) + C_{\text{out}} - C_{\text{in}} \right]$$

Excel solver can be used to find the root:

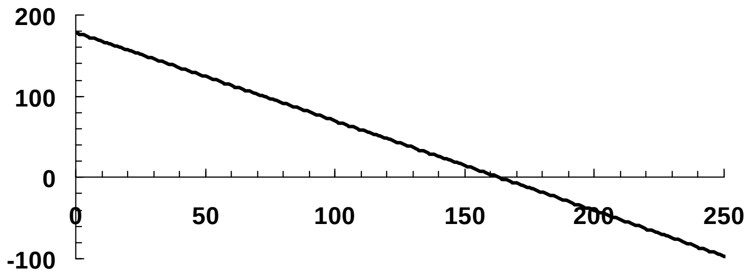




8.14 The function to be solved is

$$f(P/A) = \frac{250}{1 + 0.4/\cos[25\sqrt{(P/A)/200{,}000}\,]} - \frac{P}{A}$$

A plot of the function indicates a root at about $P/A = 163$.

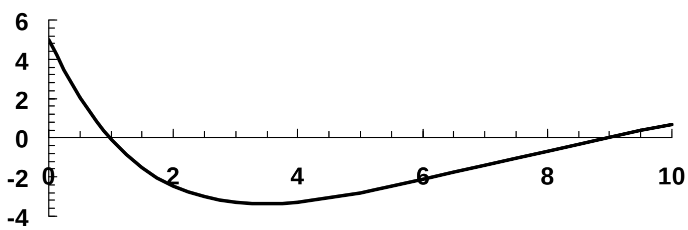


A numerical method can be used to determine that the root is 163.4429.

8.15 (a) This problem can be solved by determining the root of

$$f(x) = 10 - 20\left(e^{-0.15x} - e^{-0.5x}\right) - 5 = 0$$

A plot of the function indicates a root at about $x = 1$ km.



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

i	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0	5	2.5	5	-3.01569	-15.0784	
2	0	2.5	1.25	5	-0.87535	-4.37677	100.00%
3	0	1.25	0.625	5	1.422105	7.110527	100.00%
4	0.625	1.25	0.9375	1.422105	0.139379	0.198212	33.33%
5	0.9375	1.25	1.09375	0.139379	-0.39867	-0.05557	14.29%

After 10 iterations, the root is determined as $x = 0.971679688$ with an approximate error of 0.5%.

(b) The location of the minimum can be determined by differentiating the original function to yield

$$f'(x) = -0.15e^{-0.15x} + 0.5e^{-0.5x} = 0$$

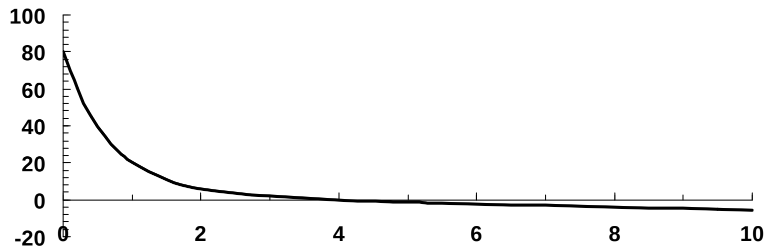
The root of this function can be determined as $x = 3.44$ km. The value of the minimum concentration can then be computed as

$$c = 10 - 20\left(e^{-0.15(3.44)} - e^{-0.5(3.44)}\right) = 1.6433$$

8.16 (a) This problem can be solved by determining the root of

$$f(t) = 75e^{-1.5t} + 20e^{-0.075t} - 15 = 0$$

A plot of the function indicates a root at about $t = 4$.



The Newton-Raphson method can be formulated as

$$t_{i+1} = t_i - \frac{75e^{-1.5t_i} + 20e^{-0.075t_i} - 15}{-112.5e^{-1.5t_i} - 1.5e^{-0.075t_i}}$$

Using the initial guess of $t = 6$, an accurate root determination can be obtained in a few iterations:

i	x_i	$f(x_i)$	$f'(x_i)$	εa
0	6	-2.23818	-0.97033	
1	3.693371	0.455519	-1.57879	62.45%
2	3.981896	0.02752	-1.39927	7.25%
3	4.001563	9.84E-05	-1.3893	0.49%

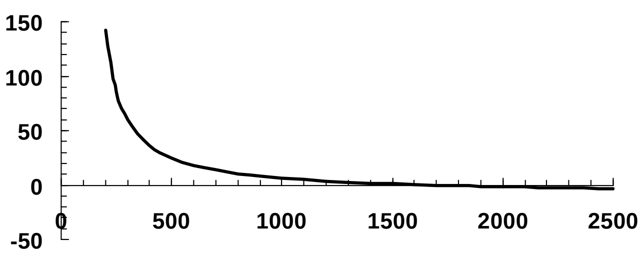
The result can be checked by substituting it back into the original equation to yield a prediction close to 15:

$$c = 75e^{-1.5(4.001563)} + 20e^{-0.075(4.001563)} = 15.0001$$

8.17 The function to be solved is

$$f(T_A) = \frac{T_A}{12}\cosh\left(\frac{600}{T_A}\right) + 6 - \frac{T_A}{12} - 15$$

A plot of the function indicates a root at about $T_A = 1700$.



A numerical method can be used to determine that the root is 1684.365.

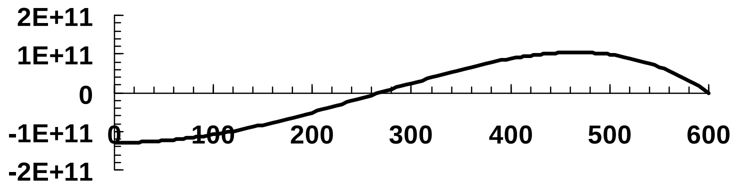
8.18 This problem can be solved by determining the root of the derivative of the elastic curve

$$\frac{dy}{dx} = 0 = \frac{w_0}{120EIL}\left(-5x^4 + 6L^2x^2 - L^4\right)$$

Therefore, after substituting the parameter values, we must determine the root of

$$f(x) = -5x^4 + 2{,}160{,}000x^2 - 1.296 \times 10^{11} = 0$$

A plot of the function indicates a root at about $x = 270$.



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

i	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0	500	250	-1.3E+11	-1.4E+10	1.83E+21	
2	250	500	375	-1.4E+10	7.53E+10	-1.1E+21	33.33%
3	250	375	312.5	-1.4E+10	3.37E+10	-4.8E+20	20.00%
4	250	312.5	281.25	-1.4E+10	9.97E+09	-1.4E+20	11.11%
5	250	281.25	265.625	-1.4E+10	-2.1E+09	2.95E+19	5.88%

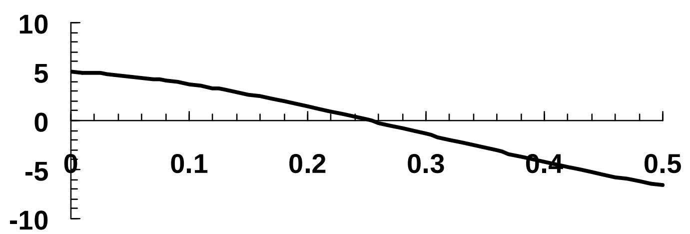
After 20 iterations, the root is determined as $x = 268.328$. This value can be substituted into Eq. (P8.18) to compute the maximum deflection as

$$y = \frac{2.5}{120(50{,}000)30{,}000(600)}(-(268.328)^5 + 720{,}000(268.328)^3 - 1.296 \times 10^{11}(268.328)) = -0.51519$$

8.19 (a) The function to be solved is

$$f(t) = 9e^{-0.7t}\cos(4t) - 3.5$$

A plot of the function indicates a root at about $t = 0.25$



(b) The Newton-Raphson method can be set up as

$$t_{i+1} = t_i - \frac{9e^{-0.7t_i}\cos(4t_i) - 3.5}{-36e^{-0.7t_i}\sin(4t_i) - 6.3\cos(4t_i)e^{-0.7t_i}}$$

Using an initial guess of 0.3,

i	t	$f(t)$	$f'(t)$	ε_a
0	0.3	-0.85651	-29.0483	
1	0.270514	-0.00335	-28.7496	10.899824%
2	0.270398	-1.2E-07	-28.7476	0.043136%
3	0.270398	0	-28.7476	0.000002%

(c) The secant method can be implemented with initial guesses of 0.3,

i	$t_i - 1$	$f(t_i - 1)$	t_i	$f(t_i)$	ε_a
0	0.2	1.951189	0.4	-3.69862	
1	0.4	-3.69862	0.269071	0.038125	48.66%
2	0.269071	0.038125	0.270407	-0.00026	0.49%
3	0.270407	-0.00026	0.270398	1.07E-07	0.0034%

