CHAPTER 9

9.1 The flop counts for the tridiagonal algorithm in Fig. 9.6 can be summarized as

	Mult/Div	Add/Subtr	Total
Forward elimination	$3(n-1)$	$2(n-1)$	$5(n-1)$
Back substitution	$2n-1$	$n-1$	$3n-2$
Total	$5n-4$	$3n-3$	$\mathbf{8n-7}$

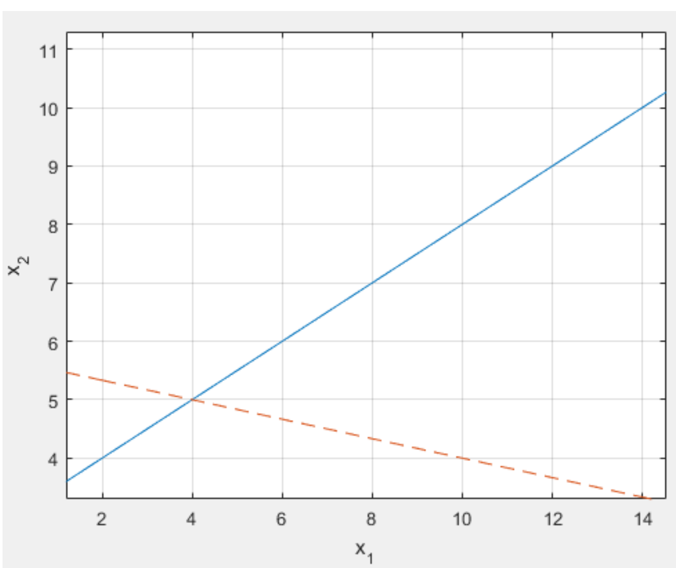
Thus, as n increases, the effort is much, much less than for a full matrix solved with Gauss elimination which is proportional to n^3.

9.2 The equations can be expressed in a format that is compatible with graphing x_2 versus x_1:

$$x_2 = 0.5x_1 + 3$$
$$x_2 = -\frac{1}{6}x_1 + \frac{34}{6}$$

which can be plotted as

```
clear,clc,clf
format compact
a11=4;a12=-8;b1=-24;
a21=1;a22=6;b2=34;
x1=[0:20];x21=(b1-a11*x1)/a12;x22=(b2-a21*x1)/a22;
plot(x1,x21,x1,x22,'--'),grid
xlabel('x_1'),ylabel('x_2')
```



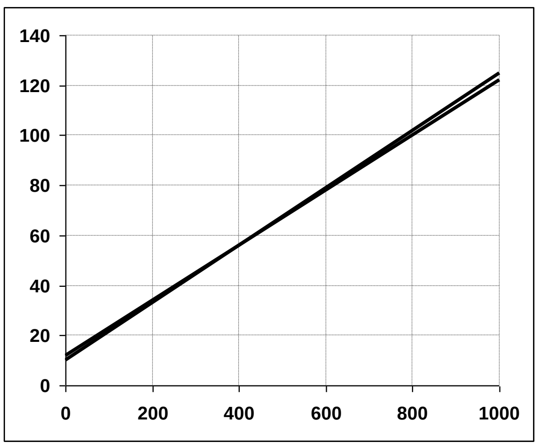
Thus, the solution is $x_1 = 4$, $x_2 = 5$. The solution can be checked by substituting it back into the equations to give

$$4(4) - 8(5) = 16 - 40 = -24$$
$$4 + 6(5) = 4 + 30 = \ 34$$

9.3 **(a)** The equations can be expressed in a format that is compatible with graphing x_2 versus x_1:

$$x_2 \quad = 0.11x_1 + 12$$
$$x_2 = 0.114943x_1 + 10$$

which can be plotted as



Thus, the solution is approximately $x_1 = 400$, $x_2 = 60$. The solution can be checked by substituting it back into the equations to give

$$-1.1(400) \ + \ 10(60) \ = 160 \approx 120$$
$$-2(400) + 17.4(60) = 244 \approx 174$$

Therefore, the graphical solution is not very good.

(b) Because the lines have very similar slopes, you would expect that the system would be ill-conditioned

(c) The determinant can be computed as

$$\begin{vmatrix} -1.1 & 10 \\ -2 & 17.4 \end{vmatrix} = -1.1(17.4) - 10(-2) = -19.14 + 20 = 0.86$$

This result is relatively low suggesting that the solution is ill-conditioned.

9.4 (a) The determinant can be evaluated as

$$D = 0\begin{bmatrix} 2 & -1 \\ -2 & 0 \end{bmatrix} - (-3)\begin{bmatrix} 1 & -1 \\ 5 & 0 \end{bmatrix} + 7\begin{bmatrix} 1 & 2 \\ 5 & -2 \end{bmatrix}$$

$$D = 0(-2) + 3(5) + 7(-12) = -69$$

(b) Cramer's rule

$$x_1 = \frac{\begin{vmatrix} 4 & -3 & 7 \\ 0 & 2 & -1 \\ 3 & -2 & 0 \end{vmatrix}}{-69} = 0.5942 \qquad x_2 = \frac{\begin{vmatrix} 0 & 4 & 7 \\ 1 & 0 & -1 \\ 5 & 3 & 0 \end{vmatrix}}{-69} = -0.0145 \qquad x_3 = \frac{\begin{vmatrix} 0 & -3 & 4 \\ 1 & 2 & 0 \\ 5 & -2 & 3 \end{vmatrix}}{-69} = 0.5652$$

