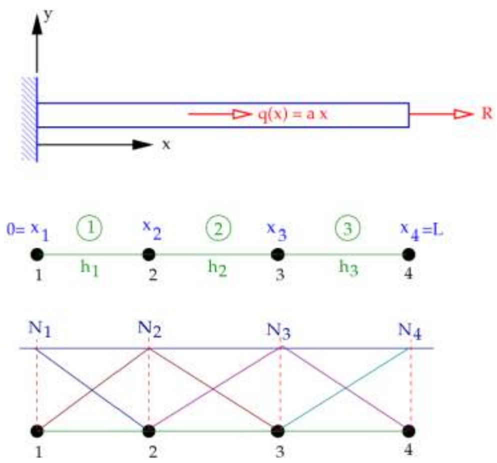
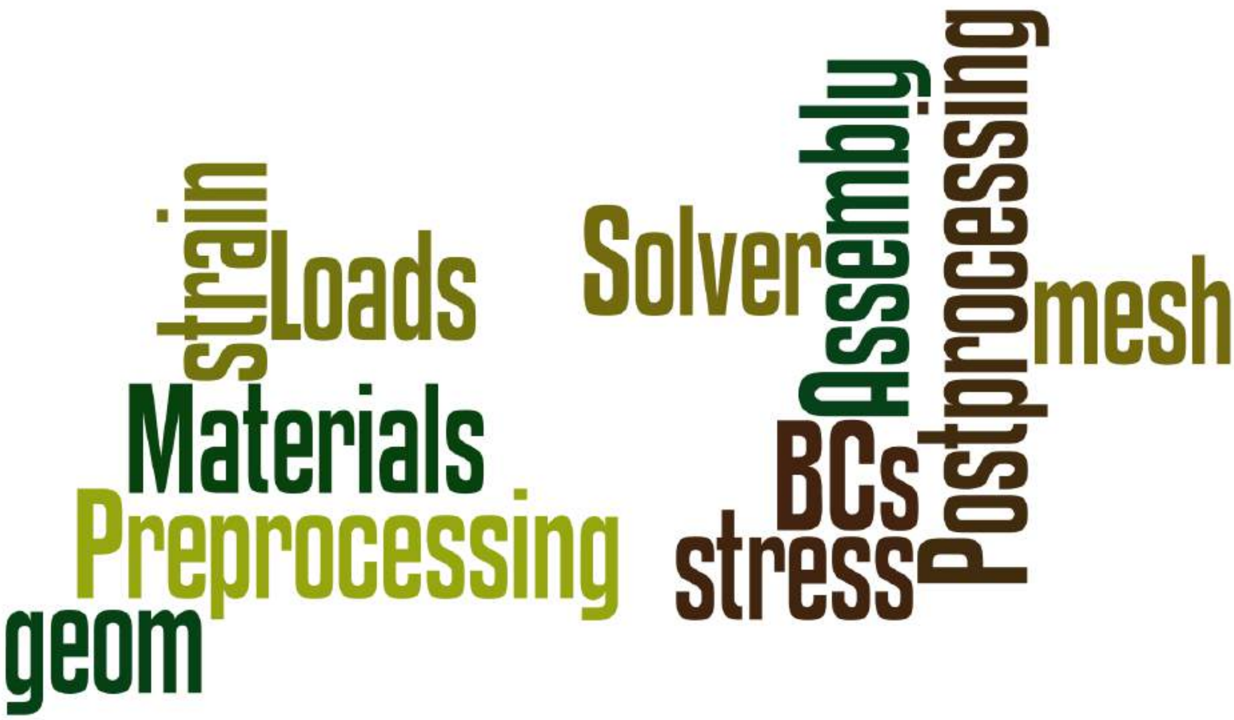
1MAE701 Exercices to be done/started in class

