Duke University
Department of Civil and Environmental Engineering
CE 131L. Matrix Structural Analysis
Henri P. Gavin
Fall, 2011

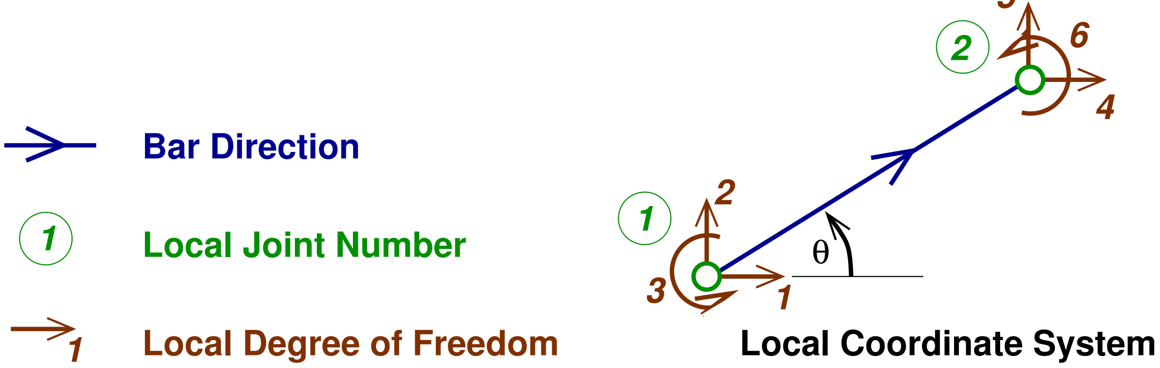
The Matrix Stiffness Method for 2D Frames

1. Identify Degrees of Freedom. Number the displacement coordinates and reaction coordinates in your frame. In a planar frame, every joint has three coordinates: one in the global X-direction, one in the global Y-direction, and one rotation about the global Z-axis (counter-clockwise). Using the method in this handout, every joint gets three coordinates.

2. Number all of the elements.

3. Joint Coordinates.
   Write the coordinates of each joint using units consistent with E and I. In other words, if E and I are given in kN/cm^2 and cm^4, write the coordinates, (x, y), in terms of centimeters.

4. Define each element.
   Draw each element of your frame individually and draw the local coordinates in the global directions. For example if element number N is a diagonal frame element, and the global directions are X: horizontal and Y: vertical, draw element number N like this:



   where 1,2,3,4,5,6 are the LOCAL coordinates of the frame element in the GLOBAL directions. The local coordinates are always numbered 1,2,3,4 with 1 and 4 pointing in the global X direction (to the right), with 2 and 5 pointing in the global Y direction (up), and with 3 and 6 rotating about the global Z-axis (counter-clockwise). All of these six coordinates will line up with with the global structural coordinates that you identified in step 1., above.

5. Element Stiffness Matrices in Global Coordinates, $\mathbf{K}$.
   For each element, find it's (6×6) element stiffness matrix, by evaluating the equations on pages 4 - 7 of the handout. 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 coordinates. In this step we will fill up the structural stiffness matrix using terms from the element stiffness matrices in global directions (from step 5.) This procedure is called matrix assembly.

   Recall from step 4. how the LOCAL element coordinates (1,2,3,4,5,6) line up with the GLOBAL structural coordinates. For example, local coordinates (1,2,3,4,5,6) might line up with structural coordinates (13,14,15,7,8,9) of the frame. In that case:

   `K(1,1)` is added to `Ks(13,13)`,
   `K(1,2)` is added to `Ks(13,14)`,
   ...
   `K(2,6)` is added to `Ks(14,9)`,
   ...
   `K(5,6)` is added to `Ks(8,9)`,
   `K(6,6)` is added to `Ks(9,9)`,

   Add each element into the structural stiffness matrix in this way to build up $\mathbf{K_s}$.

7. External Loads, $\mathbf{p} - \mathbf{p}_f$.
   Create the load vector $\mathbf{p} - \mathbf{p}_f$, by finding the equivalent forces and moments (negative fixed end forces) of each internal load, and their components in the directions of the structural displacement coordinates. Assemble the equivalent forces and moments for each internal load into the vector of concentrated joint loads, $\mathbf{p}$.

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: `d(q) = Ks(q,q) \ p(q)`, where the vector `q` lists the coordinates without reactions.

