Engineer's Guide to Finite Element Analysis using Matlab and Nastran

Prof. Dr. Ing. Joseph MORLIER
https://github.com/jomorlier/feacourse2018

The key idea is to subdivide the structure into elements and nodes (finite numbers), displacements and rotations (if they exist) are the unknowns, results in huge systems of algebraic equations

- Assemble local element stiffness matrices into a global
- What is different is that we use elements other than frame elements (e.g. membrane, plate or shell elements)
- The finite element method generally provides an approximate solution whose accuracy increases as more element, and hence unknowns, are used.

Name of First Author
Name, Address of Institute, e-mail: name@email.address

Name of Second Author
Name, Address of Institute e-mail: name@email.address

Let's try to make a difference between Hand vs. computer analysis of structure.

Let's take the example of semicircular 2-pinned arch.

In this case the level of structure indetermination is of degree one.

• use three equilibrium equations and one compatibility equation to solve for reactions, then cut free body and determine N,V,M

• process is difficult to automate

for example if we had 3 pinned arch, so statically determinate four equations and zero compatibility equation !!

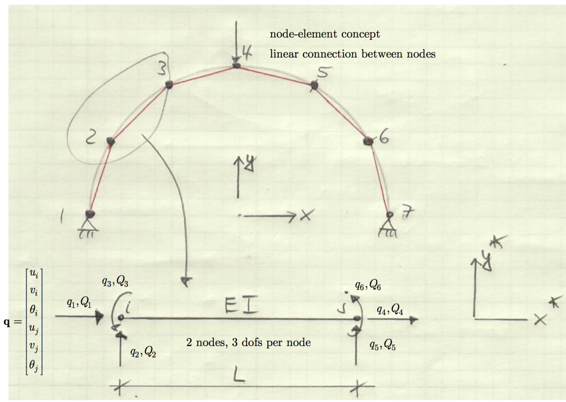

Fig. 1 Semicircular discrete approach .

FE method can be used for fluid dynamics (CFD), thermal analysis and/or any multiphysics analysis. Visit Matlab website:
https://fr.mathworks.com/help/pde/applications.html
In fact FE method is capable of Solving Partial Differential Equations (see First course). This manuscript is dedicated to Computational Structural Mechanics (CSM) only.

1 A bit of theory

subdivide structure into nodes and elements (probably straight), node-element concept, internal forces are N,V,M (see strength of materials/aeronautical structures) leads into the matrix problem (assume linear structural behavior) :

$\mathbf{K}q = f$ or $\overline{K}\widetilde{q} = \widetilde{f}$ or $[K]\{q\} = \{f\}$ my favourite is $\underline{\underline{K}} * \underline{q} = \underline{f}$

with q vector of element nodal displacements, f vector of element nodal forces, K element stiffness matrix. The above equation is a force-displacement relation in matrix form, which forms the basis of CSM. In order to get K , we need:

(1) stress-strain relation (constitutive relation)

(2) strain-displacement relation (kinematic relation)

(3) Equilibrium

Those three fundamental relations of structural analysis are part of the force method, the slope-deflection method, the solution to the differential equation or any conventional method of structural analysis, i.e. we can use any of these methods to derive K .

Note that this course will demonstrate bar, beam and membrane element. For plate/shell let's see another advanced course or ref.

NOTE —

The time derivative of v will be noted : $\dot{v}$

The space derivative of v will be noted: v'

Let's take the simplest example in 1D, Bar in traction, solve the PDE with Matlab mefeqd.m

2 (Re)view of Rayleigh-Ritz method

Assume displaced shape (trial functions) and use minimum potential energy to find the best solution among the trial functions

Mathematically this step consists of finding the constants c of the trial functions

Trial functions must satisfy geometric boundary conditions

If the set of trial functions contains the exact solution, then the method will find that solution

We start by writing a short review of beam's theory.

The internal energy stored in a beam element (neglecting shear deformation) is:
$W_i = \frac{1}{2}\int_0^L \sigma\varepsilon dV$

$$W_i = \frac{1}{2}\int_0^L \sigma\varepsilon dV$$

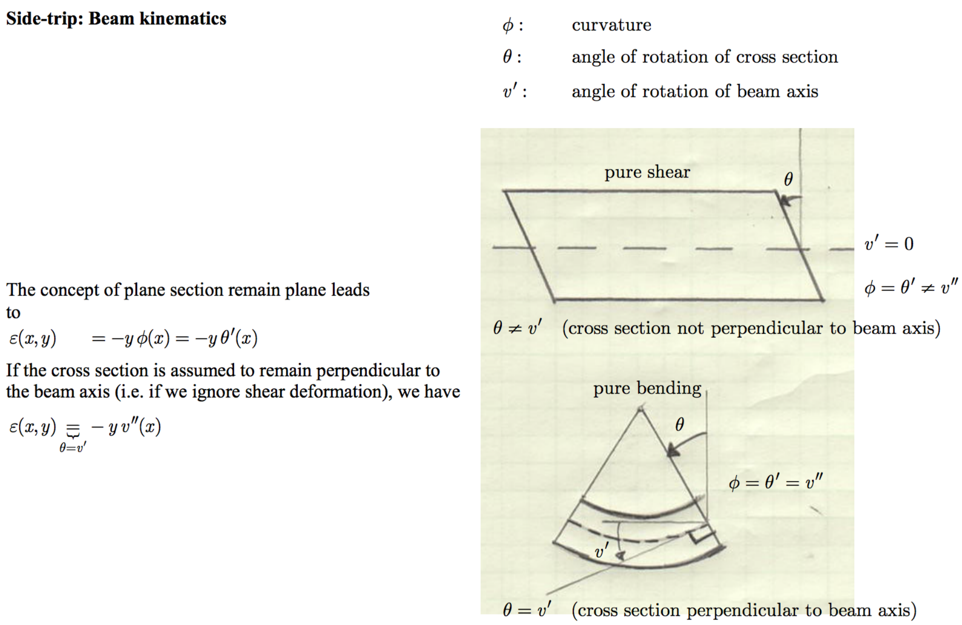

Fig. 2 Review of linear beam static's theory .

then the work done by external element loading along the element displacements :

$W_e = \frac{1}{2} \int_0^L v(x)\, p(x)\, dx$

Let's try to solve a Cantilever with uniformly end load Problem. We'll find the deflected shape and the moment diagram of structure. The demonstration will be made in course for a quadratic trial function. You'll nee to test your skills on a cubic trial function ??.

Let's try to solve the problem of the figure with $v(x) = cx^2$

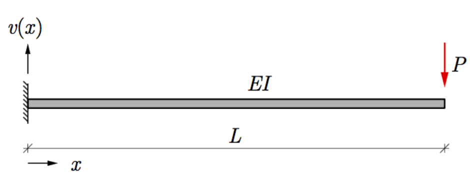

Fig. 3 Cantilever beam example .

so:
$v'(x) = 2cx$
$v''(x) = 2c = \phi(x) = $ curvature !!!

$$
\begin{aligned}
W_i &= \frac{1}{2}\int_0^L \sigma\varepsilon dV = \frac{1}{2}\int_0^L \varepsilon E\varepsilon dV \\
&= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV = \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV \\
&= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydAdx = \frac{1}{2}\int_0^L EI\phi(x)^2 dx \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
W_i &= \frac{1}{2}\int_0^L \sigma\varepsilon dV = \frac{1}{2}\int_0^L \varepsilon E\varepsilon dV \qquad (2a)\\
&= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV = \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV \qquad (2b)\\
&= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydAdx = \frac{1}{2}\int_0^L EI\phi(x)^2 dx \qquad (2c)
\end{aligned}
$$

Using Hooke's law and minimum of potential energy, we can write
$W = W_i - W_e = \dfrac{1}{2}\int_0^L EI\phi(x)^2 dx - Pv(L)$
by integrating and subtituing $\phi(x)$ it gives $W = \dfrac{1}{2}EI4cL^2 + PcL^2$
Thus deriving
$\frac{\partial W}{\partial c} = 4EIcL + PL^2 = 0 \rightarrow c = -\frac{PL}{4EI}$
and expressing $M(x) = EIv''(x) = EI(-\frac{PL}{4EI}) = -\frac{PL}{2}$

Note that we could have used chain rules for avoiding integrate the square of Phi:
$\frac{\partial W_i}{\partial c} = \dfrac{2}{2}EI\int_0^L \phi(x)\frac{\partial\phi(x)}{c}dx = EI * 2c * 2L$;)
Finally we got: $v(L) = -\frac{PL^3}{4EI}$.

The exact value is $v(L) = -\frac{PL^3}{3EI}$, try cubic trial function $c_1x^2 + c_2x^3$: It means 2 unknowns... see homework 1, Illustration of potential energy too.

We have seen that we obtain a more accurate solution is to increase the degree of the polynomial of the trial functions. We have a second option: That is decrease the size of the element for which each trial functions is valid, that is subdivide the beam into several elements and assume independent trial functions for each of those elements. In the RR method with free constants c it is difficult to enforce the boundary conditions between adjacent elements. A natural way to enforce the boundary conditions is to express the trial function in terms of nodal parameters (displacement and rotation) rather the free constants as in the Rayleigh-Ritz method. This is the key idea of the finite element method. We will demonstrate this concept in the next chapter using beam elements, the simplest family of finite elements.