C1: Complete a simple FEA code





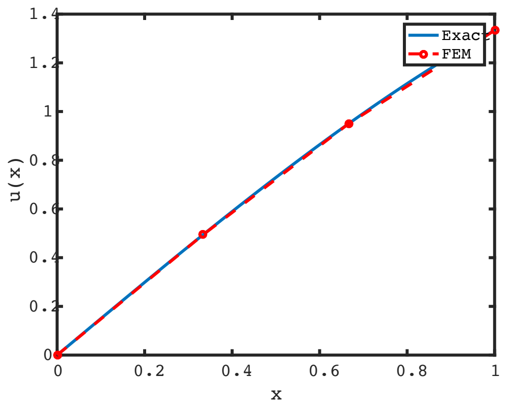
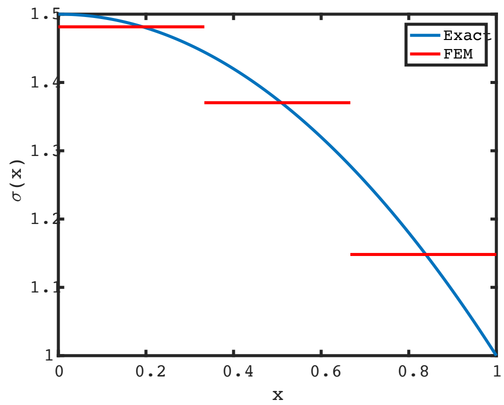
Fill the comments with above pictures wordles , specify the units

Copy paste part of the code in the ?? at the right place

B*u	[-1/h 1/h]	inv(Kred)*fred	(A*E/h)*[[1 -1];[-1 1]]	E*eps

```
function AxialBarFEM
%
%Material and Length
%
      A = 1.0; %cross section mm2
      L = 1.0; %length mm2
      E = 1.0; %Young's modulus MPa (N/mm2)
%loads
      a = 1.0; %distributed force N.m
      R = 1.0; %Force (N)

%
%Prepostfem mesh
%
      e = 3;
      h = L/e;
      n = e+1;

      for i=1:n
        node(i) = (i-1)*h;
      end
      for i=1:e
        elem(i,:) = [i i+1];
      end
%
%Assembly
%
      K = zeros(n);
      f = zeros(n,1);
      for i=1:e
        node1 = elem(i,1);
        node2 = elem(i,2);
        Ke = elementStiffness(A, E, h);
        fe =
elementLoad(node(node1),node(node2), a, h);
        K(node1:node2,node1:node2) =
K(node1:node2,node1:node2) + Ke;
        f(node1:node2) = f(node1:node2) + fe;
      end

%
%BCs &Loads
%
      f(n) = f(n) + 1.0;
      Kred = K(2:n,2:n);
      fred = f(2:n);
%
%Solvers
%
      d = ??;
      dsol = [0 d']; %known solution at x=0
      fsol = K*dsol';
      sum(fsol)
%
%Postprocessing
%
      figure;
      p0 = plotDisp(E, A, L, R, a);
      p1 = plot(node, dsol, 'ro--',
'LineWidth', 3); hold on;
      legend([p0 p1],'Exact','FEM');

      for i=1:e
        node1 = elem(i,1);
        node2 = elem(i,2);
        u1 = dsol(node1);
        u2 = dsol(node2);
        [eps(i), sig(i)] =
elementStrainStress(u1, u2, E, h);
      end

      figure;
      p0 = plotStress(E, A, L, R, a);
      for i=1:e
        node1 = node(elem(i,1));
        node2 = node(elem(i,2));
        p1 = plot([node1 node2], [sig(i)
sig(i)], 'r-','LineWidth',3); hold on;
      end
      legend([p0 p1],'Exact','FEM');


    function [p] = plotDisp(E, A, L, R, a)
      dx = 0.01;
      nseg = L/dx;
      for i=1:nseg+1
        x(i) = (i-1)*dx;
        u(i) = (1/6*A*E)*(-a*x(i)^3 + (6*R +
3*a*L^2)*x(i));
      end
      p = plot(x, u, 'LineWidth', 3); hold
on;
      xlabel('x', 'FontName', 'palatino',
'FontSize', 18);
      ylabel('u(x)', 'FontName', 'palatino',
'FontSize', 18);
      set(gca, 'LineWidth', 3, 'FontName',
'palatino', 'FontSize', 18);

    function [p] = plotStress(E, A, L, R, a)
      dx = 0.01;
      nseg = L/dx;
      for i=1:nseg+1
        x(i) = (i-1)*dx;
        sig(i) = (1/2*A*E)*(-a*x(i)^2 + (2*R
+ a*L^2));
      end
      p = plot(x, sig, 'LineWidth', 3); hold
on;
      xlabel('x', 'FontName', 'palatino',
'FontSize', 18);
      ylabel('\sigma(x)', 'FontName',
'palatino', 'FontSize', 18);
      set(gca, 'LineWidth', 3, 'FontName',
'palatino', 'FontSize', 18);

    function [Ke] = elementStiffness(A, E, h)

      Ke = ??;

    function [fe] = elementLoad(node1, node2,
a, h)
      x1 = node1;
      x2 = node2;
      fe1 = a*x2/(2*h)*(x2^2-x1^2) -
a/(3*h)*(x2^3-x1^3);
      fe2 = -a*x1/(2*h)*(x2^2-x1^2) +
a/(3*h)*(x2^3-x1^3);
      fe = [fe1;fe2];

    function [eps, sig] =
elementStrainStress(u1, u2, E, h)

      B = ??
      u = [u1; u2];
      eps = ??
      sig = ??;
```

Please also comment some lines %? And the results under
Play with e. Conclude

	
?	?

