Master's Degree Numerical Methods in Engineering

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Computational Structural Mechanics and Dynamics

Homework 6: Beams designment

Author:
Mariano Tomás Fernandez

Professor:
Miguel Cervera
Francisco Zárate

March 23^{rd}, 2020
Academic Year 2019-2020

Contents

Assignment 6.1

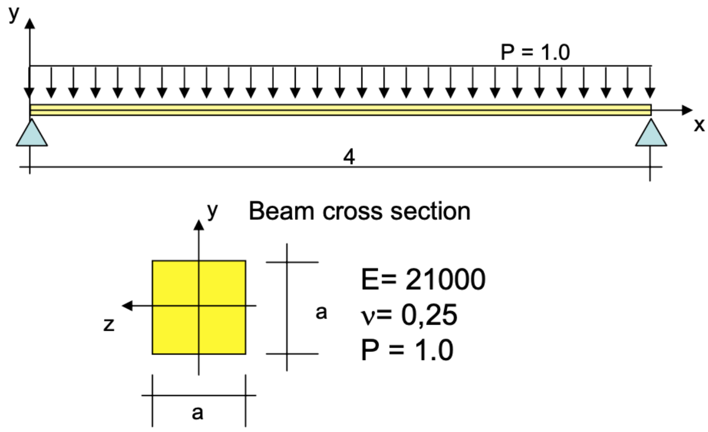
Program in MATLAB the Timoshenko 2-Nodes beam element with reduce integration for the shear stiffness matrix.

Assignment 6.2

Solve the following problem with a 64 element mesh with the:

- 2 nodes Bernoulli element;
- 2 nodes Timoshenko full integration element;
- 2 nodes Timoshenko reduced integration element.

Compare the maximum displacements, moments and shear for the 3 elements against the a/L relationship.
With a = [0.001, 0.005, 0.010, 0.020, 0.050, 0.100, 0.200, 0.400].



1 Assignment 6.1

Using the formulation from *MATFEM*, a matrix from *Beam_Timoshenko_v1_3.m* was updated to perform the reduced integration Timoshenko beam element. The matrix used is shown below, and in the Appendix A.1 the complete code can be seen.

$$K_S = \frac{G \cdot A*^{(e)}}{l} \cdot \begin{bmatrix} 1 & l^{(e)}/2 & -1 & l^{(e)}/2 \\ & (l^{(e)})^2/2 & -l^{(e)}/2 & (l^{(e)})^2/4 \\ & & 1 & -l^{(e)}/2 \\ & & & (l^{(e)})^2/4 \end{bmatrix}$$

