6.3.1 Define these matrices and column vectors used to define a truss, the loading on it, the bar tensions , the reactions, and the coefficients in the matrix form of the joint equilibrium equations:

a) $[J]$

b) $[B]$

c) $[R]$

d) $[F]$

e) $[T]$

f) $[L]$

g) $[A]$

a) [J] - Contains Joint Information

Columns : Joint Number, X-Coordinate, Y-Coordinate

b) [B] - Contains Bar Information

Columns : Bar Number, Joint-1, Joint-2

c) [R] - Contains Reaction Forces Information

Columns : Reaction Force Number, Joint, X-Component, Y-Component

d) [F] - Contains External Force Information

Columns : Joint, X-Component, Y-Component

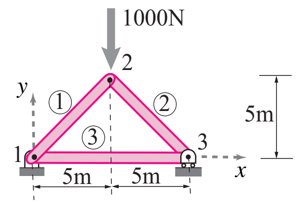
e) [T] - Contains Value of Tension on each Bar

Column : Tension

f) [L] - Contains List of negative of loads applied in x and y directions.

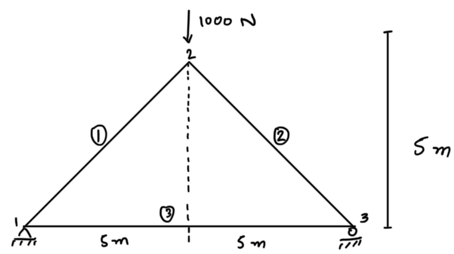
g) [A] - Contains cosines and sines of the bar angles.

6.3.2 By hand, with no use of a computer, find all of the matrices and column vectors above for this truss.



Problem 6.2

Sketch:



$$[J] = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 5 & 5 \\ 3 & 10 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 1 & 3 \end{bmatrix}$$

$$[R] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 3 & 0 & 1 \end{bmatrix} \qquad [F] = \begin{bmatrix} 2 & 0 & -1000 \end{bmatrix}$$

$$[A] = \begin{bmatrix} 1/\sqrt{2} & 0 & 1 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & 0 & 0 & 1 & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & -1/\sqrt{2} & -1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 0 & 0 & 1 \end{bmatrix} \qquad [L] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1000 \\ 0 \\ 0 \end{bmatrix}$$

$$[T] = [A]^{-1} [L]$$

$$[T] = \begin{bmatrix} 1000/\sqrt{2} \\ 1000/\sqrt{2} \\ -500 \\ 0 \\ -500 \\ -500 \end{bmatrix}$$

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

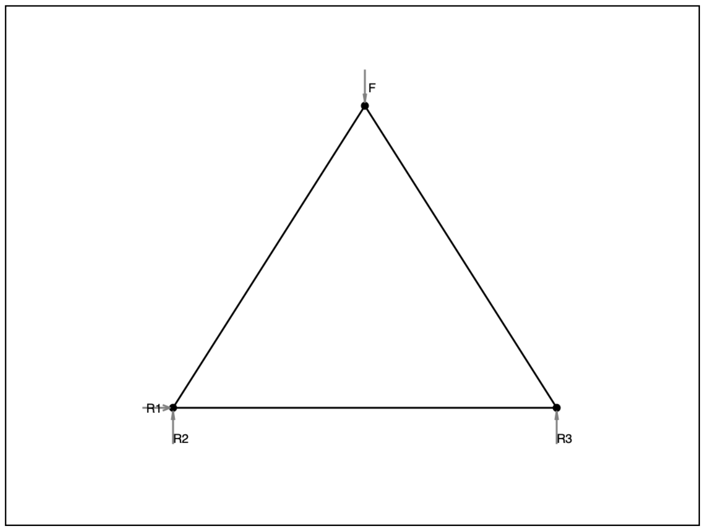
6.3.3 When does the numerical recipe presented here succeed and when does it fail? When it fails, how does it fail?

The numerical method of analysing trusses succeeds when [A] is a square matrix, ie a determinate truss. The method fails when the truss is non-determinate, ie [A] is a non-square matrix. In the case of a non-determinate truss, an error message is displayed by the program.

6.3.4

a) Write a computer program, using your preferred language or package, that takes as input the matrices $[J]$, $[B]$, $[R]$, and $[F]$ and calculates $[T]$.

b) Test this program on the truss of problem 6.3.2.