C2: Fill up a more “complex” FEA code* publish needed on LMS by pair

A cantilever beam of length Lp = 3 m and define by a squared section of aera Ap = 30×30 cm2.
The BCs are C-F, and the vertical load is imposed at the free boundary conditions such as F_o = 75 KN.
The materials used is concrete of Young’s modulus Eb = 32000 MPa.

Matlab will help you A LOT to accelerate learning and deepen understanding. The prof wants that you zip your publish or livescript with m files and upload on LMS.

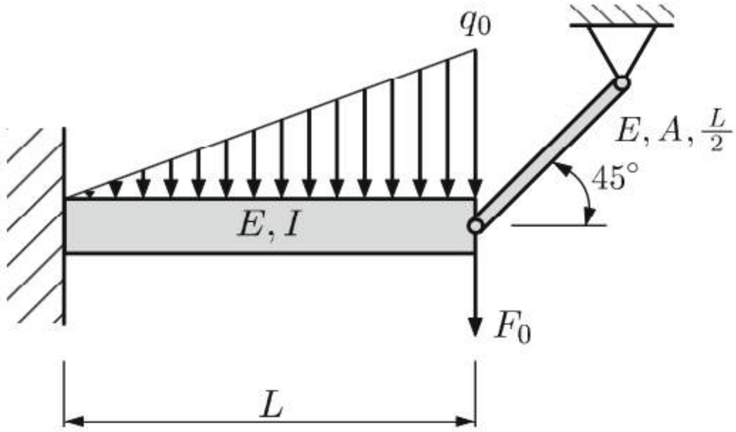


FIGURE 1: STRUCTURE TO BE DISCRETIZED USING 1 BEAM AND 1 BAR ELEMENT

Part 1 : Cantilever ONLY with Load case F_o

1) You can then compute the analytical beam deflection (to check)
2) Compute the rigidity matrix Kp and handwrite the system of equations respecting the nodes of figure above
3) Apply the BC Cantilver and solve the reduced system. Check using Matlab the solution and the resultant forces at the BC using the file “assignement2_student.m” on LMS

Part 2 : Cantilever beam+ inclined bar with the 2 load cases

The free end of the beam is connected to an inclined steel bar as shown in the figure1.
The section of the bar is Ab = 25 cm2 and the modulus of elasticity of the steel is Es = 200000 MPa.

0) Compute first the equivalent load for q_0 with max(q_0)=10kN
1) Calculate the stiffness matrix Kb of the inclined bar, taking into account the degrees of freedom of transverse displacement and rotation.
2) Introduce the axial displacement into the stiffness matrix Kp of the beam.
3) Assemble the two matrices Kb and Kp according to the numbering of the nodes given in the fiigure and write the global force vector (for all nodes of the structure).
4) Apply the boundary conditions and write the system of equations to solve.
5) Solve the system and give the displacements and rotations of the nodes. Check reactions
6) Calculate the axial force in the bar and the beam.

A SMALL HELP

CONCEPTS OF GAUSSIAN INTEGRATION

In Gaussian integration, also called Gaussian quadrature, the integral of a function in natural coordinates is substituted by an equivalent sum of this function evaluated at special points multiplied by a corresponding weight. Commonly, these special points are referred to as sampling points ξ_i.