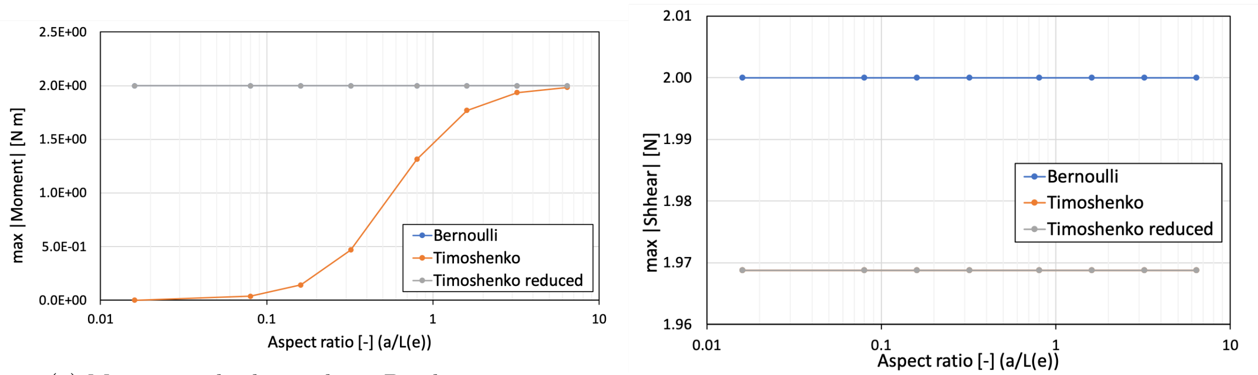
2 Assignment 6.2

To represent the conditions for this problem, GiD software was used to set the loads, constraints, sections and material properties, as well as the mesh for it.
Once the first set of conditions was established, MAT-FEM module in GiD allows to generate an input file for analyzing the beam using MATLAB. Once this file was generated, the analysis was performed for the different types of descriptions namely *Bernoulli, Timoshenko with full integration* and *Timoshenko with reduced integration.*
The element length is set to $L^{(e)} = 4.0m/64 = 0.0625m$ then, when changing the value of both width and height (a parameter), the aspect ratio $a_r = a/L$ varies. Using this parameter, the absolute values of displacement, shear force and bending moment were plotted.
The input file generated with the help of GiD was modified manually as the only property modified was the *area* and the *intertia* for the different sections. Then these files were analyzed with MATLAB. In Table 1 the eight models to be analyzed with the three beam description *Bernoulli, Timoshenko with full integration* and *Timoshenko with reduced integration* are presented.
The bending moment and shear forces registered in each of the beam descriptions are presented in Figure 1a and Figure 1b respectively. The displacements can be seen in Figure 2.
From the plots it is easy to notice that the three methods for all the aspect ratios registered:

- shear forces are in complete agreement with the theory of beams ($Q = q \cdot L/2 = 2.00\ N$), as the maximum error is below 2% for both *Timoshenko with full integration* and *Timoshenko with reduced integration*, and for *Bernoulli* the shear registered is exact.
- bending moments registered important errors in the *Timoshenko with full integration* the parameter $\beta = 4G\alpha/E \cdot \lambda^2$, where α is a shape factor related to the shear stresses, G, E are material parameters and λ is the inverse of the aspect ratio $a/L^{(e)}$, tend to infinite and shear blocking occurred for this kind of element and bending moment tend to *zero*. It is important to notice that λ varied from 62.5 to 0.16 and errors varying from 99.9% to 2%. For a $\lambda = 0.31$ the error is bigger than 10%, therefore this formulation does not seem to be reliable. In the case of *Bernoulli* and *Timoshenko with reduced integration* the error below 0.1% for every aspect ratio or λ. For aspect ratio above 3, the bending moment error is below 10%.
- theoretically displacements in a simply supported beam are $\omega = (5 \cdot P \cdot L^4)/(384 \cdot EI)$, this means that for every value of a different displacement are to be expected. To this end Figure 3 is introduced, and as it can be seen, *Timoshenko with full integration* description has an error of approximately 100% in the displacement prediction which is reduced to approximately 5% when the aspect ratio is around 2. In the case of *Timoshenko with reduced integration* and *Bernoulli* the error is approximately constant equal to 5%, which is acceptable.

Model	a [m]	a/L$^{(e)}$	A $[m^2]$	I $[m^4]$
1	0.001	1.60 10^{-02}	1.00 10^{-06}	8.33 10^{-14}
2	0.005	8.00 10^{-02}	2.50 10^{-05}	5.21 10^{-11}
3	0.010	1.60 10^{-01}	1.00 10^{-04}	8.33 10^{-10}
4	0.020	3.20 10^{-01}	4.00 10^{-04}	1.33 10^{-08}
5	0.050	8.00 10^{-01}	2.50 10^{-03}	5.21 10^{-07}
6	0.100	1.60	1.00 10^{-02}	8.33 10^{-06}
7	0.200	3.20	4.00 10^{-02}	1.33 10^{-04}
8	0.400	6.40	1.60 10^{-01}	2.13 10^{-03}

Table 1: Each of the models to be analysed.



(a) Maximum absolute values: Bending moments

(b) Maximum absolute values: Shear force

Figure 1: Bending moments and shear force for all the models with the three beam element descriptions.

