Sorbonne Université

Introduction to the finite element method in 1D

Authors :
Alexandre RICHARD
Valentin Duvivier

Supervisors :
Maurini Corrado
Angela Vincenti

December 23, 2020

Contents

1 Abstract **2**

2 Introduction to the 2D structure **3**
2.1 Material . . . 3
2.2 Domain . . . 3
2.3 Equations . . . 3

3 Equations of the problem - Case A **4**
3.1 Variational formulation - Energy and Admissible fields . . . 4
3.2 Weak form - stationary condition of the energy . . . 5
3.3 Strong formulation . . . 7

4 Discretisation **8**
4.1 Parametrical representation . . . 8
4.2 Displacement interpolation . . . 12
4.3 Alternate modeling - T6 . . . 13

5 Strains, stiffness matrix and nodal forces **17**
5.1 Strains . . . 17
5.2 Elementary stiffness matrix . . . 20
5.3 Elementary right hand member . . . 21

6 Assembly **25**

7 Solution **29**

8 Conclusion **32**

References **33**

1 Abstract

Finite element method (FEM) is one of the most commonly used numerical technique to find approximate solution for various problems in the field of mechanical engineering. It's based on creating a model of a structure through a mesh, to then run numerical analyses on it by considering a set of forces and displacements to which the model as to fit to be physically reliable.

In this paper we are presenting the overview of solving mechanical 2D structure problem by Finite Element Method (FEM). We will gradually go from a work on an elementary element to the building of the entire structure and the analysis of it (FEA).

Doing so, we will present the process used to model complex structure by extrapolation from simpler element and discretisation. We will then provide several theoretical results that one can compare to numerical one through the code provided. Please then note that a process similar to the following one as be run through a jupyter notebook, doing the link between theory from this paper and numerical results.

This paper is a combined work from 4 people : Muhammad Faizoud Din ACKBARALLY, Jiayu WANG, Valentin DUVIVIER, Alexandre RICHARD.

Indeed, even though this paper is a final product of only the last two mentioned people, a part of the results showed here has been obtained by a group of 4, group that has later been splitted into 2 sub-groups.

2 Introduction to the 2D structure

2.1 Material

The part is made of an isotropic homogeneous linear elastic material. Its elastic rigidity tensor is referred to as $\mathbb{A}$, while $\mathbb{S}$ stands for its elastic compliance tensor and finally its thermal expansion coefficient is noted α_T.

2.2 Domain

The structure at stake in this problem is called Ω. It has a rectangle shape whose dimension ℓ in direction $\underline{e}_3$ is way larger than both dimensions H and $2L$ of its rectangular section in plane $(\underline{e}_1, \underline{e}_2)$.

This domain is therefore to be reduced to the 2D domain $x1 \times x2 \equiv [-L, L] \times [0, H]$.

At last symmetries of the problem in terms of geometry, material properties and load allow to perform the study on only half of the part, namely of subpart located in the half space $x_1 > 0$.

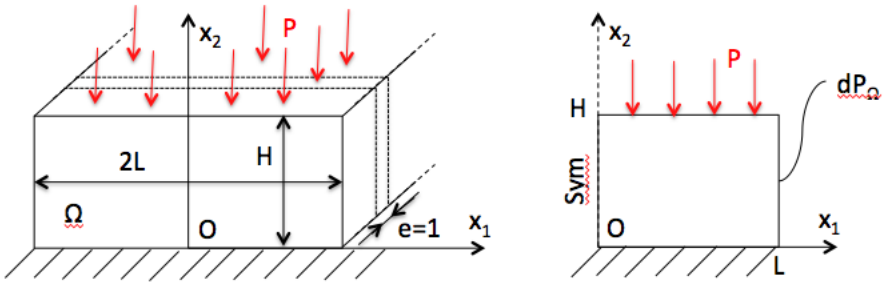


Figure 1: Schematic representation of the structure studied and its pressure load

2.3 Equations

The part lays on a rigid horizontal stand. The part is supposed to remain in contact with the stand all along its lower plane surface located at $(x_2 = 0)$.

The part is first submitted to gravity as well as a pressure load on its upper boundary $(x_2 = H)$, as represented on Figure 1.

Contact between the part and the stand can supposedly be characterized by an elastic restoring force, i.e. a surface density force is applied in the plane of the contact surface in a direction opposed to movement : $\underline{F} = -k_e \underline{u}_T$, where $\underline{F}$ is the restoring force, $\underline{u}_T = u_1 \underline{e}_1$ is the tangential component of the displacement vector $\underline{u}$, and k is a positive constant $(k > 0)$.

We will develop this point by introducing 2 possible cases for the FEA.

From the material subsection and this one, one sees that Hooke's law is to be applied. Moreover, we assume to be under HPP. These hypotheses will later lead to the admissible fields $\mathcal{U}_{ad}$ and Σ_{ad}.

3 Equations of the problem - Case A

3.1 Variational formulation - Energy and Admissible fields

When working on 2D case, one consider an extra dimension with respect to the 1D case. Doing so changes as well the system's properties such as its energy. Therefore, the variational formulation being defined as the total potential energy, it will be affected and its shape will be modified.

