CHAPTER 5

5.1 The function to evaluate is

$$f(c_d)=\sqrt{\frac{gm}{c_d}}\tanh\left(\sqrt{\frac{gc_d}{m}}t\right)-v(t)$$

or substituting the given values

$$f(c_d)=\sqrt{\frac{9.81(65)}{c_d}}\tanh\left(\sqrt{\frac{9.81c_d}{65}}4.5\right)-35$$

The first iteration is

$$x_r=\frac{0.2+0.3}{2}=0.25$$

$$f(0.2)\,f(0.25)=1.913648(0.5266859)=1.007891$$

Therefore, the root is in the second interval and the lower guess is redefined as x_l = 0.25. The second iteration is

$$x_r=\frac{0.25+0.3}{2}=0.275$$

$$\varepsilon_a=\left|\frac{0.275-0.25}{0.275}\right|100\%=9.09\%$$

$$f(0.25)\,f(0.275)=0.5266859(-0.1196732)=-0.06303$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = 0.275. The remainder of the iterations are displayed in the following table:

i	x_l	$f(x_l)$	x_u	$f(x_u)$	x_r	$f(x_r)$	$\lvert\varepsilon_a\rvert$
1	0.2	1.91365	0.3	-0.73745	0.25	0.52669	
2	0.25	0.52669	0.3	-0.73745	0.275	-0.11967	9.09%
3	0.25	0.52669	0.275	-0.11967	0.2625	0.19981	4.76%
4	0.2625	0.19981	0.275	-0.11967	0.26875	0.03916	2.33%
5	0.26875	0.03916	0.275	-0.11967	0.271875	-0.04048	1.15%

Thus, after five iterations, we obtain a root estimate of **0.271875** with an approximate error of 1.15%.

5.2
```
function root = bisectnew(func,xl,xu,Ead)
```

```
% bisectnew(xl,xu,es,maxit):
%   uses bisection method to find the root of a function
%   with a fixed number of iterations to attain
%   a prespecified tolerance
% input:
%   func = name of function
%   xl, xu = lower and upper guesses
%   Ead = (optional) desired tolerance (default = 0.000001)
% output:
%   root = real root

if func(xl)*func(xu)>0  %if guesses do not bracket a sign change
  error('no bracket')    %display an error message and terminate
end
% if necessary, assign default values
if nargin<4, Ead = 0.000001; end   %if Ead blank set to 0.000001
% bisection
xr = xl;
% compute n and round up to next highest integer
n = round(1 + log2((xu - xl)/Ead) + 0.5);
for i = 1:n
  xrold = xr;
  xr = (xl + xu)/2;
  if xr ~= 0, ea = abs((xr - xrold)/xr) * 100; end
  test = func(xl)*func(xr);
  if test < 0
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    ea = 0;
  end
end
root = xr;
```

The following is a MATLAB session that uses the function to solve Prob. 5.1 with $E_{a,d} = 0.0001$.

```
>> format long
>> fcd=@(cd) sqrt(9.81*65/cd)*tanh(sqrt(9.81*cd/65)*4.5)-35;
>> bisectnew(fcd,0.2,0.3,0.0001)

ans =
   0.27026367187500
```

5.3 The function to evaluate is

$$f(c_d) = \sqrt{\frac{9.81(65)}{c_d}} \tanh\left(\sqrt{\frac{9.81 c_d}{65}} 4.5\right) - 35$$

The first iteration is

$$x_r = 0.3 - \frac{-0.7374478(0.2-0.3)}{1.913648-(-0.7374478)} = 0.2721833$$

$$f(0.2)\,f(0.2721833) = 1.913648(-0.0483144) = -0.09246$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = 0.272183. The second iteration is

$$x_r = 0.272183 - \frac{-0.0483144(0.2-0.272183)}{1.913648-(-0.0483144)} = 0.2704057$$

$$\varepsilon_a = \left|\frac{0.2704057-0.2721833}{0.2704057}\right| 100\% = 0.66\%$$

Therefore, after only two iterations we obtain a root estimate of 0.2704057 with an approximate error of 0.66% which is below the stopping criterion of 2%.

5.4

```
function root = falsepos(func,xl,xu,es,maxit)
% falsepos(func,xl,xu,es,maxit):
%   uses the false position method to find the root
%   of the function func
% input:
%   func = name of function
%   xl, xu = lower and upper guesses
%   es = (optional) stopping criterion (%) (default = 0.001)
%   maxit = (optional) maximum allowable iterations (default = 50)
% output:
%   root = real root
if func(xl)*func(xu)>0  %if guesses do not bracket a sign change
  error('no bracket')    %display an error message and terminate
end
% default values
if nargin<5, maxit=50; end
if nargin<4, es=0.001; end
% false position
iter = 0;
xr = xl;
while (1)
  xrold = xr;
  xr = xu - func(xu)*(xl - xu)/(func(xl) - func(xu));
  iter = iter + 1;
  if xr ~= 0, ea = abs((xr - xrold)/xr) * 100; end
  test = func(xl)*func(xr);
  if test < 0
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    ea = 0;
  end
```

```
  if ea <= es | iter >= maxit, break, end
end
root = xr;
```

The following is a MATLAB session that uses the function to solve Prob. 5.1:

```
>> format long
>> fcd=@(cd) sqrt(9.81*65/cd)*tanh(sqrt(9.81*cd/65)*4.5)-35;
>> falsepos(fcd,0.2,0.3,2)

ans =
   0.27040572999122
```

5.5 Solve for the reactions:

R_1=265 lbs. R_2= 285 lbs.