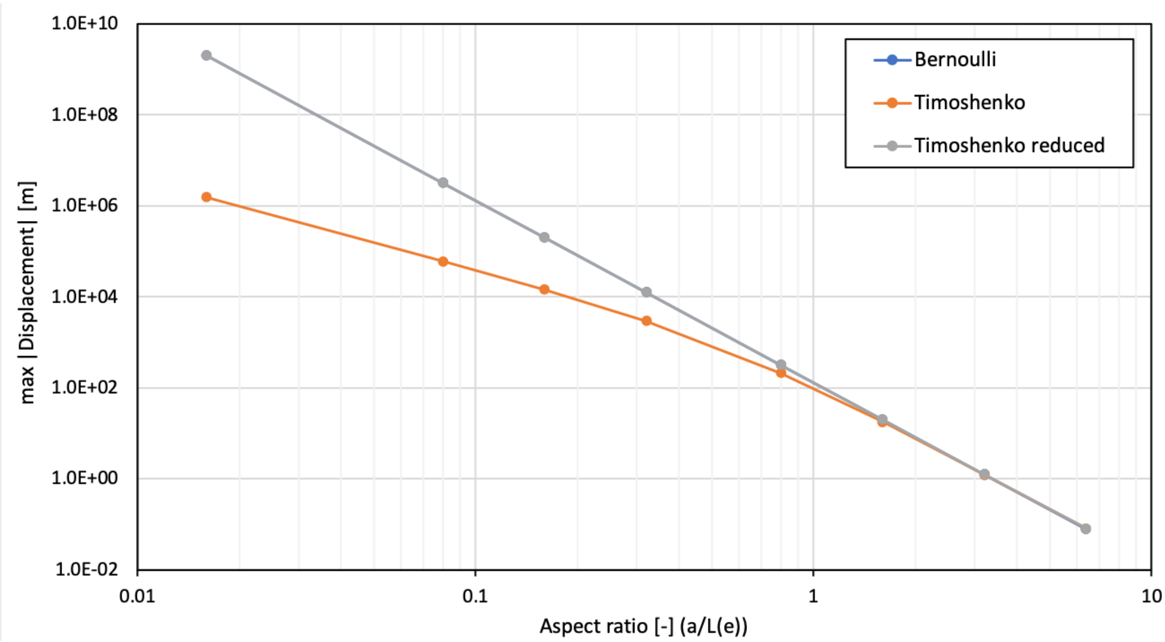


Figure 2: Maximum absolute value of registered displacements for all the models with the three beam element descriptions.