In fact, this new variational formulation will be more general (one might then get back to the 1D case through simplifications) :

$$\begin{aligned} I : \underline{u} \in U_{ad} &\to \mathbb{R} \\ \underline{u} &\to I(\underline{u}) \end{aligned}$$

$$I(\underline{u}) = \underbrace{\int_\Omega \phi(\underline{\underline{\epsilon}}(\underline{u}))dV}_{\text{kinematic energy deformation}} \quad \underbrace{-\int_\Omega \underline{f}\cdot\underline{u}dV - \int_{\partial\Omega}\underline{T_d}\cdot\underline{u}dS}_{\text{- exterior work}} \quad \forall \underline{u} \in U_{ad}\}$$

At this stage, one must introduce the 2 cases mentioned during introduction. Indeed, we have :

- Case A : there is no restoring force in $x_2 = 0$ but rather an embedded connection, avoiding displacement along e_2 ;

- Case B : we replace the embedded connection by a combination between a condition of no vertical movement and a restoring force $\underline{F} = -k_e\underline{u}_T$ in $x_2 = 0$.

These conditions mainly on the displacement here will allow us to get a unique solution when solving the problem. At sight of these contradictory cases, we are to run separately two analyses on each case. Indeed, they both go together with a different set of admissible fields and obviously the obtained results will thereby defer.

To summarize, while case B will be treated as an extra, we will in the most part of this report **only treat case A**.

Associated kinematically admissible field :

$$\begin{aligned} \mathcal{U}_{ad} = \mathcal{U}^0_{ad} = \{\underline{u} : \mathbf{x} \in \Omega \subset \mathbb{R}^2 \to \underline{u}(\mathbf{x}) \in \mathbb{R}^2, \text{smooth} \ \mid \ [|\underline{u}(\mathbf{x})|] = \underline{0}, \\ u_1 = 0 \ \ \forall \mathbf{x} \in \Sigma_{x_1=0} \\ u_2 = 0 \ \ \forall \mathbf{x} \in \Sigma_{x_2=0}\} \end{aligned}$$

In details :

$$\begin{cases} \phi(\underline{\underline{\epsilon}}) = \frac{1}{2}(\underline{\underline{\epsilon}}(\underline{u}) : \underline{\underline{\underline{\underline{\mathbb{A}}}}}) : \underline{\underline{\epsilon}}(\underline{u}) = \frac{1}{2}\underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{u})) : \underline{\underline{\epsilon}}(\underline{u}) & \text{(strain energy density)} \\ \underline{f} = \rho \underline{g} \quad \forall \underline{x} \in \Omega & \text{(volumic forces)} \\ \underline{T_d} = -P\underline{e_2} \quad \forall \underline{x} \in \Sigma_{x_2=H} & \text{(areal forces)} \end{cases}$$

with Ω the domain and $\partial\Omega = \Sigma_{x_1=0} \cup \Sigma_{x_1=L} \cup \Sigma_{x_2=0} \cup \Sigma_{x_2=H}$ its edges (see figure **??**) :

$$\begin{cases} \Omega = (x_1, & x_2) & \in [0, L] \times [0, H] \\ \Sigma_{x_1=0} = & x_2 & \in [0, H] \\ \Sigma_{x_1=L} = & x_2 & \in [0, H] \\ \Sigma_{x_2=0} = & x_1 & \in [0, L] \\ \Sigma_{x_2=H} = & x_1 & \in [0, L] \end{cases}$$

Note : as we are in 2D, the domain is a surface while the boundaries are 1D domains (lines).

Furthermore, as the material is isotropic:

$$\begin{aligned} \underline{\underline{\sigma}}(\underline{x}) &= \lambda(tr(\underline{\underline{\epsilon}}))\underline{\underline{\mathbb{1}}} + 2\mu\underline{\underline{\epsilon}} \\ \underline{\underline{\epsilon}}(\underline{u}) &= \frac{1}{2}(\underline{\underline{\nabla u}} + \underline{\underline{\nabla u}}^T) \\ \phi(\underline{\underline{\epsilon}}) = \frac{1}{2}\underline{\underline{\sigma}} : \underline{\underline{\epsilon}}(\underline{u}) &= \frac{\lambda}{2}(tr(\underline{\underline{\epsilon}}))^2 + \mu\ \underline{\underline{\epsilon}} : \underline{\underline{\epsilon}} \end{aligned}$$

and moreover :

$$\begin{aligned} \Sigma_{ad} = \{\underline{\underline{\tau}} : \mathbf{x} \in \Omega \subset \mathbb{R}^2 \to \underline{\underline{\tau}}(\mathbf{x}) \in \mathbb{R}^2, \text{smooth} \ \mid \ [|\underline{\underline{\tau}}(\mathbf{x})|] \cdot \underline{n}(\mathbf{x}) = \underline{0}, \\ \underline{\underline{\tau}}(\underline{x}) = \lambda(tr(\underline{\underline{\epsilon}}))\underline{\underline{\mathbb{1}}} + 2\mu\underline{\underline{\epsilon}} \\ \underline{div}(\underline{\underline{\tau}}) + \rho\underline{f} = \rho\underline{\gamma} = \underline{0} \\ \underline{\underline{\epsilon}}(\underline{u}) = \frac{1}{2}(\underline{\underline{\nabla u}} + \underline{\underline{\nabla u}}^T) \\ \\ \underline{\underline{\tau}}(\mathbf{x}) \cdot \underline{n}(\mathbf{x}) = -P\underline{e_2} \quad \forall \mathbf{x} \in \Sigma_{x_2=H} \\ \underline{\underline{\tau}}(\mathbf{x}) \cdot \underline{n}(\mathbf{x}) = \underline{0} \quad \forall \mathbf{x} \in \Sigma_{x_1=L}\} \end{aligned}$$

3.2 Weak form - stationary condition of the energy

To get the weak form, one would go from the variational formulation, and as in the 1D case, search for $\underline{u}$ minimizing $I(\underline{u})$. We will thus derive the above formulation using geometric symmetries as well as the different formulation of our tensors, and then conclude on WF.