Write beam equations:

0<*x*<3

$$M + (16.667x^2)\frac{x}{3} - 265x = 0$$

$$(1) \quad M = 265x - 5.5555556x^3$$

3<*x*<6

$$M + 100(x-3)(\frac{x-3}{2}) + 150(x - \frac{2}{3}(3)) - 265x = 0$$

$$(2) \quad M = -50x^2 + 415x - 150$$

6<*x*<10

$$M = 150(x - \frac{2}{3}(3)) + 300(x - 4.5) - 265x$$

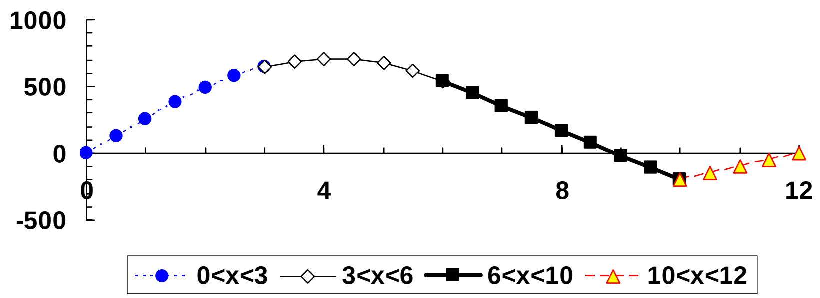
$$(3) \quad M = -185x + 1650$$

10<*x*<12

$$M + 100(12 - x) = 0$$

$$(4) \quad M = 100x - 1200$$

A plot of these equations can be generated:



Combining Equations:

Because the curve crosses the axis between 6 and 10, use (3).

(3) $M = -185x + 1650$

Set $x_L = 6; x_U = 10$

$M(x_L) = 540$
$M(x_U) = -200$ $\qquad x_r = \dfrac{x_L + x_U}{2} = 8$
$M(x_R) = 170 \rightarrow \textit{replaces } x_L$

$M(x_L) = 170$
$M(x_U) = -200$ $\qquad x_r = \dfrac{8+10}{2} = 9$
$M(x_R) = -15 \rightarrow \textit{replaces } x_U$

$M(x_L) = 170$
$M(x_U) = -15$ $\qquad x_r = \dfrac{8+9}{2} = 8.5$
$M(x_R) = 77.5 \rightarrow \textit{replaces } x_L$

$M(x_L) = 77.5$
$M(x_U) = -15$ $\qquad x_r = \dfrac{8.5+9}{2} = 8.75$
$M(x_R) = 31.25 \rightarrow \textit{replaces } x_L$

$M(x_L) = 31.25$
$M(x_U) = -15$ $\qquad x_r = \dfrac{8.75+9}{2} = 8.875$
$M(x_R) = 8.125 \rightarrow \textit{replaces } x_L$

$M(x_L) = 8.125$
$M(x_U) = -15$ $\qquad x_r = \dfrac{8.875+9}{2} = 8.9375$
$M(x_R) = -3.4375 \rightarrow \textit{replaces } x_U$

$M(x_L) = 8.125$
$M(x_U) = -3.4375$ $\qquad x_r = \dfrac{8.875+8.9375}{2} = 8.90625$
$M(x_R) = 2.34375 \rightarrow \textit{replaces } x_L$

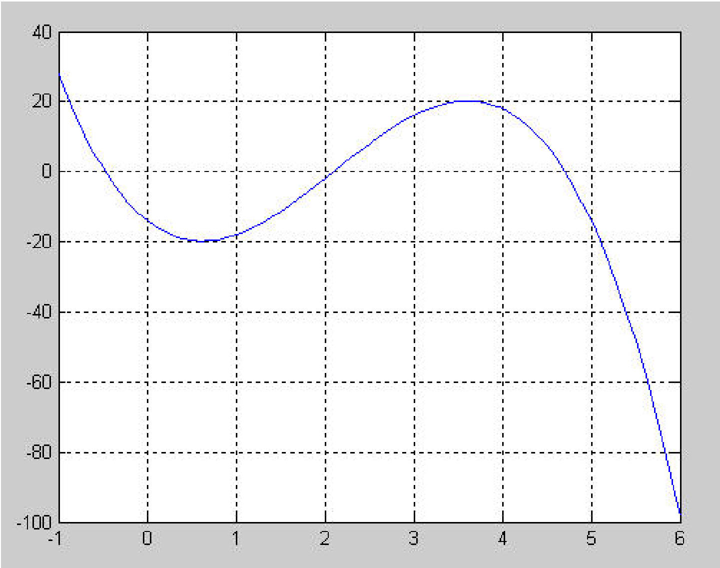
$M(x_L) = 2.34375$
$M(x_U) = -3.4375$ $\qquad x_r = \dfrac{8.90625+8.9375}{2} = 8.921875$
$M(x_R) = -0.546875 \rightarrow \textit{replaces } x_U$