9. Internal frame element forces, $\mathbf{q}$.
   Again, recall how the global structural coordinates correspond to with each element's coordinates (1,2,3,4,5,6). For example, in element number "N" from step 6., the local element deflections v_1,v_2,v_3,v_4,v_5,v_6 line up with the global deflections d_{13},d_{14},d_{15},d_7,d_8,d_9. The internal bar forces can then be computed from:

$$\mathbf{q} = \mathbf{q}_f + \mathbf{T}^T \mathbf{K} \ \mathbf{v}$$

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
... $\mathbf{p} = \mathbf{K_s}\ \mathbf{d}$

$\mathbf{q}_f$ = the vector of fixed end forces from internal applied loads
in the local element coordinate system.

$\mathbf{f}_f$ = the vector of fixed end forces from internal applied loads
in the global element coordinate system
$\ldots \mathbf{f}_f = \mathbf{T}^T \mathbf{q}_f$

	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	-	$\mathbf{K_s}$

Example 1

```
hudson10% matlab


>> E=30e3;                                    % modulus of elasticity

>> I1=2000; I2=3000; I3=4000;                 % moments of inertia

>> L1 = 120; L2 = 150; L3 = 120;              % member lengths

>> Ks = [ 4*E*I1/L1 + 4*E*I2/L2       2*E*I2/L2          6*E*I1/L1^2 ;
>          2*E*I2/L2           4*E*I2/L2+4*E*I3/L3        6*E*I3/L3^2 ;
>          6*E*I1/L1^2            6*E*I3/L3^2     12*E*I1/L1^3 + 12*E*I3/L3^3 ]

Ks =                                          % structural stiffness matrix

  4400000  1200000    25000
  1200000  6400000    50000
    25000    50000     1250

>> w = 2;                      % uniform distributed load on element 2
>> P = 25;                     % lateral point load at joint 1

>> p = [ -w*L2^2/12 w*L2^2/12  P ]'
p =
                               % force vector ( fixed end forces )
  -3750
   3750
     25

>> d = inv(Ks)*p
d =
                               % structural displacements
   -1.1244e-03
    6.7611e-04
    1.5444e-02

% Now compute the Member Forces.

      % Moment in member 1 at joint 1 due to total deflections of member 1:

>> Ma = 4*E*I1/L1 * d(1) + 6*E*I1/L1^2 * d(3)
```

```
Ma = -1862.7

      % Moment in member 2 at joint 1 due to total deflections of member 2:

>> Mb = 4*E*I2/L2 * d(1) + 2*E*I2/L2 * d(2)
Mb = -1887.3

                  % Ma and Mb are not equal, although they should be!
                  % The difference is the fixed end force at joint 1.
>> M1 = Mb - p(1)
M1 = 1862.7                          % This is the actual moment at joint 1.


% ... Likewise for joint 2 ...

      % Moment in member 3 at joint 2 due to total deflections of member 3:

>> Ma = 4*E*I3/L3 * d(2) + 6*E*I3/L3^2 * d(3)
Ma = 3476.6

      % Moment in member 2 at joint 2 due to total deflections of member 2:

>> Mb = 4*E*I2/L2 * d(2) + 2*E*I2/L2 * d(1)
Mb = 273.36

                  % Ma and Mb are not equal, although they should be!
                  % The difference is the fixed end force at joint 2.
>> M2 = Mb - F(2)
M2 = -3476.7                         % This is the actual moment at joint 2.


% What about the ractions?

>> V = (M1+M2)/L2                    % shear in member 2 due to moments
V = -10.760

>> Vl = w*L2/2 + V                   % vertial reaction on the left
Vl = 139.24

>> Vr = w*L2/2 - V                   % vertical reaction on the right
Vr = 160.76
```

```
                        % horizontal reaction on the right
>> Hr = 12*E*I3/L3^3 * d(3) + 6*E*I3/L3^2 * d(2)
Hr = 46.675

                        % horizontal reaction on the left
>> Hl = 12*E*I1/L1^3 * d(3) + 6*E*I1/L1^2 * d(1)
Hl = -21.675

>> Hl + Hr                       % horizontal equilibrium check.  (P = 25)
ans = 25.000

                                 % moment reaction on the left
>> Ml = 6*E*I1/L1^2 * d(3) + 2*E*I1/L1 * d(1)
Ml = -738.32

                                 % moment reaction on the right
>> Mr = 6*E*I3/L3^2 * d(3) + 2*E*I3/L3 * d(2)
Mr = 2124.4

                                 % check global equilibrium
>> -P*L1 - w*L2^2/2 + Mr + Ml + Vr*L2

ans = 0.098131                   % a pretty small number, as compared to 2000
```