3 Rod finite element

It is possible to use symbolic computing in order to express the stiffness matrix of a rod. Here is the matlab code.

```
%% Derivation of the stiffness matrix of a 2-node bar element

%clear workspace
clear all; close all,

%% Define symbolic quantities
%%
syms r x L
syms E_mod A_sec
syms q
%the derived quantities will automatically be initialized ...
    as symbolic variables as well

%% Derive shape functions N
%%
A = [1,r]
E = [1,-1;1,1]
N = A*E^-1
%% Derive strain-displacement matrix B
%%
Jac = diff(2*x/L-1,x)
det_Jac = Jac^-1
B = diff(N)*Jac
%% Construct stiffness matrix
%%
K_2node = E_mod * A_sec * int(B'*B,r,-1,1)*det_Jac
matrix_factor = gcd(K_2node), K_2node/matrix_factor
%% Construct loading vector
%%
F_2node = int(N'*q,r,-1,1)*det_Jac
vector_factor = gcd(F_2node), F_2node/vector_factor
%%
```

Starting by Deriving shape functions N. It comes

```
A = [1,r]
```

$A = \begin{pmatrix} 1 & r \end{pmatrix}$

```
E = [1,-1;1,1]
```

E = 1.00000000000000e+000 -1.00000000000000e+000 1.00000000000000e+000 1.00000000000000e+000

```
N = A*E^-1
```

N = $\left(\frac{1}{2} - \frac{r}{2} \;\; \frac{r}{2} + \frac{1}{2} \right)$

Then we derive strain-displacement matrix B:

```
Jac = diff(2*x/L-1,x)
```

Jac = $\frac{2}{L}$

```
det_Jac = Jac^-1
```

det$_{J}ac = \frac{L}{2}$

```
B = diff(N)*Jac
```

B = $\left(-\frac{1}{L} \;\; \frac{1}{L} \right)$

Finally, we construct the stiffness matrix:

```
K_2node = E_mod * A_sec * int(B'*B,r,-1,1)*det_Jac
```

K$_{2}node = \begin{pmatrix} \frac{A_{\mathrm{sec}} E_{\mathrm{mod}}}{L} & -\frac{A_{\mathrm{sec}} E_{\mathrm{mod}}}{L} \\ -\frac{A_{\mathrm{sec}} E_{\mathrm{mod}}}{L} & \frac{A_{\mathrm{sec}} E_{\mathrm{mod}}}{L} \end{pmatrix}$

```
matrix_factor = gcd(K_2node), K_2node/matrix_factor
```

matrix$_{f}actor = \frac{A_{\mathrm{sec}} E_{\mathrm{mod}}}{L}$ ans = $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$

Then we construct the loading vector:

```
F_2node = int(N'*q,r,-1,1)*det_Jac
```

F$_{2}node = \begin{pmatrix} \frac{Lq}{2} \\ \frac{Lq}{2} \end{pmatrix}$

```
vector_factor = gcd(F_2node), F_2node/vector_factor
```

vector$_{f}actor = \frac{Lq}{2}$ ans = $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

4 Beam finite element

First course introduce the direct stiffness method to find the stiffness matrix of a bar (1D, works in traction/compression only).

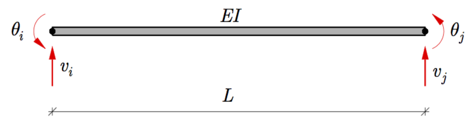

Fig. 4 4DOFs beam .

In this chapter we demonstrate the basics of the finite element method using the simple beam above.

The four nodal parameters allow us to interpolate the deflection by a cubic polynomial. As in the Rayleigh-Ritz method, we start by using free constants c.

Let's first define ξ adimensional (see mapping...) length equal to x/L.

Then we can write (the second is easy to derive function of x;)

v$(\xi) = c_1 + c_2 * \xi + c_3 * \xi^2 + c_4 * \xi^3$
$v'(\xi) = (1/L) * (c_2 + c_3 * \xi + c_4 * \xi^2)$ (3)

or in matrix form: $v(\xi) = [1, \xi, \xi^2, \xi^3]\underline{c}$

with $\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$

Next, we use the four boundary conditions of the previous figure to write:
$v(x=0) = v_i$
$v'(x=0) = \theta_i$
$v(x=L) = v_j$
$v'(x=L) = \theta_i$
In matrix form using
$v(\xi) = [1, \xi, \xi^2, \xi^3]\underline{c}$
it gives:

$$\begin{matrix} v_i \\ \theta_i \\ v_j \\ \theta_j \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/L & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1/L & 2/L & 3/L \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

or in matrix form:
$\underline{q} = \underline{\underline{G}} * \underline{c}$

thus $G^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & L & 0 & 0 \\ -3 & -2L & 3 & -L \\ 2 & L & -2 & L \end{bmatrix}$

We substitute in (3) the inverse of this relation $\underline{c} = \underline{\underline{G}}^{-1} * \underline{q}$ to obtain:
$v(\xi) = [1, \xi, \xi^2, \xi^3]\underline{\underline{G^{-1}q}}$
so
$v(\xi) = N(\xi)\underline{q}$
with
$N(\xi) = [1 - 3\xi^2 + 2\xi^3, L(\xi - 2\xi^2 + \xi^3), 3\xi^2 - 2\xi^3, L(\xi^3 - \xi^2)]$

where N is a 4x1 row vector of shape functions. Note that the concept of shape functions for beam elements is to fit a curve between two points at which both the ordinate (q1,q3) and the slope (q2,q4) are known (four data items define a cubic polynomial).

Shape functions for beam element with four degrees of freedom can be computed using this code untitled shape1.m on LMS: x is xi?

```
syms x L real % symbolic variable
P = inline('[1 x x^2 x^3]') % third order polynomial
dP = inline(diff(P(x))) % symbolic derivation
Pn = [ P(0); dP(0); P(L) ; dP(L) ] % Nodes evaluation
N = inline(( P(x) * inv(Pn))) % Shape Functions
dN = inline((dP(x) * inv(Pn))) % Derivatives
% graphs
x = 0:0.01:1;
subplot(2,1,1), plot(x, N(1,x')), title(' Shape functions ...
    N1 N2 N3 N4 ')
subplot(2,1,2), plot(x,dN(1,x')), title(' Derivative dN1 ...
    dN2 dN3 dN4 ')
end
```

BUT —

how using this, can I postprocess the strain/Stress to obtain nice colors?

In structural analysis of beams, we work with a generalized strain, the curvature. In linear structural analysis (small displacement theory) and ignoring shear deformation, the curvature (x) is the second derivative of the deflected shape v(x), thus

$$\phi(x) = \frac{d^2 v(\xi)}{dx^2} = \frac{d^2}{dx^2} N(\xi) q = \frac{1}{L^2} \frac{d^2}{dx^2} N(\xi) q = B(\xi) q \qquad (4)$$

with $\xi = x/L$

The 4x1 vector B can be written:

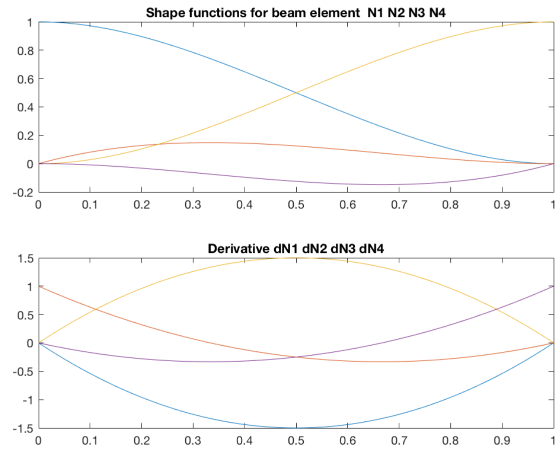

Fig. 5 Shape functions, easy to analytically derive, isn't it.

$$B(\xi) = \frac{1}{L^2}\frac{d^2}{dx^2}N(\xi) = [\frac{1}{L^2}(-6+12\xi); \frac{1}{L}(-4+6\xi); \frac{1}{L^2}(6-12\xi); \frac{1}{L}(-2+6\xi)] \quad (5)$$

is commonly referred to as strain-displacement transformation vector since it relates the strain (curvature) and nodal displacements q.

Then we should use the law of linear elasticity (HOOKE) which relates the curvature, the generalized strain for beam elements, to the bending moment (the corresponding generalized stress)

$M(\xi) = EI\phi(\xi) = EIB(\xi)q$

BUT —

We discuss a lot about $\underline{\underline{K}} * \underline{q} = \underline{f}$, How can I construct $\underline{\underline{K}}$?