$M(x_L) = 2.34375$
$M(x_U) = -0.546875$ $\qquad x_r = \dfrac{8.90625+8.921875}{2} = 8.9140625$
$M(x_R) = 0.8984$ $\qquad$ **Therefore,** $x = 8.91$ *feet*

5.6 (a) The graph can be generated with MATLAB

```
>> x=[-1:0.1:6];
>> f=-14-20*x+19*x.^2-3*x.^3;
```

```
>> plot(x,f)
>> grid
```



This plot indicates that roots are located at about –0.5, 2 and 4.7.

(b) Using bisection, the first iteration is

$$x_r = \frac{-1+0}{2} = -0.5$$

$$f(-1)\,f(-0.5) = 28(1.125) = 31.5$$

Therefore, the root is in the second interval and the lower guess is redefined as x_l = –0.5. The second iteration is

$$x_r = \frac{-0.5+0}{2} = -0.25$$

$$\varepsilon_a = \left|\frac{-0.25-(-0.5)}{-0.25}\right| 100\% = 100\%$$

$$f(-0.5)\,f(-0.25) = 1.125(-7.7653) = -8.73633$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = –0.25. The remainder of the iterations are displayed in the following table:

i	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	$\lvert\varepsilon_a\rvert$
1	-1.00000	0.00000	-0.50000	28.00000	1.12500	31.50000	100.00%
2	-0.50000	0.00000	-0.25000	1.12500	-7.76563	-8.73633	100.00%
3	-0.50000	-0.25000	-0.37500	1.12500	-3.66992	-4.12866	33.33%

4	-0.50000	-0.37500	-0.43750	1.12500	-1.36206	-1.53232	14.29%
5	-0.50000	-0.43750	-0.46875	1.12500	-0.14120	-0.15886	6.67%
6	-0.50000	-0.46875	-0.48438	1.12500	0.48619	0.54697	3.23%
7	-0.48438	-0.46875	-0.47656	0.48619	0.17107	0.08317	1.64%
8	-0.47656	-0.46875	-0.47266	0.17107	0.01458	0.00249	0.83%

Thus, after eight iterations, we obtain a root estimate of **–0.47266** with an approximate error of 0.83%, which is below the stopping criterion of 1%.

(c) Using false position, the first iteration is

$$x_r = 0 - \frac{-14(-1-0)}{28-(-14)} = -0.33333$$

$$f(-1)\,f(-0.33333) = 28(-5.11111) = -143.11111$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = –0.33333. The second iteration is

$$x_r = -0.33333 - \frac{-5.11111(-1-(-0.33333))}{28-(-5.11111)} = -0.43624$$

$$\varepsilon_a = \left|\frac{-0.43624-(-0.33333)}{-0.43624}\right| 100\% = 23.59\%$$

$$f(-1)\,f(-0.43624) = 28(-1.41028) = -39.48785$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = –0.43624. The remainder of the iterations are displayed in the following table:

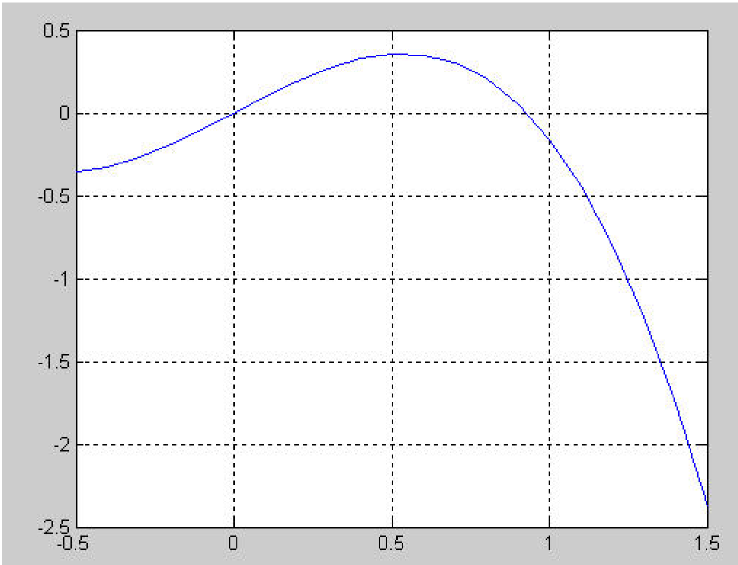
i	x_l	$f(x_l)$	x_u	$f(x_u)$	x_r	$f(x_r)$	$f(x_l)\times f(x_r)$	$\lvert\varepsilon_a\rvert$
1	-1	28.00000	0.00000	-14.00000	-0.33333	-5.11111	-143.11111	
2	-1	28.00000	-0.33333	-5.11111	-0.43624	-1.41028	-39.48785	23.590%
3	-1	28.00000	-0.43624	-1.41028	-0.46327	-0.35836	-10.03415	5.835%
4	-1	28.00000	-0.46327	-0.35836	-0.47006	-0.08914	-2.49593	1.443%
5	-1	28.00000	-0.47006	-0.08914	-0.47174	-0.02206	-0.61754	0.357%

Therefore, after five iterations we obtain a root estimate of **–0.47174** with an approximate error of 0.357%, which is below the stopping criterion of 1%.