(c) Pivoting is necessary, so switch the first and third rows,

$$\begin{aligned} 5x_1 - 2x_2 \qquad &= 3 \\ x_1 + 2x_2 - x_3 &= 0 \\ -3x_2 + 7x_3 &= 4 \end{aligned}$$

Multiply pivot row 1 by $1/5$ and subtract the result from the second row to eliminate the a_{21} term.

$$\begin{aligned} 5x_1 - 2x_2 \qquad &= 3 \\ 2.4x_2 - x_3 &= -0.6 \\ -3x_2 + 7x_3 &= 4 \end{aligned}$$

Pivoting is necessary so switch the second and third row,

$$\begin{aligned} 5x_1 - 2x_2 \qquad &= 3 \\ -3x_2 + 7x_3 &= 4 \\ 2.4x_2 - x_3 &= -0.6 \end{aligned}$$

Multiply pivot row 2 by $2.4/(-3)$ and subtract the result from the third row to eliminate the a_{32} term.

$$\begin{aligned} 5x_1 - 2x_2 \qquad &= 3 \\ -3x_2 + 7x_3 &= 4 \\ 4.6x_3 &= 2.6 \end{aligned}$$

The solution can then be obtained by back substitution

$$x_3 = \frac{2.6}{4.6} = 0.565217$$
$$x_2 = \frac{4 - 7(0.565217)}{-3} = -0.01449$$
$$x_1 = \frac{3 + 2(-0.01449)}{5} = 0.594203$$

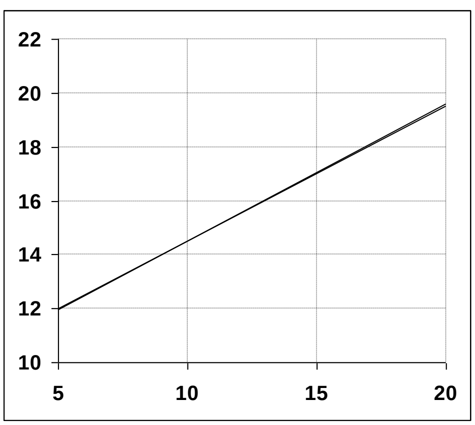
(d)

$$-3(-0.01449) + 7(0.565217) = 4$$
$$0.594203 + 2(-0.01449) - (0.565217) = 0$$
$$5(0.594203) - 2(-0.01449) = 3$$

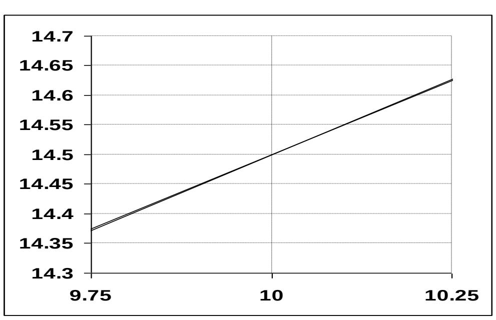
9.5 (a) The equations can be expressed in a format that is compatible with graphing x_2 versus x_1:

$$x_2 = 0.5x_1 + 9.5$$
$$x_2 = 0.51x_1 + 9.4$$

The resulting plot indicates that the intersection of the lines is difficult to detect:



Only when the plot is zoomed is it at all possible to discern that solution seems to lie at about $x_1 = 10$ and x_2 $=14.5$.



(b) The determinant can be computed as

$$\begin{vmatrix} 0.5 & -1 \\ 1.02 & -2 \end{vmatrix} = 0.5(-2) - (-1)(1.02) = 0.02$$

which is close to zero.

(c) Because the lines have very similar slopes and the determinant is so small, you would expect that the system would be ill-conditioned

(d) Multiply the first equation by $1.02/0.5$ and subtract the result from the second equation to eliminate the x_1 term from the second equation,

$$\begin{aligned} 0.5x_1 - x_2 &= -9.5 \\ 0.04x_2 &= 0.58 \end{aligned}$$

The second equation can be solved for

$$x_2 = \frac{0.58}{0.04} = 14.5$$

This result can be substituted into the first equation which can be solved for

$$x_1 = \frac{-9.5 + 14.5}{0.5} = 10$$

(e) Multiply the first equation by $1.02/0.52$ and subtract the result from the second equation to eliminate the x_1 term from the second equation,

$$\begin{aligned} 0.52x_1 \qquad - x_2 &= -9.5 \\ -0.03846x_2 &= -0.16538 \end{aligned}$$

The second equation can be solved for

$$x_2 = \frac{-0.16538}{-0.03846} = 4.3$$

This result can be substituted into the first equation which can be solved for

$$x_1 = \frac{-9.5 + 4.3}{0.52} = -10$$

Interpretation: The fact that a slight change in one of the coefficients results in a radically different solution illustrates that this system is very ill-conditioned.