Remember the RR approach:

$W_i = \frac{1}{2}\int_0^L \sigma\varepsilon dV = \frac{1}{2}\int_0^L \varepsilon E\varepsilon dV$

$= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV = \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydV$

$= \frac{1}{2}\int_0^L \phi(x)yE\phi(x)ydAdx = \frac{1}{2}\int_0^L EI\phi(x)^2dx = \frac{1}{2}\int_0^L B(x)*q*B(x)*q*EIdx$

using matrix.vector product:

$$\frac{1}{2}\int_0^L q^T * B^T(x) * B(x) * q * EIdx \quad (6)$$

Note that $\varepsilon(y) = \phi * y$ and $\int y^2 dA = I$

The external work done on a beam element is the work done by the nodal forces Q along the nodal displacements (general force and moment matrix) and the work done by external element loading F along the element displacements (if element loading p(x) is present).

$$We = q^T * Q + \int_O^L N(x) * q * p(x)dx = q^T * Q + q^T \int_O^L N^T(x) * p(x)dx = q^T * Q + q^T * F \tag{7}$$

with $F = \int_O^L N^T(x) * p(x)dx$

The vector F contains forces applied to the nodes that are equivalent (in an energy sense) to the element loading.It is thus often referred to as vector of equivalent nodal forces. Note that the potential energy is now a function of the nodal displacements q , in the Rayleigh-Ritz method it is a function of the free constants c . Taking derivative with respect to q yields

$$\frac{\partial W}{\partial q} = \begin{bmatrix} \frac{\partial W}{\partial q_1} \\ \frac{\partial W}{\partial q_2} \\ \vdots \\ \frac{\partial W}{\partial q_n} \end{bmatrix} = \int_0^L B^T(x) \cdot EI \cdot B(x)dx \cdot q - Q - F = 0 \tag{8}$$

Isn't it $\underline{\underline{K}} * \underline{q} = \underline{f}$ with $\underline{f} = Q + F$
and

$$\underline{\underline{K}} = \int_0^L B^T(x) * EI * B(x)dx \tag{9}$$

We then try to construct the first element of this 4x4 matrix (4DOFs)=
it comes:

$$k_{11} = \int_0^L B_1^T(x) * EI * B_1(x)dx = \int_0^1 B^T(\xi) * EI * B(\xi)Ld\xi$$

=$\int_0^1 (1/L^2)(6-12\xi) * EI * (1/L^2)(6-12\xi) * Ld\xi$
=(EI/L^3) $\int_0^1 (6-12*\xi)^2 d\xi$
=(EI/L^3)$(-1/(12*3))\left[(6-12\xi)^3\right]_0^1$
= $\frac{12EI}{L^3}$(10)
finally,

$$\underline{\underline{K}} = (EI/L^3) \begin{bmatrix} 12 & 6L & -12 & 6L \\ & 4L^2 & -6L & 2L^2 \\ sym & & 12 & -6L \\ & & & 4L^2 \end{bmatrix} \tag{11}$$

```
N:=matrix([N_1,N_2,N_3,N_4])
```

$$\begin{pmatrix} \frac{2X^3}{L^3} - \frac{3X^2}{L^2} + 1 \\ X\left(\frac{X}{L} - 1\right)^2 \\ \frac{3X^2}{L^2} - \frac{2X^3}{L^3} \\ -L\left(\frac{X^2}{L^2} - \frac{X^3}{L^3}\right) \end{pmatrix}$$

```
B:=hold(transpose(diff(diff(N,X),X)))
```

$$\left(\frac{\partial}{\partial X}\frac{\partial}{\partial X} N\right)^t$$

```
B:=(transpose(diff(diff(N,X),X)))
```

$$\left(\frac{12X}{L^3} - \frac{6}{L^2} \quad \frac{2X}{L^2} + \frac{4\left(\frac{X}{L}-1\right)}{L} \quad \frac{6}{L^2} - \frac{12X}{L^3} \quad L\left(\frac{6X}{L^3} - \frac{2}{L^2}\right) \right)$$

```
K=hold(EI*int(transpose(B)*B,X=0..L));
```

$$K = \mathrm{EI}\int_0^L B^t\, B\, \mathrm{d}X$$

```
K=(EI*int(transpose(B)*B,X=0..L));
```

$$K = \begin{pmatrix} \sigma_2 & \sigma_3 & -\sigma_2 & \sigma_3 \\ \sigma_3 & \frac{4\,\mathrm{EI}}{L} & \sigma_1 & \frac{2\,\mathrm{EI}}{L} \\ -\sigma_2 & \sigma_1 & \sigma_2 & \sigma_1 \\ \sigma_3 & \frac{2\,\mathrm{EI}}{L} & \sigma_1 & \frac{4\,\mathrm{EI}}{L} \end{pmatrix}$$

Fig. 6 Again K symbolically with $\sigma_1 = -\sigma_3$ and $\sigma_2 = 12EI/L^3$ and $\sigma_3 = 6EI/L^2$.

Similarly one can obtain Mass matrix

$$\underline{\underline{M}} = \int_0^L N^T(x) * \rho * A * N(x) dx \tag{12}$$

with final computation leads to:

$$\underline{\underline{M}} = (\rho * A * L/420) \begin{bmatrix} 156 & 22L^2 & 54 & -13L \\ & 4L^2 & 13L & -3L^2 \\ sym & & 156 & -22L \\ & & & 4L^2 \end{bmatrix} \tag{13}$$

Only valid If the beam is made of the same homogeneous material and the same section.

One other remark concerning the mass matrix is that we can associate half of the total mass of the element with the degrees of translation of each node. We only take the translations because the rotations do not produce forces of inertia. The concentrated mass matrix is thus written as follows:

$$\underline{\underline{M}} = (\rho * A * L/420) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

It should be noted that the two concentrated and distributed matrices give the same total mass with the sum of the components associated with degrees of freedom of translation: M(1, 1) M(1, 3) M(3, 1) M(3, 3)

BUT —

Verify ?

By writing Potential and Kinematic energy of the beam we can directly identify $\underline{\underline{K}}$ and $\underline{\underline{M}}$ using the quadratic form per element.

$$V_e = \frac{1}{2}\int_0^L EI * v''^2 dx = \frac{1}{2}\int_0^L q^T * B^T(x) * EI * B(x) * q dx = \frac{1}{2} q_e^T * \underline{\underline{K_e}} * q_e \tag{15}$$

$$T_e = \frac{1}{2}\int_0^L \rho * A * \dot{v}^2 dx = \frac{1}{2}\int_0^L q^T * N^T(x) * \rho * A * N(x) * q = \frac{1}{2}\dot{q_e}^T * \underline{\underline{M_e}} * \dot{q_e} \tag{16}$$

$$\begin{array}{c} \\ \mathbf{k} = \end{array} \begin{array}{c} \begin{array}{cccccc} \quad u_i & \qquad v_i & \qquad \theta_i & \qquad u_j & \qquad v_j & \qquad \theta_j \end{array} \\ \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \end{array}$$

Fig. 7 You can find on the slides the "augmented" generalized beam for space truss/beam assembly

4.1 Equivalent nodal forces and examples, see exercices

As an example, we calculate the first component of the vector of equivalent nodal forces F due to a trapezoidal load with intensities p1 and p2

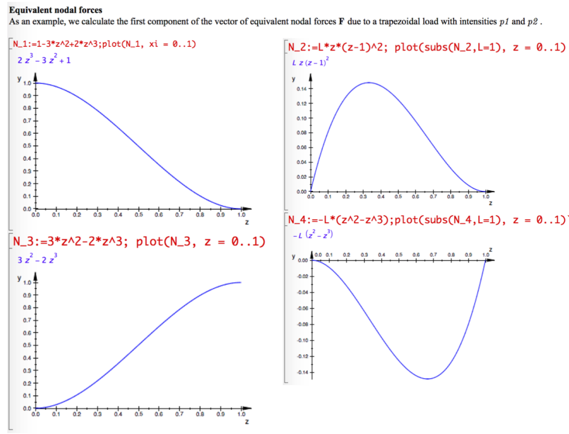

Fig. 8 Shape functions, easy to analytically integrate, isn't it.

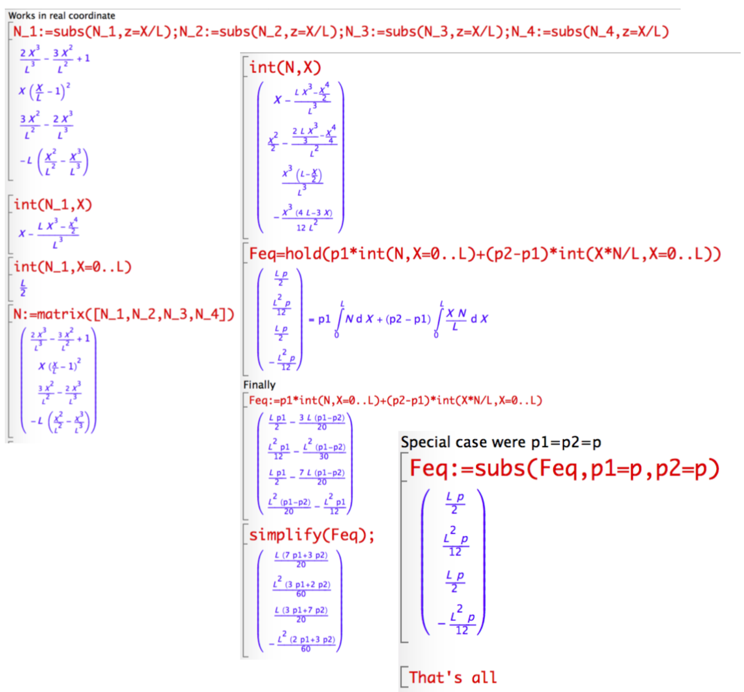

Fig. 9 Symbolic computation of equivalent nodal forces.