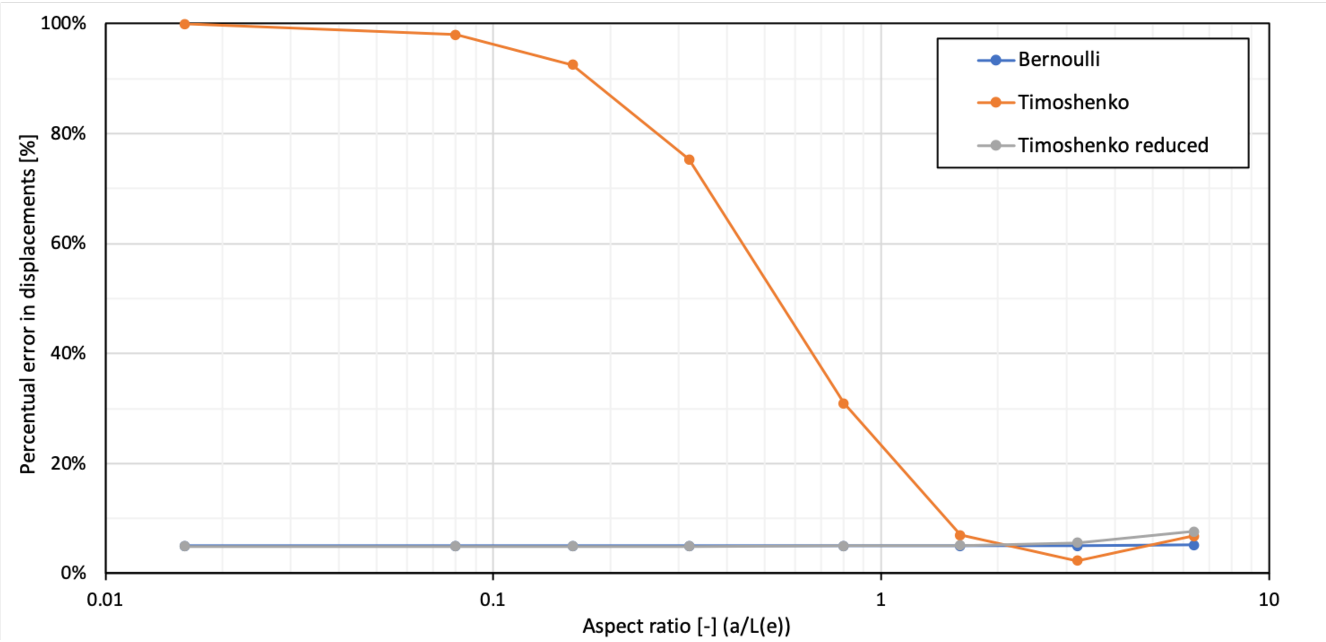


Figure 3: Percentual error between the maximum absolute value of registered displacements for all the models with the three beam element descriptions, compared to the theoretically correct maximum absolute displacement.

3 Conclusions

Eight different geometric sections were analyzed in the same beam designment using three different beam element description, namely *Bernoulli*, *Timoshenko with full integration* and *Timoshenko with reduced integration*. This last element beam description was added modifying a MAT-FEM file to correctly represent it. The results shown that both *Bernoulli* and *Timoshenko with reduced integration* represented correctly the stress, reaction forces and displacements in beams for most of the different sections and their slenderness (*aspect ration*). In the case of *Timoshenko with full integration* the results are not in agreement with the theory for aspect ratios below approximately 3 considering both displacement and bending moment, therefore this description is not reliable because of it mesh dependency.