9.6 (a) Multiply the first equation by –3/10 and subtract the result from the second equation to eliminate the x_1 term from the second equation. Then, multiply the first equation by 1/10 and subtract the result from the third equation to eliminate the x_1 term from the third equation.

$$\begin{aligned}10x_1+2x_2\quad -x_3&=27\\ -4.4x_2+1.7x_3&=-53.4\\ 0.8x_2\quad +6.1x_3&=-24.2\end{aligned}$$

Multiply the second equation by 0.8/(–4.4) and subtract the result from the third equation to eliminate the x_2 term from the third equation,

$$\begin{aligned}10x_1+2x_2\qquad -x_3&=27\\ -4.4x_2\qquad +1.7x_3&=-53.4\\ 6.409091x_3&=-33.9091\end{aligned}$$

Back substitution can then be used to determine the unknowns

$$x_3=\frac{-33.9091}{6.409091}=-5.29078$$
$$x_2=\frac{(-53.4-1.7(-5.29078))}{-4.4}=10.0922$$
$$x_1=\frac{(27-5.29078-2(10.0922))}{10}=0.152482$$

(b) Check:

$$10(0.152482)+2(10.0922)-(-5.29078)=27$$
$$-3(0.152482)-5(10.0922)+2(-5.29078)=-61.5$$
$$0.152482+10.0922+5(-5.29078)=-21.5$$

9.7 (a) Pivoting is necessary, so switch the first and third rows,

$$-8x_1+x_2-2x_3=-20$$
$$-3x_1-x_2+7x_3=-34$$
$$2x_1-6x_2-x_3=-38$$

Multiply the first equation by $-3/(-8)$ and subtract the result from the second equation to eliminate the a_{21} term from the second equation. Then, multiply the first equation by $2/(-8)$ and subtract the result from the third equation to eliminate the a_{31} term from the third equation.

$$\begin{aligned} -8x_1 \quad + x_2 \quad - 2x_3 &= -20 \\ -1.375x_2 + 7.75x_3 &= -26.5 \\ -5.75x_2 \quad -1.5x_3 &= -43 \end{aligned}$$

Pivoting is necessary so switch the second and third row,

$$\begin{aligned} -8x_1 \quad + x_2 \quad - 2x_3 &= -20 \\ -5.75x_2 \quad -1.5x_3 &= -43 \\ -1.375x_2 + 7.75x_3 &= -26.5 \end{aligned}$$

Multiply pivot row 2 by $-1.375/(-5.75)$ and subtract the result from the third row to eliminate the a_{32} term.

$$\begin{aligned} -8x_1 \quad + x_2 \quad - 2x_3 &= -20 \\ -5.75x_2 \quad -1.5x_3 &= -43 \\ 8.108696x_3 &= -16.21739 \end{aligned}$$

At this point, the determinant can be computed as

$$D = -8 \times -5.75 \times 8.108696 \times (-1)^2 = 373$$

The solution can then be obtained by back substitution

$$x_3 = \frac{-16.21739}{8.108696} = -2$$
$$x_2 = \frac{-43 + 1.5(-2)}{-5.75} = 8$$
$$x_1 = \frac{-20 + 2(-2) - 1(8)}{-8} = 4$$

(b) Check:

$$\begin{aligned} 2(4) - 6(8) - (-2) &= -38 \\ -3(4) - (8) + 7(-2) &= -34 \\ -8(4) + (8) - 2(-2) &= -20 \end{aligned}$$

9.8 Multiply the first equation by –0.4/0.8 and subtract the result from the second equation to eliminate the x_1 term from the second equation.

$$\begin{bmatrix} 0.8 & -0.4 & \\ & 0.6 & -0.4 \\ & -0.4 & 0.8 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 41 \\ 45.5 \\ 105 \end{Bmatrix}$$

Multiply pivot row 2 by –0.4/0.6 and subtract the result from the third row to eliminate the x_2 term.

$$\begin{bmatrix} 0.8 & -0.4 & \\ & 0.6 & -0.4 \\ & & 0.533333 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 41 \\ 45.5 \\ 135.3333 \end{Bmatrix}$$

The solution can then be obtained by back substitution

$$x_3 = \frac{135.3333}{0.533333} = 253.75$$
$$x_2 = \frac{45.5 - (-0.4)253.75}{0.6} = 245$$
$$x_1 = \frac{41 - (-0.4)245}{0.8} = 173.75$$

(b) Check:

$$0.8(173.75) - 0.4(245) = 41$$
$$-0.4(173.75) + 0.8(245) - 0.4(253.75) = 25$$
$$-0.4(245) + 0.8(253.75) = 105$$

9.9 The mass balances can be written as

$$Q_{21}c_2 + 500 = Q_{12}c_1 + Q_{13}c_1$$
$$Q_{12}c_1 = Q_{21}c_2 + Q_{23}c_2$$
$$Q_{13}c_1 + Q_{23}c_2 + 500 = Q_{33}c_3$$

or collecting terms

$$\begin{aligned} (Q_{12} + Q_{13})c_1 \quad & - Q_{21}c_2 & & = 500 \\ -Q_{12}c_1 \quad & + (Q_{21} + Q_{23})c_2 & & = 0 \\ -Q_{13}c_1 \quad & - Q_{23}c_2 & + Q_{33}c_3 & = 500 \end{aligned}$$

Substituting the values for the flows and expressing in matrix form

$$\begin{bmatrix} 130 & -30 & 0 \\ -90 & 90 & 0 \\ -40 & -60 & 120 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} = \begin{Bmatrix} 500 \\ 0 \\ 500 \end{Bmatrix}$$