5 Assembly process: from local to global

See slides and last year assignement 2

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 = 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.

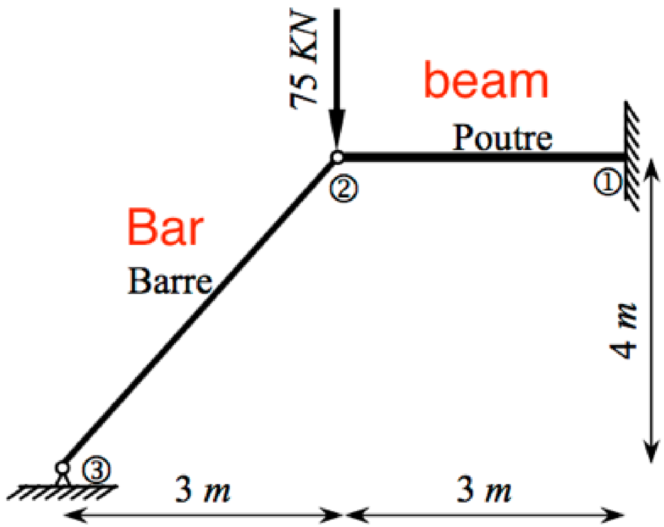

Fig. 10 Problem C2.

- Part 1 : Cantilever ONLY
- 1) Compute Kp the rigidity matrix Kp and handwrite the system of equations respecting the nodes of figure above
- 2) Apply the BC Cantilver and solve the reduced system. Check using Matlab the solution and the resultant forces at the BC using the file “Assignement$_2$.m”$onLMS.Part2:$ $Cantileverbeam+inclinedbarThefreeendofthebeamisconnectedtothegroundbyaninclinedsteelbarasshowninthefig$ $25cm2andthemodulusofelasticityofthesteelisEs = 200000MPa.$
- 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 Figure 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.

- 6) Calculate the axial force in the bar and the beam.

6 Gauss Integration

In finite element analysis, the element stiffness matrix is obtained by evaluating an integral. For line element (also termed 1-D elements, like rod, truss, beam, frame), we integrate over the length of the element, since the integration over the cross section has already been carried out by using cross section properties like the area A and the second moment of inertia I . For 2D elements, the integral is over the area of the element, for volume elements we integrate over the volume.
In finite element analysis, the integrand usually becomes quite complicated and it is not possible to solve the integrals in closed form. We have to use numerical procedures in these situations. In finite element calculations, simple numerical integration schemes such as the trapezoidal rule or Simpson's formula do not work very well.

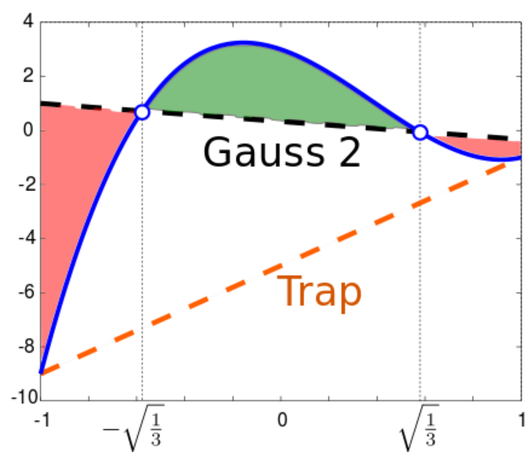

Fig. 11 Example of Gaussian Quadrature ref WIKIPEDIA.

The blue line is the polynomial $y(x) = 7x^3 - 8x^2 - 3x + 3$, whose integral in [1, 1] is 2/3. The trapezoidal rule returns the integral of the orange dashed line, equal to $y(-1) + y(1) = -10$. The 2-point Gaussian quadrature rule returns the integral of the black dashed curve, equal to $y(-\sqrt{1/3}) + y(\sqrt{1/3}) = 2/3$. Such a result is exact, since the green region has the same area as the red regions

The Gauss numerical integration scheme or Gauss quadrature, has becomes the standard tool to calculate element stiffness matrices. We derive the Gauss formula for one-dimensional integrals in the following section and then easily extend it to two and three-dimensional integrals.

As in other numerical integration schemes, the basic idea is to represent the integral in the following form (an integral is replaced with a sum). Let's take a 1D example:

$$I = \int_{-1}^{1} f(g)\,dg = \sum_{i=1}^{n} w_i f(g_i) \tag{17}$$

and n is ??

in which g_i are the n sample points at which we evaluate the integrand. These location are commonly referred to as Gauss points. The coefficients wi are the corresponding weights. Thus, to calculate I , we evaluate the integrand at each of several Gauss points g_i (the sample points) to obtain ordinates $f(g_i)$. We multiply each $f(g_i)$ by an appropriate weight and add.

$$I = \int_{-1}^{1} f(g)\,dg \approx \sum_{i=1}^{2} w_i f(g_i) = f(g_1)w_1 + f(g_2)w_2 \tag{18}$$

We determine the four unknowns (two Gauss points and two weights) by requiring that the formula give the exact result if the integrand is equal to the first four terms in a polynomial, hence

$$f(g) = 1; \int_{-1}^{1} f(g)\,dg = \int_{-1}^{1} 1\,dg = 2 \rightarrow w_1 + w_2 = 2$$

f(g)=g; $\int_{-1}^{1} f(g)\,dg = \int_{-1}^{1} g\,dg = 0 \rightarrow gw_1 + gw_2 = 0$

f(g)=g^2; $\int_{-1}^{1} f(g)\,dg = \int_{-1}^{1} g^2\,dg = 2/3 \rightarrow g^2 w_1 + g^2 w_2 = 2/3$

f(g)=g^3; $\int_{-1}^{1} f(g)\,dg = \int_{-1}^{1} g^3\,dg = 0 \rightarrow g^3 w_1 + g^3 w_2 = 0$

(19)

Solving the previous four simultaneous nonlinear equations, we obtain:
$w_1 = w_2$ and $g_1 = -1/\sqrt{(}3) = -g_2$
hence

$$I = \int_{-1}^{1} f(g)\,dg \approx \sum_{i=1}^{2} w_i f(g_i) = f(-1/\sqrt{(}3)) \cdot 1 + f(1/\sqrt{(}3)) \cdot 1 \tag{20}$$

Using Gauss integration with two points gives us the exact result up to a polynomial of order three. For other functions the results will be approximate.

BUT —

What happens in 2D?

When the region of integration is a 2x2 square, the one-dimensional Gauss formula easily extends to two dimensions.

$$I = \int_{-1}^{1}\int_{-1}^{1} f(g_1,g_2)\,dg_1dg_2 = \sum_{i=1}^{m}\sum_{i=1}^{n}(g_1,g_2)w_iw_j \tag{21}$$

Usually the number of integration points along the two directions is the same such that m = n

For example, let's evaluate the integral

$$I = \int_{-1}^{1}\int_{-1}^{1} f(g_1,g_2)\,dg_1dg_2 = \sum_{i=1}^{m}\sum_{i=1}^{n}(g_1,g_2)w_iw_j \tag{22}$$

Here is the 2 exercices of the assignement 3:

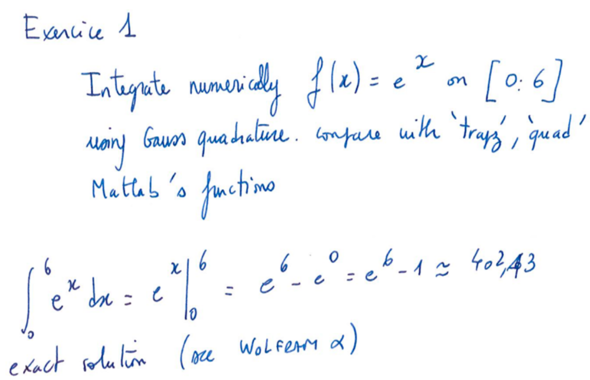

Fig. 12 Exercice 1.

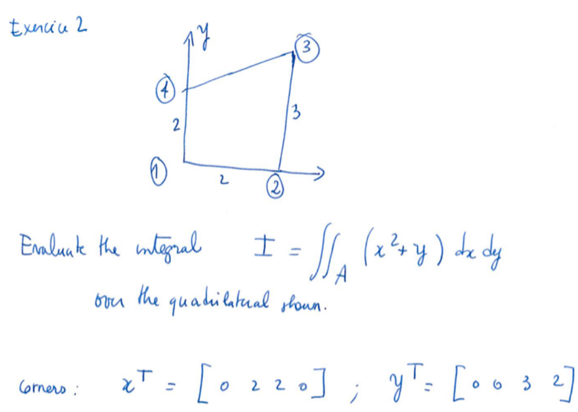

Fig. 13 Exercice 2.

and the important table for Gaussian integration (QUAD)

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	

Fig. 14 Quad1.