Code:

a)

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces
%
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
```

```
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-
J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
```

```
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

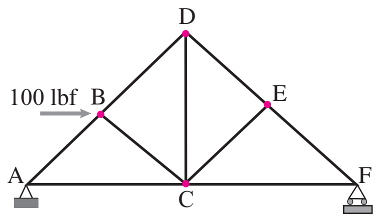
b)

```
J = [1 0 0;
     2 5 5;
     3 10 0;];

B = [1 1 2;
     2 2 3;
     3 1 3;];

R = [1 1 1 0;
     2 1 0 1;
     3 3 0 1;];

F = [2 0 -1000];

T = TrussAnalyser(B,J,R,F)
```

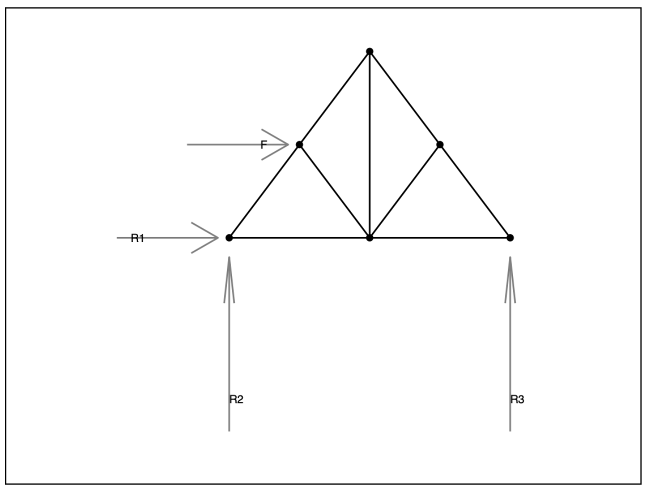
Output:

$$T = \begin{bmatrix} 707.1068 \\ 707.1068 \\ -500 \\ 0 \\ -500 \\ -500 \end{bmatrix}$$

6.3.5 All of the bars in the symmetric truss below are either level or at 30° from the horizontal. Find all the bar forces and reactions.



Problem 6.5



Code:

```
J = [1 0 0;
     2 1/2 1/(2*sqrt(3));
     3 1 0;
     4 1 1/sqrt(3);
     5 1.5 1/(2*sqrt(3));
     6 2 0];

B = [1 1 2;
     2 2 4;
     3 4 5;
     4 5 6;
     5 6 3;
     6 1 3;
     7 2 3;
     8 3 4;
     9 3 5];

R = [1 1 1 0;
     2 1 0 1;
     3 6 0 1;];

```

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

```
F = [2 100 0];

TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces
%
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
```

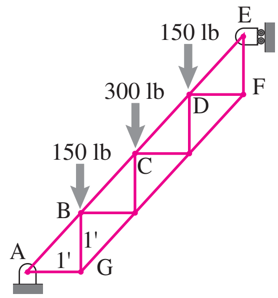
```
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

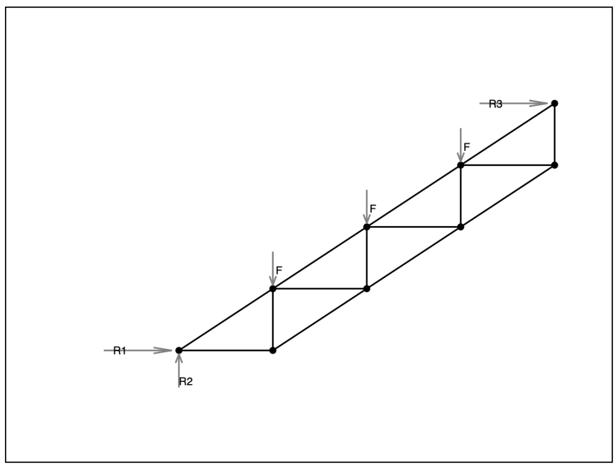
Output:

$$T = \begin{bmatrix} -28.8675 \\ 28.8675 \\ 28.8675 \\ 28.8675 \\ 25 \\ -75 \\ 57.7350 \\ 28.8675 \\ 0 \\ 100 \\ 14.4338 \\ -14.4338 \end{bmatrix}$$

6.3.6 Find the force in each bar of the staircase truss shown in the figure by writing the required number of equilibrium equations and then solving them on a computer.



Problem 6.6



Number of equilibrium Equations : 18

Code :

```
J = [1 0 0;
     2 1 1;
     3 2 2;
     4 3 3;
     5 4 4;
     6 4 3;
     7 3 2;
     8 2 1;
     9 1 0];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 4 5;
     5 5 6;
     6 6 7;
     7 7 8;
     8 8 9;
```

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

```
     9 1 9;
     10 2 9;
     11 2 8;
     12 3 8;
     13 3 7;
     14 7 4;
     15 4 6;];

R = [1 1 1 0;
     2 1 0 1;
     3 5 1 0;];

F = [2 0 -150;
     3 0 -300;
     4 0 -150;];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces
%
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
```

```
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
```

```
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

Output:

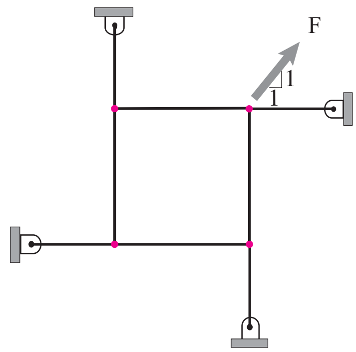
$$T = 10^3 \times \begin{bmatrix} 0.8485 \\ 1.0607 \\ 0.8485 \\ 0.4243 \\ 0.3 \\ 0.4243 \\ 0.6364 \\ 0.4243 \\ -0.3 \\ -0.3 \\ -0.15 \\ -0.15 \\ 0.15 \\ 0.15 \\ 0.3 \\ -0.3 \\ -0.6 \\ 0.3 \end{bmatrix}$$

