CHAPTER 8

8.1

```
>> Aug = [A eye(size(A))]
```

Here's an example session of how it can be employed.

```
>> A = rand(3)

A =
    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214

>> Aug = [A eye(size(A))]
Aug =

    0.9501    0.4860    0.4565    1.0000         0         0
    0.2311    0.8913    0.0185         0    1.0000         0
    0.6068    0.7621    0.8214         0         0    1.0000
```

8.2 **(a)** $[A]$: 3×2 $[B]$: 3×3 $\{C\}$: 3×1 $[D]$: 2×4
$[E]$: 3×3 $[F]$: 2×3 $\lfloor G \rfloor$: 1×3

(b) square: $[B]$, $[E]$; column: $\{C\}$, row: $\lfloor G \rfloor$

(c) $a_{12} = 7$, $b_{23} = 7$, $d_{32} =$ undefined, $e_{22} = 2$, $f_{12} = 0$, $g_{12} = 6$

(d)

(1) $[E]+[B]=\begin{bmatrix} 5 & 8 & 15 \\ 8 & 4 & 10 \\ 6 & 0 & 10 \end{bmatrix}$

(2) $[A] + [F] =$ undefined

(3) $[B]-[E]=\begin{bmatrix} 3 & -2 & -1 \\ -6 & 0 & 4 \\ -2 & 0 & -2 \end{bmatrix}$

(4) $7[B]=\begin{bmatrix} 28 & 21 & 49 \\ 7 & 14 & 49 \\ 14 & 0 & 28 \end{bmatrix}$

(5) $[C]^T = \lfloor 3 \quad 6 \quad 1 \rfloor$

(6) $[E][B]=\begin{bmatrix} 25 & 13 & 74 \\ 36 & 25 & 75 \\ 28 & 12 & 52 \end{bmatrix}$

(7) $[B][A]=\begin{bmatrix} 54 & 76 \\ 41 & 53 \\ 28 & 38 \end{bmatrix}$

(8) $[D]^T = \begin{bmatrix} 9 & 2 \\ 4 & -1 \\ 3 & 7 \\ -6 & 5 \end{bmatrix}$

(9) $[A][C]$ = undefined

(10) $[I][B]=\begin{bmatrix}4 & 3 & 7\\1 & 2 & 7\\2 & 0 & 4\end{bmatrix}$

(11) $[E]^T[E]=\begin{bmatrix}66 & 19 & 53\\19 & 29 & 46\\53 & 46 & 109\end{bmatrix}$

(12) $[C]^T[C]=46$

8.3 The terms can be collected to give

$$\begin{bmatrix}0 & -7 & 5\\0 & 4 & 7\\-4 & 3 & -7\end{bmatrix}\begin{Bmatrix}x_1\\x_2\\x_3\end{Bmatrix}=\begin{Bmatrix}50\\-30\\40\end{Bmatrix}$$

Here is the MATLAB script:

```
A = [0 -7 5;0 4 7;-4 3 -7];
b = [50;-30;40];
x = A\b
AT = A'
AI = inv(A)
```

Results:

```
x =
  -15.1812
   -7.2464
   -0.1449
AT =
     0     0    -4
    -7     4     3
     5     7    -7
AI =
   -0.1775   -0.1232   -0.2500
   -0.1014    0.0725         0
    0.0580    0.1014         0
```

8.4 (a) Here are all the possible multiplications:

```
A=[6 -1;12 8;-5 4];
B=[4 0;0.5 2];
C=[2 -2;3 1];
AB=A*B
AC=A*C
BC=B*C
CB=C*B
```

```
AB =
   23.5000   -2.0000
   52.0000   16.0000
  -18.0000    8.0000

AC =
     9   -13
    48   -16
     2    14

BC =
     8    -8
     7     1

CB =
    7.0000   -4.0000
   12.5000    2.0000
```

(b) $[B][A]$ and $[C][A]$ are impossible because the inner dimensions do not match:

$(2\times2) * (3\times2)$

(c) According to **(a)**, $[B][C] \neq [C][B]$

8.5

```
>> A=[3+2*i 4;-i 1]
>> b=[2+i;3]
>> z=A\b

z =
  -0.5333 + 1.4000i
   1.6000 - 0.5333i
```