7 plane stress and strain

TBD

see slides

8 Deriving differential equation

TBD

see slides

9 The 4-node membrane element

TBD

see slides

10 Plates shells

TBD

see slides

11 Formal approach

Readers can find lot of example for Axially loaded bar using The Finite Element Solution. The Finite Element Method is a type of Galerkin method that has the following advantages:

- The functions φ_i are found in a systematic manner.
- The functions φ_i are chosen such that they can be used for arbitrary domains.
- The functions φ_i are piecewise polynomials.
- The functions φ_i are non-zero only on a small part of the domain.

As a result, computations can be done in a modular manner that is suitable for computer implementation.

The first step in the finite element approach is to divide the domain into 'elements' and ' nodes', i.e., to create the 'finite element mesh'.

Let us consider a simple situation and divide the rod into 3 elements and 4 nodes as shown in the next figure.

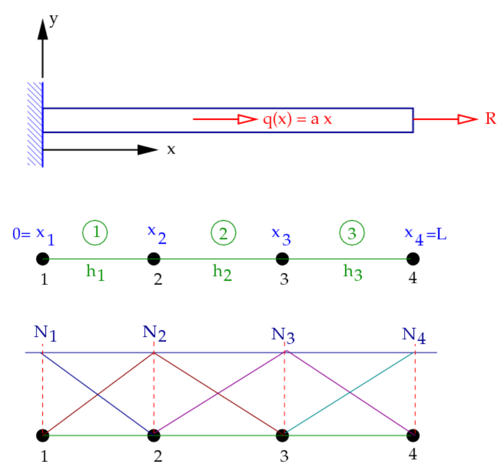

Fig. 15 Finite element mesh and basis functions for the bar .

The functions φ_i have special characteristics in finite element methods and are generally written as N_i and are called 'basis functions', ' shape functions', or ' interpolation functions'.

NOTE —

Therefore, we may write:

$$K_{ij} = \int_0^L \frac{dN_j}{dx} AE \frac{dN_i}{dx}\, dx \tag{23}$$

$$K_{ij} = \int_0^L B_i AE B_i \; dx \tag{24}$$

$$f_j = \int_0^L N_j \mathbf{q} \; dx + N_j R\big|_{x=L} \tag{25}$$

The finite element basis functions are chosen such that they have the following properties:

- The functions N_i are bounded and continuous.
- If there are n nodes, then there are n basis functions - one for each node. There are four basis functions for the mesh shown in Figure ?. function N_i is nonzero only on elements connected to node i.
- N_i is 1 at node i and zero at all other nodes.

Let us compute the values of K_{ij} for the three element mesh. We have:

$$\begin{aligned}
K_{ij} &= \int_0^L \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx \\
&= \int_0^{x_2} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx + \int_{x_2}^{x_3} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx + \int_{x_3}^{L} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx \\
&\equiv \int_{\Omega_1} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx + \int_{\Omega_2} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx + \int_{\Omega_3} \frac{dN_j}{dx} AE \frac{dN_i}{dx} \; dx
\end{aligned} \tag{26}$$

The components of $\mathbf{K}$ are:

$$
\begin{aligned}
K_{11} &= \int_{\Omega_1} \frac{dN_1}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_2} \frac{dN_1}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_3} \frac{dN_1}{dx} AE \frac{dN_1}{dx}\, dx \\
K_{12} &= \int_{\Omega_1} \frac{dN_2}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_2} \frac{dN_2}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_3} \frac{dN_2}{dx} AE \frac{dN_1}{dx}\, dx \\
K_{13} &= \int_{\Omega_1} \frac{dN_3}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_2} \frac{dN_3}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_3} \frac{dN_3}{dx} AE \frac{dN_1}{dx}\, dx \\
K_{14} &= \int_{\Omega_1} \frac{dN_4}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_2} \frac{dN_4}{dx} AE \frac{dN_1}{dx}\, dx + \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_1}{dx}\, dx \\
K_{22} &= \int_{\Omega_1} \frac{dN_2}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_2} \frac{dN_2}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_3} \frac{dN_2}{dx} AE \frac{dN_2}{dx}\, dx \\
K_{23} &= \int_{\Omega_1} \frac{dN_3}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_2} \frac{dN_3}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_3} \frac{dN_3}{dx} AE \frac{dN_2}{dx}\, dx \\
K_{24} &= \int_{\Omega_1} \frac{dN_4}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_2} \frac{dN_4}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_2}{dx}\, dx \\
K_{33} &= \int_{\Omega_1} \frac{dN_3}{dx} AE \frac{dN_3}{dx}\, dx + \int_{\Omega_2} \frac{dN_3}{dx} AE \frac{dN_3}{dx}\, dx + \int_{\Omega_3} \frac{dN_3}{dx} AE \frac{dN_3}{dx}\, dx \\
K_{34} &= \int_{\Omega_1} \frac{dN_4}{dx} AE \frac{dN_3}{dx}\, dx + \int_{\Omega_2} \frac{dN_4}{dx} AE \frac{dN_3}{dx}\, dx + \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_3}{dx}\, dx \\
K_{44} &= \int_{\Omega_1} \frac{dN_4}{dx} AE \frac{dN_4}{dx}\, dx + \int_{\Omega_2} \frac{dN_4}{dx} AE \frac{dN_4}{dx}\, dx + \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_4}{dx}\, dx \,.
\end{aligned}
\tag{27}
$$

The matrix **K** is symmetric, so we don't need to explicitly compute the other terms. From Figure 6, we see that N_1 is zero in elements 2 and 3, N_2 is zero in element 3, N_3 is zero in element 1, and N_4 is zero in elements 1 and 2. The same holds for dN_i/dx.

Therefore, the coefficients of the **K** matrix become :

$$K_{11} = \int_{\Omega_1} \frac{dN_1}{dx} AE \frac{dN_1}{dx}\, dx \;;\; K_{12} = \int_{\Omega_1} \frac{dN_2}{dx} AE \frac{dN_1}{dx}\, dx \;;\; K_{13} = 0 \;;\; K_{14} = 0 \tag{28}$$

$$K_{22} = \int_{\Omega_1} \frac{dN_2}{dx} AE \frac{dN_2}{dx}\, dx + \int_{\Omega_2} \frac{dN_2}{dx} AE \frac{dN_2}{dx}\, dx \;;\; K_{23} = \int_{\Omega_2} \frac{dN_3}{dx} AE \frac{dN_2}{dx}\, dx \;;\; K_{24} = 0 \tag{28}$$

$$K_{33} = \int_{\Omega_2} \frac{dN_3}{dx} AE \frac{dN_3}{dx}\, dx + \int_{\Omega_3} \frac{dN_3}{dx} AE \frac{dN_3}{dx}\, dx \;;\; K_{34} = \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_3}{dx}\, dx \tag{28}$$

$$K_{44} = \int_{\Omega_3} \frac{dN_4}{dx} AE \frac{dN_4}{dx}\, dx \,. \tag{28}$$

(28)

We can simplify our calculation further by letting N_k^e be the shape functions over an element e. For example, the shape functions over element 2 are N_1^2 and N_2^2 where the local nodes 1 and 2 correspond to global nodes 2 and 3, respectively. Then we can write, :

$$\begin{aligned}
&K_{11} = \int_{\Omega_1} \frac{dN_1^1}{dx} AE \frac{dN_1^1}{dx}\, dx \;;\; K_{12} = \int_{\Omega_1} \frac{dN_2^1}{dx} AE \frac{dN_1^1}{dx}\, dx \;;\; K_{13} = 0 \;;\; K_{14} = 0 \\
&K_{22} = \int_{\Omega_1} \frac{dN_2^1}{dx} AE \frac{dN_2^1}{dx}\, dx + \int_{\Omega_2} \frac{dN_1^2}{dx} AE \frac{dN_1^2}{dx}\, dx \;;\; K_{23} = \int_{\Omega_2} \frac{dN_2^2}{dx} AE \frac{dN_1^2}{dx}\, dx \;;\; K_{24} = 0 \\
&K_{33} = \int_{\Omega_2} \frac{dN_2^2}{dx} AE \frac{dN_2^2}{dx}\, dx + \int_{\Omega_3} \frac{dN_1^3}{dx} AE \frac{dN_1^3}{dx}\, dx \;;\; K_{34} = \int_{\Omega_3} \frac{dN_2^3}{dx} AE \frac{dN_1^3}{dx}\, dx \\
&K_{44} = \int_{\Omega_3} \frac{dN_2^3}{dx} AE \frac{dN_2^3}{dx}\, dx \,.
\end{aligned} \tag{29}$$

Let K_{kl}^e be the part of the value of K_{ij} that is contributed by element e. The indices kl are local and the indices ij are global. Then, :

$$\begin{aligned}
&K_{11} = K_{11}^1 \;;\; K_{12} = K_{12}^1 \;;\; K_{13} = 0 \;;\; K_{14} = 0 \\
&K_{22} = K_{22}^1 + K_{11}^2 \;;\;\; K_{23} = K_{12}^2 \;;\;\; K_{24} = 0 \\
&K_{33} = K_{22}^2 + K_{11}^3 \;;\; K_{34} = K_{12}^3 \\
&K_{44} = K_{22}^3
\end{aligned} \tag{30}$$

We can therefore see that if we compute the stiffness matrices over each element and assemble them in an appropriate manner, we can get the global stiffness matrix **K**.