A solution can be obtained with a MATLAB script as

```
clear, clc
format compact
Q33=120;Q13=40;Q12=90;Q23=60;Q21=30;
A=[Q12+Q13 -Q21 0;-Q12 Q21+Q23 0;-Q13 -Q23 Q33]
b = [500 0 500]';
c = A\b

A =
   130   -30     0
   -90    90     0
   -40   -60   120
c =
    5.0000
    5.0000
    8.3333
```

9.10 Let x_i = the volume taken from pit i. Therefore, the following system of equations must hold

$$\begin{bmatrix} 0.55 & 0.25 & 0.25 \\ 0.30 & 0.45 & 0.20 \\ 0.15 & 0.30 & 0.55 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 4800 \\ 5800 \\ 5700 \end{Bmatrix}$$

MATLAB can be used to solve this system of equations for

```
clear, clc
A=[0.55 0.25 0.25;0.3 0.45 0.20;0.15 0.30 0.55];
b=[4800;5800;5700];
x=A\b

x =
   1.0e+03 *
    2.4167
    9.1933
    4.6900
```

Therefore, we take x_1 = 2,416.7, x_2 = 9,193.3, and x_3 = 4,690.0 m^3 from pits 1, 2 and 3 respectively.

9.11 Let c_i = component *i*. Therefore, the following system of equations must hold

$$\begin{bmatrix} 15 & 17 & 19 \\ 0.3 & 0.4 & 0.55 \\ 1 & 1.2 & 1.5 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} = \begin{Bmatrix} 3890 \\ 95 \\ 282 \end{Bmatrix}$$

The solution can be developed with MATLAB:

```
clear, clc
A=[15 17 19;0.3 0.4 0.55;1 1.2 1.5];
b=[3890;95;282];
c=A\b

c =
   90.0000
   60.0000
   80.0000
```

Therefore, $c_1 = 90$, $c_2 = 60$, and $c_3 = 80$.

9.12 Centered differences (recall Chap. 4) can be substituted for the derivatives to give

$$0 = D\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} - U\frac{c_{i+1} - c_{i-1}}{2\Delta x} - kc_i$$

collecting terms yields

$$-(D + 0.5U\Delta x)c_{i-1} + (2D + k\Delta x^2)c_i - (D - 0.5U\Delta x)c_{i+1} = 0$$

Assuming $\Delta x = 1$ and substituting the parameters gives

$$-2.5c_{i-1} + 4.2c_i - 1.5c_{i+1} = 0$$

For the first interior node ($i = 1$),

$$4.2c_1 - 1.5c_2 = 200$$

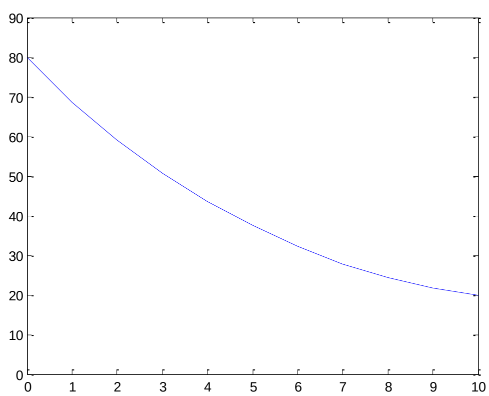
For the last interior node ($i = 9$)

$$-2.5c_8 + 4.2c_9 = 30$$

The following script generates and plots the solution:

```
clear, clc, clf, format compact
D=2;U=1;k=0.2;c0=80;c10=20;dx=1;
diag=(2*D+k*dx^2);
sup=-(D-0.5*U*dx);
sub=-(D+0.5*U*dx);
r1=-sub*c0;
rn=-sup*c10;
A=[diag sup 0 0 0 0 0 0 0
     sub diag sup 0 0 0 0 0 0
     0 sub diag sup 0 0 0 0 0
     0 0 sub diag sup 0 0 0 0
     0 0 0 sub diag sup 0 0 0
     0 0 0 0 sub diag sup 0 0
     0 0 0 0 0 sub diag sup 0
     0 0 0 0 0 0 sub diag sup
     0 0 0 0 0 0 0 sub diag];
b=[r1 0 0 0 0 0 0 0 rn]';
c=A\b
c=[80 c' 20];
x=0:1:10;
plot(x,c)
ylim([0 90])

c =
   68.6755
   58.9581
   50.6236
   43.4825
   37.3783
   32.1884
   27.8305
   24.2779
   21.5940
```