5.7 A graph of the function can be generated with MATLAB

```
>> x=[-0.5:0.1:1.5];
>> f=sin(x)-x.^3;
>> plot(x,f);grid
```

This plot indicates that a nontrivial root (i.e., nonzero) is located at about 0.9.



Using bisection, the first iteration is

$$x_r = \frac{0.5+1}{2} = 0.75$$

$$f(0.5)f(0.75) = 0.354426(0.2597638) = 0.092067$$

Therefore, the root is in the second interval and the lower guess is redefined as x_l = 0.75. The second iteration is

$$x_r = \frac{0.75+1}{2} = 0.875 \qquad \varepsilon_a = \left|\frac{0.875-0.75}{0.875}\right| 100\% = 14.29\%$$

$$f(0.75)f(0.875) = 0.259764(0.0976216) = 0.025359$$

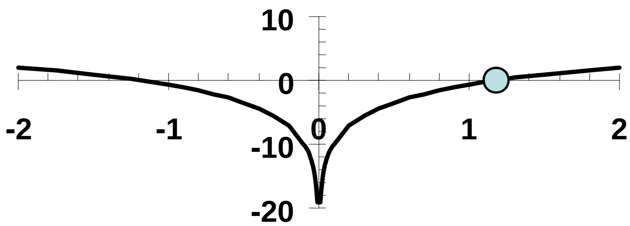
Because the product is positive, the root is in the second interval and the lower guess is redefined as x_l = 0.875. All the iterations are displayed in the following table:

i	x_l	$f(x_l)$	x_u	$f(x_u)$	x_r	$f(x_r)$	ε_a
1	0.5	0.354426	1	-0.158529	0.75	0.2597638	
2	0.75	0.259764	1	-0.158529	0.875	0.0976216	14.29%
3	0.875	0.097622	1	-0.158529	0.9375	-0.0178935	6.67%
4	0.875	0.097622	0.9375	-0.0178935	0.90625	0.0429034	3.45%
5	0.90625	0.042903	0.9375	-0.0178935	0.921875	0.0132774	1.69%

Consequently, after five iterations we obtain a root estimate of **0.921875** with an approximate error of 1.69%, which is below the stopping criterion of 2%. The result can be checked by substituting it into the original equation to verify that it is close to zero.

$$f(x) = \sin(x) - x^3 = \sin(0.921875) - 0.921875^3 = 0.0132774$$

5.8 (a) A graph of the function indicates a positive real root at approximately *x* = 1.2.



(b) Using bisection, the first iteration is

$$x_r = \frac{0.5+2}{2} = 1.25 \qquad \varepsilon_a = \left|\frac{2-0.5}{2+0.5}\right| 100\% = 60\%$$

$$f(0.5)f(1.25) = -3.47259(0.19257) = -0.66873$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = 1.25. The second iteration is

$$x_r = \frac{0.5+1.25}{2} = 0.875 \qquad \varepsilon_a = \left|\frac{0.875-1.25}{0.875}\right| 100\% = 42.86\%$$

$$f(0.5)f(0.875) = -3.47259(-1.23413) = 4.28561$$

Consequently, the root is in the second interval and the lower guess is redefined as x_l = 0.875. All the iterations are displayed in the following table:

i	x_l	x_u	x_r	$f(x_l)$	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0.50000	2.00000	1.25000	-3.47259	0.19257	-0.66873	
2	0.50000	1.25000	0.87500	-3.47259	-1.23413	4.28561	42.86%
3	0.87500	1.25000	1.06250	-1.23413	-0.4575	0.56461	17.65%

Thus, after three iterations, we obtain a root estimate of **1.0625** with an approximate error of 17.65%.

(c) Using false position, the first iteration is

$$x_r = 2 - \frac{2.07259(0.5-2)}{-3.47259-2.07259} = 1.43935$$

$$f(0.5)f(1.43935) = -3.47259(0.75678) = -2.62797$$

Therefore, the root is in the first interval and the upper guess is redefined as x_u = 1.43935. The second iteration is

$$x_r = 1.43935 - \frac{0.75678(0.5 - 1.43935)}{-3.47259 - 0.75678} = 1.27127$$

$$\varepsilon_a = \left| \frac{1.27127 - 1.43935}{1.27127} \right| 100\% = 13.222\%$$

$$f(0.5)\,f(1.27127) = -3.47259(0.26007) = -0.90312$$

Consequently, the root is in the first interval and the upper guess is redefined as x_u = 1.27127. All the iterations are displayed in the following table:

iteration	x_l	x_u	$f(x_l)$	$f(x_u)$	x_r	$f(x_r)$	$f(x_l)\times f(x_r)$	ε_a
1	0.5	2.00000	-3.47259	2.07259	1.43935	0.75678	-2.62797	
2	0.5	1.43935	-3.47259	0.75678	1.27127	0.26007	-0.90312	13.222%
3	0.5	1.27127	-3.47259	0.26007	1.21753	0.08731	-0.30319	4.414%

After three iterations we obtain a root estimate of **1.21753** with an approximate error of 4.414%.