For our problem, if we consider an element e with two nodes, the local hat shape functions have the form :

$$N_1^e(\mathbf{x}) = \frac{\mathbf{x}_2 - \mathbf{x}}{h_e} \;;\;\; N_2^e(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{x}_1}{h_e} \tag{31}$$

where h_e is the length of the element.

Then, the components of the "' element stiffness matrix"' are :

$$
\begin{aligned}
K_{11}^e &= \int_{x_1}^{x_2} \frac{dN_1^e}{dx} AE \frac{dN_1^e}{dx}\, dx = \int_{x_1}^{x_2} \left(-\frac{1}{h_e}\right) AE \left(-\frac{1}{h_e}\right)\, dx = \frac{AE}{h_e} \\
K_{12}^e &= \int_{x_1}^{x_2} \frac{dN_2^e}{dx} AE \frac{dN_1^e}{dx}\, dx = \int_{x_1}^{x_2} \left(\frac{1}{h_e}\right) AE \left(-\frac{1}{h_e}\right)\, dx = -\frac{AE}{h_e} \\
K_{22}^e &= \int_{x_1}^{x_2} \frac{dN_2^e}{dx} AE \frac{dN_2^e}{dx}\, dx = \int_{x_1}^{x_2} \left(\frac{1}{h_e}\right) AE \left(\frac{1}{h_e}\right)\, dx = \frac{AE}{h_e}
\end{aligned} \tag{32}
$$

In matrix form, :

$$
\mathbf{K}^e = \frac{AE}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{33}
$$

The components of the "' global stiffness matrix"' are :

$$
\begin{aligned}
& K_{11} = \frac{AE}{h_1}\,;\;\; K_{12} = -\frac{AE}{h_1}\,;\;\; K_{13} = 0\,;\;\; K_{14} = 0 \\
& K_{22} = \frac{AE}{h_1} + \frac{AE}{h_2}\,;\;\; K_{23} = -\frac{AE}{h_2}\,;\;\; K_{24} = 0 \\
& K_{33} = \frac{AE}{h_2} + \frac{AE}{h_3}\,;\;\; K_{34} = -\frac{AE}{h_3} \\
& K_{44} = \frac{AE}{h_3}
\end{aligned} \tag{34}
$$

In matrix form, :

$$
\mathbf{K} = AE \begin{bmatrix} \frac{1}{h_1} & -\frac{1}{h_1} & 0 & 0 \\ & \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & 0 \\ & & \frac{1}{h_2} + \frac{1}{h_3} & -\frac{1}{h_3} \\ \text{Symm.} & & & \frac{1}{h_3} \end{bmatrix} \tag{35}
$$

Similarly, for the 'load vector' **f**, we have :

$$
\begin{aligned}
f_j &= \int_0^L N_j \mathbf{q}\, dx + N_j R\big|_{x=L} \\
&= \int_{\Omega_1} N_j \mathbf{q}\, dx + \int_{\Omega_2} N_j \mathbf{q}\, dx + \int_{\Omega_3} N_j \mathbf{q}\, dx + N_j R\big|_{x=L}
\end{aligned} \tag{36}
$$

The components of the load vector are :

$$
\begin{aligned}
f_1 &= \int_{\Omega_1} N_1\mathbf{q}\,dx + \int_{\Omega_2} N_1\mathbf{q}\,dx + \int_{\Omega_3} N_1\mathbf{q}\,dx + N_1R|_{x=L} \\
f_2 &= \int_{\Omega_1} N_2\mathbf{q}\,dx + \int_{\Omega_2} N_2\mathbf{q}\,dx + \int_{\Omega_3} N_2\mathbf{q}\,dx + N_2R|_{x=L} \\
f_3 &= \int_{\Omega_1} N_3\mathbf{q}\,dx + \int_{\Omega_2} N_3\mathbf{q}\,dx + \int_{\Omega_3} N_3\mathbf{q}\,dx + N_3R|_{x=L} \\
f_4 &= \int_{\Omega_1} N_4\mathbf{q}\,dx + \int_{\Omega_2} N_4\mathbf{q}\,dx + \int_{\Omega_3} N_4\mathbf{q}\,dx + N_4R|_{x=L}
\end{aligned}
\tag{37}
$$

Once again, since N_1 is zero in elements 2 and 3, N_2 is zero in element 3, N_3 is zero in element 1, and N_4 is zero in elements 1 and 2, we have :

$$
\begin{aligned}
f_1 &= \int_{\Omega_1} N_1\mathbf{q}\,dx + N_1R|_{x=L} \\
f_2 &= \int_{\Omega_1} N_2\mathbf{q}\,dx + \int_{\Omega_2} N_2\mathbf{q}\,dx + N_2R|_{x=L} \\
f_3 &= \int_{\Omega_2} N_3\mathbf{q}\,dx + \int_{\Omega_3} N_3\mathbf{q}\,dx + N_3R|_{x=L} \\
f_4 &= \int_{\Omega_3} N_4\mathbf{q}\,dx + N_4R|_{x=L}
\end{aligned}
\tag{38}
$$

Now, the boundary $\mathbf{x} = L$ is at node 4 which is attached to element 3. The only non-zero shape function at this node is N_4. Therefore, we have :

$$
\begin{aligned}
f_1 &= \int_{\Omega_1} N_1\mathbf{q}\,dx \\
f_2 &= \int_{\Omega_1} N_2\mathbf{q}\,dx + \int_{\Omega_2} N_2\mathbf{q}\,dx \\
f_3 &= \int_{\Omega_2} N_3\mathbf{q}\,dx + \int_{\Omega_3} N_3\mathbf{q}\,dx \\
f_4 &= \int_{\Omega_3} N_4\mathbf{q}\,dx + N_4R|_{x=L}
\end{aligned}
\tag{39}
$$

In terms of element shape functions, the above equations can be written as :

$$
\begin{aligned}
f_1 &= \int_{\Omega_1} N_1^1\mathbf{q}\,dx = f_1^1 \\
f_2 &= \int_{\Omega_1} N_2^1\mathbf{q}\,dx + \int_{\Omega_2} N_1^2\mathbf{q}\,dx = f_2^1 + f_1^2 \\
f_3 &= \int_{\Omega_2} N_2^2\mathbf{q}\,dx + \int_{\Omega_3} N_1^3\mathbf{q}\,dx = f_2^2 + f_1^3 \\
f_4 &= \int_{\Omega_3} N_2^3\mathbf{q}\,dx + N_2^3R\big|_{x=L} = f_2^3 + R
\end{aligned}
\tag{40}
$$

The above shows that the global load vector can also be assembled from the element load vectors if we use finite element shape functions.

Using the linear shape functions discussed earlier and replacing $\mathbf{q}$ with $a\mathbf{x}$, the components of the element load vector $\mathbf{f}^e$ are :

$$\begin{aligned} f_1^e &= \int_{x_1}^{x_2} N_1^e a\mathbf{x}\, dx = \int_{x_1}^{x_2} \left(\frac{x_2 - x}{h_e}\right) a\mathbf{x}\, dx = \frac{a}{h_e}\left(\frac{x_2(x_2^2 - x_1^2)}{2} - \frac{x_2^3 - x_1^3}{3}\right) \\ f_2^e &= \int_{x_1}^{x_2} N_2^e a\mathbf{x}\, dx = \int_{x_1}^{x_2} \left(\frac{x - x_1}{h_e}\right) a\mathbf{x}\, dx = -\frac{a}{h_e}\left(\frac{x_1(x_2^2 - x_1^2)}{2} - \frac{x_2^3 - x_1^3}{3}\right) \end{aligned} \tag{41}$$

In matrix form, the ' element load vector' is written :

$$\mathbf{f}^e = \frac{a}{h_e}\begin{bmatrix} \dfrac{x_2(x_2^2 - x_1^2)}{2} - \dfrac{x_2^3 - x_1^3}{3} \\ \dfrac{x_2^3 - x_1^3}{3} - \dfrac{x_1(x_2^2 - x_1^2)}{2} \end{bmatrix} \tag{42}$$

Therefore, the components of the 'global load vector' are :

$$\begin{aligned} f_1 &= \frac{a}{h_1}\left(\frac{x_2(x_2^2 - x_1^2)}{2} - \frac{x_2^3 - x_1^3}{3}\right) \\ f_2 &= \frac{a}{h_1}\left(\frac{x_2^3 - x_1^3}{3} - \frac{x_1(x_2^2 - x_1^2)}{2}\right) + \frac{a}{h_2}\left(\frac{x_3(x_3^2 - x_2^2)}{2} - \frac{x_3^3 - x_2^3}{3}\right) \\ f_3 &= \frac{a}{h_2}\left(\frac{x_3^3 - x_2^3}{3} - \frac{x_2(x_3^2 - x_2^2)}{2}\right) + \frac{a}{h_3}\left(\frac{x_4(x_4^2 - x_3^2)}{2} - \frac{x_4^3 - x_3^3}{3}\right) \\ f_4 &= \frac{a}{h_3}\left(\frac{x_4^3 - x_3^3}{3} - \frac{x_3(x_4^2 - x_3^2)}{2}\right) + R \end{aligned} \tag{43}$$