9.13 For the first stage, the mass balance can be written as (note: $x_{\text{out}} = x_1$ and $y_{\text{out}} = y_5$)

$$F_1 y_{\text{in}} + F_2 x_2 = F_2 x_1 + F_1 y_1$$

Substituting $x = Ky$ and rearranging gives

$$-\left(1+\frac{F_2}{F_1}K\right)y_1 + \frac{F_2}{F_1}Ky_2 = -y_{\text{in}}$$

Using a similar approach, the equation for the last stage is

$$y_4 - \left(1+\frac{F_2}{F_1}K\right)y_5 = -\frac{F_2}{F_1}x_{\text{in}}$$

For interior stages,

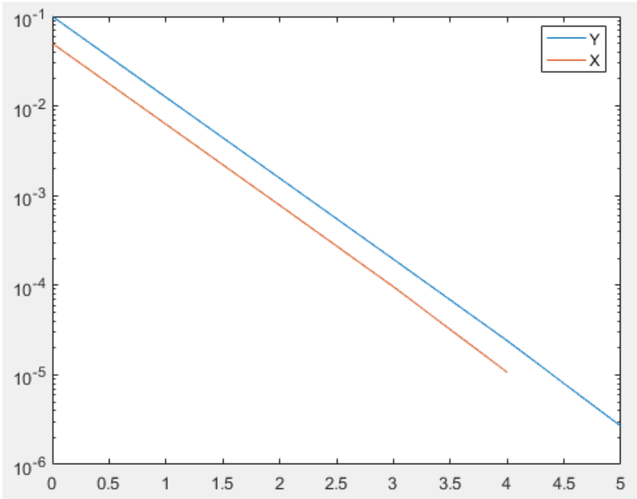
$$y_{i-1} - \left(1+\frac{F_2}{F_1}K\right)y_i + \frac{F_2}{F_1}Ky_{i+1} = 0$$

The solution can be developed in a number of ways. For example, using a MATLAB script

```
clear,clc,clf,format short g
format compact
F1=500;F2=1000;yin=0.1;xin=0;K=4;
diag=1+F2/F1*K; super=-F2/F1*K; sub=-1;
r1=yin; rn=F2/F1*xin;
A=[diag super 0 0 0
     sub diag super 0 0
     0 sub diag super 0
     0 0 sub diag super
     0 0 0 sub diag];
b=[r1 0 0 0 rn]';
Y=A\b
X=K*Y
x=0:1:5; Y=[yin Y']; X=[X' xin];
semilogy(x,Y,x,X),legend('Y','X','location','best')
```

The results are

```
Y =
       0.0125
    0.0015621
   0.00019493
   2.4033e-05
   2.6703e-06
X =
     0.049999
    0.0062485
   0.00077973
   9.6131e-05
   1.0681e-05
```



Therefore, $y_{\text{out}} = 2.67\times10^{-6}$ and $x_{\text{out}} = 1.0681\times10^{-5}$.

9.14 Assuming a unit flow for Q_1, the simultaneous equations can be written in matrix form as

$$\begin{bmatrix} -2 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & -2 & 3 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \begin{Bmatrix} Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ Q_7 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix}$$

These equations can then be solved with MATLAB,

```
>> A=[-2 1 2 0 0 0;
0 0 -2 1 2 0;
0 0 0 0 -2 3;
1 1 0 0 0 0;
0 1 -1 -1 0 0;
0 0 0 1 -1 -1];
>> B=[0 0 0 1 0 0 ]';
>> Q=A\B

Q =
    0.5059
    0.4941
    0.2588
    0.2353
    0.1412
    0.0941
```

9.15 The solution can be generated with MATLAB,

```
>> A=[1 0 0 0 0 0 0 0 1 0;
      0 0 1 0 0 0 0 1 0 0;
      0 1 0 3/5 0 0 0 0 0 0;
      -1 0 0 -4/5 0 0 0 0 0 0;
      0 -1 0 0 0 0 3/5 0 0 0;
      0 0 0 0 -1 0 -4/5 0 0 0;
      0 0 -1 -3/5 0 1 0 0 0 0;
      0 0 0 4/5 1 0 0 0 0 0;
      0 0 0 0 0 -1 -3/5 0 0 0;
      0 0 0 0 0 0 4/5 0 0 1];
>> B=[0 0 -74 0 0 24 0 0 0 0]';
>> x=A\B

x =
   37.3333
  -46.0000
   74.0000
  -46.6667
   37.3333
   46.0000
  -76.6667
  -74.0000
  -37.3333
   61.3333
```

Therefore, in kN

$AB = 37.3333$ $\quad BC = -46$ $\quad AD = 74$ $\quad BD = -46.6667$ $\quad CD = 37.3333$

$DE = 46$ $\quad CE = -76.6667$ $\quad A_x = -74$ $\quad A_y = -37.33333$ $\quad E_y = 61.3333$

9.16

```
function x=pentasol(A,b)
% pentasol: pentadiagonal system solver banded system
%   x=pentasol(A,b):
%      Solve a pentadiagonal system Ax=b
% input:
%   A = pentadiagonal matrix
%   b = right hand side vector
% output:
%   x = solution vector

% Error checks
[m,n]=size(A);
if m~=n,error('Matrix must be square');end
if length(b)~=m,error('Matrix and vector must have the same
number of rows');end
x=zeros(n,1);

% Extract bands
d=[0;0;diag(A,-2)];
```

```
e=[0;diag(A,-1)];
f=diag(A);
g=diag(A,1);
h=diag(A,2);
delta=zeros(n,1);
epsilon=zeros(n-1,1);
gamma=zeros(n-2,1);
alpha=zeros(n,1);
c=zeros(n,1);
z=zeros(n,1);

% Decomposition
delta(1)=f(1);
epsilon(1)=g(1)/delta(1);
gamma(1)=h(1)/delta(1);
alpha(2)=e(2);
delta(2)=f(2)-alpha(2)*epsilon(1);
epsilon(2)=(g(2)-alpha(2)*gamma(1))/delta(2);
gamma(2)=h(2)/delta(2);
for k=3:n-2
  alpha(k)=e(k)-d(k)*epsilon(k-2);
  delta(k)=f(k)-d(k)*gamma(k-2)-alpha(k)*epsilon(k-1);
  epsilon(k)=(g(k)-alpha(k)*gamma(k-1))/delta(k);
  gamma(k)=h(k)/delta(k);
end
alpha(n-1)=e(n-1)-d(n-1)*epsilon(n-3);
delta(n-1)=f(n-1)-d(n-1)*gamma(n-3)-alpha(n-1)*epsilon(n-2);
epsilon(n-1)=(g(n-1)-alpha(n-1)*gamma(n-2))/delta(n-1);
alpha(n)=e(n)-d(n)*epsilon(n-2);
delta(n)=f(n)-d(n)*gamma(n-2)-alpha(n)*epsilon(n-1);
% Forward substitution
c(1)=b(1)/delta(1);
c(2)=(b(2)-alpha(2)*c(1))/delta(2);
for k=3:n
  c(k)=(b(k)-d(k)*c(k-2)-alpha(k)*c(k-1))/delta(k);
end
% Back substitution
x(n)=c(n);
x(n-1)=c(n-1)-epsilon(n-1)*x(n);
for k=n-2:-1:1
  x(k)=c(k)-epsilon(k)*x(k+1)-gamma(k)*x(k+2);
end
```