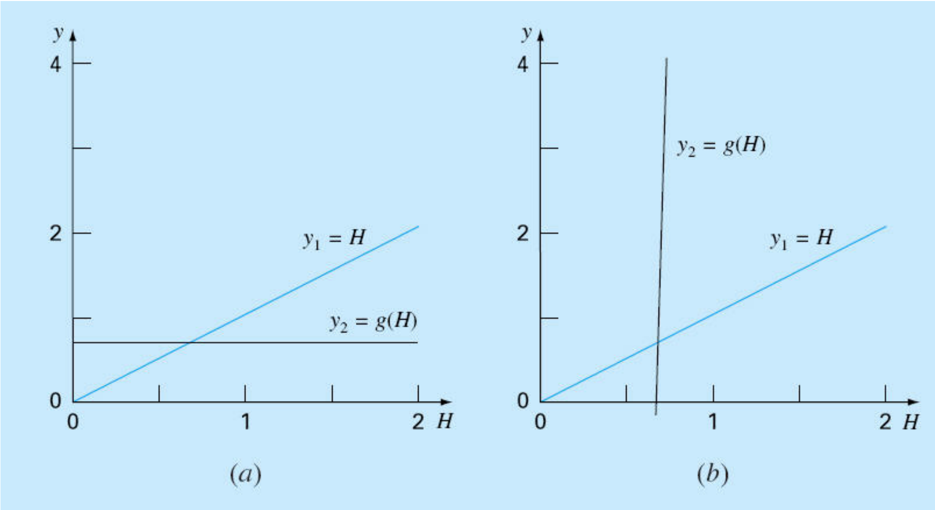
8.20 Two solutions are immediately apparent. We can either solve for the H in the numerator

$$H = \frac{(Qn)^{3/5}(B+2H)^{2/5}}{BS^{3/10}}$$

or the denominator

$$H = \frac{1}{2}\left[\frac{S^{3/4}(BH)^{5/2}}{(Qn)^{3/2}} - B\right]$$

Physical reasoning can be helpful in choosing between these alternatives. For most rivers and streams, the width is much greater than the depth. Thus, the quantity $B + 2H$ should not vary much. In fact, it should be roughly equal to B. In comparison, BH is directly proportional to H. Consequently, the first alternative should home in more rapidly on the root. This can be verified by substituting the brackets of $H = 0$ and 10 into both equations. For the first equation, the results are 0.6834 and 0.9012, which are both close to the true value of 0.7023. In contrast, the results for the second alternative are −10 and 8,178, which clearly are distant from the root. The superiority of the first version is further supported by component plots:



As in (a), the $g(H)$ component for the first version is almost flat. Thus, it will not only converge, but should do so rapidly. In contrast, as in (b), the $g(H)$ component for the second version is almost vertical, connoting strong and rapid divergence.

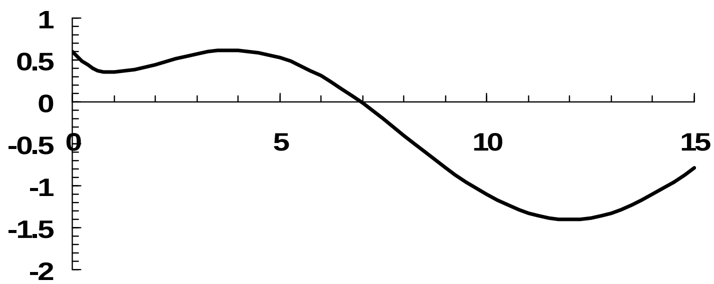
8.21 The solution can be formulated as

$$0.5 = \sin\left(\frac{2\pi x}{16}\right)\cos\left(\frac{2\pi(12)48}{16}\right) + e^{-x}$$

or

$$f(x) = \sin\left(\frac{\pi}{8}x\right) + e^{-x} - 0.4$$

A plot of this function suggests a root at about 7:

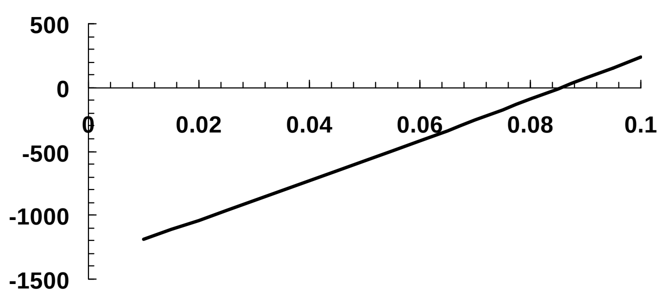


A numerical method can be used to determine that the root is 6.954732.

8.22 The solution can be formulated as

$$f(i) = 25{,}000\frac{i(1+i)^6}{(1+i)^6 - 1} - 5{,}500$$

A plot of this function suggests a root at about 0.086:

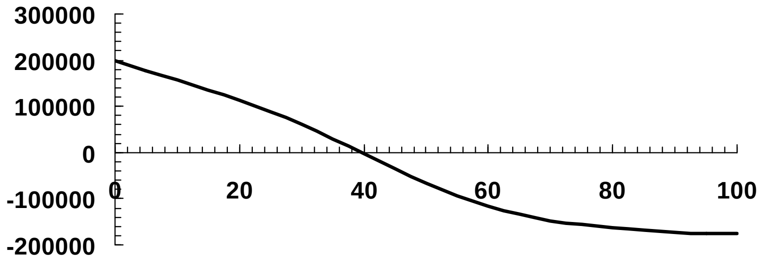


A numerical method can be used to determine that the root is 0.085595.

8.23 (a) The solution can be formulated as

$$f(t) = 1.2\left(75{,}000e^{-0.045t} + 100{,}000\right) - \frac{300{,}000}{1 + 29e^{-0.08t}}$$

A plot of this function suggests a root at about 40:



(b) The false-position method can be implemented with the results summarized as

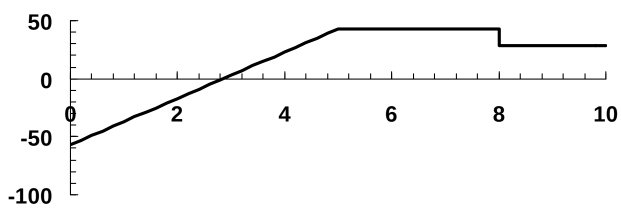
i	t_l	t_u	$f(t_l)$	$f(t_u)$	t_r	$f(t_r)$	$f(t_l)\times f(t_r)$	ε_a
1	0	100.0000	200000	-176110	53.1760	-84245	-1.685E+10	
2	0	53.1760	200000	-84245	37.4156	14442.8	2.889E+09	42.123%
3	37.4156	53.1760	14443	-84245	39.7221	-763.628	-1.103E+07	5.807%
4	37.4156	39.7221	14443	-763.628	39.6063	3.545288	5.120E+04	0.292%
5	39.6063	39.7221	4	-763.628	39.6068	0.000486	1.724E-03	0.001%

(c) The modified secant method (with δ = 0.01) can be implemented with the results summarized as

i	t_i	$f(t_i)$	δt_i	$t_i+\delta t_i$	$f(t_i+\delta t_i)$	$f'(t_i)$	ε_a
0	50	-66444.8	0.50000	50.5	-69357.6	-5825.72	
1	38.5946	6692.132	0.38595	38.98053	4143.604	-6603.33	29.552%
2	39.6080	-8.14342	0.39608	40.00411	-2632.32	-6625.36	2.559%
3	39.6068	-0.00345	0.39607	40.00287	-2624.09	-6625.35	0.003%

For both parts **(b)** and **(c)**, the root is determined to be t = 39.6068. At this time, the ratio of the suburban to the urban population is 135,142.5/112,618.7 = 1.2.

8.24 First, we can generate a plot of the function:



Thus, a zero value occurs at approximately x = 2.8. A numerical solution can be developed in a number of ways. Using MATLAB, we would first formulate an M-file for the shear function as:

```
function f = V(x)
f=20*(sing(x,0,1)-sing(x,5,1))-15*sing(x,8,0)-57;
```

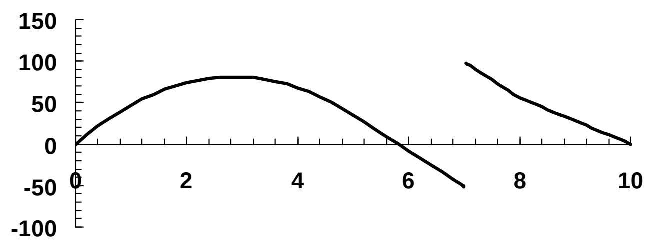
In addition, the singularity function can be set up as

```
function s = sing(x, a, n)
if x > a
  s = (x - a) ^ n;
else
  s = 0;
end
```

We can then either design our own M-file or use MATLAB's built-in capabilities like the `fzero` function. A session using the `fzero` function would yield a root of 2.85 as shown here,

```
>> x=fzero(@V,5)
x =
    2.8500
```

8.25 First, we can generate a plot of the moment function:



Thus, a zero value occurs at approximately $x = 5.8$. A numerical solution can be developed in a number of ways. Using MATLAB, we would first formulate an M-file for the moment function as:

```
function f = Mx(x)
f=-10*(sing(x,0,2)-sing(x,5,2))+15*sing(x,8,1)+150*sing(x,7,0)+57*x;
```