Recall that we assumed that the displacement can be written as :

$$\mathbf{u}_h(\mathbf{x}) = a_1\varphi_1(\mathbf{x}) + a_2\varphi_2(\mathbf{x}) + \cdots + a_n\varphi_n(\mathbf{x}) = \sum_{i=1}^{n} a_i\varphi_i(\mathbf{x}) \ . \tag{44}$$

If we use finite element shape functions, we can write the above as :

$$\mathbf{u}_h(\mathbf{x}) = a_1\, N_1(\mathbf{x}) + a_2\, N_2(\mathbf{x}) + \cdots + a_n\, N_n(\mathbf{x}) = \sum_{i=1}^{n} a_i\, N_i(\mathbf{x}) \tag{45}$$

where n is the total number of nodes in the domain. Also, recall that the value of N_i is 1 at node i and zero elsewhere. Therefore, we have :

$$
\begin{aligned}
u_1 &:= \mathbf{u}_h(\mathbf{x}_1) = a_1\, N_1(\mathbf{x}_1) + a_2\, N_2(\mathbf{x}_1) + \cdots + a_n\, N_n(\mathbf{x}_1) = a_1 \\
u_2 &:= \mathbf{u}_h(\mathbf{x}_2) = a_1\, N_1(\mathbf{x}_2) + a_2\, N_2(\mathbf{x}_2) + \cdots + a_n\, N_n(\mathbf{x}_2) = a_2 \\
&\vdots \\
u_n &:= \mathbf{u}_h(\mathbf{x}_n) = a_1\, N_1(\mathbf{x}_n) + a_2\, N_2(\mathbf{x}_n) + \cdots + a_n\, N_n(\mathbf{x}_n) = a_n
\end{aligned} \tag{46}
$$

Therefore, the 'trial function' can be written as :

$$
\mathbf{u}_h(\mathbf{x}) = u_1\, N_1(\mathbf{x}) + u_2\, N_2(\mathbf{x}) + \cdots + u_n\, N_n(\mathbf{x}) = \sum_{i=1}^{n} u_i\, N_i(\mathbf{x}) \tag{47}
$$

where u_i are the 'nodal displacements'.

If all the elements are assumed to be of the same length h, the finite element system of equations ($\mathbf{Ka} = \mathbf{f}$) can then be written as :

$$
\frac{AE}{h}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \frac{a}{h}\begin{bmatrix} \left(\frac{x_2(x_2^2-x_1^2)}{2} - \frac{x_2^3-x_1^3}{3}\right) \\ \left(\frac{x_2^3-x_1^3}{3} - \frac{x_1(x_2^2-x_1^2)}{2}\right) + \left(\frac{x_3(x_3^2-x_2^2)}{2} - \frac{x_3^3-x_2^3}{3}\right) \\ \left(\frac{x_3^3-x_2^3}{3} - \frac{x_2(x_3^2-x_2^2)}{2}\right) + \left(\frac{x_4(x_4^2-x_3^2)}{2} - \frac{x_4^3-x_3^3}{3}\right) \\ \left(\frac{x_4^3-x_3^3}{3} - \frac{x_3(x_4^2-x_3^2)}{2}\right) + R\frac{h}{a} \end{bmatrix} \tag{48}
$$

To solve this system of equations we have to apply the 'essential boundary condition' $\mathbf{u} = 0$ at $\mathbf{x} = 0$. This is equivalent to setting $u_1 = 0$. The reduced system of equations is :

$$
\frac{AE}{h}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \frac{a}{h}\begin{bmatrix} \left(\frac{x_2^3-x_1^3}{3} - \frac{x_1(x_2^2-x_1^2)}{2}\right) + \left(\frac{x_3(x_3^2-x_2^2)}{2} - \frac{x_3^3-x_2^3}{3}\right) \\ \left(\frac{x_3^3-x_2^3}{3} - \frac{x_2(x_3^2-x_2^2)}{2}\right) + \left(\frac{x_4(x_4^2-x_3^2)}{2} - \frac{x_4^3-x_3^3}{3}\right) \\ \left(\frac{x_4^3-x_3^3}{3} - \frac{x_3(x_4^2-x_3^2)}{2}\right) + R\frac{h}{a} \end{bmatrix} \tag{49}
$$

This system of equations can be solved for u_2, u_3, and u_4. Let us do that.

Assume that A, E, L, a, and R are all equal to 1. Then $x_1 = 0$, $x_2 = 1/3$, $x_3 = 2/3$, $x_4 = 1$, and $h = 1/3$. The system of equations becomes :

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0.037 \\ 0.074 \\ 0.383 \end{bmatrix} \qquad \Longrightarrow \qquad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.494 \\ 0.951 \\ 1.333 \end{bmatrix} \tag{50}$$

From the above, it is clear that the displacement field within an element e is given by :

$$\mathbf{u}^e = u_1^e N_1^e(\mathbf{x}) + u_2^e N_2^e(\mathbf{x}) \ . \tag{51}$$

Therefore, the strain within an element is :

$$\varepsilon^e = \frac{\partial \mathbf{u}^e}{\partial x} = u_1^e \frac{\partial N_1^e}{\partial x} + u_2^e \frac{\partial N_2^e}{\partial x} \ . \tag{52}$$

In matrix notation, :

$$\varepsilon^e = \mathbf{B}^e \mathbf{u}^e = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial x} \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \begin{bmatrix} -\frac{1}{h} & \frac{1}{h} \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} \tag{53}$$

The stress in the element is given by :

$$\sigma^e = E\varepsilon^e \ . \tag{54}$$

For our discretization, the element stresses are :

$$\begin{aligned} \sigma^1 &= 1.48 \\ \sigma^2 &= 1.37 \\ \sigma^3 &= 1.15 \end{aligned} \tag{55}$$

A plot of this solution is shown in the next Figure

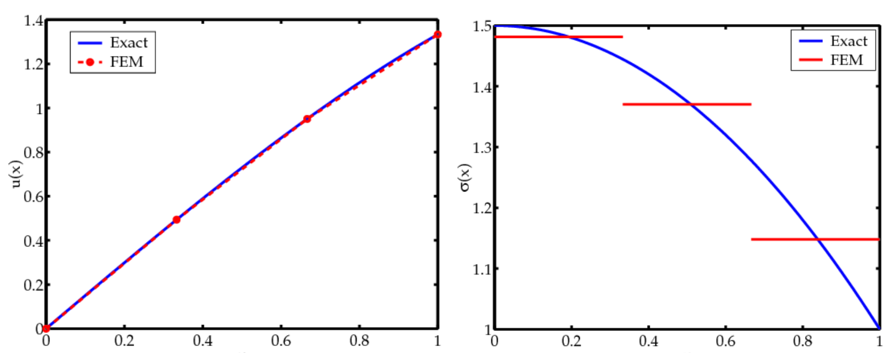

Fig. 16 FEM vs exact solutions for displacements/stress of an axially loaded bar .

The finite element code (Matlab) must be completed in assignement 1.

```
function AxialBarFEM
%Comment please with the right word
%Copy paste part of the code in the ?? at the right place
B*u
[-1/h 1/h]
d = inv(Kred)*fred
(A*E/h)*[[1 -1];[-1 1]]
E*eps



%
%
%
 A = 1.0;
 L = 1.0;
 E = 1.0;
 a = 1.0;
 R = 1.0;
%
%
%

 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

%
%
%

 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);



%
%
%

 d = ??;

 dsol = [0 d'];

 fsol = K*dsol';
 sum(fsol)




%
%
%
 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 = ??;
```

12 Patran/Nastran/What else?

NASA STRuctural ANalysis, founded in the 1960s, marketed in 1969 (COSMIC) is the most widely used EF program in the aerospace industry. MSC-Nastran (Macneal Swendler Corp.) is the most popular version. The software will create the stiffness matrix K, the mass matrix M from user input (E, nu, rho, formulation and type of elements - bar -beam-plate-shell, plus thicknesses). Then Nastran will launch the resolution of the mechanical problem (inversion of K in the static case, eigenvalue problem for modal / buckling analysis).

PATRAN: pre and post-processor of MSC.Software for NASTRAN (similar to a CAD program).
NASTRAN: calculation code to solve problems by the finite element method. NASTRAN is not a graphical tool, to perform a calculation it is necessary to create an ASCII data file which describes the problem = it is the role of Patran.

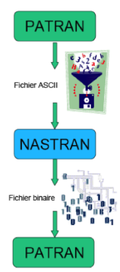

Fig. 17 Chaine de calcul patran/nastran .

13 Hello Doc?

First, the Linear Static User Guide is a wealth of information: Type of elements, materials, modeling guides, Model verification, organization of a Nastran dataset ...