6.3.7 Find the tensions in all the bars, and all the reactions for these structures.

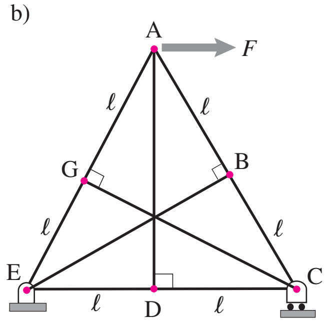
a) A square supported by four bars. This is perhaps the simplest rigid structure that has no triangles.

b) The 9-bar structure shown. This structure also has no triangles in that there is no closed circuit that involves only three bars (for example, from D to A to B to C and back to D involves 4 bars).

a)



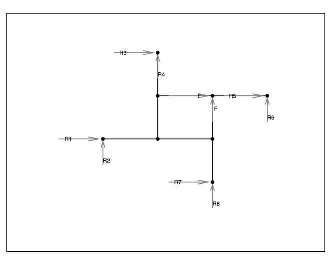
b)



no pin at the center, total 9 bars

Problem 6.7

a)



```
J = [1 0 0;
     2 1 0;
     3 1 1;
     4 1 2;
     5 2 1;
     6 3 1;
     7 2 0;
     8 2 -1];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 3 5;
     5 5 6;
     6 5 7;
     7 7 8;
     8 7 2;];

R = [1 1 1 0;
     2 1 0 1;
     3 4 1 0;
     4 4 0 1;
     5 6 1 0;
     6 6 0 1;
```

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

```
     7 8 1 0;
     8 8 0 1;];

F = [5 1 1];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces %
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

```

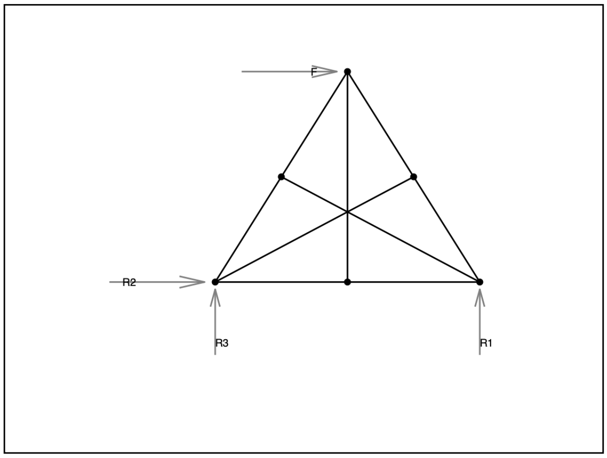
```
    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

Output:

$$T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

b)



Code:

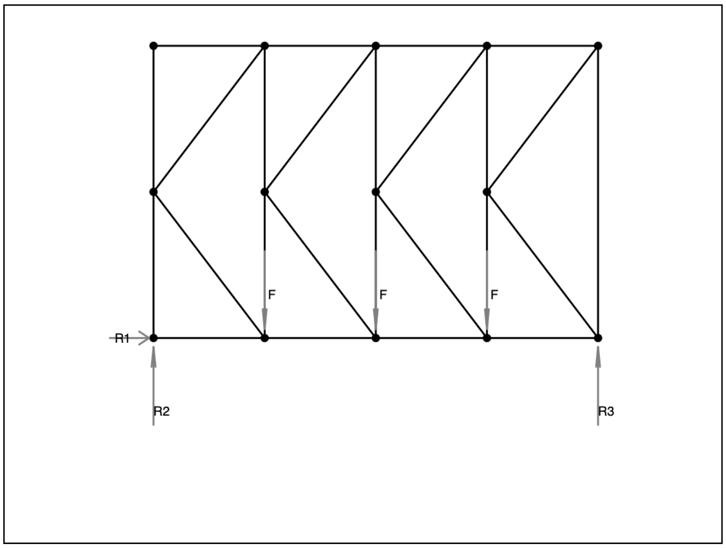
```
l = 1;
F = 1;

J = [1 2*l*cosd(60) 2*l*sind(60);
     2 sqrt(5)*l*cosd(30) sqrt(5)*l*sind(30);
     3 2*l 0;
     4 l 0;
     5 0 0;
     6 l*cosd(60) l*sind(60);];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 4 5;
     5 5 6;
     6 1 6;
     7 1 4;
     8 3 6;
     9 2 5];

R = [1 3 0 1;
     2 5 1 0;
     3 5 0 1;];

```

```
F = [1 F 0];