5.9 (a) Equation (5.6) can be used to determine the number of iterations

$$n = \log_2\left(\frac{\Delta x^0}{E_{a,d}}\right) = \log_2\left(\frac{40}{0.05}\right) = 9.6439$$

which can be rounded up to 10 iterations.

(b) Here is an M-file that evaluates the temperature in ^{o}C using 10 iterations of bisection based on a given value of the oxygen saturation concentration in freshwater:

```
function TC = TempEval(osf)
% function to evaluate the temperature in degrees C based
% on the oxygen saturation concentration in freshwater (osf).
xl = 0 + 273.15;
xu = 40 + 273.15;
if fTa(xl,osf)*fTa(xu,osf)>0  %if guesses do not bracket
  error('no bracket')   %display an error message and terminate
end
xr = xl;
for i = 1:10
  xrold = xr;
  xr = (xl + xu)/2;
  if xr ~= 0, ea = abs((xr - xrold)/xr) * 100; end
  test = fTa(xl,osf)*fTa(xr,osf);
  if test < 0
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    ea = 0;
```

```
  end
end
TC = xr - 273.15;

function f = fTa(Ta, osf)
f = -139.34411 + 1.575701e5/Ta - 6.642308e7/Ta^2;
f = f + 1.2438e10/Ta^3 - 8.621949e11/Ta^4;
f = f - log(osf);
```

The function can be used to evaluate the test cases:

```
>> TempEval(8)
ans =
   26.7578

>> TempEval(10)
ans =
   15.3516

>> TempEval(12)
ans =
    7.4609
```

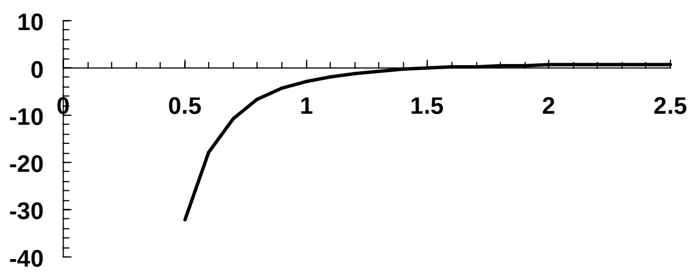
Note that these values can be compared with the true values to verify that the errors are less than 0.05:

o_{sf}	Approximation	Exact	Error
8	26.75781	26.78017	0.0224
10	15.35156	15.38821	0.0366
12	7.46094	7.46519	0.0043

5.10 (a) The function to be evaluated is

$$f(y) = 1 - \frac{400}{9.81(3y + y^2/2)^3}(3+y)$$

A graph of the function indicates a positive real root at approximately 1.5.



(b) Using bisection, the first iteration is

$$x_r = \frac{0.5 + 2.5}{2} = 1.5$$

$$f(0.5)\,f(1.5) = -32.2582(-0.030946) = 0.998263$$

Therefore, the root is in the second interval and the lower guess is redefined as $x_l = 1.5$. The second iteration is

$$x_r = \frac{1.5 + 2.5}{2} = 2 \qquad \varepsilon_a = \left|\frac{2 - 1.5}{2}\right| 100\% = 25\%$$

$$f(1.5)\,f(2) = -0.030946(0.601809) = -0.018624$$

Therefore, the root is in the first interval and the upper guess is redefined as $x_u = 2$. All the iterations are displayed in the following table:

i	x_l	$f(x_l)$	x_u	$f(x_u)$	x_r	$f(x_r)$	ε_a
1	0.5	–32.2582	2.5	0.813032	1.5	–0.030946	
2	1.5	–0.03095	2.5	0.813032	2	0.601809	25.00%
3	1.5	–0.03095	2	0.601809	1.75	0.378909	14.29%
4	1.5	–0.03095	1.75	0.378909	1.625	0.206927	7.69%
5	1.5	–0.03095	1.625	0.206927	1.5625	0.097956	4.00%
6	1.5	–0.03095	1.5625	0.097956	1.53125	0.036261	2.04%
7	1.5	–0.03095	1.53125	0.036261	1.515625	0.003383	1.03%
8	1.5	–0.03095	1.515625	0.003383	1.5078125	–0.013595	0.52%

After eight iterations, we obtain a root estimate of **1.5078125** with an approximate error of 0.52%.