Test of function:

```
>> A=[8 -2 -1 0 0
-2 9 -4 -1 0
-1 3 7 -1 -2
0 -4 -2 12 -5
0 0 -7 -3 15];
>> b=[5 2 1 1 5]';
>> x=pentasol(A,b)

x =
    0.7993
    0.5721
```

```
    0.2503
    0.5491
    0.5599
```

9.17 Here is the M-file function based on Fig. 9.5 to implement Gauss elimination with partial pivoting

```
function [x, D] = GaussPivotNew(A, b, tol)
% GaussPivotNew: Gauss elimination pivoting
%   [x, D] = GaussPivotNew(A,b,tol): Gauss elimination with
pivoting.
% input:
%   A = coefficient matrix
%   b = right hand side vector
%   tol = tolerance for detecting "near zero"
% output:
%   x = solution vector
%   D = determinant

[m,n]=size(A);
if m~=n, error('Matrix A must be square'); end
nb=n+1;
Aug=[A b];
npiv=0;
% forward elimination
for k = 1:n-1
  % partial pivoting
  [big,i]=max(abs(Aug(k:n,k)));
  ipr=i+k-1;
  if ipr~=k
    npiv=npiv+1;
    Aug([k,ipr],:)=Aug([ipr,k],:);
  end
  if abs(Aug(k,k))<=tol
    D=0;
    error('Singular or near singular system')
  end
  for i = k+1:n
    factor=Aug(i,k)/Aug(k,k);
    Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
  end
end
for i = 1:n
  if abs(Aug(i,i))<=tol
    D=0;
    error('Singular or near singular system')
  end
end
% back substitution
x=zeros(n,1);
x(n)=Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
  x(i)=(Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
D=(-1)^npiv;
for i=1:n
```

```
  D=D*Aug(i,i);
end
```

Here is a script to solve Prob. 9.5 for the two cases of tol:

```
clear; clc; format short g
A=[0.5 -1;1.02 -2];
b=[-9.5;-18.8];
disp('Solution and determinant calculated with GaussPivotNew:')
[x, D] = GaussPivotNew(A,b,1e-5)
disp('Determinant calculated with det:')
D=det(A)
```

The resulting output is

```
Solution and determinant calculated with GaussPivotNew:
x =
           10
         14.5
D =
         0.02

Determinant calculated with det:
D =
         0.02
```

9.18 **(a)** Centered finite differences for the first and second derivatives are

$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x} \qquad \frac{d^2c}{dx^2} = \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2}$$

Substituting these into the differential equation gives,

$$0 = D\frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} - U\frac{c_{i+1} - c_{i-1}}{2\Delta x} - kc_i$$

Collecting terms gives

$$-\left(\frac{D}{\Delta x^2} - \frac{U}{2\Delta x}\right)c_{i+1} + \left(\frac{2D}{\Delta x^2} + k\right)c_i - \left(\frac{D}{\Delta x^2} + \frac{U}{2\Delta x}\right)c_{i-1} = 0$$

(b)

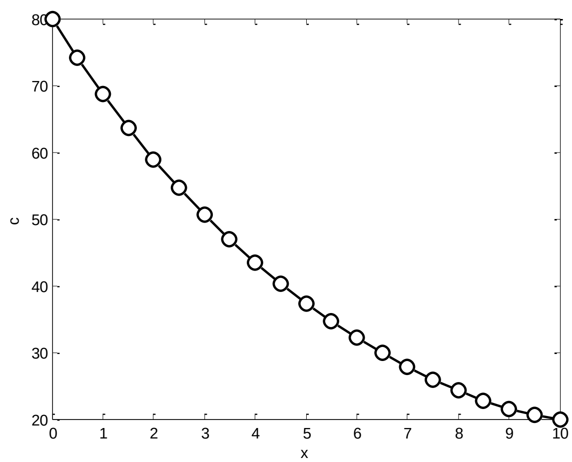
```
function [x, c] = YourLastName_Reactor(D, U, k, c0, cL, L, dx)
x=[0:dx:L]';
ns=L/dx-1;
A=zeros(ns);
b=zeros(ns,1);
A(1,1)=2*D/dx^2+k;A(1,2)=-(D/dx^2-U/2/dx);
```

```
b(1)=(D/dx^2+U/2/dx)*c0;
for i=2:ns-1
  A(i,i-1)=-(D/dx^2+U/2/dx);
  A(i,i)=2*D/dx^2+k;
  A(i,i+1)=-(D/dx^2-U/2/dx);
end
A(ns,ns-1)=-(D/dx^2+U/2/dx);
A(ns,ns)=2*D/dx^2+k;
b(ns)=(D/dx^2-U/2/dx)*cL;
c=A\b;
c=[c0;c;cL];
```