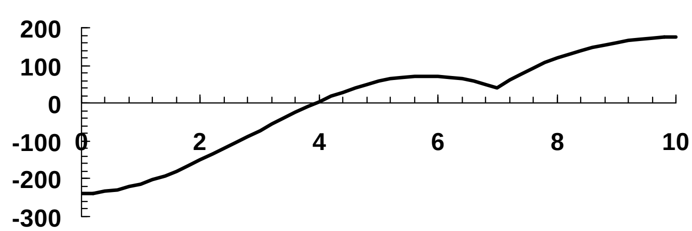
In addition, the singularity function can be set up as

```
function s = sing(x, a, n)
if x > a
  s = (x - a) ^ n;
else
  s = 0;
end
```

We can then either design our own M-file implementing one of the numerical methods in the book or use MATLAB's built-in capabilities like the `fzero` function. A session using the `fzero` function would yield a root of 5.814 as shown here,

```
>> x=fzero(@Mx,5)
x =
    5.8140
```

8.26 First, we can generate a plot of the slope function:



Thus, a zero value occurs at approximately $x = 3.9$. A numerical solution can be developed in a number of ways. Using MATLAB, we would first formulate an M-file for the slope function as:

```
function f = duydx(x)
f=-10/3*(sing(x,0,3)-sing(x,5,3))+7.5*sing(x,8,2)+150*sing(x,7,1)+57/2*x^2-238.25;
```

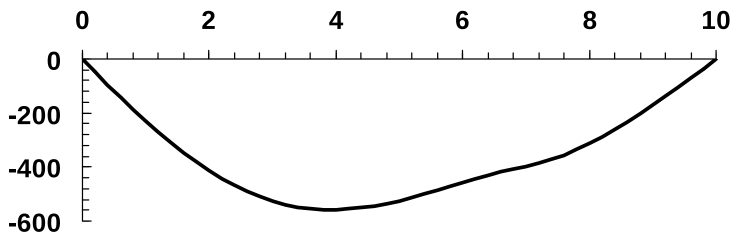
In addition, the singularity function can be set up as

```
function s = sing(x, a, n)
if x > a
  s = (x - a) ^ n;
else
  s = 0;
end
```

We can then either design our own M-file implementing one of the numerical methods in the book or use MATLAB's built-in capabilities like the `fzero` function. A session using the `fzero` function would yield a root of 3.9357 as shown here,

```
>> x=fzero(@duydx,5)
x =
    3.9357
```

8.27 (a) First, we can generate a plot of the slope function:



Therefore, other than the end supports, there are no points of zero displacement.

(b) The location of the minimum can be determined by locating the zero of the slope function as described in Prob. 8.26.

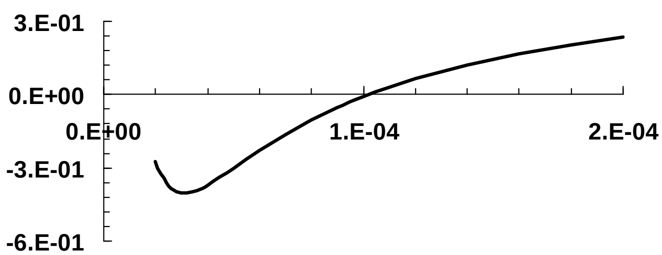
8.28 (a) The solution can be formulated as

$$f(C) = e^{-280(0.05)/(2(7.5))} \cos\left[\sqrt{\frac{1}{7.5C} - \left(\frac{280}{2(7.5)}\right)^2}(0.05)\right] - 0.01$$

or

$$f(C) = 0.393241 \cos\left[\sqrt{\frac{1}{7.5C} - 348.4444}(0.05)\right] - 0.01$$

A plot of this function indicates a root at about $C = 1\times10^{-4}$.



(b) Bisection:

i	C_l	C_u	C_r	$f(C_l)$	$f(C_r)$	$f(C_l)\times f(C_r)$	ε_a
1	5.0000E-05	1.5000E-04	1.0000E-04	-3.02E-01	-9.35E-03	0.002823	
2	1.0000E-04	1.5000E-04	1.2500E-04	-9.35E-03	8.00E-02	-0.00075	20.00%
3	1.0000E-04	1.2500E-04	1.1250E-04	-9.35E-03	3.88E-02	-0.00036	11.11%
4	1.0000E-04	1.1250E-04	1.0625E-04	-9.35E-03	1.57E-02	-0.00015	5.88%
5	1.0000E-04	1.0625E-04	1.0313E-04	-9.35E-03	3.44E-03	-3.2E-05	3.03%

After 14 iterations, the root is determined as 0.000102277 with an approximate error of 0.006%.

(c) In order to use MATLAB, we can first set up a function to hold the equation to be solved,

```
function f = prob0828(C)
t = 0.05; R = 280; L = 7.5; goal = 0.01;
f=exp(-R*t/(2*L))*cos(sqrt(1/(L*C)-(R/(2*L))^2)*t)-goal;
```

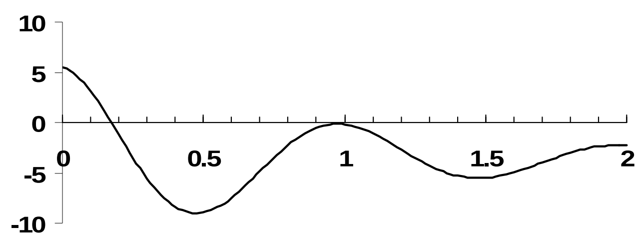
Here is the session that then determines the root,

```
>> format long
>> fzero(@prob0828,1e-4)
ans =
    1.022726852565315e-004
```

8.29 The solution can be formulated as

$$f(t) = 9e^{-t}\cos(2\pi t) - 3.5$$

A plot of this function indicates a root at $t = 0.175$.



Using the Excel Solver and an initial guess of 0 yields a root of $t = 0.173467$.

8.30 The solution can be formulated as

$$f(N) = 0 = \frac{2}{q\left(N + \sqrt{N^2 + 4n_i^2}\right)\mu} - \rho$$

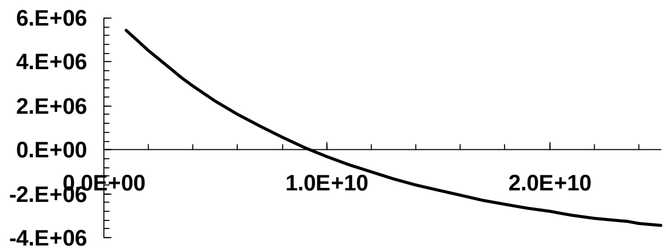
where

$$\mu = 1350\left(\frac{1000}{300}\right)^{-2.42} = 73.2769$$

Substituting this value along with the other parameters gives

$$f(N) = 0 = \frac{2}{1.24571\times10^{-17}\left(N + \sqrt{N^2 + 1.54256\times10^{20}}\right)} - 6.5\times10^{6}$$

A plot of this function indicates a root at about $N = 9\times10^9$.



(b) The bisection method can be implemented with the results for the first 5 iterations summarized as

i	N_l	N_u	N_r	$f(N_l)$	$f(N_r)$	$f(N_l)\times f(N_r)$	ε_a
1	5.000E+09	1.500E+10	1.000E+10	2.23E+06	-3.12E+05	-7E+11	
2	5.000E+09	1.000E+10	7.500E+09	2.23E+06	7.95E+05	1.77E+12	33.333%
3	7.500E+09	1.000E+10	8.750E+09	7.95E+05	2.06E+05	1.63E+11	14.286%
4	8.750E+09	1.000E+10	9.375E+09	2.06E+05	-6.15E+04	-1.3E+10	6.667%
5	8.750E+09	9.375E+09	9.063E+09	2.06E+05	6.99E+04	1.44E+10	3.448%

After 15 iterations, the root is 9.228×10^9 with a relative error of 0.003%.

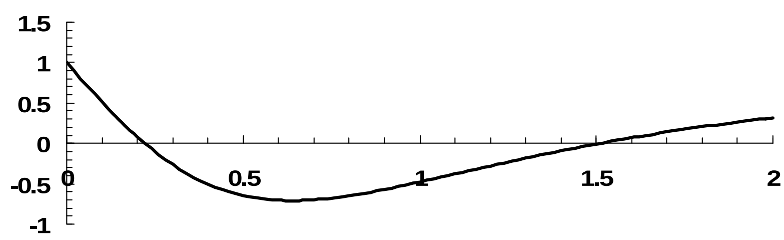
(c) The modified secant method (with $\delta = 0.01$) can be implemented with the results summarized as

i	N_i	$f(N_i)$	δN_i	$N_i+\delta N_i$	$f(N_i+\delta N_i)$	$f'(N_i)$	ε_a
0	9.000E+09	9.672E+04	9.000E+07	9.09E+09	5.819E+04	-0.0004	
1	9.226E+09	6.749E+02	9.226E+07	9.32E+09	-3.791E+04	-0.0004	2.449%
2	9.228E+09	-3.160E+00	9.228E+07	9.32E+09	-3.858E+04	-0.0004	0.017%
3	9.228E+09	1.506E-02	9.228E+07	9.32E+09	-3.858E+04	-0.0004	0.000%

8.31 Using the given values, the roots problem to be solved is

$$f(x) = 0 = 1 - 3.59672\frac{x}{(x^2+0.81)^{3/2}}$$

A plot indicates roots at about 0.22 and 1.5.