We get from the non projected form :

$$\begin{aligned} I'(\underline{u})(\underline{v}) &= \int_\Omega \frac{\partial \phi}{\partial(\underline{\underline{\epsilon}}(\underline{u}))} : (\underline{\underline{\epsilon}}(\underline{v})dV + \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV + \int_{\Sigma_{x_2=H}} P\underline{e2} \cdot \underline{v} dS \\ &= \int_\Omega \underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{(u)}) : (\underline{\underline{\epsilon}}(\underline{v})dV + \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV + \int_{\Sigma_{x_2=H}} P\underline{e2} \cdot \underline{v} dS \end{aligned}$$

Yet from the product of an asymmetric variable by a symetric one :

$$\begin{aligned} A_{ij} \cdot B_{ij} &= -A_{ji} \cdot B_{ji} \\ &= -A_{kl} \cdot B_{kl} \\ &= -A_{ij} \cdot B_{ij} \\ &\rightarrow 2A_{ij} \cdot B_{ij} = 0 \end{aligned}$$

with A and B two order two tensors, A symmetric and B anti-symmetric.

By decomposition of $\underline{\underline{\nabla} u}$ into its symmetric and anti-symmetric part, and by recalling that tensor $\underline{\underline{\sigma}}$ is symmetric, we have :

$$\begin{aligned} & \underline{\underline{\nabla} u} = (\underline{\underline{\epsilon}}(\underline{u}) - \underline{\underline{\Omega}}(\underline{u})) \\ \rightarrow \ & \underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{u})) : \underline{\underline{\epsilon}}(\underline{v}) = \underline{\underline{\sigma}} : \underline{\underline{\nabla} v} \end{aligned}$$

We now implement this relation in the previous one to eventually get the weak formulation

$$\begin{aligned} I'(\underline{u})(\underline{v}) &= \int_\Omega \underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{u})) : \underline{\underline{\nabla} v} dV + \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV + \int_{\Sigma_{x_2=H}} P\underline{e_2} \cdot \underline{v} dS = 0 \\ \rightarrow & \underbrace{\int_\Omega \underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{u})) : \underline{\underline{\nabla} v} dV}_{\text{a(u,v)}} = \underbrace{- \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV - \int_{\Sigma_{x_2=H}} P\underline{e_2} \cdot \underline{v} dS}_{\text{l(v)}} \end{aligned}$$

This last equality defines the weak formulation :

Find $u \in U_{ad} : a(u,v) = l(v) \forall \underline{v} \in U_{ad}^0 = Uad$

$a(u,v) = \int_\Omega \underline{\underline{\sigma}}(\underline{\underline{\epsilon}}(\underline{u})) : \underline{\underline{\nabla} v} dV$

$l(v) = - \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV - \int_{\Sigma_{x_2=H}} P\underline{e_2} \cdot \underline{v} dS$

3.3 Strong formulation

To now get the strong formulation, the point is to do an integration by part and to then use the FLCV.

First : IPP on left hand side term :

$$\int_\Omega \underline{\underline{\sigma}} : \underline{\underline{\nabla v}} dV = \int_\Omega \underline{\underline{\sigma n}} \cdot \underline{v} dS - \int_\Omega \underline{\underline{\nabla \sigma}} \cdot \underline{v} dV$$

and by identification : LFCV :

$$\int_\Omega \underline{\underline{\nabla \sigma}} \cdot \underline{v} dV = \int_\Omega \rho g \underline{e_2} \cdot \underline{v} dV$$
$$\int_\Omega \underline{\underline{\sigma n}} \cdot \underline{v} dS = - \int_{\Sigma_{x_2=H}} P \underline{e_2} \cdot \underline{v} dS$$

$$\rightarrow \underline{\underline{\nabla \sigma}} = \rho g \underline{e_2} \quad \forall \mathbf{x} \in \Omega$$
$$\rightarrow \underline{\underline{\sigma n}} = -P \underline{e_2} \quad \forall \mathbf{x} \in \Sigma_{x_2=H}$$

These two last equalities stand as our strong formulation, and more specifically the equilibrium equation of the 2D problem, with thus associated considered stresses.

Thus, the kinematic field is fulfilled by composition of : the BC, the equations driving the problem and the smoothness of the function throughout the domain ; while the smoothness of σ is being fulfilled through the smoothness of vector v in U_{ad}.