(c) Using false position, the first iteration is

$$x_r = 2.5 - \frac{0.81303(0.5 - 2.5)}{-32.2582 - 0.81303} = 2.45083$$

$$f(0.5)\,f(2.45083) = -32.25821(0.79987) = -25.80248$$

Therefore, the root is in the first interval and the upper guess is redefined as $x_u = 2.45083$. The second iteration is

$$x_r = 2.45083 - \frac{0.79987(0.5 - 2.45083)}{-32.25821 - 0.79987} = 2.40363 \qquad \varepsilon_a = \left|\frac{2.40363 - 2.45083}{2.40363}\right| 100\% = 1.96\%$$

$$f(0.5)\,f(2.40363) = -32.2582(0.78612) = -25.35893$$

The root is in the first interval and the upper guess is redefined as $x_u = 2.40363$. All the iterations are displayed in the following table:

i	x_l	$f(x_l)$	x_u	$f(x_u)$	x_r	$f(x_r)$	ε_a
1	0.5	–32.2582	2.50000	0.81303	2.45083	0.79987	
2	0.5	–32.2582	2.45083	0.79987	2.40363	0.78612	1.96%
3	0.5	–32.2582	2.40363	0.78612	2.35834	0.77179	1.92%
4	0.5	–32.2582	2.35834	0.77179	2.31492	0.75689	1.88%
5	0.5	–32.2582	2.31492	0.75689	2.27331	0.74145	1.83%
6	0.5	–32.2582	2.27331	0.74145	2.23347	0.72547	1.78%
7	0.5	–32.2582	2.23347	0.72547	2.19534	0.70900	1.74%
8	0.5	–32.2582	2.19534	0.70900	2.15888	0.69206	1.69%
9	0.5	–32.2582	2.15888	0.69206	2.12404	0.67469	1.64%
10	0.5	–32.2582	2.12404	0.67469	2.09077	0.65693	1.59%

After ten iterations we obtain a root estimate of **2.09077** with an approximate error of 1.59%. Thus, after ten iterations, the false position method is converging at a very slow pace and is still far from the root in the vicinity of 1.5 that we detected graphically.

Discussion: This is a classic example of a case where false position performs poorly and is inferior to bisection. Insight into these results can be gained by examining the plot that was developed in part **(a)**. This function violates the premise upon which false position was based–that is, if $f(x_u)$ is much closer to zero than $f(x_l)$, then the root is closer to x_u than to x_l (recall Figs. 5.8 and 5.9). Because of the shape of the present function, the opposite is true.

5.11

Errata: In the first printing, the solution to the differential equation should have been:

$$S = S_0 - v_m t + k_s \ln(S_0 / S)$$

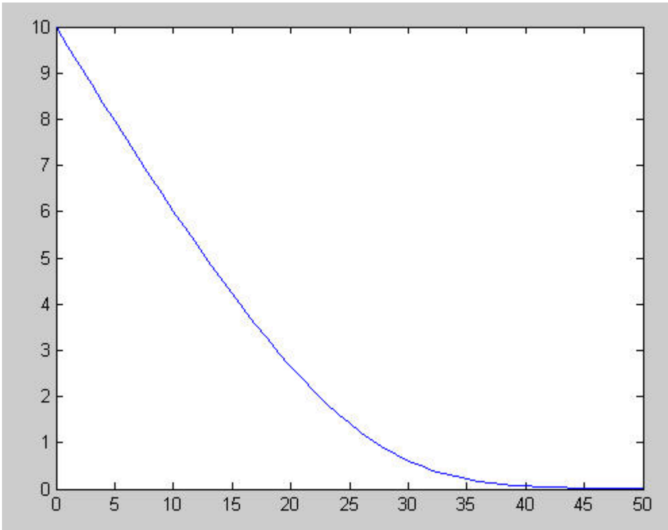
Subsequent printings should show the correct solution.

The problem amounts to determining the root of

$$f(S) = S_0 - v_m t + k_s \ln(S_0 / S) - S$$

The following script calls the bisect function (Fig. 5.7) for various values of t in order to generate the solution.

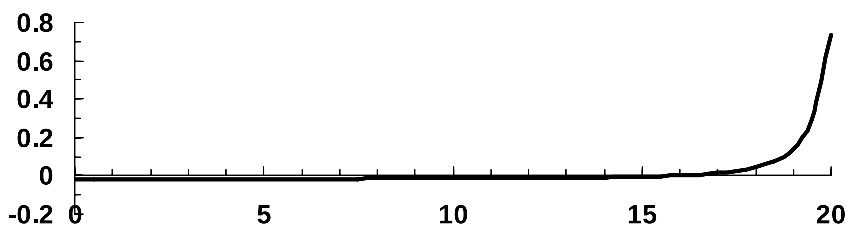
```
S0=10;vm=0.5;ks=2;
f = @(S,t) S0 - vm * t + ks * log(S0 / S) - S;
t=0:50;S=0:50;
for i = 1:n
  S(i)=bisect(f,0.00001,10.01,1e-6,100,t(i));
end
plot(t,S)
```



5.12 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. 5.9). 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	8.42×10^{-5}	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	-1.8×10^{-5}	7.692%
5	15	16.25	15.625	-0.00585	-0.00228	1.33×10^{-5}	4.000%
6	15.625	16.25	15.9375	-0.00228	0.000123	-2.8×10^{-7}	1.961%
7	15.625	15.9375	15.78125	-0.00228	-0.00114	2.59×10^{-6}	0.990%
8	15.78125	15.9375	15.85938	-0.00114	-0.00052	5.98×10^{-7}	0.493%