Example 2

```
% frame_debug.m  --- helpful in de-bugging frame analysis m-files
% --------------------------------- set up the problem definition matrices ...

E = 3e4;                       % modulus of elasticity
A = 10;                        % area of cross section
I = 1000;                      % moment of inertia

EAI = [ E E E ; A A A ; I I I ];  % matrix of E,A,I, one column for each frame element

EA  = prod(EAI([1 2],:));        % the product of E and A for each elmnt
EI  = prod(EAI([1 3],:));        % the product of E and I for each elmnt

XY = [   0   0 120 240 ;         % joint coordinates in global X direction
         0 150 150   0 ];        % joint coordinates in global Y direction

JTS = [   1   2   3   ;          % frame element location starting joint
          2   3   4   ];         % frame element location ending   joint

RCT = [   1   0   0  0  ;        % reaction in the "x"     direction
          1   0   0  1  ;        % reaction in the "y"     direction
          1   0   0  0  ];       % reaction in the "theta" direction

P = [   0   20    0   0 ;        % point force  in the "x"     direction
        0   30    0   0 ;        % point force  in the "y"     direction
        0    0   40   0 ];       % point moment in the "theta" direction

W = [   -0.03 -0.04  -0.05 ];    % uniformly distributed transverse load

D = zeros(3,4);                  % prescribed displacements

% ------------------- compute the element stiffness matrix for each element ...

% NOTE: DoF's 1,2,3 are at joint 1; DOF's 4,5,6 are at joint 2, etc.

format bank

L = zeros(1,3);
```

```
                            % element number 1: joint 1 -> joint 2 ... vertical
[K1, L(1)] = frame_element_2d ( XY(1,1),XY(2,1),XY(1,2),XY(2,2), EA(1), EI(1) )

% K1 =
%
%    106.67        0     -8000.00    -106.67        0     -8000.00
%       0      2000.00        0          0     -2000.00        0
%  -8000.00        0    800000.00    8000.00        0    400000.00
%   -106.67        0      8000.00     106.67        0      8000.00
%       0     -2000.00        0          0      2000.00        0
%  -8000.00        0    400000.00    8000.00        0    800000.00

                           % element number 2: joint 2 -> joint 3 ... horizontal
[K2, L(2)] = frame_element_2d ( XY(1,2),XY(2,2),XY(1,3),XY(2,3), EA(2), EI(2) )

% K2 =
%
%    2500.00         0           0      -2500.00         0           0
%        0        208.33    12500.00         0       -208.33    12500.00
%        0      12500.00  1000000.00         0     -12500.00   500000.00
%   -2500.00         0           0       2500.00         0           0
%        0       -208.33   -12500.00         0        208.33   -12500.00
%        0      12500.00   500000.00         0     -12500.00  1000000.00


                           % element number 3: joint 3 -> joint 4 ... diagonal
[K3, L(3)] = frame_element_2d ( XY(1,3),XY(2,3),XY(1,4),XY(2,4), EA(3), EI(3) )

% K3 =
%
%    640.43    -737.05    3809.12    -640.43     737.05    3809.12
%   -737.05     972.10    3047.29     737.05    -972.10    3047.29
%   3809.12    3047.29  624695.05   -3809.12   -3047.29  312347.52
%   -640.43     737.05   -3809.12     640.43    -737.05   -3809.12
%    737.05    -972.10   -3047.29    -737.05     972.10   -3047.29
%   3809.12    3047.29  312347.52   -3809.12   -3047.29  624695.05
```

```
% ----------------------------------- assemble the global stiffness matrix ...

Ks = zeros(12,12);                 % make the whole matrix 0 to start with

sc1 = [ 1  2  3  4  5  6  ];       % structural coordinates for frame element # 1
Ks(sc1,sc1) = Ks(sc1,sc1) + K1;    % assemble element #1 into Ks

sc2 = [ 4  5  6  7  8  9  ];       % structural coordinates for frame element # 2
Ks(sc2,sc2) = Ks(sc2,sc2) + K2;    % assemble element #2 into Ks

sc3 = [ 7  8  9  10 11 12  ];      % structural coordinates for frame element # 3
Ks(sc3,sc3) = Ks(sc3,sc3) + K3;    % assemble element #3 into Ks

% Ks =
%
%    Columns 1 through 6:
%        106.67          0     -8000.00     -106.67          0     -8000.00
%            0     2000.00          0           0     -2000.00          0