4 Discretisation

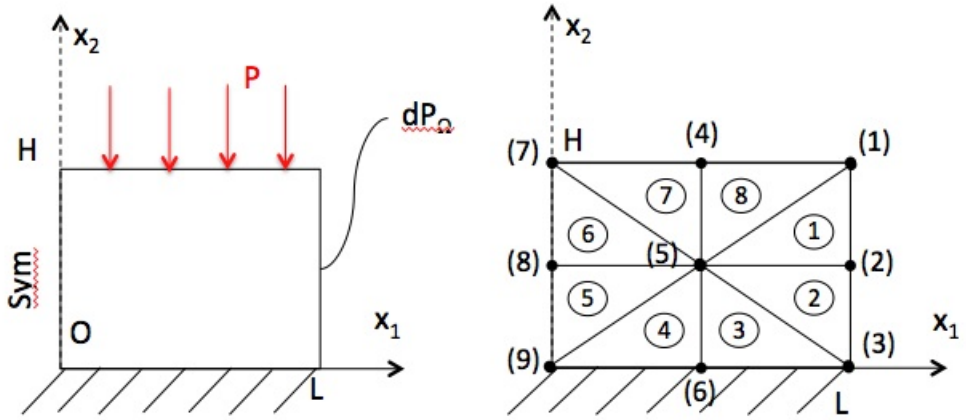


Figure 2: Schematic representation of the structure and the finite element mesh

The problem will be solved thanks to finite elements analysis (FEA). To this aim the geometry is represented by the mesh on Figure 2. It is exclusively composed of T3 elements, i.e. 3-node triangles. These elements are isoparametric (Lagrange-type interpolation).

4.1 Parametrical representation

1. Coordinate matrix : the coordinates of the nodes are represented by the numbers under parentheses, i.e. (a). As we are in 2D, each of our nodes is represented by 2 coordinates, summarized in the following matrix :

$$[Coord]^T = \begin{bmatrix} L & L & L & L/2 & L/2 & L/2 & 0 & 0 & 0 \\ H & H/2 & 0 & H & H/2 & 0 & H & H/2 & 0 \end{bmatrix}$$

2. Connectivity matrix : the connectivity matrix is defined as a matrix containing the nodes associated in one element. Because we work on T3 elements, we have 3 nodes associated to each elements. This matrix will find itself very useful to build the mesh after introducing unit T3 element. It's what makes the link between the geometry of the mesh and the nodes.

   Yet, for the method to be usable, one must consider the connectivity matrix to be anticlockwise and we choose it to be centered in common nod (5) such that the connectivity matrix of the mesh is :

$$[Connec]^T = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 2 & 3 & 6 & 9 & 8 & 7 & 4 & 1 \\ 1 & 2 & 3 & 6 & 9 & 8 & 7 & 4 \end{bmatrix}$$

3. Shape functions $N_k(\underline{a})$: the figure below represent a unit element whose coordinates are put under a matrix form through shape functions.

We systematize calculation during the solving phase using the reference mesh Δ_e. From it we derive the shape functions and indirectly the parametrical representation ($\underline{x} = f(\underline{a})$) of a real element E_e.

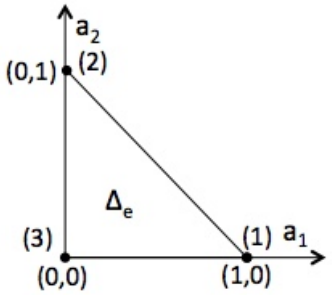


Figure 3: Reference mesh

We work on an elementary element T3. We then are to derive a matrix defining this element and which will allow us to then build the overall structure. This matrix is called shape function :

$$^{\intercal}[N_e(\underline{a})] = \begin{bmatrix} N_1(\underline{a}) & N_2(\underline{a}) & N_3(\underline{a}) \end{bmatrix}$$

This matrix is used for coordinates interpolation : we define it for local element of unit coordinates and then use it for global element of non-unit coordinates.

One can see that as we work on T3 elements, the lines between each nodes will be an order one function. Thus, the function one could derive to obtain the T3 coordinates will be of same order. With a_1 and a_2 our parameters, we will look for the coordinates' equation under the form $N_k(\underline{a}) = c_1 a_1 + c_2 a_2 + c_3$, with $c_1, c_2, c_3 \in \mathbb{R}$.

We will do the calculation for N_3 and give then the entire matrix N_k :

- This node is located in [0, 0], we then must get $N_3 = 1$ in this point and $N_3 = 0$ at the two others ;
- in (1) : $N_3(a_1 = 1, a_2 = 0) = c_1 + 0 + c_3 = 0$
- in (2) : $N_3(a_1 = 0, a_2 = 1) = 0 + c_2 + c_3 = 0$
- in (3) : $N_3(a_1 = 0, a_2 = 0) = c_3 = 1$

Solving the associated system, one get $N_3(a) = 1 - (a_1 + a_2)$

We finally get the following components for the shape function $N_k(\underline{a})$ of the reference mesh :

$$N_1(\underline{a}) = a_1$$
$$N_2(\underline{a}) = a_2$$
$$N_3(\underline{a}) = 1 - a_1 - a_2$$