A Appendix

A.1 Reduced integration Timoshenko - Code

```
%% MAT-fem_Beams
% 2 Nodes Beam using Timoshenko Theory

% Clear memory and variables
clear

% The variables are read as a MAT-fem subroutine
% young   = Young Modulus
% poiss   = Poission Ratio
% denss   = Material density
% area    = Cross section area
% inertia = Cross section inertia
% coordinates = [ x ] matrix size: nnode x ndime (1)
% elements    = [ inode , jnode ] element connectivity matrix.
%               Matrix size: nelem x nnode; nnode = 2
% fixnodes    = [ node number , dof , fixed value ] matrix with
%               Dirichlet restrictions, were dof=1 for vertical
%               displacement and dof=2 for vertical derivative
% pointload   = [ node number , dof , load value ] matrix with
%               nodal loads, were dof=1 for vertical load
%               and dof=2 for moment
% uniload     = [ uniform vertical load ] sparse matrix size: nelem x 1

file_name = input('Enter the file name: ','s');

tic;                    % Start clock
ttim = 0;               % Initialize time counter
eval(file_name);        % Read input file

% Find basic dimensions
npnod = size(coordinates,1);          % Number of nodes
nelem = size(elements,1);             % Number of elements
nnode = size(elements,2);             % Number of nodes por element
dofpn = 2;                            % Number of DOF per node
dofpe = nnode*dofpn;                  % Number of DOF per element
nndof = npnod*dofpn;                  % Number of total DOF

ttim = timing('Time needed to read the input file',ttim); %Reporting time

% Dimension the global matrices
StifMat  = sparse( nndof , nndof );  % Create the global stiffness matrix
force    = sparse( nndof , 1 );      % Create the global force vector
force1   = sparse( nndof , 1 );      % Create the global force vector
reaction = sparse( nndof , 1 );      % Create the global reaction vector
u        = sparse( nndof , 1 );      % Nodal variables

% Material properties (Constant over the domain)
D_matb = young*inertia;
D_mats = young/(2*(1+poiss))*area*5/6;

ttim = timing('Time needed to  set initial values',ttim); %Reporting time

% Element cycle
for ielem = 1 : nelem

  lnods(1:nnode) = elements(ielem,1:nnode);

  coor_x(1:nnode) = coordinates(lnods(1:nnode),1);   % Elem. X coordinate

  len = coor_x(2) - coor_x(1);  % x_j > x_i

  const = D_matb/len;

  K_b = [ 0 ,  0 ,  0 ,  0 ;
          0 ,  1 ,  0 , -1 ;
          0 ,  0 ,  0 ,  0 ;
          0 , -1 ,  0 ,  1 ];
```

```

    K_b = K_b * const;

    const = D_mats/len;

    K_s = [   1    ,   len/2 ,   -1    ,   len/2 ;
            len/2 , len^2/4 , -len/2 , len^2/4 ;
             -1    ,  -len/2 ,    1    ,  -len/2 ;
            len/2 , len^2/4 , -len/2 , len^2/4 ];

    K_s = K_s * const;

    K_elem = K_b + K_s;

    f       = (-denss*area + uniload(ielem))*len/2;
    ElemFor = [ f, 0, f, 0];

% Find the equation number list for the i-th element
    for i = 1 : nnode
      ii = (i-1)*dofpn;
      for j = 1 : dofpn
        eqnum(ii+j) = (lnods(i)-1)*dofpn + j;   % Build the eq. number list
      end
    end

% Assemble the force vector and the stiffness matrix
    for i = 1 : dofpe
      ipos = eqnum(i);
      force(ipos) = force(ipos) + ElemFor(i);
      for j = 1 : dofpe
        jpos = eqnum(j);
        StifMat(ipos,jpos) = StifMat(ipos,jpos) + K_elem(i,j);
      end
    end

  end  % End element cycle

  ttim = timing('Time to assemble the global system',ttim); %Reporting time

% Add point load conditions to the force vector
  for i = 1 : size(pointload,1)
    ieqn = (pointload(i,1)-1)*dofpn + pointload(i,2);   % Find eq. number
    force(ieqn) = force(ieqn) + pointload(i,3);         % and add the force
  end

  ttim = timing('Time for apply side and point load',ttim); %Reporting time

% Apply the Dirichlet conditions and adjust the right hand side
  for i = 1 : size(fixnodes,1)
    ieqn = (fixnodes(i,1)-1)*dofpn + fixnodes(i,2);  % Find equation number
    u(ieqn) = fixnodes(i,3);                         % and store the solution in u
    fix(i) = ieqn;                                 % and mark the eq. as a fix value
  end

  force1 = force - StifMat * u;      % Adjust the rhs with the known values

% Compute the solution by solving StifMat * u = force for the remaining
% unknown values of u
  FreeNodes = setdiff( 1:nndof , fix );             % Find the free node list
                                                    % and solve for it
  u(FreeNodes) = StifMat(FreeNodes,FreeNodes) \ force1(FreeNodes);

  ttim = timing('Time to solve the stiffness matrix',ttim); %Reporting time

% Compute the reactions on the fixed nodes as R = StifMat * u - F
  reaction(fix) = StifMat(fix,1:nndof) * u(1:nndof) - force(fix);

  ttim = timing('Time to solve the  nodal reactions',ttim); %Reporting time

% Compute the stresses
  Strnod = Stress_Beam_Timoshenko_v1_3(D_matb,D_mats,u);
```

```

  ttim = timing('Time to  solve the  nodal stresses',ttim); %Reporting time

% Graphic representation
  ToGiD_Beam_Timoshenko_v1_3(file_name,u,reaction,Strnod);

  ttim = timing('Time  used to write  the  solution',ttim); %Reporting time
  itim = toc;                                               %Close last tic
  fprintf(1,'\nTotal running time %12.6f \n\n',ttim); %Reporting final time
```