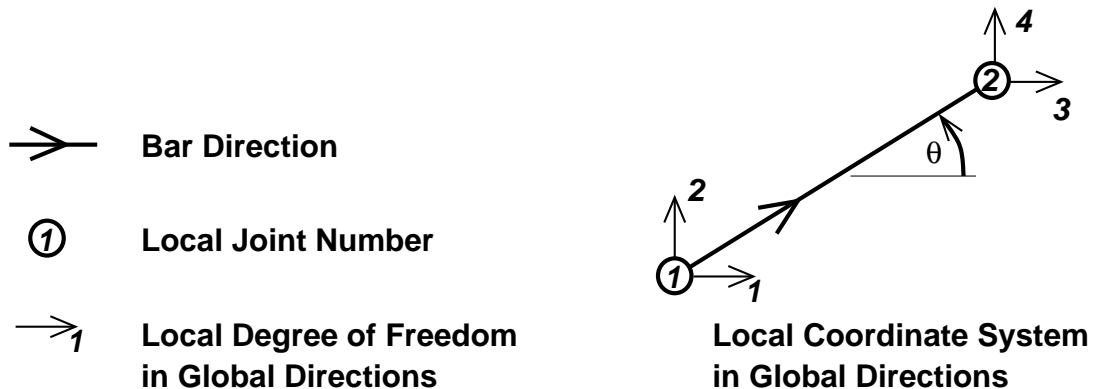


# THE DIRECT STIFFNESS METHOD FOR PLANAR TRUSSES

## CE 131 — Matrix Structural Analysis

**Henri Gavin**  
**Fall, 2006**

1. Number all of the joints and all of the elements.
2. Identify the Degrees of Freedom. Number all the global degrees of freedom in your truss. In a planar truss, each joint can have a maximum of two degrees of freedom: one in the global  $X$ -direction and one in the global  $Y$ -direction. If a degree of freedom is restrained by a reaction, then it doesn't get a number.
3. Joint Locations. Write the  $(x, y)$  coordinates of each joint using units consistent with  $E$  and  $A$ . In other words, if  $E$  and  $A$  are given in  $\text{kN}/\text{cm}^2$  and  $\text{cm}^2$ , write the  $(x, y)$  coordinates in terms of centimeters.
4. Define each element. Draw each element of your truss individually and draw the local coordinates in the global directions. For example if element number  $N$  is a diagonal truss element, and the global directions are  $X$ : horizontal and  $Y$ : vertical, draw element number  $N$  like this:



where 1,2,3,4 are the LOCAL coordinates of the truss element in the GLOBAL directions. The local coordinates are always numbered 1,2,3,4 with 1 and 3 pointing in the global  $X$  direction (to the right) and with 2 and 4 pointing in the global  $Y$  direction (up). Some or all of these four coordinates will line up with the global degrees of freedom that you identified in step 2., above.

5. Element Stiffness Matrices in Global Coordinates,  $\mathbf{K}$ .

For each element, find it's (4x4) element stiffness matrix, by evaluating the equations below:

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$c = (x_2 - x_1)/L$$

$$s = (y_2 - y_1)/L$$

$$\mathbf{K} = \frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}.$$

or by using the TRUSS script in matlab. You should understand where these equations come from, why this matrix is symmetric, why the diagonal terms are all positive, and what the off-diagonal terms mean.

6. Structural Stiffness Matrix,  $\mathbf{K}_s$ .

The structural stiffness matrix is a square, symmetric matrix with dimension equal to the number of degrees of freedom. In this step we will fill up the structural stiffness matrix using terms from the element stiffness matrices in global coordinates (from step 5.) This procedure is called matrix assembly.