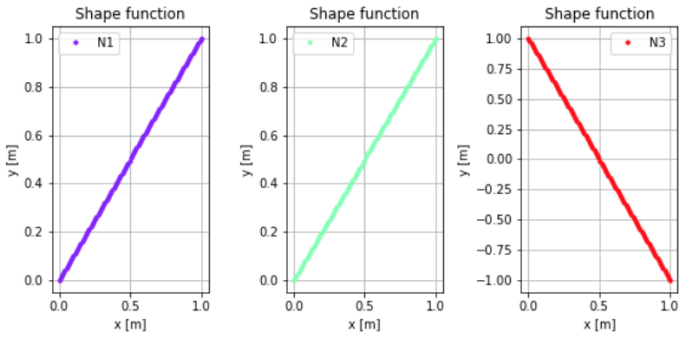


Figure 4: Shape functions T3 and associated graphs

As mentioned, we now associate this matrix to coordinates to get global coordinates. With the parametrical representation ($\underline{x} = f(\underline{a})$) :

$$\underline{x} = N_1(\underline{a})\underline{x}^{(1)} + N_2(\underline{a})\underline{x}^{(2)} + N_3(\underline{a})\underline{x}^{(3)}$$

With $\underline{x}^{(1)}$, $\underline{x}^{(2)}$, $\underline{x}^{(3)}$ coordinates of the nodes, it comes :

$$[Coord_e] = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ x_1^{(3)} & x_2^{(3)} \end{bmatrix}$$

$$\begin{bmatrix} x_1^e & x_2^e \end{bmatrix} = ^{\intercal} [N_e(\underline{a})] \cdot [Coord_e]$$

4. Jacobian matrix : we will go through two methods to derive the expression of it.

   Method 1 : using parametrical representation.

   Let's calculate the jacobian of the matrix. We have by definition :

$$J_{ij} = \frac{\partial x_i}{\partial a_j}, \quad with \quad i, j = 1, 2$$

   Thus, in 2D for T3 element the Jacobian matrix is of 2 by 2. We can then calculate each of its 4 components :

$$J_{11} = \frac{\partial x_1}{\partial a_1} = \frac{\partial (N_1(\underline{a})x_1^{(1)} + N_2(\underline{a})x_1^{(2)} + N_3(\underline{a})x_1^{(3)})}{\partial a_1}$$

$$= N_{1,1}x_1^{(1)} + N_{2,1}x_1^{(2)} + N_{3,1}x_1^{(3)} = x_1^{(1)} - x_1^{(3)}$$

$$J_{12} = \frac{\partial x_1}{\partial a_2} = N_{1,2}x_1^{(1)} + N_{2,2}x_1^{(2)} + N_{3,2}x_1^{(3)} = x_1^{(2)} - x_1^{(3)}$$

$$J_{21} = \frac{\partial x_2}{\partial a_1} = N_{1,1}x_2^{(1)} + N_{2,1}x_2^{(2)} + N_{3,1}x_2^{(3)} = x_2^{(1)} - x_2^{(3)}$$

$$J_{22} = \frac{\partial x_2}{\partial a_2} = N_{1,2}x_2^{(1)} + N_{2,2}x_2^{(2)} + N_{3,2}x_2^{(3)} = x_2^{(2)} - x_2^{(3)}$$

$$\underline{\underline{J}}(\underline{a}) = \begin{bmatrix} x_1^{(1)} - x_1^{(3)} & x_1^{(2)} - x_1^{(3)} \\ x_2^{(1)} - x_2^{(3)} & x_2^{(2)} - x_2^{(3)} \end{bmatrix}$$

Method 2 : definition as a function of $[D_N]$ (the matrix of the derivatives with respect to $\underline{a}$ of the shape functions) and $[Coord_e^{(i)}]$ the matrix of the coordinates of the nodes of element (i).

It comes :

$$\underline{\underline{J}}(\underline{a}) = [Coord_e]^t \cdot [D_N]$$

With:

$$[D_N]^t = \begin{bmatrix} N_{1,1} & N_{2,1} & N_{3,1} \\ N_{1,2} & N_{2,2} & N_{3,2} \end{bmatrix}$$

and coords as expressed above.

Consequently :

$$\underline{\underline{J}}(\underline{a}) = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} \end{bmatrix} \cdot \begin{bmatrix} N_{1,1} & N_{1,2} \\ N_{2,1} & N_{2,2} \\ N_{3,1} & N_{3,2} \end{bmatrix}$$

By application to the element (1) we get from both methods :

$$\underline{\underline{J}}(\underline{a}) = \begin{bmatrix} -L/2 & 0 \\ -H/2 & -H/2 \end{bmatrix}$$

5. Jacobian (i.e. $det(J_{ij})$)

From Jacobian matrix for element (1) we get its determinant :

$$\underline{\underline{J}}(\underline{a}) = \begin{bmatrix} -L/2 & 0 \\ -H/2 & -H/2 \end{bmatrix} \rightarrow J = det(\underline{\underline{J}}(\underline{a})) = \frac{HL}{4}$$

Yet, as $d\Omega_{Ee} = J \cdot d\Omega_{\Delta_e}$ and $d\Omega_{\Delta_e} = \frac{1}{2}$ so the area of element 1 is equal to :

$$d\Omega_{Ee} = \frac{HL}{8}$$

Because all the elements have the same area, the Jacobian is the same for all elements : $J = \frac{HL}{4}$.

This result states that when integrating, a multiplication by the jacobian allows to go from real mesh to reference one. It's the principle first mentioned in the abstract : we defined

reference element and we can through Jacobian coefficient run calculation on real elements (i.e. non-elementary nodes coordinates).