8.6

```
function X=mmult(Y,Z)
% mmult: matrix multiplication
%   X=mmult(Y,Z)
%     multiplies two matrices
% input:
%   Y = first matrix
%   Z = second matrix
% output:
%   X = product

if nargin<2,error('at least 2 input arguments required'),end
[m,n]=size(Y);[n2,p]=size(Z);
if n~=n2,error('Inner matrix dimensions must agree.'),end
for i=1:m
  for j=1:p
    s=0.;
    for k=1:n
      s=s+Y(i,k)*Z(k,j);
```

```
    end
    X(i,j)=s;
  end
end
```

Test of function for cases from Prob. 8.4:

```
>> A=[6 -1;12 8;-5 4];
>> B=[4 0;0.5 2];
>> C=[2 -2;3  1];
>> mmult(A,B)

ans =
   23.5000   -2.0000
   52.0000   16.0000
  -18.0000    8.0000

>> mmult(A,C)

ans =
     9   -13
    48   -16
     2    14

>> mmult(B,C)

ans =
8    -8
7     1

>> mmult(C,B)

ans =
    7.0000   -4.0000
   12.5000    2.0000

>> mmult(B,A)

??? Error using ==> mmult
Inner matrix dimensions must agree.

>> mmult(C,A)

??? Error using ==> mmult
Inner matrix dimensions must agree.
```

8.7

```
function AT=matran(A)
% matran: matrix transpose
%   AT=mtran(A)
%      generates the transpose of a matrix
% input:
%   A = original matrix
% output:
%   AT = transpose
```

```
[m,n]=size(A);
for i = 1:m
  for j = 1:n
    AT(j,i) = A(i,j);
  end
end
```

Test of function for cases from Prob. 8.4:

```
>> A=[6 -1;12 8;-5 4];
>> B=[4 0;0.5 2];
>> C=[2 -2;3  1];

>> matran(A)
ans =
     6    12    -5
    -1     8     4

>> matran(B)
ans =
    4.0000    0.5000
         0    2.0000

>> matran(C)
ans =
     2     3
    -2     1
```

8.8

```
function B = permut(A,r1,r2)
% permut: Switch rows of matrix A with a permutation matrix
% B = permut(A,r1,r2)
% input:
% A = original matrix
% r1, r2 = rows to be switched
% output:
% B = matrix with rows switched

[m,n] = size(A);
if m ~= n, error('matrix not square'), end
if r1 == r2 | r1>m | r2>m
  error('row numbers are equal or exceed matrix dimensions')
end
P = zeros(n);
P(r1,r2)=1;P(r2,r1)=1;
for i = 1:m
  if i~=r1 & i~=r2
    P(i,i)=1;
  end
end
B=P*A;
```

Test script:

```
clc
A=[1 2 3 4;5 6 7 8;9 10 11 12;13 14 15 16]
B = permut(A,3,1)
B = permut(A,3,5)

A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
B =
     9    10    11    12
     5     6     7     8
     1     2     3     4
    13    14    15    16

??? Error using ==> permut
row numbers are equal or exceed matrix dimensions

Error in ==> permutScript at 4
B = permut(A,3,5)
```

8.9 The mass balances can be written as

$$
\begin{aligned}
&(Q_{15}+Q_{12})c_1 && && -Q_{31}c_3 && && = Q_{01}c_{01} \\
&-Q_{12}c_1 && +(Q_{23}+Q_{24}+Q_{25})c_2 && && && = 0 \\
& && -Q_{23}c_2 && +(Q_{31}+Q_{34})c_3 && && = Q_{03}c_{03} \\
& && -Q_{24}c_2 && -Q_{34}c_3+Q_{44}c_4 && -Q_{54}c_5 && = 0 \\
&-Q_{15}c_1 && -Q_{25}c_2 && && +(Q_{54}+Q_{55})c_5 && = 0
\end{aligned}
$$

The following MATLAB script can then be used to solve for the concentrations

```
clear, clc, format compact
Q01=5;c01=10;
Q03=8;c03=20;
Q15=3;Q12=3;Q25=1;Q24=1;Q23=1;
Q34=8;Q31=1;
Q44=11;Q55=2;Q54=2;
Q = [Q15+Q12 0 -Q31 0 0;
-Q12 Q23+Q24+Q25 0 0 0;
0 -Q23 Q31+Q34 0 0;
0 -Q24 -Q34 Q44 -Q54;
-Q15 -Q25 0 0 Q54+Q55];
Qc = [Q01*c01;0;Q03*c03;;0;0];
c = Q\Qc
```

```
c =
   11.5094
   11.5094
   19.0566
   16.9983
   11.5094
```