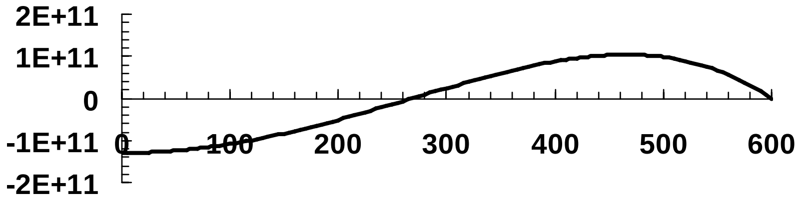
5.13 This problem can be solved by determining the root of the derivative of the elastic curve

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

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	$-1.3\text{x}10^{11}$	$-1.4\text{x}10^{10}$	$1.83\text{x}10^{21}$	
2	250	500	375	$-1.4\text{x}10^{10}$	$7.53\text{x}10^{10}$	$-1.1\text{x}10^{21}$	33.33%
3	250	375	312.5	$-1.4\text{x}10^{10}$	$3.37\text{x}10^{10}$	$-4.8\text{x}10^{20}$	20.00%
4	250	312.5	281.25	$-1.4\text{x}10^{10}$	$9.97\text{x}10^{9}$	$-1.4\text{x}10^{20}$	11.11%
5	250	281.25	265.625	$-1.4\text{x}10^{10}$	$-2.1\text{x}10^{9}$	$2.95\text{x}10^{19}$	5.88%

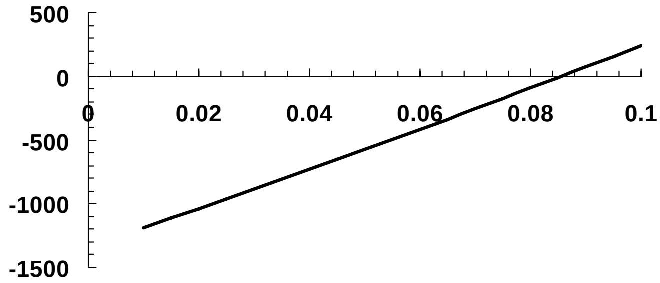
After 20 iterations, the root is determined as x = 268.328. This value can be substituted into Eq. (P5.13) to compute the maximum deflection as

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

5.14 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:

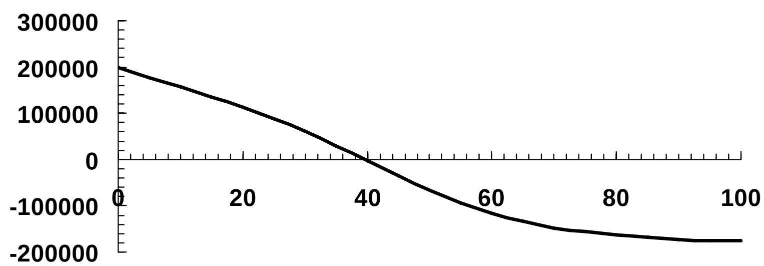


A numerical method can be used to determine that the root is 0.085595.

5.15 (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	$-1.685\text{x}10^{10}$	
2	0	53.1760	200000	-84245	37.4156	14442.8	$2.889\text{x}10^{9}$	42.123%
3	37.4156	53.1760	14443	-84245	39.7221	-763.628	$-1.103\text{x}10^{7}$	5.807%
4	37.4156	39.7221	14443	-763.628	39.6063	3.545288	$5.120\text{x}10^{4}$	0.292%
5	39.6063	39.7221	4	-763.628	39.6068	0.000486	$1.724\text{x}10^{-3}$	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.

5.16 The solution can be formulated as

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

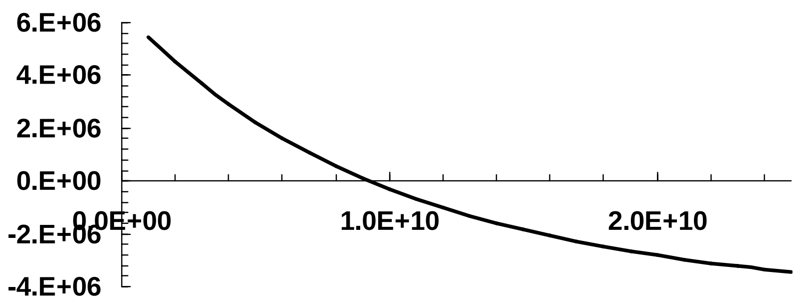
where

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

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	$5.000\text{x}10^9$	$1.500\text{x}10^{10}$	$1.000\text{x}10^{10}$	$2.23\text{x}10^6$	$-3.12\text{x}10^5$	$-7\text{x}10^{11}$	
2	$5.000\text{x}10^9$	$1.000\text{x}10^{10}$	$7.500\text{x}10^9$	$2.23\text{x}10^6$	$7.95\text{x}10^5$	$1.77\text{x}10^{12}$	33.333%
3	$7.500\text{x}10^9$	$1.000\text{x}10^{10}$	$8.750\text{x}10^9$	$7.95\text{x}10^5$	$2.06\text{x}10^5$	$1.63\text{x}10^{11}$	14.286%
4	$8.750\text{x}10^9$	$1.000\text{x}10^{10}$	$9.375\text{x}10^9$	$2.06\text{x}10^5$	$-6.15\text{x}10^4$	$-1.3\text{x}10^{10}$	6.667%
5	$8.750\text{x}10^9$	$9.375\text{x}10^9$	$9.063\text{x}10^9$	$2.06\text{x}10^5$	$6.99\text{x}10^4$	$1.44\text{x}10^{10}$	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 δ = 0.01) can be implemented with the results summarized as