T = TrussAnalyser(B,J,R,F)
```

Output:

$$T = \begin{bmatrix} 1 \\ 1 \\ 0.5 \\ 0.5 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ -0.866 \\ 1 \\ 0.866 \end{bmatrix}$$

6.3.8 Analyze the truss given in Problem **??** and solve for all bar tensions and support reactions.



Code:

```
J = [1 0 0;
     2 0 1;
     3 0 2;
     4 2 2;
     5 2 1;
     6 2 0;
     7 4 2;
     8 4 1;
     9 4 0;
     10 6 2;
     11 6 1;
     12 6 0;
     13 8 2;
     14 8 0];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 4 7;
     5 7 10;
     6 10 13;
```

```
     7 13 14;
     8 12 14;
     9 9 12;
     10 6 9;
     11 1 6;
     12 2 6;
     13 2 4;
     14 4 5;
     15 5 6;
     16 5 9;
     17 5 7;
     18 7 8;
     19 8 9;
     20 8 12;
     21 8 10;
     22 10 11;
     23 11 12;
     24 11 14;
     25 11 13;];

R = [1 1 1 0;
     2 1 0 1;
     3 14 0 1;];

F = [6 0 -2000;
     9 0 -2000;
     12 0 -2000;];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
```

```
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces %
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
```

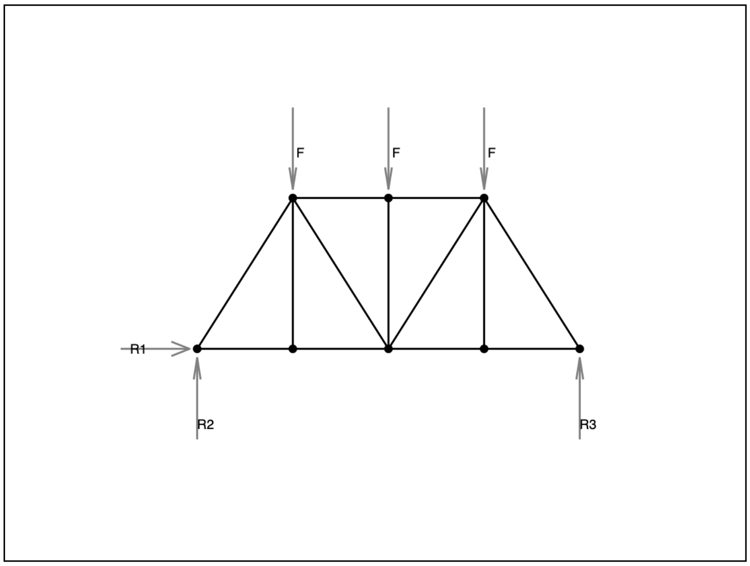
```
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

Output:

$$T = 10^3 \times \begin{bmatrix} -3 \\ 0 \\ 0 \\ 3 \\ 4 \\ 3 \\ -1.5 \\ -3 \\ -4 \\ -3 \\ 0 \\ -3.3541 \\ 3.3541 \\ 1.5 \\ 0.5 \\ -1.118 \\ 1.118 \\ 0.5 \\ 1.5 \\ 1.118 \\ -1.118 \\ -0.5 \\ 2.5 \\ 3.3541 \\ -3.3541 \\ 0 \\ -3 \\ -3 \end{bmatrix}$$

6.3.9 Solve Problem **??**.



Code:

```
J = [1 0 0;
     2 1 1;
     3 1 0;
     4 2 1;
     5 2 0;
     6 3 1;
     7 3 0;
     8 4 0;];

B = [1 1 2;
     2 2 4;
     3 4 6;
     4 6 8;
     5 7 8;
     6 5 7;
     7 3 5;
     8 1 3;
     9 2 3;
     10 2 5;
     11 4 5;
     12 5 6;
```

```
     13 6 7;];

R = [1 1 1 0;
     2 1 0 1;
     3 8 0 1;];

F = [2 0 -2000;
     4 0 -5000;
     6 0 -2000;];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces %
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
```

```
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

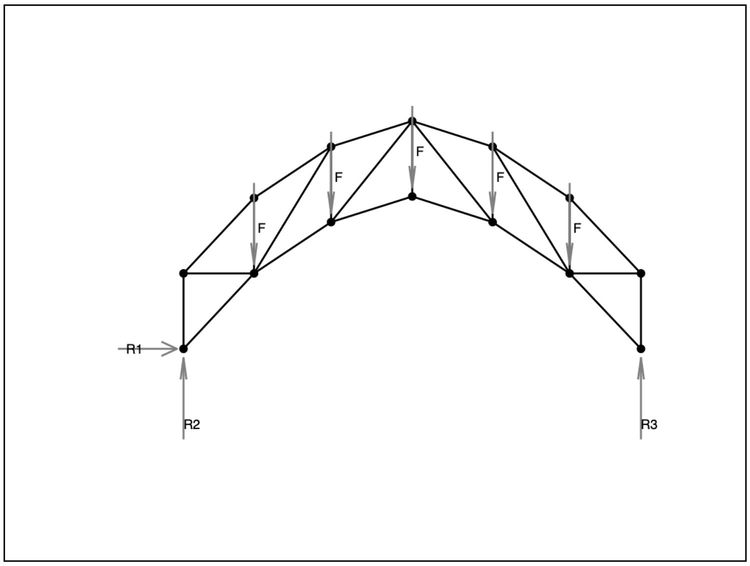
```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
```