8.10 The problem can be written in matrix form as

$$\begin{bmatrix} 0.866025 & 0 & -0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.866025 & 0 & 0 & 0 \\ -0.866025 & -1 & 0 & -1 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.866025 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ H_2 \\ V_2 \\ V_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

MATLAB can then be used to solve for the forces and reactions,

```
clc; format compact, format short g
A = [0.866025 0 -0.5 0 0 0;
0.5 0 0.866025 0 0 0;
-0.866025 -1 0 -1 0 0;
-0.5 0 0 0 -1 0;
0 1 0.5 0 0 0;
0 0 -0.866025 0 0 -1];
b = [0 -1000 0 0 0 0]';
F = A\b

F =
         -500
       433.01
      -866.03
            0
          250
          750
```

Therefore, $F_1 = -500$, $F_2 = 433.01$, $F_3 = -866.03$, $H_2 = 0$, $V_2 = 250$, and $V_3 = 750$

8.11

```
clc; format short g
k1=10;k2=30;k3=30;k4=10;
m1=1;m2=1;m3=1;
km=[(1/m1)*(k2+k1),-(k2/m1),0
    -(k2/m2),(1/m2)*(k2+k3),-(k3/m2)
    0,-(k3/m3),(1/m3)*(k3+k4)]
x=[0.05;0.04;0.03];
kmx=-km*x
```

```
km =
    40   -30     0
   -30    60   -30
     0   -30    40
kmx =
         -0.8
            -5.5511e-17
            0
```

Therefore, $\ddot{x}_1 = -0.8$, $\ddot{x}_2 = 0$, and $\ddot{x}_3 = 0$ m/s^2.

8.12 Vertical force balances can be written to give the following system of equations,

$$
\begin{aligned}
&m_1 g + k_2(x_2 - x_1) - k_1 x_1 = 0\\
&m_2 g + k_3(x_3 - x_2) - k_2(x_2 - x_1) = 0\\
&m_3 g + k_4(x_4 - x_3) - k_3(x_3 - x_2) = 0\\
&m_4 g + k_5(x_5 - x_4) - k_4(x_4 - x_3) = 0\\
&m_5 g - k_5(x_5 - x_4) = 0
\end{aligned}
$$

Collecting terms,

$$
\begin{bmatrix}
k_1 + k_2 & -k_2 & & & \\
-k_2 & k_2 + k_3 & -k_3 & & \\
 & -k_3 & k_3 + k_4 & -k_4 & \\
 & & -k_4 & k_4 + k_5 & -k_5 \\
 & & & -k_5 & k_5
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix}
=
\begin{Bmatrix} m_1 g \\ m_2 g \\ m_3 g \\ m_4 g \\ m_5 g \end{Bmatrix}
$$

After substituting the parameters, the equations can be expressed as ($g = 9.81$),

$$
\begin{bmatrix}
130 & -50 & 0 & 0 & 0 \\
-50 & 120 & -70 & 0 & 0 \\
0 & -70 & 170 & -100 & 0 \\
0 & 0 & -100 & 120 & -20 \\
0 & 0 & 0 & -20 & 20
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix}
=
\begin{Bmatrix} 539.55 \\ 735.75 \\ 588.60 \\ 735.75 \\ 882.90 \end{Bmatrix}
$$

The solution can then be obtained with the following MATLAB script:

```
clc; format short g
g=9.81;
m1=55;m2=75;m3=60;m4=75;m5=90;
k1=80;k2=50;k3=70;k4=100;k5=20;
A=[k1+k2 -k2 0 0 0
-k2 k2+k3 -k3 0 0
0 -k3 k3+k4 -k4 0
```

```
0 0 -k4 k4+k5 -k5
0 0 0 -k5 k5]
b=[m1*g m2*g m3*g m4*g m5*g]'
x=A\b

A =
   130   -50     0     0     0
   -50   120   -70     0     0
     0   -70   170  -100     0
     0     0  -100   120   -20
     0     0     0   -20    20
b =
        539.55
        735.75
         588.6
        735.75
         882.9
x =
        43.532
        102.39
        133.92
        150.11
        194.26
```