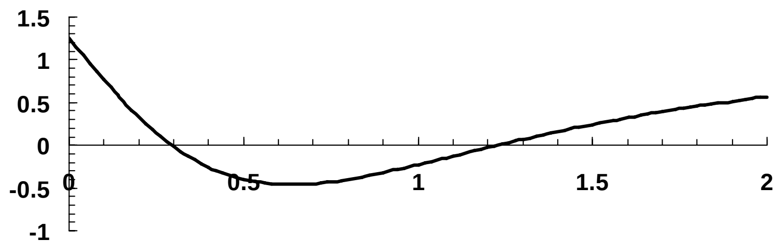
i	N_i	$f(N_i)$	δN_i	$N_i+\delta N_i$	$f(N_i+\delta N_i)$	$f'(N_i)$	ε_a

0	9.000×10^9	9.672×10^4	9.000×10^7	9.09×10^9	5.819×10^4	-0.0004	
1	9.226×10^9	6.749×10^2	9.226×10^7	9.32×10^9	-3.791×10^4	-0.0004	2.449%
2	9.228×10^9	-3.160×10^0	9.228×10^7	9.32×10^9	-3.858×10^4	-0.0004	0.017%
3	9.228×10^9	1.506×10^{-2}	9.228×10^7	9.32×10^9	-3.858×10^4	-0.0004	0.000%

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

$$f(x) = 0 = 1.25 - 3.59672 \frac{x}{(x^2 + 0.81)^{3/2}}$$

A plot indicates roots at about 0.3 and 1.23.



A numerical method can be used to determine that the roots are 0.295372 and 1.229573.

5.18 The solution can be formulated as

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

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 (5.6) 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 MATLAB function that is set up to implement 11 iterations of bisection to solve the problem.

```
function f=Fanning(func,xl,xu,varargin)
test = func(xl,varargin{:})*func(xu,varargin{:});
if test>0,error('no sign change'),end
for i = 1:11
  xr = (xl + xu)/2;
  test = func(xl,varargin{:})*func(xr,varargin{:});
  if test < 0
```

```
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    break
  end
end
f=xr;
```

This can be implemented in MATLAB. For example,

```
>> vk=@(f,Re) 4*log10(Re*sqrt(f))-0.4-1/sqrt(f);
>> format long
>> f=Fanning(vk,0.0028,0.012,2500)

f =
   0.01152832031250
```

Here are additional 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	0.0115283203125	0.0115247638118	$3.56\text{x}10^{-6}$
3000	0.0108904296875	0.0108902285840	$2.01\text{x}10^{-7}$
10000	0.0077279296875	0.0077271274071	$8.02\text{x}10^{-7}$
30000	0.0058771484375	0.0058750482511	$2.10\text{x}10^{-6}$
100000	0.0045025390625	0.0045003757287	$2.16\text{x}10^{-6}$
300000	0.0036220703125	0.0036178949673	$4.18\text{x}10^{-6}$
1000000	0.0029123046875	0.0029128191460	$5.14\text{x}10^{-7}$

5.19 The solution can be formulated as

$$f(T) = 0 = -0.10597 + 1.671\times10^{-4}T + 9.7215\times10^{-8}T^2 - 9.5838\times10^{-11}T^3 + 1.9520\times10^{-14}T^4$$

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 -0.10597];
>> roots(x)

ans =

  1.0e+003 *

  2.74833708474921 + 1.12628559147229i
  2.74833708474921 - 1.12628559147229i
 -1.13102810059654
  0.54408753765551
```

The only realistic value is 544.0875. This value can be checked using the `polyval` function,

```
>> polyval(x,544.08753765551)

ans =
    3.191891195797325e-016
```

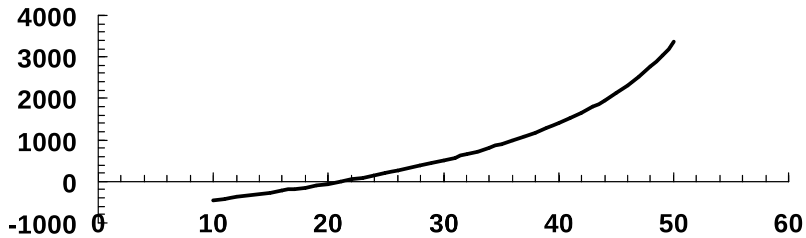
5.20 The solution can be formulated as

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

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.