```
                a = 0;
            end
        end
end
```

Output:

$$T = 10^3 \times \begin{bmatrix} 6.364 \\ 7 \\ 7 \\ 6.364 \\ -4.5 \\ -4.5 \\ -4.5 \\ -4.5 \\ 0 \\ -3.5355 \\ -5 \\ -3.5355 \\ 0 \\ 0 \\ -4.5 \\ -4.5 \end{bmatrix}$$

6.3.10 Solve Problem **??**.



Code:

```
J = [1 0 0;
     2 0 0.5;
     3 0.86 1;
     4 1.8 1.34;
     5 2.79 1.51;
     6 3.77 1.34;
     7 4.71 1;
     8 5.58 0.5;
     9 5.58 0;
     10 4.71 0.5;
     11 3.77 0.84;
     12 2.79 1.01;
     13 1.8 0.84;
     14 0.86 0.5];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 4 5;
     5 5 6;
     6 6 7;
```

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

```
     7 7 8;
     8 8 9;
     9 9 10;
     10 10 11;
     11 11 12;
     12 12 13;
     13 13 14;
     14 1 14;
     15 2 14;
     16 3 14;
     17 4 13;
     18 5 12;
     19 6 11;
     20 7 10;
     21 8 10;
     22 4 14;
     23 5 13;
     24 5 11;
     25 6 10;];

R = [1 1 1 0;
     2 1 0 1;
     3 9 0 1;];

F = [10 0 -20*1000;
     11 0 -20*1000;
     12 0 -20*1000;
     13 0 -20*1000;
     14 0 -20*1000;];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
```

```
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces
%
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-
J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
```

```
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

Output:

$$T = 1.0e+05 \times \begin{bmatrix} -0.5007 \\ 0.9962 \\ 0.9158 \\ 1.4475 \\ 1.4528 \\ 0.9238 \\ 1.0020 \\ -0.4993 \\ 0 \\ 1.5222 \\ 1.6503 \\ 1.6498 \\ 1.5170 \\ 0 \\ -0.8612 \\ 0.1892 \\ 0.5717 \\ 0.7613 \\ 0.5687 \\ 0.1851 \\ 0.8688 \\ -0.7582 \\ -0.2408 \\ 0.2357 \\ 0.7546 \\ 0 \\ -0.5007 \\ -0.4993 \end{bmatrix}$$

6.3.11 Using your program from problem 6.3.4 solve each of the following problems:

a) Problem **??**

b) Problem **??**

c) Problem **??**

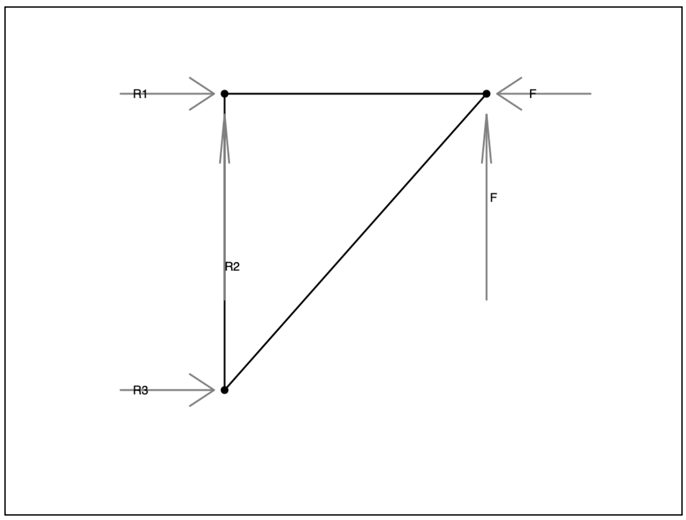
d) Problem **??**

e) Problem **??**

f) Problem **??**

Code:

a)




```
J = [1 0 0.86;
     2 2 0.86;
     3 0 0;];

B = [1 1 2;
     2 2 3;
     3 1 3;];

R = [1 1 1 0;
     2 1 0 1;
     3 3 1 0;];

F = [2 173*cosd(120) 173*sind(120)];

T = TrussAnalyser(B,J,R,F)
```

```
function T = TrussAnalyser(B,J,R,F)

    % Receiving Matrix A from mat() %
    A = TrussMatrix(B,J,R);
```

```
    [sr,g] = size(A);

    % Finding Matrix Containing External Forces %
    L = zeros(sr,1);
    [sf, g] = size(F);
    for i = 1:sf
        L((F(i)*2)-1,1) = F(i,2);
        L((F(i)*2),1) = F(i,3);
    end

    % Finding Tensions on Each Rod Based on External Forces %
    T = A\L;

    % Checking if values of Tensions are Valid %
    if MatrixCheck(A,T,L) == 1
        disp("Values are Valid")
    else
        disp("Values are not Valid")
    end
end
```

```
function A = TrussMatrix(B,J,R)
    % Finding Number of Rows in Matrix A%
    [num_reactions, ~] = size(R);
    [sr,g] = size(B);
    A = zeros(sr+num_reactions,sr+num_reactions);

    % Finding Matrix A Based on Joint and Rod Data%
    for i = 1:sr
        % Finding Joints connecting to ith Rod %
        jnt1 = B(i,2);
        jnt2 = B(i,3);

        % Finding Angle between ith Rod and Horizontal
        theta = atan((J(jnt1,3)-J(jnt2,3))/(J(jnt1,2)-J(jnt2,2)));
        % Adding values of cos and sin theta into A Matrix %
        A((jnt1*2)-1,i) = cos(theta);
        A(jnt1*2,i) = sin(theta);
```

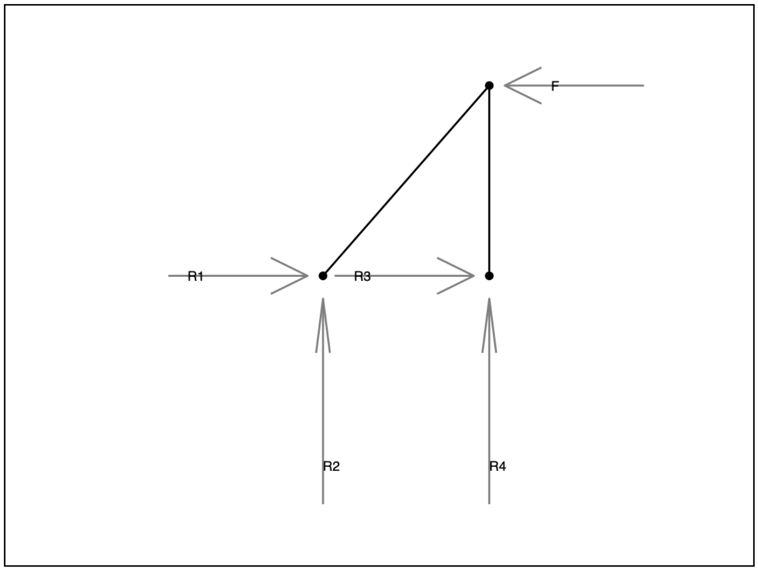
```
        A((jnt2*2)-1,i) = -cos(theta);
        A(jnt2*2,i) = -sin(theta);
    end
    for i = 1:num_reactions
        if R(i,3) == 1
            A((R(i,2)*2)-1,size(A)-num_reactions+i) = 1;
        elseif R(i,4) == 1
            A((R(i,2)*2),size(A)-num_reactions+i) = 1;
        end
    end
end
```

```
function a = MatrixCheck(A,T,L)
    [sr, g] = size(A);
    a = 1;
    tol = 0.0001;     % Setting Tolerance Value
    for i = 1:sr
        tmp = A(i,:)*T;
        disc = abs(tmp-L(i));

        % If the Difference is Greater than the Tolerance %
        if (disc >= tol)
            disp(strcat("Not valid at Joint:",num2str(i)))
            a = 0;
        end
    end
end
```

Output:

$$T = \begin{bmatrix} 434.9242 \\ 379.2706 \\ -149.8224 \\ -434.9242 \\ 149.8224 \\ 348.4242 \end{bmatrix}$$

b)

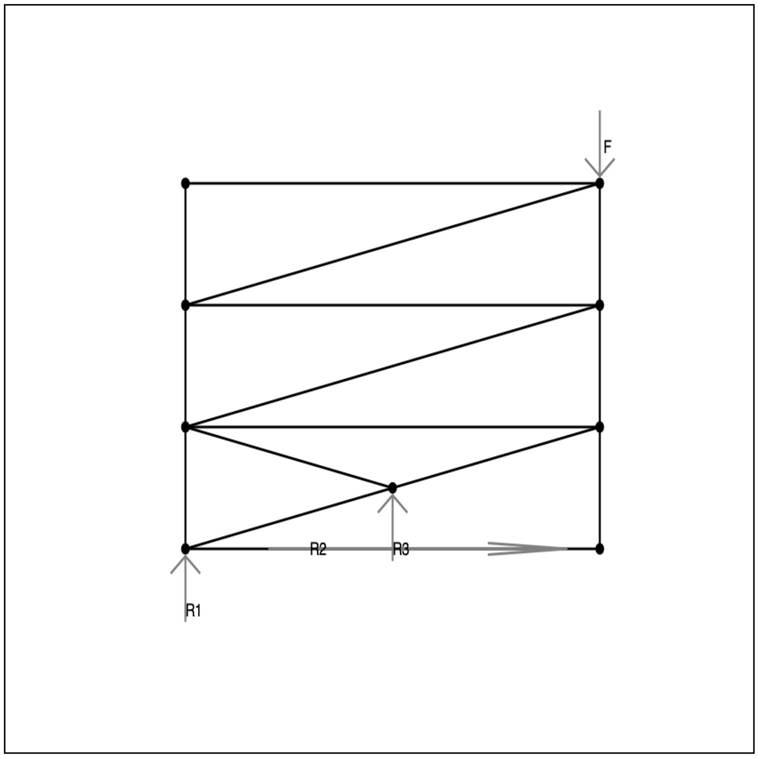



```
J = [1 -0.86 0;
     2 0 0.5;
     3 0 0;];

B = [1 1 2;
     2 2 3;];

R = [1 1 1 0;
     2 1 0 1;
     3 3 1 0;
     4 3 0 1;];

F = [2 -300 0];

T = TrussAnalyser(B,J,R,F)
```