8.13 The position of the three masses can be modeled by the following steady−state force balances

$$
\begin{aligned}
0 &= k(x_2 - x_1) + m_1 g - kx_1 \\
0 &= k(x_3 - x_2) + m_2 g - k(x_2 - x_1) \\
0 &= m_3 g - k(x_3 - x_2)
\end{aligned}
$$

Terms can be combined to yield

$$
\begin{aligned}
2kx_1 - kx_2 \qquad\quad &= m_1 g \\
-kx_1 + 2kx_2 - kx_3 &= m_2 g \\
-kx_2 + kx_3 &= m_3 g
\end{aligned}
$$

Substituting the parameter values

$$
\begin{bmatrix} 20 & -10 & 0 \\ -10 & 20 & -10 \\ 0 & -10 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 19.62 \\ 29.43 \\ 24.525 \end{Bmatrix}
$$

A MATLAB script can be used to obtain the solution for the displacements

```
clc; format short g
g=9.81;k=10;
K=[2*k -k 0;-k 2*k -k;0 -k k]
```

```
m=[2;3;2.5];
mg=m*g
x=K\mg

K =
    20   -10     0
   -10    20   -10
     0   -10    10
mg =
         19.62
         29.43
        24.525
x =
        7.3575
        12.753
        15.205
```

8.14 Just as in Sec. 8.3, the simultaneous equations can be expressed in matrix form as

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & R_{52} & -R_{32} & 0 & -R_{54} & -R_{43} \\ R_{12} & -R_{52} & 0 & -R_{65} & 0 & 0 \end{bmatrix} \begin{Bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_1 - V_6 \end{Bmatrix}$$

or substituting the resistances

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 5 & -20 & 0 & -5 & -2 \\ 10 & -5 & 0 & -25 & 0 & 0 \end{bmatrix} \begin{Bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 150 \end{Bmatrix}$$

This system can be solved with MATLAB,

```
clc; format short g, format compact
format compact
R12=10;R52=5;R32=20;R65=25;R54=5;R43=2;
V1=150;V6=0;
A=[1 1 1 0 0 0;
0 -1 0 1 -1 0;
0 0 -1 0 0 1;
0 0 0 0 1 -1;
0 R52 -R32 0 -R54 -R43;
R12 -R52 0 -R65 0 0]
```

```
B=[0 0 0 0 0 V1-V6]'
I=A\B

A =
     1     1     1     0     0     0
     0    -1     0     1    -1     0
     0     0    -1     0     0     1
     0     0     0     0     1    -1
     0     5   -20     0    -5    -2
    10    -5     0   -25     0     0
B =
     0
     0
     0
     0
     0
   150
I =
        3.8247
       -3.2271
      -0.59761
       -3.8247
      -0.59761
      -0.59761
```

The results are $i_{21} = 3.8247$, $i_{52} = -3.2271$, $i_{32} = -0.59761$, $i_{65} = -3.8247$, $i_{54} = -0.59761$, and $i_{43} = -$ 0.59761.

8.15 The current equations can be written as

$$-i_{21} - i_{23} + i_{52} = 0$$
$$i_{23} - i_{35} + i_{43} = 0$$
$$-i_{43} + i_{54} = 0$$
$$i_{35} - i_{52} + i_{65} - i_{54} = 0$$

Voltage equations:

$$i_{21} = \frac{V_2 - 20}{35} \qquad i_{54} = \frac{V_5 - V_4}{15}$$
$$i_{23} = \frac{V_2 - V_3}{30} \qquad i_{35} = \frac{V_3 - V_5}{7}$$
$$i_{43} = \frac{V_4 - V_3}{8} \qquad i_{52} = \frac{V_5 - V_2}{10}$$
$$i_{65} = \frac{140 - V_5}{5}$$

$$\begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 35 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} i_{21} \\ i_{23} \\ i_{52} \\ i_{35} \\ i_{43} \\ i_{54} \\ i_{65} \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -20 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 140 \end{Bmatrix}$$