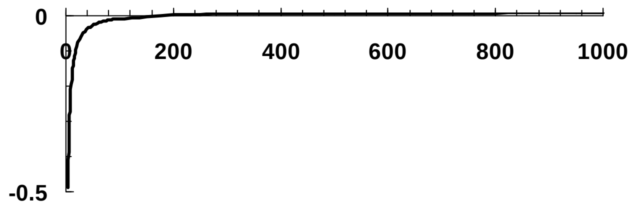


A numerical method can be used to determine that the roots are 0.22135 and 1.50979.

8.32 The solution can be formulated as

$$f(\omega) = 0 = 0.01333333 - \sqrt{1.97531\times10^{-5} + \left(6\times10^{-7}\omega - \frac{2}{\omega}\right)^2}$$

A plot of this function indicates a root at about $\omega = 150$.



Note that the shape of the curve suggests that it may be ill-suited for solution with the false-position method (refer to Fig. 5.14). This conclusion is borne out by the following results for bisection and false position.

(b) The bisection method can be implemented with the results for the first 5 iterations summarized as

i	ω_l	ω_u	ω_r	$f(\omega_l)$	$f(\omega_r)$	$f(\omega_l)\times f(\omega_r)$	ε_a
1	1	1000	500.5	-1.98667	0.007553	-0.01501	
2	1	500.5	250.75	-1.98667	0.004334	-0.00861	99.601%
3	1	250.75	125.875	-1.98667	-0.00309	0.006144	99.206%
4	125.875	250.75	188.3125	-0.00309	0.001924	-6E-06	33.156%
5	125.875	188.3125	157.0938	-0.00309	-6.2E-05	1.93E-07	19.873%

After 13 iterations, the root is 157.9474 with an approximate relative error of 0.077%.

(c) The false-position method can be implemented with the results for the first 5 iterations summarized as

i	ωl	ωu	$f(\omega l)$	$f(\omega u)$	ωr	$f(\omega r)$	$f(\omega l)\times f(\omega r)$	ε_a
1	1	1000.0	-1.98667	0.008674	995.7	0.00867	-0.01722	
2	1	995.7	-1.98667	0.00867	991.3	0.008667	-0.01722	0.436%
3	1	991.3	-1.98667	0.008667	987.0	0.008663	-0.01721	0.436%
4	1	987.0	-1.98667	0.008663	982.8	0.00866	-0.01720	0.436%
5	1	982.8	-1.98667	0.00866	978.5	0.008656	-0.01720	0.435%

After 578 iterations, the root is 189.4316 with an approximate error of 0.0998%. Note that the true error is actually about 20%. Therefore, the false position method is a very poor choice for this problem.

8.33 The solution can be formulated as

$$f(f) = 4\log_{10}\left(\text{Re}\sqrt{f}\right) - 0.4 - \frac{1}{\sqrt{f}}$$

We want our program to work for Reynolds numbers between 2,500 and 1,000,000. Therefore, we must determine the friction factors corresponding to these limits. This can be done with any root location method to yield 0.011525 and 0.002913. Therefore, we can set our initial guesses as $x_l = 0.0028$ and $x_u = 0.012$. Equation (5.5) can be used to determine the number of bisection iterations required to attain an absolute error less than 0.000005,

$$n = \log_2\left(\frac{\Delta x^0}{E_{a,d}}\right) = \log_2\left(\frac{0.012 - 0.0028}{0.000005}\right) = 10.8454$$

which can be rounded up to 11 iterations. Here is a VBA function that is set up to implement 11 iterations of bisection to solve the problem. Note that because VBA does not have a built-in function for the common logarithm, we have developed a user-defined function for this purpose.

```
Function Bisect(xl, xu, Re)
Dim xrold As Double, test As Double
Dim xr As Double, iter As Integer, ea As Double
Dim i As Integer
iter = 0
For i = 1 To 11
  xrold = xr
  xr = (xl + xu) / 2
  iter = iter + 1
  If xr <> 0 Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  test = f(xl, Re) * f(xr, Re)
  If test < 0 Then
    xu = xr
  ElseIf test > 0 Then
    xl = xr
  Else
    ea = 0
  End If
Next i
Bisect = xr
End Function

Function f(x, Re)
f = 4 * log10(Re * Sqr(x)) - 0.4 - 1 / Sqr(x)
End Function

Function log10(x)
log10 = Log(x) / Log(10)
End Function
```

This can be implemented in Excel. Here are the results for a number of values within the desired range. We have included the true value and the resulting error to verify that the results are within the desired error criterion of $E_a < 5\times10^{-6}$.

Re	Root	Truth	E_t
2500	0.011528320	0.011524764	3.56E-06
3000	0.010890430	0.010890229	2.01E-07
10000	0.007727930	0.007727127	8.02E-07
30000	0.005877148	0.005875048	2.10E-06
100000	0.004502539	0.004500376	2.16E-06
300000	0.003622070	0.003617895	4.18E-06
1000000	0.002912305	0.002912819	5.14E-07

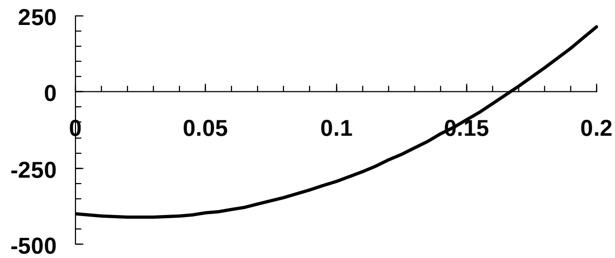
8.34 The solution can be formulated as

$$f(d) = 0 = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2}k_1 d^2 - mgd - mgh$$

Substituting the parameter values gives

$$f(d) = 0 = 16d^{5/2} + 20{,}000d^2 - 931.95d - 400.7385$$

A plot of this function indicates a root at about $d = 0.168$.



A numerical method can be used to determine that the root is $d = 0.166724$.

8.35 The solution can be formulated as

$$f(T) = 0 = -0.20597 + 1.671 \times 10^{-4} T + 9.7215 \times 10^{-8} T^2 - 9.5838 \times 10^{-11} T^3 + 1.9520 \times 10^{-14} T^4$$

MATLAB can be used to determine all the roots of this polynomial,

```
>> format long
>> x=[1.952e-14 -9.5838e-11 9.7215e-8 1.671e-4 -0.20597];
>> roots(x)
ans =
  1.0e+003 *
  2.536837119097375 + 0.910501037132716i
  2.536837119097375 - 0.910501037132716i
 -1.289950382479250
  1.126009750841876
```

The only realistic value is 1,126. This value can be checked using the `polyval` function,

```
>> polyval(x,1126)
ans =
   -1.296516228432854e-006
```

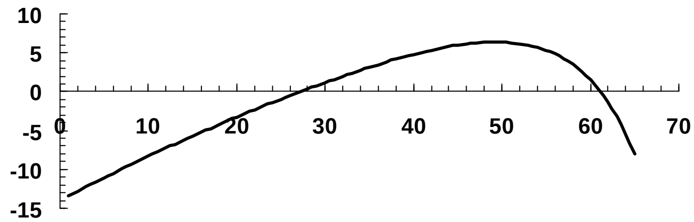
8.36 The solution can be formulated as

$$f(\theta_0) = (\tan \theta_0)x - \frac{g}{2v_0^2 \cos^2 \theta_0} x^2 + y_0 - y$$

Substituting the parameter values gives

$$f(\theta_0) = 0 = 35 \tan(\pi\theta_0 / 180) - \frac{15.0215625}{\cos^2(\pi\theta_0 / 180)} + 1$$

where θ_0 is expressed in degrees. A plot of this function indicates roots at about $\theta_0 = 27^\circ$ and 61°.



The Excel solver can then be used to determine the roots to higher accuracy. Using an initial guesses of 27° and 61° yields θ_0 = 27.2036° and 61.1598°, respectively. Therefore, two angles result in the desired outcome. Note that the lower angle would probably be preferred as the ball would arrive at the catcher faster.

8.37 This problem was solved using the `roots` command in MATLAB.

```
>> I=10+7+16;
>> II=10*7+10*16+7*16-14^2-25^2-15^2;
>> III=10*7*16-10*15^2-7*25^2-16*14^2+2*14*25*15;
>> c=[1 -I II -III]
c =
           1         -33        -704       -1859
>> roots(c)
ans =
  48.354283925405085
 -12.204072966723880
  -3.150210958681167
```

Therefore,

$\sigma_1 = 48.4$ Mpa $\qquad \sigma_2 = -3.15$ MPa $\qquad \sigma_3 = -12.20$ MPa

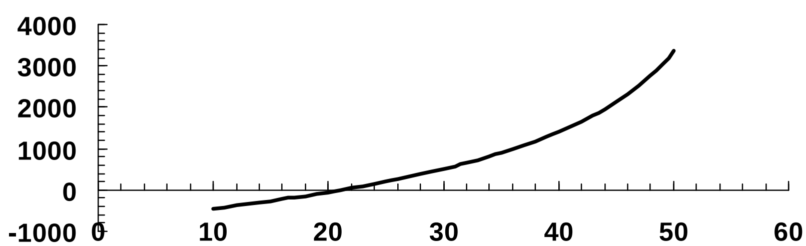
8.38 The solution can be formulated as

$$f(t) = u \ln \frac{m_0}{m_0 - qt} - gt - v$$

Substituting the parameter values gives

$$f(t) = 2{,}000 \ln \frac{150{,}000}{150{,}000 - 2{,}700t} - 9.81t - 750$$

A plot of this function indicates a root at about $t = 21$.