Numerical application 4.1

Define in `python` the following objects:

- `coords` (the coordinate matrix).
- `elements` (the connectivity matrix).
- The function `J(i,coords=coords,elements=elements)` returning the value of the jacobian of the i-th element.
- The function `N(a,i)` returning the value of the i-th shape function at the point $a = [a_0, a_1]$ in the reference element.
- `D_N` (the matrix of the derivatives with respect to $\underline{a}$ of the shape functions).
- Plot the shape functions.
- Define a function `meshplot`$(coord, elements)$. Optionally, include labels for elements and nodes.

4.2 Displacement interpolation

1. Elementary nodal displacement

   $\{U_e^{(3)}\}$ stands for the elementary nodal displacements vector for element (3). As for the matrices we've been defining so far, its shape depends on the domain and the elementary element considered. As we have T3 element in 2D and that we aim to be under vectorized form : its shape will be (6×1) :

   $$^{\intercal}\{U_e^{(3)}\} = \begin{bmatrix} U_1^{(5)} & U_2^{(5)} & U_1^{(6)} & U_2^{(6)} & U_1^{(3)} & U_2^{(3)} \end{bmatrix}$$

   Then, we have the following general shape of the interpolation matrix : with $N_k(\underline{x}) = N_k(\underline{x}(\underline{a}))$:

   $$[N_e] = \begin{bmatrix} N_1(\underline{a}) & 0 & N_2(\underline{a}) & 0 & N_3(\underline{a}) & 0 \\ 0 & N_1(\underline{a}) & 0 & N_2(\underline{a}) & 0 & N_3(\underline{a}) \end{bmatrix}$$

   And we get the associated displacement interpolation in element (3) :

   $$\{u(\underline{x})\} = [N_e] \cdot \{U_e^{(3)}\}$$

2. Global nodal displacement

   $\{U\}$ stands for the global nodal displacements vector of the structure. It's a combination of every elementary displacements. We can for instance do this through localisation matrix.

Yet, this would ask for a too big matrix of localisation (as we have here global nodal displacement's shape being (18×1) due to having 9 nodes with 2 coordinates each and that we want a vectorized form).

Then, instead of using this localisation matrix, one would prefer to look for interpolation from local to global coordinates. It's the method that we will use to go from local to global coordinates leading eventually to $\{U\}$ after assembly process.

Boundary conditions

You may know that as this, the system is ill-defined as it brings no unique solution. To avoid this and get unique solution, one might then need to apply boundary conditions. Doing so, we have a well-define problem and we reduce the number of dof, defined here as the number of unknown components in $\{U\}$.

The boundary conditions are as follows:

$$U_1^7 = U_1^8 = U_1^9 = 0 \quad \text{(i.e. on } \Sigma_{x_1=0})$$

$$U_2^9 = U_2^6 = U_2^3 = 0 \quad \text{(i.e. on } \Sigma_{x_2=0})$$

We then here reduce our problem to a problem with $18 - 6 = 12$ dof.

3. Example

   The fifth dof refers to node **3**. Moreover, its the displacement along $\underline{e_x}$ of the third nod.

4. Global interpolation function : $\tilde{N}$ associated to this node using the straight lines method.

$$\tilde{N}_3(\underline{x}) = \begin{cases} \frac{2}{L}(x_1 - \frac{L}{2}) & \underline{x} \in \textcircled{3} \\ \frac{2}{H}(\frac{H}{2} - x_2) & \underline{x} \in \textcircled{2} \\ 0 \quad elsewhere \end{cases}$$

4.3 Alternate modeling - T6

On this part the mesh chosen is geometrically simpler but made of second order elements. Indeed, we replace in previous problem the mesh's element by T6 elements. The reference element and overall mesh associated with those T6 are represented on Figure 5.

The coordinates matrix now is given by :

$$[connec] = \begin{bmatrix} (7) & (9) & (1) & (8) & (5) & (4) \\ (9) & (3) & (1) & (6) & (2) & (5) \end{bmatrix}$$

We indeed have 2 T6 with 6 nodes each.

1. Shape function [2]

   Consider the reference element in figure 5. First and foremost, we calculate the shape function $N_1(\underline{a})$ for the first node (1). As we know, this shape function should be equal to

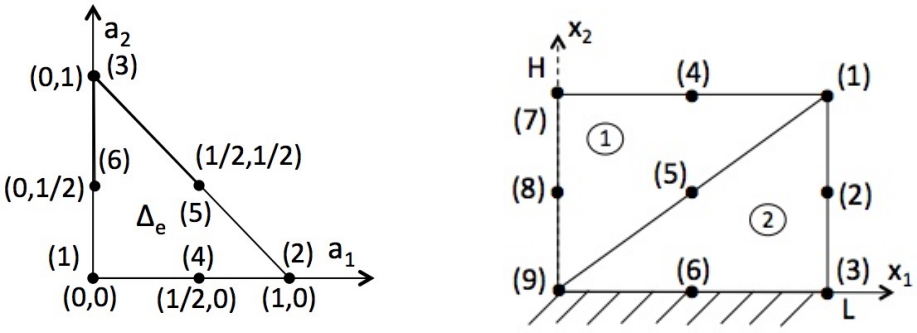


Figure 5: Reference mesh for the 6-node triangles and alternate mesh of the structure.