You can find relevant information about: Nastran: MSC.Nastran 2016 - Quick Reference Guide - MSC.Software that can be downloaded: Type on google ”MSC nastran 2016 quick reference guide 2016”. You will find it on the LMS on the SM201 course page, along with a number of other official supports of the 2016 version. Regarding the use of Patran, you can online help directly in the software .

MSC does not provide information on the design of its elements, you can find documentation on the more theoretical part:

- Modélisation des structures – Calcul par éléments finis, Jean-Charles Craveur, Dunod, 2e édition, 2001.
- Analyse des structures par éléments finis de J.-f. Imbert, 1991, Cépaduès, SUP'AERO
- "The Finite Element Method third edition", O.C. Zienkiewitch, McGraw-Hill, London, 1977
- "Concepts and Applications of Finite Element Analysis", Robert D. Cook and al., , John Wiley and Sons, Inc., 2001.

To finish on internet, we can find a number of examples Nastran, Patran on the site MSC.

14 Units system?

The FE software does not have a predefined unit system. They are universal. It is the user who defines his system of units and ensures its consistency. A consistent system defines the units of all properties required for the model: Length, Area, Volume, Section Moment Force, Moment, Acceleration, Mass, Density Young Modulus, Constraint, etc. In Nastran there are no default units. It is up to the user to always work with textbf consistent units: Patran only works with numbers !!! For structural mechanics problems the user chooses the units of length and force. For problems where time intervenes, it will have to choose the unit of time.

SI Units : Lenght -(meters), Force - (Newtons) Mass (Kg)
Then one will analyze constraints in Pa and the density must be specified in Kg / m^3 to obtain s^-1 Hz

Lenght-(mm), Force - (Newtons) Mass (Kg)
Then we will analyze constraints in MPa and the density must be specified in t / mm^3 to obtain s^-1 or Hz

15 Types of analysis

Each type of analysis available is called a "solution sequence" (SOL). The most famous "SOL" are:

- **101 - Linear Static**

- **103 - Modal**
- **105 - Linear Buckling**
- 106 - Static nonlinear
- 108 - Frequency response (direct)
- 109 - Transient response (direct)
- 111 - Frequency response (modal)
- 112 - Transient response (modal)
- 200 - Sensitivity and design analysis
- 400 - The new BIG solver (implicit)
  Those used in this course are in bold.

16 Let's start...

- Copy the Patran and Nastran shortcuts to the desktop.
- Create a working folder on the local disk (textbf not on the network)
- Specify the path of your working folder to Patran when you create the 1st template.
- Sotcker all templates to use in this directory

All I/O files generated by Patran / Nastran will be stored in this working directory.

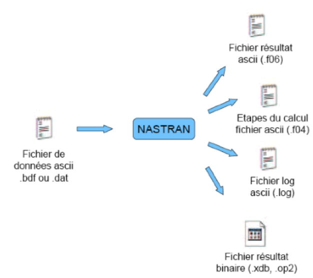

Fig. 18 Fichiers générés .

.f06
Results in ascii format. Also indicates errors and warnings, always look at it (look for the words ”fatal” and ”warning” in this file).

.f04
Resumes the steps of resolution (matrix assembly, elimination Blocked d.d.l, ...) with their execution time.

.log
Summarizes the configuration of the calculation (paths of files and executables, configuration of machine, ...)

.xdb, .op2
Results files in binary format. The .xdb and .op2 are readable by Patran But we can also create op2 readable by IDEAS, ...

.ses, .jou
All The nastran / patran commands are stored. This is the equivalent of macro under excel for example.

17 Step by step

CAD on Patran

For simple models it is entirely possible to create entirely the model under patran. It is then enough to follow from left to right the icons of the horizontal menu.

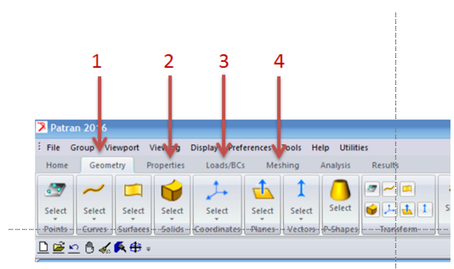

Fig. 19 Pre-treatment steps on Patran .

- Step 1: Creating the geometry: point, curve, surface, set of axis
- Step 2: Creation of the material and physic properties:
  Plate thickness, beam section, and assignment to geometric entities (line, surface, ..)

- Step 3: Definition of the boundary conditions in movement and loading
  Creation then assignment to geometric entities which will be the supports of the nodes and elements on which these BCs will bear. It is possible to create different load cases from different BCs.

- Step 4: Creation of the mesh:
  Control of the mesh by the Meshseeds definition, then creation of the mesh with association to the physical properties and materials desired. Do not forget to do equivalence to ensure the correct junction of the meshes between the different entities that have been meshed.

Starting the computation

- Launching the computation.

  In the section **Analysis** :
  `analysis > entire model > full run` The following topics are used for:
  * Translation parameters: specific parameters of the calculation, choice on the creation of the file (long or short formats, what output files, ...)
  * Typical solution: choice of solution type (static, buckling, or modal analysis)
  * Direct text input: section to add specific commands (options for "home" solutions for example)
  * Subcases: choice of what we will ask in computation results for each subcase (constraints, displacements, efforts, ...)
  * Subcase select: choice of load cases to be used for the calculation

- Checking

  In the compute directory look for the NASTRAN calculation files and open the file `.f06`
  Search in this file the term "**FATAL**", if it is present in the file: the calculation does not pass it needs to correct the model according to the information contained in this file
  * Also search for "**warning**", it may indicate problems when calculating
  * Search in file "**epsilon**", there is a residue for the calculation, if the value is too high (greater than 1e-4) it can be assumed that there is a problem in the mesh and Must be analyzed with great caution.

  * Search the term **SPCFORCE RESULTANT**: one finds there the sum of the reactions of support, it is necessary to find the sum of the forces applied. Always verify that the sum of the forces applied is equal to the sum of the forces measured at the boundary conditions (mechanical equilibrium).

* Request the **OLOADs** Result (load applied) and the **SPCFORCEs** Result (Reaction to embedding points):
If all these points are validated, you can look at the results with confidence.

- Postprocessing

Before the post-processing under Patran we must reread the result file generated by Nastran (in our case the .xdb file): * In the "Analysis" section: action "attach xdb" By default, Patran averaged the constraints between two neighboring elements to create a continuous pattern. It is a "trap" because this visualization can smooth local peaks of constraints and lead to an underestimation of the value in an element. To avoid this, you should choose not to use the "**Fringe**" option in an option and to see the values at the integration points.

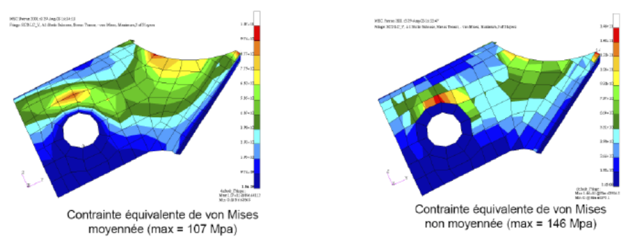

Fig. 20 Fringe? .

Finally, in the "**Results**" section:
* **Quick Plot**: fast isovate plot (suitable for displacements but limited for constraints)
* **Deformation**: view the distorted
* **Fringe**: a more advanced isov value plot (useful for viewing non-averaged constraints)
* **Marker**: plotting forces in the form of tensors (arrows)
* **Cursor**: Lets you know either a Scalar, Vector or Tensor value
* **Graphs**: Allows you to draw a nodal or elementary result along a path
* **Report**: display results in a file for use in excel for example.

Case of triangles: The calculations made with the linear interpolation triangle elements must lead to the following remark: constraints are constant in a triangle element. This observation is quite normal: the stress is calculated from the interpolation gradient, and for the triangles with linear interpolation the interpolation gradient is constant:

$N(x,y) = [1, -x, -y, x, y]$

$\nabla N(x,y) = [-1,1,0;-1,0,1]$

18 Finally, some tips

* Pay attention to the use of triangles (and tetrahedrons for 3D meshes) which allow only a uniform field of stresses in the element (cf. above). In sensitive areas it is necessary to mesh finer
* Always take a critical look at the results of a calculation, they only reflect the instructions you gave to the calculation software

Patran and Nastran allow to display on the model of the structure the maps of visualization of the fields of constraints. The maps giving the variation of the stress components (sx, sy ...) give only a partial image and do not allow any interpretation on the risks of exceeding Re. Moreover, they depend on the reference point in which they are Expressed. The equivalent constraint of Von Mises is a combination of these components and does not depend on the benchmark.

Only the image of the field of the equivalent constraints of Von Misès makes it possible to clearly visualize the zone (s) subjected to the risk of plastification. Be careful, however, the Von Mises constraint does not apply to composite materials.

The display of stresses or deformations requires special attention insofar as Patran proposes various options, in particular for calculating averages.

Patran uses a number of options by ”**default**” which can show values different from those calculated by the solver. If we consider that the values provided by the solver are the reference values, it is good to know how to find these values in Patran. Patran’s online documentation can help.

`Book: Results Postprocessing > Chapter: 13 Numerical Methods >`