Because two initial guesses are given, a bracketing method like bisection can be used to determine the root,

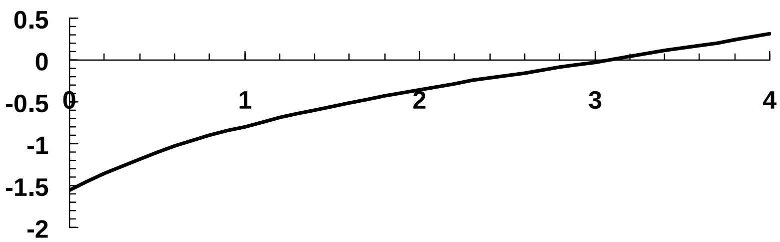
i	t_l	t_u	t_r	$f(t_l)$	$f(t_r)$	$f(t_l)\times f(t_r)$	ε_a
1	10	50	30	-451.198	508.7576	-229550	
2	10	30	20	-451.198	-53.6258	24195.86	50.00%
3	20	30	25	-53.6258	200.424	-10747.9	20.00%
4	20	25	22.5	-53.6258	67.66275	-3628.47	11.11%
5	20	22.5	21.25	-53.6258	5.689921	-305.127	5.88%
6	20	21.25	20.625	-53.6258	-24.2881	1302.471	3.03%
7	20.625	21.25	20.9375	-24.2881	-9.3806	227.8372	1.49%
8	20.9375	21.25	21.09375	-9.3806	-1.8659	17.50322	0.74%

Thus, after 8 iterations, the approximate error falls below 1% with a result of $t = 21.09375$. Note that if the computation is continued, the root can be determined as 21.13242.

8.39 The solution can be formulated as

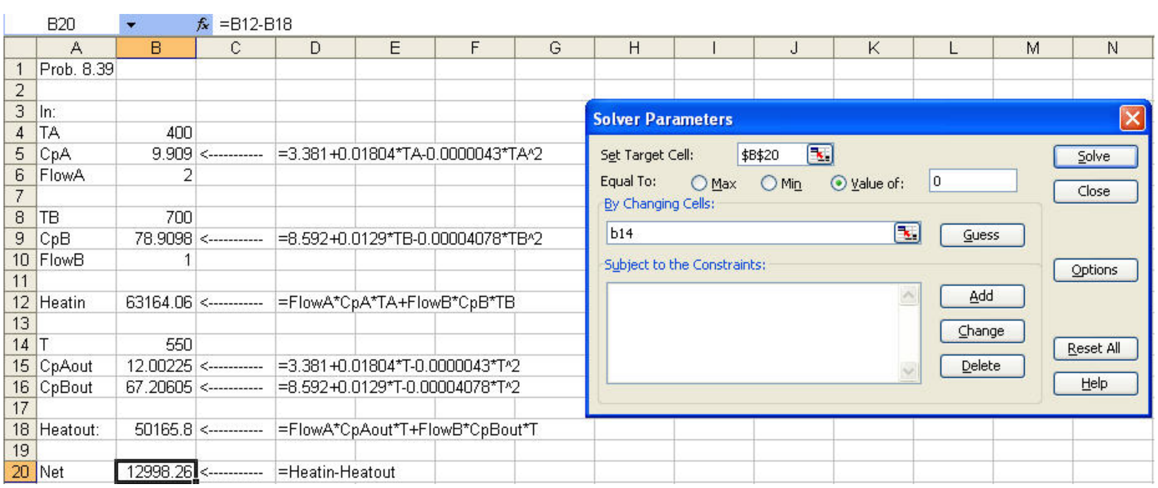
$$f(\omega) = \tan(\omega/3 - 1) - \frac{0.007158\omega}{1 - (\omega/34.119887)^2}$$

A plot of this function indicates a root at about $\omega = 3.1$.



A numerical method can be used to determine that the root is $\omega = 3.06637$.

8.40 Excel Solver solution:



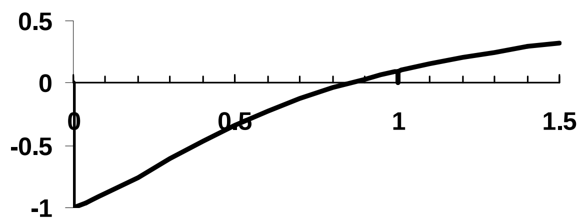
	A	B	C	D
1	Prob. 8.39			
2				
3	In:			
4	TA	400		
5	CpA	9.909	<----------	=3.381+0.01804*TA-0.0000043*TA^2
6	FlowA	2		
7				
8	TB	700		
9	CpB	78.9098	<----------	=8.592+0.0129*TB-0.00004078*TB^2
10	FlowB	1		
11				
12	Heatin	63164.06	<----------	=FlowA*CpA*TA+FlowB*CpB*TB
13				
14	T	550		
15	CpAout	12.00225	<----------	=3.381+0.01804*T-0.0000043*T^2
16	CpBout	67.20605	<----------	=8.592+0.0129*T-0.00004078*T^2
17				
18	Heatout:	50165.8	<----------	=FlowA*CpAout*T+FlowB*CpBout*T
19				
20	Net	12998.26	<----------	=Heatin-Heatout

	A	B	C	D
1	Prob. 8.39			
2				
3	In:			
4	TA	400		
5	CpA	9.909	<-----------	=3.381+0.01804*TA-0.0000043*TA^2
6	FlowA	2		
7				
8	TB	700		
9	CpB	78.9098	<-----------	=8.592+0.0129*TB-0.00004078*TB^2
10	FlowB	1		
11				
12	Heatin	63164.06	<-----------	=FlowA*CpA*TA+FlowB*CpB*TB
13				
14	T	631.9315		
15	CpAout	13.06389	<-----------	=3.381+0.01804*T-0.0000043*T^2
16	CpBout	73.82618	<-----------	=8.592+0.0129*T-0.00004078*T^2
17				
18	Heatout:	63164.06	<-----------	=FlowA*CpAout*T+FlowB*CpBout*T
19				
20	Net	1.27E-07	<-----------	=Heatin-Heatout

8.41 The problem reduces to finding the value of n that drives the second part of the equation to 1. In other words, finding the root of

$$f(n) = \frac{n}{n-1}\left(R_c^{(n-1)/n} - 1\right) - 1 = 0$$

Inspection of the equation indicates that singularities occur at x = 0 and 1. A plot indicates that otherwise, the function is smooth.

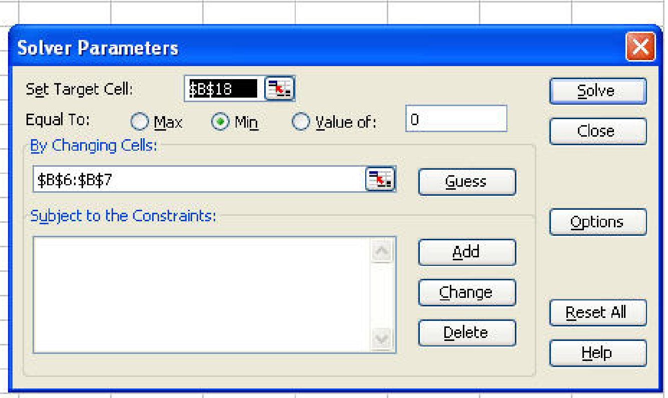


A tool such as the Excel Solver can be used to locate the root at n = 0.8518.