1 at node (1) and equal to 0 at other 5 nodes. This formulation recalls saying that $N_1 = 0$ on the line connecting points (5), (6) (equation: $1 - a_1 - a_2 = 0$) and the line connecting points (3), (5), (2) (equation: $\frac{1}{2} - a_1 - a_2 = 0$). So the shape function is written as:

$$N_1(\underline{a}) = \frac{(\frac{1}{2} - a_1 - a_2)(1 - a_1 - a_2)}{(\frac{1}{2} - 0 - 0)(1 - 0 - 0)} = 2(\frac{1}{2} - a_1 - a_2)(1 - a_1 - a_2)$$

The denominator is calculated by replacing a_1, a_2 by their values on node (1) to ensure N_1 equals 1 on this node.

By similar analyses, we calculate shape functions for all 6 nodes:

$$N_1(\underline{a}) = 2(\frac{1}{2} - a_1 - a_2)(1 - a_1 - a_2)$$
$$N_2(\underline{a}) = a_1(2a_1 - 1)$$
$$N_3(\underline{a}) = a_2(2a_2 - 1)$$
$$N_4(\underline{a}) = 4a_1(1 - a_1 - a_2)$$
$$N_5(\underline{a}) = 4a_1a_2$$
$$N_6(\underline{a}) = 4a_2(1 - a_1 - a_2)$$

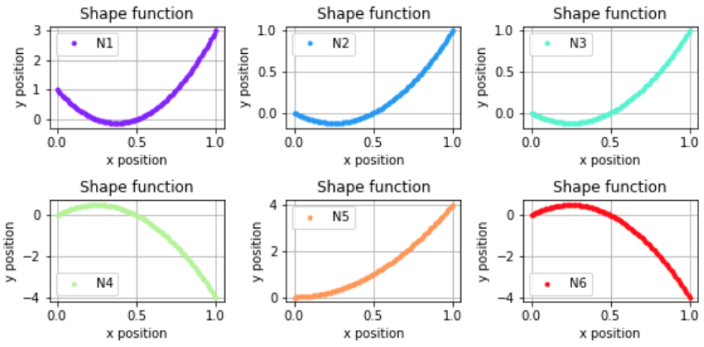


Figure 6: Shape functions T6 and associated graphs

2. Jacobian matrix - element ①

$$\underline{\underline{J}}(\underline{a}) = [Coord_e]^t \cdot [D_N]$$

$$[D_N]^t = \begin{bmatrix} N_{1,1} & N_{2,1} & N_{3,1} & N_{4,1} & N_{5,1} & N_{6,1} \\ N_{1,2} & N_{2,2} & N_{3,2} & N_{4,2} & N_{5,2} & N_{6,2} \end{bmatrix}$$

$$[Coord_e]^t = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & x_1^{(4)} & x_1^{(5)} & x_1^{(6)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & x_2^{(4)} & x_2^{(5)} & x_2^{(6)} \end{bmatrix}$$

With $x_k^{(ij)} = x_k^{(i)} - x_k^{(j)}$, we represent the general form of $\underline{\underline{J}}$

$$
\begin{aligned}
J_{11} &= 4a_1(x_1^{(14)} + x_1^{(24)}) + 4a_2(x_1^{(14)} + x_1^{(56)}) - 3x_1^{(1)} + 4x_1^{(4)} - x_1^{(2)} \\
J_{12} &= 4a_1(x_1^{(14)} + x_1^{(56)}) + 4a_2(x_1^{(16)} + x_1^{(36)}) - 3x_1^{(1)} + 4x_1^{(6)} - x_1^{(3)} \\
J_{21} &= 4a_1(x_2^{(14)} + x_2^{(24)}) + 4a_2(x_2^{(14)} + x_2^{(56)}) - 3x_2^{(1)} + 4x_2^{(4)} - x_2^{(2)} \\
J_{22} &= 4a_1(x_2^{(14)} + x_2^{(56)}) + 4a_2(x_2^{(16)} + x_2^{(36)}) - 3x_2^{(1)} + 4x_2^{(6)} - x_2^{(3)}
\end{aligned}
$$

Contrary to previous case, the Jacobian Jacobian matrix depends on the $\underline{a}$.

3. Jacobian - element (1)

   For element (1), we observe that $x_1^{(14)} = x_1^{(24)} = x_1^{(56)} = 0$, $x_2^{(16)} + x_2^{(36)} = 0$, $-3x_1^{(1)} + 4x_1^{(6)} - x_1^{(3)} = L$, etc. All terms depending on a_1 and a_2 then become zero while the one which don't either get non-zero value or zero value such that eventually the Jacobian matrix of element (1) is :

$$
\underline{\underline{J}}(\underline{a}) = \begin{bmatrix} 0 & L \\ -H & 0 \end{bmatrix}
$$

   The Jacobian of element (1) is therefore $det(\underline{\underline{J}}) = HL$.

   As $d\Omega_{Ee} = J \cdot d\Omega_{\Delta_e}$, the area of the element of reference is $d\Omega_{\Delta_e} = \frac{1}{2}$ the area of element 1 is $d\Omega_{Ee} = \frac{HL}{2}$, the Jacobian is the fraction between theses 2 values:

$$
J = \frac{\Omega_{Ee}}{\Omega_{\Delta_e}} = HL
$$

   The result is verified.