Recall from step 4. how the LOCAL element degrees of freedom (1,2,3,4) line up with the GLOBAL degrees of freedom in your problem. For example, coordinates (1,2,3,4) might line up with degrees of freedom (3,4,7,8) of the truss. In this case, to assemble element number  $N$  into the structural stiffness matrix,

$$\begin{array}{cccc|cccc|cccc|cccc} K(1,1) & +> & K_s(3,3) & & K(1,2) & +> & K_s(3,4) & & K(1,3) & +> & K_s(3,7) & & K(1,4) & +> & K_s(3,8) & \\ \hline K(2,1) & +> & K_s(4,3) & & K(2,2) & +> & K_s(4,4) & & K(2,3) & +> & K_s(4,7) & & K(2,4) & +> & K_s(4,8) & \\ \hline K(3,1) & +> & K_s(7,3) & & K(3,2) & +> & K_s(7,4) & & K(3,3) & +> & K_s(7,7) & & K(3,4) & +> & K_s(7,8) & \\ \hline K(4,1) & +> & K_s(8,3) & & K(4,2) & +> & K_s(8,4) & & K(4,3) & +> & K_s(8,7) & & K(4,4) & +> & K_s(8,8) & \end{array}$$

where the +> is short-hand for “is added to” ....  $K(1,3)$  is added to  $K_s(3,7)$ .

Add each element into the structural stiffness matrix in this way to get  $\mathbf{K}_s$ . Note that if one end of the truss element is fully restrained in both the the X- and Y- directions, you will need to place only four of the sixteen terms of the element's 4x4 stiffness matrix.

7. Forces,  $\mathbf{p}$ .

Create the force vector  $\mathbf{p}$ , by finding the components of each applied force in the directions of the global degrees of freedom. Create the force vector by placing these force components into the force vector at the proper coordinates.

8. Deflections,  $\mathbf{d}$ .

Find the deflections by inverting the stiffness matrix and multiplying it by the load vector. You can do this easily in matlab:  $\mathbf{d} = \mathbf{K}_s \setminus \mathbf{p}$

9. Internal bar forces,  $n$ .

Again, recall how the global degrees of freedom line up with each element's coordinates (1,2,3,4). For example, in element number  $N$  from step 6., the local element deflections in the global directions,  $v_1, v_2, v_3, v_4$  line up with the structural deflections  $d_3, d_4, d_7, d_8$ . The internal bar forces can be computed from:

$$n = \frac{EA}{L} [c(v_3 - v_1) + s(v_4 - v_2)]$$

where  $c$  and  $s$  are the direction cosine and sine for the element from step 5.

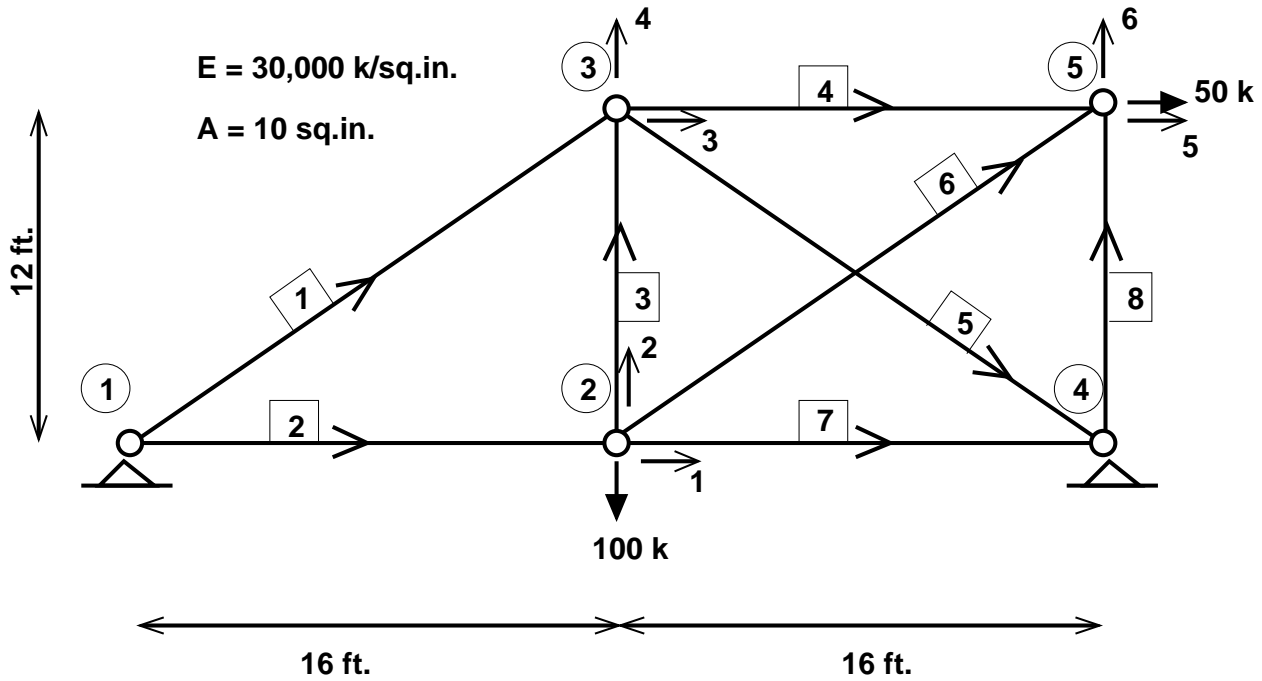
You should be able to derive this equation.