%      -8000.00          0    800000.00     8000.00          0    400000.00
%       -106.67          0      8000.00     2606.67          0      8000.00
%            0    -2000.00          0           0      2208.33     12500.00
%      -8000.00          0    400000.00     8000.00   12500.00   1800000.00
%            0          0          0      -2500.00          0           0
%            0          0          0           0      -208.33    -12500.00
%            0          0          0           0     12500.00    500000.00
%            0          0          0           0          0           0
%            0          0          0           0          0           0
%            0          0          0           0          0           0
%
%    Columns 7 through 12:
%            0          0          0           0          0           0
%            0          0          0           0          0           0
%            0          0          0           0          0           0
%      -2500.00          0          0           0          0           0
%            0     -208.33   12500.00           0          0           0
%            0   -12500.00  500000.00           0          0           0
%       3140.43     -737.05    3809.12     -640.43     737.05      3809.12
%       -737.05     1180.43   -9452.71      737.05    -972.10      3047.29
%       3809.12    -9452.71 1624695.05    -3809.12   -3047.29    312347.52
%       -640.43      737.05   -3809.12      640.43    -737.05     -3809.12
%        737.05     -972.10   -3047.29     -737.05     972.10     -3047.29
%       3809.12     3047.29  312347.52    -3809.12   -3047.29    624695.05
```

```
% check frame_assemble_2d.m
% Ks_difference should be practically all zero  (within roundoff error)

Ks_difference = Ks  - frame_assemble_2d ( XY,JTS,EA,EI)


% ---------------- add the equivalent joint loads to the point joint loads ...

Pv = P(:) + frame_loads_2d(XY,JTS,L,W)

% Pv =
%
%    2.25
%     0
%  -56.25
%   22.25
%   27.60
%    8.25
%   -3.75
%   -5.40
%  -65.75
%   -3.75
%   -3.00
%  153.75


% ------------ sort out which coordinates have reactions, and which don't ...

r = find(RCT);                          % Degrees of freedom with    reactions

% r =   1  2  3  11

q = find(~RCT);                         % Degrees of freedom without reactions

% q =   4  5  6  7  8  9  10  12

% compute the structural displacements, Dv, reactions, Rv, and frame element end forces, Q

format                       % change formats for more significant figures

Dv = zeros(12,1);
Rv = zeros(12,1);
Dv(q) = Ks(q,q) \ ( Pv(q) - Ks(q,r)*D(r) )    % compute the joint displacements
```

```
% Dv =
%    0                        % joint 1: D1, D2, D3, are all zero (fixed)
%    0
%    0
%   0.30925                   % joint 2: D4, D5, D6, are all non-zero (free)
%   0.01478
%  -0.00228
%   0.30625                   % joint 3: D7, D8, D9, are all non-zero (free)
%  -0.12343
%  -0.00018
%   0.45225                   % joint 4: D10, D11, D12, roller, X & rotation
%    0
%   0.00183

Rv(r) = Ks(r,q)*Dv(q)  + Ks(r,r)*D(r);    % compute the reactions
Rv    = Rv - frame_loads_2d(XY,JTS,L,W);  % Subtract equivalent frame loads from reactions
Rv(q) = 0.0;                              % Reactions are zero at joints "q"

% Rv =
%
%   -17.00                     % joint 1: D1, D2, D3, are all non-zero (fixed)
%   -29.55
%  1618.43
%      0                       % joint 2: D4, D5, D6, are all zero (free)
%      0
%      0
%      0                       % joint 3: D7, D8, D9, are all zero (free)
%      0
%      0
%      0                       % joint 4: D10, D11, D12, roller, Y-only
%    10.35
%      0

Q = frame_forces_2d ( XY, JTS, EA, EI, Dv, W )

% Q =
%   -29.55     7.50     8.08        % N1 - axial   -  joint #1
%    17.00     0.45     3.14        % V1 - shear   -  joint #1
%  1618.43  -594.07  -319.79        % M1 - moment  -  joint #1
%    29.55    -7.50    -8.08        % N2 - axial   -  joint #2
%   -12.50     4.35     6.47        % V2 - shear   -  joint #2
%   594.07   359.79      0          % M2 - moment  -  joint #2
```