Output :

$$T = \begin{bmatrix} 347.0185 \\ 174.4186 \\ -300 \\ -174.4186 \\ 0 \\ 174.4186 \end{bmatrix}$$

c)



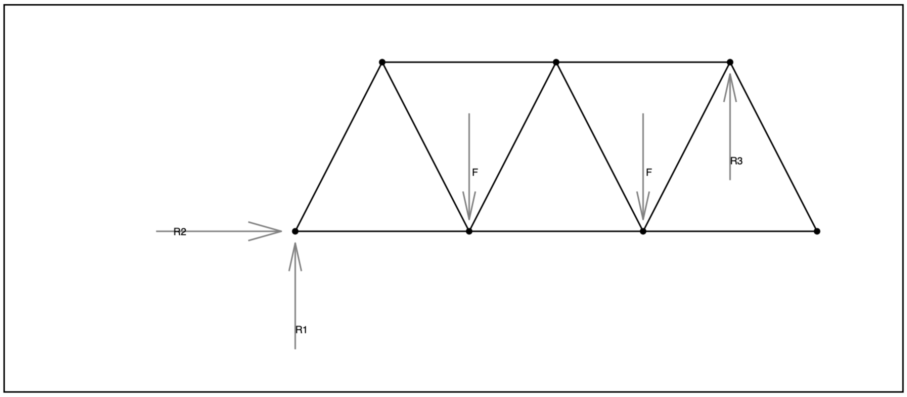

```
J = [1 0 3;
     2 1 3;
     3 0 2;
     4 1 2;
     5 0 1;
     6 1 1;
     7 0 0;
     8 0.5 0.5;
     9 1 0;];

B = [1 1 2;
     2 2 4;
     3 4 6;
     4 6 9;
     5 7 9;
     6 5 7;
```

```
     7 3 5;
     8 1 3;
     9 2 3;
     10 3 4;
     11 4 5;
     12 5 6;
     13 5 8;
     14 6 8;
     15 7 8;];

R = [1 7 0 1;
     2 9 1 0;
     3 8 0 1;];

F = [2 0 -2];

T = TrussAnalyser(B,J,R,F)
```

Output:

$$T = \begin{bmatrix} 0 \\ -2 \\ -2 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \\ 2.8284 \\ -2.8284 \\ 0 \\ 2 \\ 0 \\ -4 \end{bmatrix}$$

d)




```
J = [1 0 0;
     2 0.5 0.86;
     3 1 0;
     4 1.5 0.86;
     5 2 0;
     6 2.5 0.86;
     7 3 0;];

B = [1 1 2;
     2 2 4;
     3 4 6;
     4 6 7;
     5 5 7;
     6 3 5;
     7 1 3;
     8 2 3;
     9 3 4;
     10 4 5;
     11 5 6;];

R = [1 1 0 1;
     2 1 1 0;
     3 6 0 1;];

F = [3 0 -4000;
     5 0 -1000;];
```

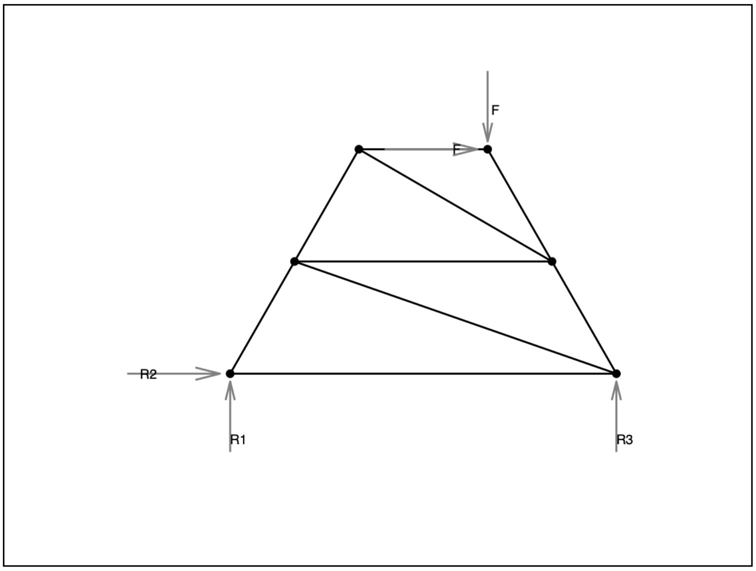
Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

```

T = TrussAnalyser(B,J,R,F)
```