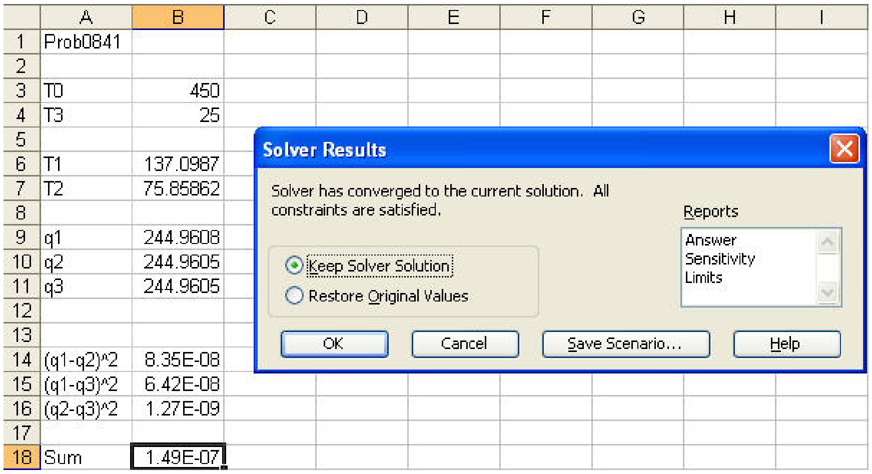
8.42 The following application of Excel Solver can be set up:

B18 =SUM(B14:B16)

	A	B
1	Prob0841	
2		
3	T0	450
4	T3	25
5		
6	T1	200
7	T2	100
8		
9	q1	223.1909
10	q2	400
11	q3	411.1734
12		
13		
14	(q1-q2)^2	31261.44
15	(q1-q3)^2	35337.41
16	(q2-q3)^2	124.8454
17		
18	Sum	66723.7



The solution is:



	A	B
1	Prob0841	
2		
3	T0	450
4	T3	25
5		
6	T1	137.0987
7	T2	75.85862
8		
9	q1	244.9608
10	q2	244.9605
11	q3	244.9605
12		
13		
14	(q1-q2)^2	8.35E-08
15	(q1-q3)^2	6.42E-08
16	(q2-q3)^2	1.27E-09
17		
18	Sum	1.49E-07

8.43 For this problem, two continuity conditions must hold. First, the flows must balance,

$$Q_1 = Q_2 + Q_3 \tag{1}$$

Second, the energy balance must hold. That is, the head losses in pipes 1 and 3 must balance the elevation drop between reservoirs A and C,

$$H_{L,1} + H_{L,3} = E_A - E_C \tag{2}$$

The head losses for each pipe can be computed with

$$H_{L,i} = f_i \frac{L_i}{D_i} \frac{V_i^2}{2g} \tag{3}$$

The flows and velocities are related by the continuity equation, which for a circular pipe is

$$V_i = \frac{4Q_i}{\pi D_i^2} \tag{4}$$

Finally, the Colebrook equation relates the friction factor to the pipe characteristics as in

$$\frac{1}{\sqrt{f_i}} = -2.0 \log\left(\frac{\varepsilon_i}{3.7D_i} + \frac{2.51}{\text{Re}_i \sqrt{f_i}}\right)$$

where ε = the roughness (m), and Re = the *Reynolds number*,

$$\text{Re}_i = \frac{V_i D_i}{\nu_i}$$

where ν = kinematic viscosity (m^2/s).

These equations can be combined to reduce the problem to two equations with 2 unknowns. First, Eq. 4 can be solved for Q and substituted into Eq. 1 to give

$$\frac{\pi D_1^2}{4}V_1 - Q_2 - \frac{\pi D_3^2}{4}V_3 = 0 \tag{5}$$

Then, Eq. 3 can be substituted into Eq. 2 to yield

$$f_1 \frac{L_1}{2gD_1}V_1^2 + f_3 \frac{L_3}{2gD_3}V_3^2 - (E_A - E_C) = 0 \tag{6}$$

Therefore, if we knew the friction factors, these are two equations with two unknowns, V_1 and V_3. If we could solve for these velocities, we could then determine the flows in pipes 1 and 3 via Eq. 4. Further, we could then determine the head losses in each pipe and compute the elevation of reservoir B as

$$E_B = E_A - H_{L,1} - H_{L,2} \tag{7}$$

There are a variety of ways to obtain the solution. One nice way is to use the Excel Solver. First the calculation can be set up as

	A	B	C	D	E	F
1						
2	e	0.0012				
3	g	9.81				
4	nu	1.00E-06				
5						
6	Reservoir					
7	ElevA	200	**ElevB**	**181.147169**	ElevC	172.5
8						
9	Parameters:					
10	L1	1800	L2	500	L3	1400
11	D1	0.4	D2	0.25	D3	0.2
12	**V1**	**1**	V2	2.03718327	**V3**	**1**
13						
14	Q1	0.125663706	Q2	0.1	Q3	0.03141593
15						
16	f1	0.026509711	f2	0.03019179	f3	0.03253974
17						
18	Re1	400000	Re2	509295.818	Re3	200000
19						
20	hL1	6.080208857	hL2	12.7726225	hL3	11.609491
21						
22		Res	Res^2			
23	Flow	-0.00575222	3.31E-05			
24	Head	-9.81030018	96.24199			
25						
26		Target Cell	96.24202			

The shaded cells are input data and the bold cells are the unknowns. The remaining cells are computed with formulas as outlined below. Note that we have named the cells so that the formulas are easier to understand.

	A	B	C	D	E	F
1						
2	e	0.0012				
3	g	9.81				
4	nu	0.000001				
5						
6	Reservoir					
7	ElevA	200	**ElevB**	**=ElevA-hL1_-hL2_**	ElevC	172.5
8						
9	Parameters:					
10	L1	1800	L2	500	L3	1400
11	D1	0.4	D2	0.25	D3	0.2
12	**V1**	**1.12317013769616**	V2	=4*Q2_/PI()/D2_^2	**V3**	**1.30927138993214**
13						
14	Q1	=PI()*D1_^2/4*V1_	Q2	0.1	Q3	=PI()*D3_^2/4*V3_
15						
16	f1	=ff(e,D1_,B18)	f2	=ff(e,D2_,D18)	f3	=ff(e,D3_,F18)
17						
18	Re1	=D1_*V1_/nu	Re2	=D2_*V2_/nu	Re3	=D3_*V3_/nu
19						
20	hL1	=f1_*L1_/D1_*V1_^2/(2*g)	hL2	=f2_*L2_/D2_*V2_^2/(2*g)	hL3	=f3_*L3_/D3_*V3_^2/(2*g)
21						
22		Res	Res^2			
23	Flow	=Q1_-Q2_-Q3_	=B23^2			
24	Head	=hL1_+hL3_-(ElevA-ElevC)	=B24^2			
25						
26		Target Cell	=SUM(C23:C24)			

Notice that we have set up the flow and head loss balances (Eqs. 5 and 6) in cells b23 and b24. We form a target cell (c26) as the summation of the squares of the balances (c23 and c24). It is this target cell that must be minimized to solve the problem.

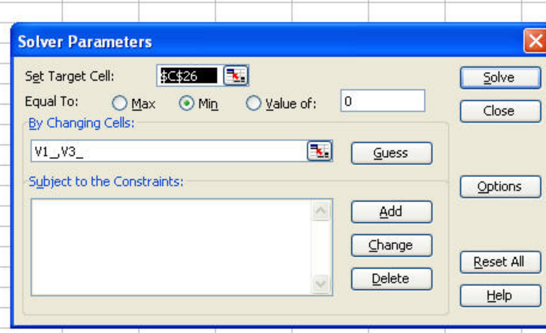
An important feature of the solution is that we use a VBA worksheet function, ff, to solve for the friction factors in cells b16, d16 and f16. This function uses the modified secant method to solve the Colebrook equation for the friction factor. As shown below, it uses the Blasius formula to obtain a good initial guess:

```
Option Explicit

Function ff(e, D, Re)
Dim imax As Integer, iter As Integer
Dim es As Double, ea As Double
Dim xr As Double, xrold As Double, fr As Double
Const del As Double = 0.01
es = 0.01
imax = 20
'Blasius equation
xr = 0.316 / Re ^ 0.25
iter = 0
Do
  xrold = xr
  fr = f(xr, e, D, Re)
  xr = xr - fr * del * xr / (f(xr + del * xr, e, D, Re) - fr)
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
ff = xr
End Function

Function f(x, e, D, Re)
'Colebrook equation
f = -2 * Log(e / (3.7 * D) + 2.51 / Re / Sqr(x)) / Log(10) - 1 / Sqr(x)
```

```
End Function
```

The Excel Solver can then be used to drive the target cell to a minimum by varying the cells for V_1 (cell b12) and V_3 (cell f12).

	A	B	C	D	E	F
1						
2	e	0.0012				
3	g	9.81				
4	nu	1.00E-06				
5						
6	Reservoir					
7	ElevA	200	**ElevB**	**181.147169**	ElevC	172.5
8						
9	Parameters:					
10	L1	1800	L2	500	L3	1400
11	D1	0.4	D2	0.25	D3	0.2
12	**V1**	**1**	V2	2.03718327	**V3**	**1**
13						
14	Q1	0.125663706	Q2	0.1	Q3	0.03141593
15						
16	f1	0.026509711	f2	0.03019179	f3	0.03253974
17						
18	Re1	400000	Re2	509295.818	Re3	200000
19						
20	hL1	6.080208857	hL2	12.7726225	hL3	11.609491
21						
22		Res	Res^2			
23	Flow	-0.00575222	3.31E-05			
24	Head	-9.81030018	96.24199			
25						
26		Target Cell	96.24202			



The results of Solver are shown below:

	A	B	C	D	E	F
1						
2	e	0.0012				
3	g	9.81				
4	nu	1.00E-06				
5						
6	Reservoir					
7	ElevA	200	**ElevB**	**179.564187**	ElevC	172.5
8						
9	Parameters:					
10	L1	1800	L2	500	L3	1400
11	D1	0.4	D2	0.25	D3	0.2
12	**V1**	**1.123442064**	V2	2.03718327	**V3**	**1.30915804**
13						
14	Q1	0.141175893	Q2	0.1	Q3	0.04112841
15						
16	f1	0.026472486	f2	0.03019179	f3	0.03244092
17						
18	Re1	449376.8256	Re2	509295.818	Re3	261831.608
19						
20	hL1	7.66319003	hL2	12.7726225	hL3	19.8370151
21						
22		Res	Res^2			
23	Flow	4.74805E-05	2.25E-09			
24	Head	0.000205128	4.21E-08			
25						
26		Target Cell	4.43E-08			

Therefore, the solution is $V_1 = 1.12344$ and $V_3 = 1.309158$. Equation 4 can then be used to compute $Q_1 = 0.14118$ and $Q_3 = 0.041128$. Finally, Eq. 7 can be used to compute the elevation of reservoir B as 179.564.