A MATLAB script can be developed to generate the solution,

```
clc; format short g
R12=35;R52=10;R32=30;R34=8;R45=15;R35=7;R25=10;R65=5;
V1=20;V6=140;
A=[-1   -1  1   0   0   0   0   0  0  0  0;
    0   1   0   -1  1   0   0   0  0  0  0;
    0   0   0   0   -1  1   0   0  0  0  0;
    0   0   -1  1   0   -1  1   0  0  0  0;
    R12 0   0   0   0   0   0   -1 0  0  0;
    0   R32 0   0   0   0   0   -1 1  0  0;
    0   0   0   0   R34 0   0   0  1  -1 0;
    0   0   0   0   0   R45 0   0  0  1  -1;
    0   0   0   R35 0   0   0   0  -1 0  1;
    0   0   R25 0   0   0   0   1  0  0  -1;
    0   0   0   0   0   0   R65 0  0  0  1];
B=[0 0 0 0 -V1 0 0 0 0 0 V6]';
I=A\B

I =
        2.5107
      -0.55342
        1.9573
      -0.42429
       0.12913
       0.12913
        2.5107
        107.87
        124.48
        125.51
        127.45
```

Thus, the solution is

$i_{21} = 2.5107$ $i_{23} = -0.55342$ $i_{52} = 1.9573$ $i_{35} = -0.42429$ $i_{43} = 0.12913$

$i_{54} = 0.12913$ $i_{65} = 2.5107$ $V_2 = 107.87$ $V_3 = 124.48$ $V_4 = 125.51$
$V_5 = 127.45$

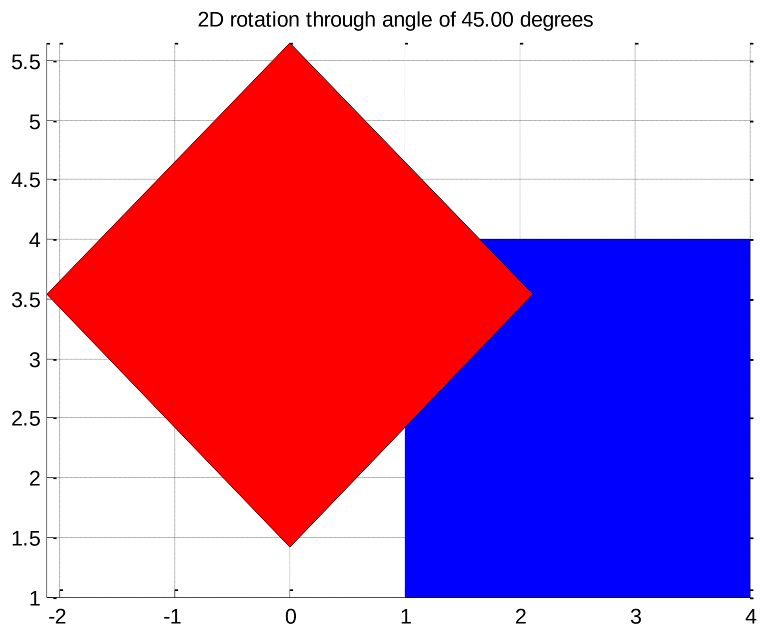
8.16

```
clc;clf;format compact
x = [1 4 4 1]; y = [1 1 4 4];
[xt, yt] = Rotate2D(45, x, y);

function [xr, yr] = Rotate2D(thetad, x, y)
% two dimensional rotation 2D rotate Cartesian
% [xr, yr] = rot2d(thetad, x, y)
% Rotation of a two-dimensional object the Cartesian coordinates
% of which are contained in the vectors x and y.
% input:
% thetad = angle of rotation (degrees)
% x = vector containing objects x coordinates
% y = vector containing objects y coordinates
% output:
% xr = vector containing objects rotated x coordinates
% yr = vector containing objects rotated y coordinates

% set up rotation matrix
thetar = thetad*pi/180;
r = [cos(thetar) -sin(thetar);sin(thetar) cos(thetar)];
% close shape
x = [x x(1)]; y = [y y(1)];
% plot original object
hold on, grid on
fill(x, y,'b')
% rotate shape
z = r*[x;y];
% plot rotated object
xr = z(1,:); yr = z(2,:);
fill(xr, yr,'r');
title(sprintf('2D rotation through angle of %3.2f
degrees',thetad))
xmin=min([x xr]); ymin=min([y yr]); xmax=max([x xr]);
ymax=max([y yr])
axis([xmin xmax ymin ymax])
hold off

ymax =
    5.6569
```


2D rotation through angle of 45.00 degrees
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
-2
-1
0
1
2
3
4