(c) Script:

```
clear,clc,clf
L=10;dx=0.5;D=2;U=1;k=0.2;c0=80;cL=20;
[x, c]=YourLastName_Reactor(D, U, k, c0, cL, L, dx);
plot(x,c,'-ok','LineWidth',2,...
     'MarkerSize',10,...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','w')
xlabel('x');ylabel('c')
```

(d) Output:



Alternative solution: Another way to do this is to use the tridiagonal function from the book. This has the advantage that it is more efficient than using LU factorization on the full matrix

(b) For this version we also need a function to implement the tridiagonal solver.

```
function [x, c] = YourLastName_Reactor(D, U, k, c0, cL, L, dx)
x=[0:dx:L]';
ns=L/dx-1;
e=ones(ns,1)*(-(D/dx^2+U/2/dx));
f=ones(ns,1)*(2*D/dx^2+k);
```

```
g=ones(ns,1)*(-(D/dx^2-U/2/dx));
r=zeros(ns,1);
r(1)=(D/dx^2+U/2/dx)*c0;
r(ns)=(D/dx^2-U/2/dx)*cL;
% solve the system
c = Tridiag(e,f,g,r);
c=[c0;c';cL];

function x = Tridiag(e,f,g,r)
% Tridiag: Tridiagonal equation solver banded system
%   x = Tridiag(e,f,g,r): Tridiagonal system solver.
% input:
%   e = subdiagonal vector
%   f = diagonal vector
%   g = superdiagonal vector
%   r = right hand side vector
% output:
%   x = solution vector
n=length(f);
% forward elimination
for k = 2:n
  factor = e(k)/f(k-1);
  f(k) = f(k) - factor*g(k-1);
  r(k) = r(k) - factor*r(k-1);
end
% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
  x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```

(c) and (d) The script and resulting plot are identical to the first version.

9.19 **(a)** Centered finite differences for the second derivative is

$$\frac{d^2y}{dx^2} = \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2}$$

Substituting this into the differential equation gives,

$$0 = EI\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} - \frac{wLx_i}{2} + \frac{wx_i^2}{2}$$

Collecting terms yields

$$y_{i-1} - 2y_i + y_{i+1} = \frac{w\Delta x^2}{2EI}(Lx_i - x_i^2)$$

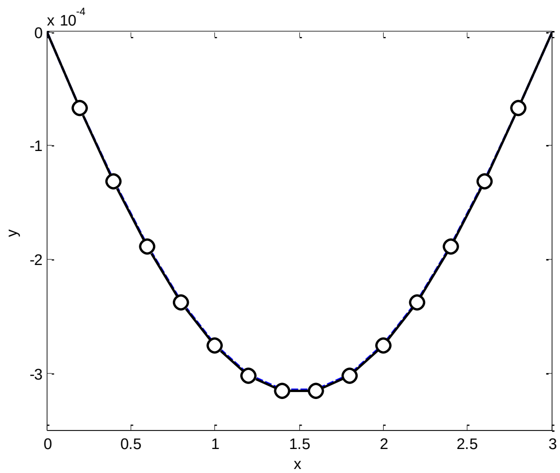
(b) The following solution also includes the analytical solution for comparison:

```
function [x, y] = YourLastName_Beam(E, I, w, y0, yL, L, dx)
ns=L/dx-1;
x=[dx:dx:L-dx]';
% set up system of equations
A=zeros(ns); b=zeros(ns,1);
A(1,1)=-2;A(1,2)=1;
b(1)=-y0+w*dx^2/(2*E*I)*(L*x(1)-x(1)^2);
for i=2:ns-1
  A(i,i-1)=1; A(i,i)=-2; A(i,i+1)=1;
  b(i)=w*dx^2/(2*E*I)*(L*x(i)-x(i)^2);
end
A(ns,ns-1)=1; A(ns,ns)=-2;
b(ns)=-yL+w*dx^2/(2*E*I)*(L*x(ns)-x(ns)^2);
% solve the system
y=A\b;
x=[0;x;L]; y=[y0;y;yL];
```

(c) Script:

```
clear,clc,clf
L=3;dx=0.2;E=250e9;I=3e-4;w=22500;y0=0;yL=0;
[x, y]= YourLastName_Beam(E, I, w, y0, yL, L, dx);
yanal=w*L*x.^3/(12*E*I)-w*x.^4/(24*E*I)-w*L^3*x/(24*E*I);
plot(x,yanal,'--',x,y,'-ok','LineWidth',2,...
    'MarkerSize',10,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','w')
xlabel('x');ylabel('y')
```

(d) Output:



Alternative solution: Another way to do this is to use the tridiagonal function from the book. This has the advantage that it is more efficient than using LU factorization on the full matrix

(b) For this version we also need a function to implement the tridiagonal solver.