Knowing what each bar force is, the reactions can be easily computed with equilibrium equations.

**Notation**

- $\mathbf{u}$  = Element deflection vector in the Local coordinate system
- $\mathbf{q}$  = Element force vector in the Local coordinate system
- $\mathbf{k}$  = Element stiffness matrix in the Local coordinate system  
...  $\mathbf{q} = \mathbf{k} \mathbf{u}$
- $\mathbf{T}$  = Coordinate Transformation Matrix  
...  $\mathbf{T}^{-1} = \mathbf{T}^T$
- $\mathbf{v}$  = Element deflection vector in the Global coordinate system  
...  $\mathbf{u} = \mathbf{T} \mathbf{v}$
- $\mathbf{f}$  = Element force vector in the Global coordinate system  
...  $\mathbf{q} = \mathbf{T} \mathbf{f}$
- $\mathbf{K}$  = Element stiffness matrix in the Global coordinate system  
...  $\mathbf{K} = \mathbf{T}^T \mathbf{k} \mathbf{T}$
- $\mathbf{d}$  = Structural deflection vector in the Global coordinate system
- $\mathbf{p}$  = Structural load vector in the Global coordinate system
- $\mathbf{K}_s$  = Structural stiffness matrix in the Global coordinate system  
...  $\mathbf{p} = \mathbf{K}_s \mathbf{d}$

	Local	Global
Element Deflection	$\mathbf{u}$	$\mathbf{v}$
Element Force	$\mathbf{q}$	$\mathbf{f}$
Element Stiffness	$\mathbf{k}$	$\mathbf{K}$
Structural Deflection	-	$\mathbf{d}$
Structural Loads	-	$\mathbf{p}$
Structural Stiffness	-	$\mathbf{K}_s$



① Global Joint Number

□ 2 Bar Number

➤ Bar Direction

➤<sub>1</sub> Global Degree of Freedom

➤ Applied Force

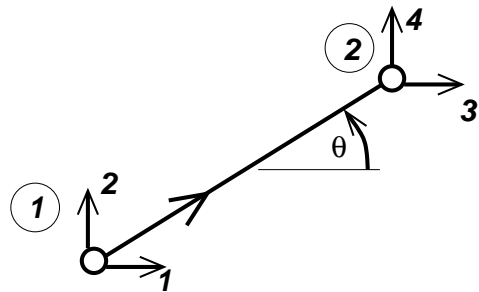
① Local Joint Number

➤<sub>1</sub> Local Degree of Freedom

System Equation:  $\{f\} = [K]\{d\}$

$\{d\}$  : displacement vector  
 $= \{ d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \}$

$\{p\}$  : load vector =  $\{ 0 \ -100 \ 0 \ 0 \ 50 \ 0 \}$



Local Coordinate System

```

function K = truss( x1,y1,x2,y2,E,A )
% K = TRUSS (X1,Y1,X2,Y2,E,A)
% Returns a 2-D truss element stiffness matrix in global coordinates
%
%       X1,Y1 are the coordiniates of joint 1 of the truss element
%       X2,Y2 are the coordiniates of joint 2 of the truss element
%       E       is the elastic modulus
%       A       is the cross sectional area

L = sqrt( (x2-x1)^2 + (y2-y1)^2 );
c = (x2-x1) / L;
s = (y2-y1) / L;

K = (E*A/L) * [ c^2    c*s   -c^2   -c*s ;
               c*s    s^2   -c*s   -s^2 ;
               -c^2   -c*s    c^2    c*s ;
               -c*s   -s^2    c*s    s^2 ];

% ----- TRUSS

hudson17% matlab

>> help truss

K = TRUSS (X1,Y1,X2,Y2,E,A)
Returns a 2-D truss element stiffness matrix in global coordinates

       X1,Y1 are the coordiniates of joint 1 of the truss element
       X2,Y2 are the coordiniates of joint 2 of the truss element
       E       is the elastic modulus
       A       is the crosssectional area

>> format bank                               % two decimal places after the .
>> E = 3e4;                                   % modulus of elasticity
>> A = 10;                                     % area of cross section

>> K1 = truss(0,0,12*16,12*12,E,A)           % member 1
K1 =

      800.00      600.00     -800.00     -600.00
      600.00      450.00     -600.00     -450.00
     -800.00     -600.00      800.00      600.00
     -600.00     -450.00      600.00      450.00