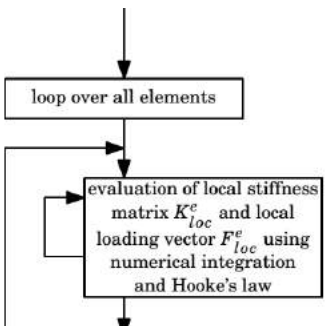
$$\int_{-1}^{1} F(\xi)d\xi = \sum_{i=1}^{NG} w_i F(\xi_i)$$

Although there is usually an error term associated with numerical integration, Gaussian quadrature has been shown to yield exact results for polynomials of degree 2m-1 or lower, where m is equal to the amount of weights / sampling points. This case directly applied to the common shape functions and their associated derivatives. Below is a table summarizing the position of the sampling points for Gaussian quadrature and their associated weights.

rule	point	coordinate	numerical value	weight	numerical value	order
1 point	1	0	0.0000000000	2	2.0000000000	1
2 points	1	$-\sqrt{\frac{1}{3}}$	-0.5773502692	1	1.0000000000	3
	2	$\sqrt{\frac{1}{3}}$	0.5773502692	1	1.0000000000	
3 points	1	$-\sqrt{\frac{3}{5}}$	-0.7745966692	$\frac{5}{9}$	0.5555555556	5
	2	0	0.0000000000	$\frac{8}{9}$	0.8888888889	
	3	$\sqrt{\frac{3}{5}}$	0.7745966692	$\frac{5}{9}$	0.5555555556	
4 points	1	$-\sqrt{\frac{3+2\sqrt{\frac{6}{5}}}{7}}$	-0.8611363116	$\frac{18-\sqrt{30}}{36}$	0.3478548452	7
	2	$-\sqrt{\frac{3-2\sqrt{\frac{6}{5}}}{7}}$	-0.3399810436	$\frac{18+\sqrt{30}}{36}$	0.6521451548	
	3	$\sqrt{\frac{3-2\sqrt{\frac{6}{5}}}{7}}$	0.3399810436	$\frac{18+\sqrt{30}}{36}$	0.6521451548	
	4	$\sqrt{\frac{3+2\sqrt{\frac{6}{5}}}{7}}$	0.8611363116	$\frac{18-\sqrt{30}}{36}$	0.3478548452	

SAMPLING POINTS AND CORRESPONDING WEIGHTS FOR GAUSSIAN QUADRATURE

PSEUDO CODE FOR IMPLEMENTATION IN A FINITE ELEMENT SOFTWARE



Processing:
for *each element* **do**
  recall element specific properties (length, cross-section, material)
  recall numerical integration points ξ_i and their weights w_i
  evaluate numerical integration of local stiffness matrix and load vector:
  for *each numerical integration point* **do**
    recall evaluated shape functions H_i and derivatives B_i
    evaluate Jacobian J and its determinant $|J|$
    evaluate local stiffness matrix $\mathbf{K^e_{loc}} = \mathbf{K^e_{loc}} + \mathbf{B^T D B |J|}\big|_{\xi_i} \cdot w_i$
    evaluate local loading vector $\mathbf{F^e_{loc}} = \mathbf{F^e_{loc}} + \mathbf{H^T f}\big|_{\xi_i} \cdot w_i$
  end

PSEUDO CODE FOR NUMERICAL INTEGRATION WITHIN A FINITE ELEMENT SOFTWARE

EXPECTED RESULTS

After first summation

K_local 4x4 double

	1	2	3	4
1	27.6000	43.5349	-27.6000	11.6651
2	43.5349	68.6697	-43.5349	18.4000
3	-27.6000	-43.5349	27.6000	-11.6651
4	11.6651	18.4000	-11.6651	4.9303

F_local 4x1 doub

	1
1	8.8490
2	2.6289
3	1.1510
4	-0.7044

After second summation

K_local 4x4 double

	1	2	3	4
1	55.2000	55.2000	-55.2000	55.2000
2	55.2000	73.6000	-55.2000	36.8000
3	-55.2000	-55.2000	55.2000	-55.2000
4	55.2000	36.8000	-55.2000	73.6000