```
function [x, y] = YourLastName_Beam(E, I, w, y0, yL, L, dx)
ns=L/dx-1;
x=[dx:dx:L-dx]';
% set up system of equations
e=ones(ns,1);f=-2*ones(ns,1);g=ones(ns,1);
r=w*dx^2/(2*E*I)*(L*x-x.^2);
r(1)=r(1)-y0;r(ns)=r(ns)-yL;
% solve the system
y = Tridiag(e,f,g,r);
x=[0;x;L]; y=[y0;y';yL];

function x = Tridiag(e,f,g,r)
% Tridiag: Tridiagonal equation solver banded system
%   x = Tridiag(e,f,g,r): Tridiagonal system solver.
% input:
%   e = subdiagonal vector
%   f = diagonal vector
%   g = superdiagonal vector
%   r = right hand side vector
% output:
%   x = solution vector
n=length(f);
% forward elimination
for k = 2:n
  factor = e(k)/f(k-1);
  f(k) = f(k) - factor*g(k-1);
  r(k) = r(k) - factor*r(k-1);
end
% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
  x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```

(c) and (d) The script and resulting plot are identical to the first version.

9.20 (a) A centered finite difference approximation for the second derivative is

$$\frac{d^2T}{dx^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}$$

Substituting this into the differential equation gives,

$$0 = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + h'(T_\infty - T_i)$$

Collecting terms yields

$$-\frac{1}{\Delta x^2}T_{i-1}+\left(\frac{2}{\Delta x^2}+h'\right)T_i-\frac{1}{\Delta x^2}T_{i+1}=h'T_\infty$$

or multiplying by Δx^2,

$$-T_{i-1}+(2+h'\Delta x^2)T_i-T_{i+1}=h'\Delta x^2 T_\infty$$

(b)

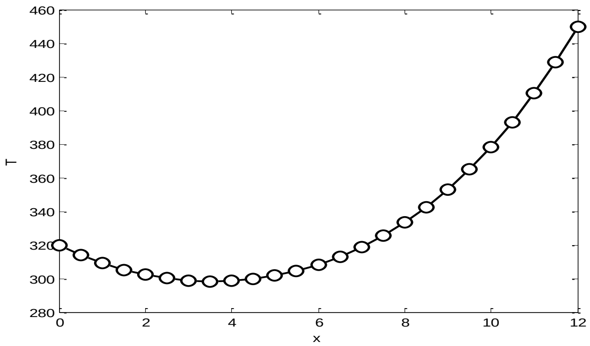
```
function [x, T] = YourLastName_rod(hp, Tinf, T0, TL, L, dx)
ns=L/dx-1;
x=[dx:dx:L-dx]';
% set up system of equations
A=zeros(ns); b=zeros(ns,1);
A(1,1)=2+hp*dx^2;A(1,2)=-1;
b(1)=hp*dx^2*Tinf+T0;
for i=2:ns-1
  A(i,i-1)=-1; A(i,i)=2+hp*dx^2; A(i,i+1)=-1;
  b(i)=hp*dx^2*Tinf;
end
A(ns,ns-1)=-1; A(ns,ns)=2+hp*dx^2;
b(ns)= hp*dx^2*Tinf+TL;
% solve the system
T=A\b;
x=[0;x;L]; T=[T0;T;TL];
```

(d) Test your script for the following parameters: $h' = 0.05\ \mathrm{m}^{-2}$, $L = 10$ m, $T_\infty = 200$ K, $T(0) = 300$ K, and $T(L) = 400$ K.

(c) Script:

```
clear,clc,clf
L=12;dx=0.5;hp=0.0425;Tinf=220;T0=320;TL=450;
[x, y]= YourLastName_rod(hp, Tinf, T0, TL, L, dx);
plot(x,y,'-ok','LineWidth',2,...
    'MarkerSize',10,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','w')
xlabel('x');ylabel('T')
```

(d) Output:



Alternative solution: Another way to do this is to use the tridiagonal function from the book. This has the advantage that it is more efficient than using LU factorization on the full matrix

(b) For this version we also need a function to implement the tridiagonal solver.

```
function [x, T] = YourLastName_rod(hp, Tinf, T0, TL, L, dx)
ns=L/dx-1;
x=[dx:dx:L-dx]';
% set up system of equations
e=-ones(ns,1);f=ones(ns,1)*(2+hp*dx^2);g=-ones(ns,1);
r=ones(ns,1)*hp*dx^2*Tinf;
r(1)=r(1)+T0;r(ns)=r(ns)+TL;
% solve the system
T = Tridiag(e,f,g,r);
x=[0;x;L]; T=[T0;T';TL];

function x = Tridiag(e,f,g,r)
% Tridiag: Tridiagonal equation solver banded system
%   x = Tridiag(e,f,g,r): Tridiagonal system solver.
% input:
%   e = subdiagonal vector
%   f = diagonal vector
%   g = superdiagonal vector
%   r = right hand side vector
% output:
%   x = solution vector
n=length(f);
% forward elimination
for k = 2:n
  factor = e(k)/f(k-1);
  f(k) = f(k) - factor*g(k-1);
  r(k) = r(k) - factor*r(k-1);
end
% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
  x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```

(c) and (d) The script and resulting plot are identical to the first version.