```

```
>> K2 = truss(0,0,12*16,0,E,A) % member 2
```

```
K2 =
```

```
1562.50    0.00 -1562.50    0.00
   0.00    0.00    0.00    0.00
-1562.50    0.00  1562.50    0.00
   0.00    0.00    0.00    0.00
```

```
>> K3 = truss(12*16,0,12*16,12*12,E,A) % member 3
```

```
K3 =
```

```
0.00    0.00    0.00    0.00
0.00  2083.33    0.00 -2083.33
0.00    0.00    0.00    0.00
0.00 -2083.33    0.00  2083.33
```

```
>> K4 = truss(12*16,12*12,12*32,12*12,E,A) % member 4
```

```
K4 =
```

```
1562.50    0.00 -1562.50    0.00
   0.00    0.00    0.00    0.00
-1562.50    0.00  1562.50    0.00
   0.00    0.00    0.00    0.00
```

```
>> K5 = truss(12*16,12*12,12*32,0,E,A) % member 5
```

```
K5 =
```

```
800.00 -600.00 -800.00  600.00
-600.00  450.00  600.00 -450.00
-800.00  600.00  800.00 -600.00
 600.00 -450.00 -600.00  450.00
```

```
>> K6 = truss(12*16,0,12*32,12*12,E,A) % member 6
```

```
K6 =
```

```
800.00  600.00 -800.00 -600.00
 600.00  450.00 -600.00 -450.00
-800.00 -600.00  800.00  600.00
-600.00 -450.00  600.00  450.00
```

```
>> K7 = truss(12*16,0,12*32,0,E,A)           % member 7
K7 =
```

```
    1562.50    0.00   -1562.50    0.00
         0.00    0.00    0.00    0.00
   -1562.50    0.00    1562.50    0.00
         0.00    0.00    0.00    0.00
```

```
>> K8 = truss(12*32,0,12*32,12*12,E,A)       % member 8
K8 =
```

```
    0.00    0.00    0.00    0.00
    0.00  2083.33    0.00  -2083.33
    0.00    0.00    0.00    0.00
    0.00 -2083.33    0.00   2083.33
```

```
% ----- input the Global Stiffness Matrix by hand ...
```

```
>> Ks = [ 3925    600     0     0   -800   -600   ;
>         600  2533.33    0  -2083.33  -600   -450   ;
>         0     0   3162.5    0  -1562.5    0     ;
>         0  -2083.33    0   2983.33    0     0     ;
>        -800  -600  -1562.5    0   2362.5   600   ;
>        -600  -450     0     0     600  2533.33 ]
```

```
Ks =
```

```
    3925.00    600.00    0.00    0.00   -800.00   -600.00
    600.00    2533.33    0.00  -2083.33   -600.00   -450.00
    0.00     0.00   3162.50    0.00  -1562.50    0.00
    0.00  -2083.33    0.00   2983.33    0.00    0.00
   -800.00   -600.00  -1562.50    0.00   2362.50    600.00
   -600.00  -450.00    0.00    0.00    600.00   2533.33
```

```
>> find(Ks-Ks')           % check to see if Ks is symmetric ...
ans = [](0x0)             % It is!
```

```
>> p = [ 0 -100 0 0 50 0 ]'      % input the load vector
```

```
p =
```

```
    0.00
 -100.00
    0.00
    0.00
    50.00
    0.00
```

```
>> format                  % change formats for more sig. fig's
```

```
>> d = Ks \ p              % compute the joint displacements
```

```
d =
```

```
    0.0146067
   -0.1046405
    0.0027214
   -0.0730729
    0.0055080
   -0.0164325
```