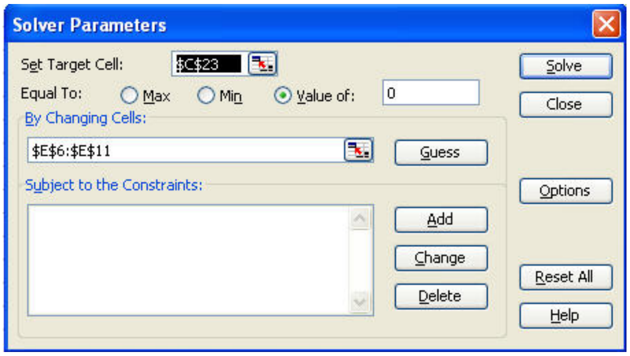
8.44 This problem can be solved in a number of ways. The following solution employs Excel and its Solver option. A worksheet is developed to solve for the pressure drop in each pipe and then determine the flow and pressure balances. Here is how the worksheet is set up,

	A	B	C	D	E	F	G	H
1	Flow1	1	m3/s					
2	rho	1.23	kg/m3					
3								
4	pipe	Length	Diameter	Area	Flow	Velocity	f	DeltaP
5	2	4	0.5	0.1963	0.5	2.546479	0.005	1.60E-01
6	3	2	0.5	0.1963	0.5	2.546479	0.005	7.98E-02
7	4	4	0.5	0.1963	0.5	2.546479	0.005	1.60E-01
8	5	2	0.5	0.1963	0.5	2.546479	0.005	7.98E-02
9	6	4	0.5	0.1963	0.5	2.546479	0.005	1.60E-01
10	7	8	0.5	0.1963	0.5	2.546479	0.005	3.19E-01
11	8	2	0.5	0.1963	0.5	2.546479	0.005	7.98E-02
12	9	2	0.5	0.1963	0.5	2.546479	0.005	7.98E-02
13								
14		Res	Res^2					
15	Flow1	0.00E+00	0.00E+00					
16	Flow2	-5.00E-01	2.50E-01					
17	Flow3	-5.00E-01	2.50E-01					
18	Head1	1.60E-01	2.54E-02					
19	Head2	1.60E-01	2.54E-02					
20	Head3	1.60E-01	2.54E-02					
21								
22		Target	5.76E-01					

The following shows the data and formulas that are entered into each cell.

	A	B	C	D	E	F	G	H
1	Flow1	1	m3/s					
2	rho	1.23	kg/m3					
3								
4	pipe	Length	Diameter	Area	Flow	Velocity	f	DeltaP
5	2	4	0.5	=PI()*(C5/2)^2	0.5	=ABS(E5/D5)	0.005	=16/PI()^2*G5*B5*rho/2/C5^5*E5^2
6	3	2	0.5	=PI()*(C6/2)^2	0.5	=ABS(E6/D6)	0.005	=16/PI()^2*G6*B6*rho/2/C6^5*E6^2
7	4	4	0.5	=PI()*(C7/2)^2	0.5	=ABS(E7/D7)	0.005	=16/PI()^2*G7*B7*rho/2/C7^5*E7^2
8	5	2	0.5	=PI()*(C8/2)^2	0.5	=ABS(E8/D8)	0.005	=16/PI()^2*G8*B8*rho/2/C8^5*E8^2
9	6	4	0.5	=PI()*(C9/2)^2	0.5	=ABS(E9/D9)	0.005	=16/PI()^2*G9*B9*rho/2/C9^5*E9^2
10	7	8	0.5	=PI()*(C10/2)^2	0.5	=ABS(E10/D10)	0.005	=16/PI()^2*G10*B10*rho/2/C10^5*E10^2
11	8	2	0.5	=PI()*(C11/2)^2	=E8	=ABS(E11/D11)	0.005	=16/PI()^2*G11*B11*rho/2/C11^5*E11^2
12	9	2	0.5	=PI()*(C12/2)^2	=E6	=ABS(E12/D12)	0.005	=16/PI()^2*G12*B12*rho/2/C12^5*E12^2
13								
14		Res	Res^2					
15	Flow1	=Flow1-E5-E6	=B15^2					
16	Flow2	=E6-E7-E8	=B16^2					
17	Flow3	=E8-E9-E10	=B17^2					
18	Head1	=H6+H12+H7-H5	=B18^2					
19	Head2	=H8+H11+H9-H7	=B19^2					
20	Head3	=H10-H9	=B20^2					
21								
22		Target	=SUM(C15:C20)					

Notice that we have set up the flow and pressure head loss balances in cells b16 through b21. We form a target cell (c23) as the summation of the squares of the residuals (c16 through c21). It is this target cell that must be minimized to solve the problem. The following shows how this was done with the Solver.



Here is the final result:

	A	B	C	D	E	F	G	H
1	Flow1	1	m3/s					
2	rho	1.23	kg/m3					
3								
4	pipe	Length	Diameter	Area	Flow	Velocity	f	DeltaP
5	2	4	0.5	0.1963	0.5316	2.707495	0.005	1.80E-01
6	3	2	0.5	0.1963	0.4684	2.385504	0.005	7.00E-02
7	4	4	0.5	0.1963	0.2513	1.279907	0.005	4.03E-02
8	5	2	0.5	0.1963	0.217	1.105374	0.005	1.50E-02
9	6	4	0.5	0.1963	0.1272	0.647611	0.005	1.03E-02
10	7	8	0.5	0.1963	0.0899	0.457797	0.005	1.03E-02
11	8	2	0.5	0.1963	0.217	1.105374	0.005	1.50E-02
12	9	2	0.5	0.1963	0.4684	2.385504	0.005	7.00E-02
13								
14		Res	Res^2					
15	Flow1	-7.99E-06	6.38E-11					
16	Flow2	4.36E-05	1.90E-09					
17	Flow3	-6.55E-06	4.29E-11					
18	Head1	-4.28E-05	1.83E-09					
19	Head2	7.60E-05	5.78E-09					
20	Head3	-5.99E-06	3.59E-11					
21								
22		Target	9.66E-09					

8.45 This problem can be solved in a number of ways. The following solution employs Excel and its Solver option. A worksheet is developed to solve for the pressure drop in each pipe and then determine the flow and pressure drop balances. Here is how the worksheet is set up,

	A	B	C	D	E	F	G	H	I
1	Flow1	1	m3/s						
2	rho	1.23	kg/m3						
3	mu	1.79E-05	N s/m2						
4									
5	pipe	Length	Diameter	Area	Flow	Velocity	Re	f	DeltaP
6	2	4	0.5	0.1963	0.5000	2.5465	87491	0.004629	1.48E-01
7	3	2	0.5	0.1963	0.5000	2.5465	87491	0.004629	7.38E-02
8	4	4	0.5	0.1963	0.5000	2.5465	87491	0.004629	1.48E-01
9	5	2	0.5	0.1963	0.5000	2.5465	87491	0.004629	7.38E-02
10	6	4	0.5	0.1963	0.5000	2.5465	87491	0.004629	1.48E-01
11	7	8	0.5	0.1963	0.5000	2.5465	87491	0.004629	2.95E-01
12	8	2	0.5	0.1963	0.5000	2.5465	87491	0.004629	7.38E-02
13	9	2	0.5	0.1963	0.5000	2.5465	87491	0.004629	7.38E-02
14									
15		Res	Res^2						
16	Flow1	0.00E+00	0.00E+00						
17	Flow2	-5.00E-01	2.50E-01						
18	Flow3	-5.00E-01	2.50E-01						
19	Head1	1.48E-01	2.18E-02						
20	Head2	1.48E-01	2.18E-02						
21	Head3	1.48E-01	2.18E-02						
22									
23		Target	5.65E-01						

The following shows the data and formulas that are entered into each cell.