Output:

$$T = 10^3 \times \begin{bmatrix} 3.0075 \\ 3.0233 \\ 1.3953 \\ 0 \\ 0 \\ -2.2093 \\ -1.5116 \\ -3.0075 \\ -1.6194 \\ 1.6194 \\ -2.7761 \\ -2.6 \\ 0 \\ -2.4 \end{bmatrix}$$

e)



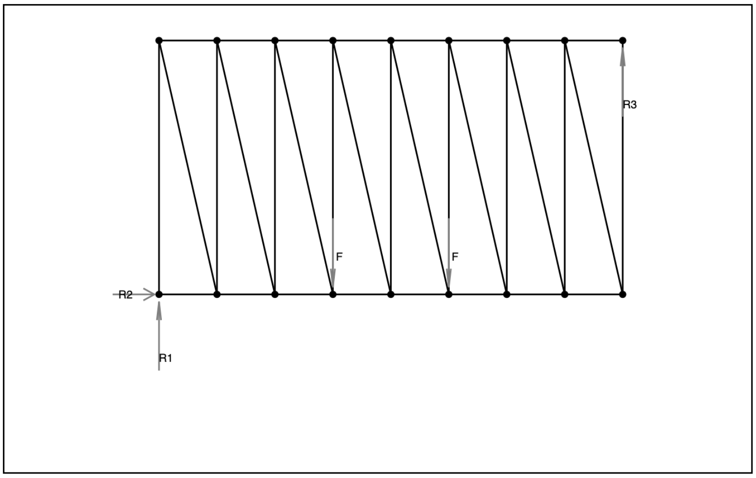

```
l = sym('l');

J = [1 0 0;
     2 l/2 0.86*l;
     3 l 2*l*0.86;
     4 2*l 2*l*0.86;
     5 (5/2)*l 0.86*l;
     6 3*l 0;];

B = [1 1 2;
     2 2 3;
     3 3 4;
     4 4 5;
     5 5 6;
     6 1 6;
     7 2 6;
     8 2 5;
     9 3 5;];

R = [1 1 0 1;
     2 1 1 0;
     3 6 0 1;];
```

```

F = [4 3000*cosd(45) -3000*sind(45)];

T = TrussAnalyser(B,J,R,F)
```

Output:

$$T = 10^3 \times \begin{bmatrix} -0.5889 \\ -0.4417 \\ -0.8880 \\ 2.4538 \\ 2.8955 \\ -1.8253 \\ 0.3913 \\ -0.4440 \\ 0.7677 \\ 0.5091 \\ 2.1213 \\ -2.6304 \end{bmatrix}$$

f)




```
J = [1 0 0;
     2 0 2;
     3 1 0;
     4 1 2;
     5 2 0;
     6 2 2;
     7 3 0;
     8 3 2;
     9 4 0;
     10 4 2;
     11 5 0;
     12 5 2;
     13 6 0;
     14 6 2;
     15 7 0;
     16 7 2;
     17 8 0;
     18 8 2;];

B = [1 1 2;
     2 2 4;
     3 4 6;
     4 6 8;
     5 8 10;
     6 10 12;
```

```
     7 12 14;
     8 14 16;
     9 16 18;
     10 17 18;
     11 15 17;
     12 13 15;
     13 11 13;
     14 9 11;
     15 7 9;
     16 5 7;
     17 3 5;
     18 1 3;
     19 2 3;
     20 3 4;
     21 4 5;
     22 5 6;
     23 6 7;
     24 7 8;
     25 8 9;
     26 9 10;
     27 10 11;
     28 11 12;
     29 12 13;
     30 13 14;
     31 14 15;
     32 15 16;
     33 16 17;];

R = [1 1 0 1;
     2 1 1 0;
     3 18 0 1;];

F = [7 0 -10*1000;
     11 0 -10*1000;];

T = TrussAnalyser(B,J,R,F)
```

Disclaimer: *We have lost track of the many people who wrote these solutions. They are not official and have not been reviewed by the authors for correctness.*

Output:

$$T = 10^4 \times \begin{bmatrix} -1 \\ 0.5 \\ 1 \\ 1.5 \\ 1.5 \\ 1.5 \\ 1 \\ 0.5 \\ 0 \\ 1 \\ -0.5 \\ -1 \\ -1.5 \\ -1.5 \\ -1.5 \\ -1 \\ -0.5 \\ 0 \\ -1.118 \\ -1 \\ -1.118 \\ -1 \\ -1.118 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1.118 \\ 1 \\ 1.118 \\ 1 \\ 1.118 \\ -1 \\ 0 \\ -1 \end{bmatrix}$$