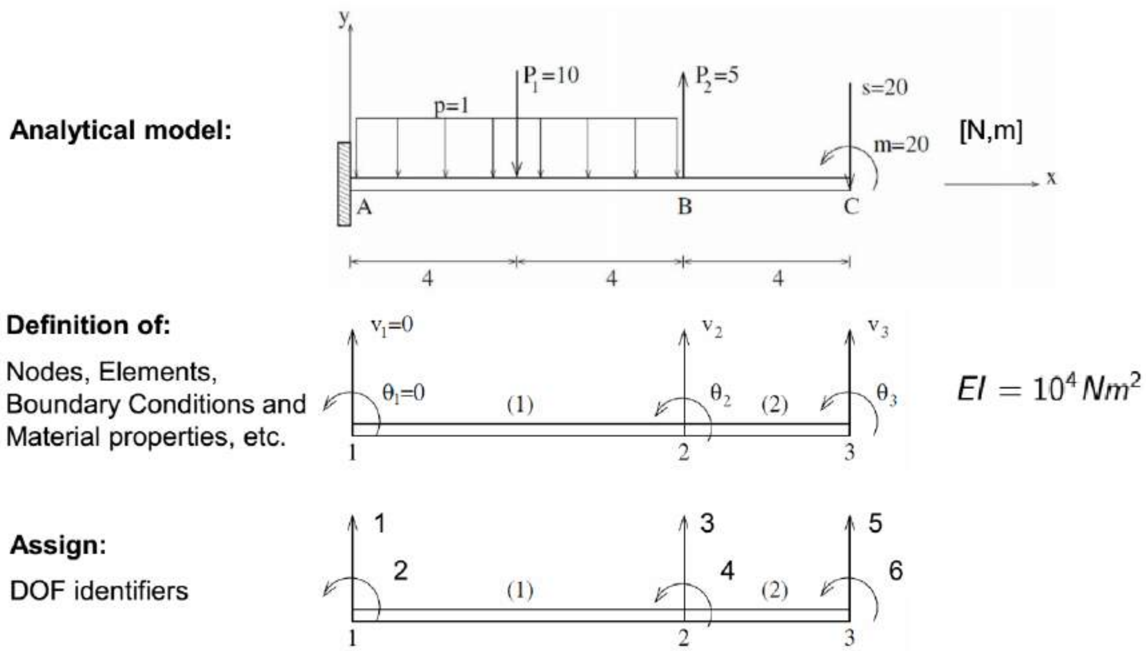
F_local 4x1 doub

	1
1	10
2	3.3333
3	10
4	-3.3333

SOLVING A SAMPLE PROBLEM

Given the following sample beam, find the displacements, moments and shear forces across the beam. Modify your numerical integration method slightly to work within a skeletal finite element program provided, for both the integration of the stiffness matrix as well as the force vector.

Take a look at the skeletal finite element code provided and complete the sections marked with comments in the code. Some intermediate results are provided below.



Some intermediate results are the following:

K_local = (Beam 1)

K_local = (Beam 2)

K_local — 4x4 double

	1	2	3	4
1	234.3750	937.5000	-234.3750	937.5000
2	937.5000	5.0000e+03	-937.5000	2.5000e+03
3	-234.3750	-937.5000	234.3750	-937.5000
4	937.5000	2.5000e+03	-937.5000	5.0000e+03

K_local — 4x4 double

	1	2	3	4
1	1.8750e+03	3.7500e+03	-1.8750e+03	3.7500e+03
2	3.7500e+03	1.0000e+04	-3.7500e+03	5.0000e+03
3	-1.8750e+03	-3.7500e+03	1.8750e+03	-3.7500e+03
4	3.7500e+03	5.0000e+03	-3.7500e+03	1.0000e+04

K_system =

K_system — 6x6 double

	1	2	3	4	5	6
1	234.3750	937.5000	-234.3750	937.5000	0	0
2	937.5000	5.0000e+03	-937.5000	2.5000e+03	0	0
3	-234.3750	-937.5000	2.1094e+03	2.8125e+03	-1.8750e+03	3.7500e+03
4	937.5000	2.5000e+03	2.8125e+03	1.5000e+04	-3.7500e+03	5.0000e+03
5	0	0	-1.8750e+03	-3.7500e+03	1.8750e+03	-3.7500e+03
6	0	0	3.7500e+03	5.0000e+03	-3.7500e+03	1.0000e+04

F_local = (distributed load)

F_local — 4x1 double

	1
1	-4
2	-5.3333
3	-4
4	5.3333

F_local = (point load)

F_local — 4x1 double

	1
1	-5
2	-10
3	-5
4	10

F_system_element = (point and distributed load)

F_system_element — 6x1 double

	1
1	-9
2	-15.3333
3	-9
4	15.3333
5	0
6	0

F_system_nodal = (all nodal loads)

F_system_nodal — 6x1 double

	1
1	0
2	0
3	5
4	0
5	-20
6	20

F_system =

F_system — 6x1 double

	1
1	-9
2	-15.3333
3	-4
4	15.3333
5	-20
6	20

d_system =

d_system — 6x1 double

	1
1	0
2	0
3	-0.5525
4	-0.1125
5	-1.0293
6	-0.1205