	A	B	C	D	E	F	G	H	I
1	Flow1	1	m3/s						
2	rho	1.23	kg/m3						
3	mu	0.0000179	N s/m2						
4									
5	pipe	Length	Diameter	Area	Flow	Velocity	Re	f	DeltaP
6	2	4	0.5	=PI()*(C6/2)^2	0.5	=ABS(E6/D6)	=rho*F6*C6/mu	=ff(G6)	=16/PI()^2*H6*B6*rho/2/C6^5*E6^2
7	3	2	0.5	=PI()*(C7/2)^2	0.5	=ABS(E7/D7)	=rho*F7*C7/mu	=ff(G7)	=16/PI()^2*H7*B7*rho/2/C7^5*E7^2
8	4	4	0.5	=PI()*(C8/2)^2	0.5	=ABS(E8/D8)	=rho*F8*C8/mu	=ff(G8)	=16/PI()^2*H8*B8*rho/2/C8^5*E8^2
9	5	2	0.5	=PI()*(C9/2)^2	0.5	=ABS(E9/D9)	=rho*F9*C9/mu	=ff(G9)	=16/PI()^2*H9*B9*rho/2/C9^5*E9^2
10	6	4	0.5	=PI()*(C10/2)^2	0.5	=ABS(E10/D10)	=rho*F10*C10/mu	=ff(G10)	=16/PI()^2*H10*B10*rho/2/C10^5*E10^2
11	7	8	0.5	=PI()*(C11/2)^2	0.5	=ABS(E11/D11)	=rho*F11*C11/mu	=ff(G11)	=16/PI()^2*H11*B11*rho/2/C11^5*E11^2
12	8	2	0.5	=PI()*(C12/2)^2	=E9	=ABS(E12/D12)	=rho*F12*C12/mu	=ff(G12)	=16/PI()^2*H12*B12*rho/2/C12^5*E12^2
13	9	2	0.5	=PI()*(C13/2)^2	=E7	=ABS(E13/D13)	=rho*F13*C13/mu	=ff(G13)	=16/PI()^2*H13*B13*rho/2/C13^5*E13^2
14									
15		Res	Res^2						
16	Flow1	=Flow1-E6-E7	=B16^2						
17	Flow2	=E7-E8-E9	=B17^2						
18	Flow3	=E9-E10-E11	=B18^2						
19	Head1	=I7+I13+I8-I6	=B19^2						
20	Head2	=I9+I12+I10-I8	=B20^2						
21	Head3	=I11-I10	=B21^2						
22									
23		Target	=SUM(C16:C21)						

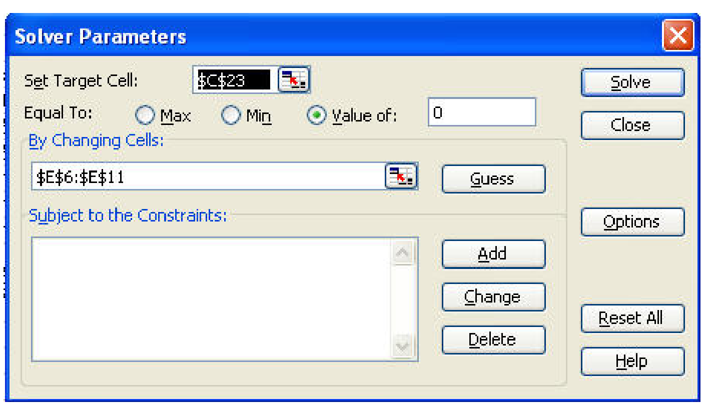
Notice that we have set up the flow and pressure head loss balances in cells b16 through b21. We form a target cell (c23) as the summation of the squares of the residuals (c16 through c21). It is this target cell that must be minimized to solve the problem.

An important feature of the solution is that we use a VBA worksheet function, ff, to solve for the friction factors in column h. This function uses the modified false position method to solve the von Karman equation for the friction factor.

```
Option Explicit

Function ff(Re)
Dim iter As Integer, imax As Integer
Dim il As Integer, iu As Integer
Dim xrold As Double, fl As Double, fu As Double, fr As Double
Dim xl As Double, xu As Double, es As Double
Dim xr As Double, ea As Double
xl = 0.00001
```

```
xu = 1
es = 0.01
imax = 40
iter = 0
fl = f(xl, Re)
fu = f(xu, Re)
Do
  xrold = xr
  xr = xu - fu * (xl - xu) / (fl - fu)
  fr = f(xr, Re)
  iter = iter + 1
  If xr <> 0 Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If fl * fr < 0 Then
    xu = xr
    fu = f(xu, Re)
    iu = 0
    il = il + 1
    If il >= 2 Then fl = fl / 2
  ElseIf fl * fr > 0 Then
    xl = xr
    fl = f(xl, Re)
    il = 0
    iu = iu + 1
    If iu >= 2 Then fu = fu / 2
  Else
    ea = 0
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
ff = xr
End Function

Function f(x, Re)
f = 4 * Log(Re * Sqr(x)) / Log(10) - 0.4 - 1 / Sqr(x)
End Function
```

The Excel Solver can then be used to drive the target cell to a minimum by varying the flows in cells e6 through e11.



Here is the final result:

	A	B	C	D	E	F	G	H	I
1	Flow1	1	m3/s						
2	rho	1.23	kg/m3						
3	mu	1.79E-05	N s/m2						
4									
5	pipe	Length	Diameter	Area	Flow	Velocity	Re	f	DeltaP
6	2	4	0.5	0.1963	0.5411	2.7559	94687	0.004552	1.70E-01
7	3	2	0.5	0.1963	0.4588	2.3366	80278	0.004714	6.33E-02
8	4	4	0.5	0.1963	0.2512	1.2794	43958	0.005380	4.33E-02
9	5	2	0.5	0.1963	0.2075	1.0567	36306	0.005620	1.54E-02
10	6	4	0.5	0.1963	0.1239	0.6310	21681	0.006349	1.24E-02
11	7	8	0.5	0.1963	0.0836	0.4256	14622	0.007002	1.25E-02
12	8	2	0.5	0.1963	0.2075	1.0567	36306	0.005620	1.54E-02
13	9	2	0.5	0.1963	0.4588	2.3366	80278	0.004714	6.33E-02
14									
15		Res	Res^2						
16	Flow1	9.09E-05	8.26E-09						
17	Flow2	8.14E-05	6.62E-09						
18	Flow3	1.63E-05	2.65E-10						
19	Head1	-1.65E-04	2.73E-08						
20	Head2	-1.41E-05	2.00E-10						
21	Head3	3.94E-05	1.55E-09						
22									
23		Target	4.42E-08						

8.46 The horizontal and vertical components of the orbiter thruster can be computed as

$$F_H = T_S \sin\theta \qquad\qquad F_V = T_S \cos\theta$$

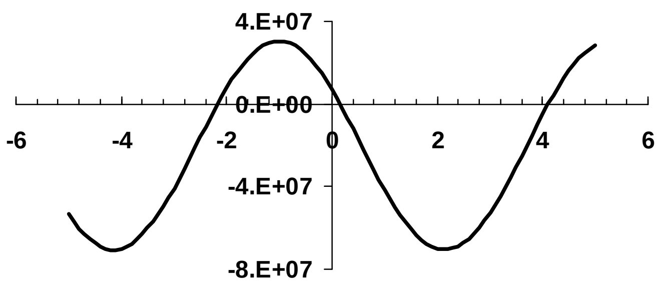
A moment balance about point G can be computed as

$$M = 4W_B - 4T_B - 24W_S + 24T_S \cos\theta - 38T_S \sin\theta$$

Substituting the parameter values yields

$$M = -20.068\times10^6 + 27\times10^6 \cos\theta - 42.75\times10^6 \sin\theta$$

This function can be plotted for the range of –5 to +5 radians



A valid root occurs at about 0.15 radians.

A MATLAB M-file called prob0846.m can be written to implement the Newton-Raphson method to solve for the root as

```
%  Shuttle Liftoff Engine Angle
%  Newton-Raphson Method of iteratively finding a single root
format long
%  Constants
```

```
LGB = 4.0; LGS = 24.0; LTS = 38.0;
WS = 0.230E6; WB = 1.663E6;
TB = 5.3E6; TS = 1.125E6;
es = 0.5E-7; nmax = 200;
%  Initial estimate in radians
x = 0.25
%Calculation loop
for i=1:nmax
  fx = LGB*WB-LGB*TB-LGS*WS+LGS*TS*cos(x)-LTS*TS*sin(x);
  dfx = -LGS*TS*sin(x)-LTS*TS*cos(x);
  xn=x-fx/dfx;
  %convergence check
  ea=abs((xn-x)/xn);
  if (ea<=es)
    fprintf('convergence: Root = %f radians \n',xn)
    theta = (180/pi)*x;
    fprintf('Engine Angle = %f  degrees \n',theta)
    break
  end
  x=xn;
  x
end
```

The program can be run with the result:

```
>> prob0846

x =
   0.15000000000000
x =
   0.15519036852630
x =
   0.15518449747863
convergence: Root = 0.155184 radians
Engine Angle = 8.891417  degrees
```

The program can be run for the case of the minimum payload, by changing W_s to 195,000 and running the M-file with the result:

```
>> prob0846

x =
   0.15000000000000
x =
   0.17333103912866
x =
   0.17321494968603
convergence: Root = 0.173215 radians
Engine Angle = 9.924486  degrees
```