   As the jacobian characterizes the geometrical transformation between the real and reference mesh, we have that integrals can be calculated by a multiplication by constant jacobian.

4. General shape of the interpolation matrix

   Similar to T_3, we use iso-parametric interpolation, which means the interpolation functions are the same as shape functions. The interpolation matrix is of dimension 2*12, as we have 6 nodes with 2 coordinates each in one element.

$$
[N_e] = \begin{bmatrix} N_1(\underline{a}) & 0 & N_2(\underline{a}) & 0 & \cdots & N_6(\underline{a}) & 0 \\ 0 & N_1(\underline{a}) & 0 & N_2(\underline{a}) & \cdots & 0 & N_6(\underline{a}) \end{bmatrix}
$$

5. Shape of $\{U\}$

   $\{U\}$'s shape is same as for T3 element as it depends on the mesh and nodes of it which are of same number here. More precisely, it will be the combination of two sub-vectors of size (12×1) each.

   After superposition we then have size($\{U\}$) = (12 + 12 - 6) with 6 being the number of shared nodes that must be accounted only once.

Therefore, $\{U\}$'s shape is as expected (18×1). Similarly, after application of BC, we reduce the number of dof to 12 as we apply BC on 6 nodes' displacement (3 at left and 3 at bottom).

What may differ is that on one element one may apply two different areal forces on two of the element's surfaces.

6. Comparison T3 - T6

   The overall differences that one could expect from T6 with respect to T3 element is that it implies quadratic displacement. Thus, an element can theoretically bend with a T6 element while it is limited to linear displacement with T3.

   This would result into more realistic movement of the mesh.

5 Strains, stiffness matrix and nodal forces

The general expression of the jacobian matrix of a T3 mesh is reminded here:

$$[J] = \begin{bmatrix} (x_1^{(1)} - x_1^{(3)}) & (x_1^{(2)} - x_1^{(3)}) \\ (x_2^{(1)} - x_2^{(3)}) & (x_2^{(2)} - x_2^{(3)}) \end{bmatrix}$$

and

$$J = det\,[J] = \left(x_1^{(1)} - x_1^{(3)}\right)\left(x_2^{(2)} - x_2^{(3)}\right) - \left(x_2^{(1)} - x_2^{(3)}\right)\left(x_1^{(2)} - x_1^{(3)}\right) = 2A$$

Where A stands for the area of the element considered.

5.1 Strains

It is reminded that the gradient matrix $[Be(\underline{a})]$ is introduced in order to obtain a direct estimation of strains from nodal values. In the case of an isoparametric element nodal values are the nodal displacements only and strains can be estimated as follows:

$$\{\varepsilon_e(\underline{x})\} = [B_e(\underline{a})]\,\{U_e\} \quad \forall \underline{x} \in E_e$$

We will seek to demonstrate that for a T3 iso-parametric element the gradient matrix $[Be(\underline{a})]$ has the following form:

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} = \frac{1}{2\mathrm{A}} \begin{bmatrix} x_2^{(23)} & 0 & x_2^{(31)} & 0 & x_2^{(12)} & 0 \\ 0 & x_1^{(32)} & 0 & x_1^{(13)} & 0 & x_1^{(21)} \\ x_1^{(32)} & x_2^{(23)} & x_1^{(13)} & x_2^{(31)} & x_1^{(21)} & x_2^{(12)} \end{bmatrix} \begin{Bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \\ u_2^{(3)} \end{Bmatrix}$$

with $x_k^{(ij)} = x_k^{(i)} - x_k^{(j)}$ and where A is the area of the element in the structure.

To this aim the matrix $[G_N]$ will be constructed with $[G_N]_{ij} = \dfrac{\partial N_i}{\partial x_j}$. Then the components of $[Be(\underline{a})]$ may be expressed from the ones of $[G_N]$.

1. Demonstration $[Be(\underline{a})]$

   In the case of an iso-parametric, it is possible to obtain a direct estimation of strains from nodal values using a grandient matrix $[B_e(\underline{a})]$ as :

   $$\{\varepsilon_e(\underline{x})\} = [B_e(\underline{a})]\,\{U_e\} \quad \forall \underline{x} \in E_e$$

   We know that:

   $$\underline{\underline{\epsilon}}(\underline{u}) = \frac{1}{2}(\underline{\underline{\nabla u}} + \underline{\underline{\nabla u}}^T)$$

In the case of an isoparametric case:

$$\underline{\underline{\nabla u}} = \frac{\partial u_i}{\partial x_j} = \sum_{k=1}^{n_e} \frac{\partial N_k}{x_j} u_i^{(k)}$$

Developping the equation and rewriting in matrix form we have:

$$\underline{\underline{\epsilon}}(\underline{u}) = \frac{1}{2}([D_N]^T[U_e][J^{-1}] + ([D_N]^T[U_e][J^{-1}])^T)$$

Introducing the matrix :

$$[G_N] = [D_N][J^{-1}]$$

$$[D_N]^t = \begin{bmatrix} N_{1,1} & N_{2,1} & N_{3,1} \\ N_{1,2} & N_{2,2} & N_{3,2} \end{bmatrix}$$

$$[J^{-1}] = \frac{1}{2A} \begin{bmatrix} x_2^{(23)} & x_1^{(31)} \\ x_2^{(31)} & x_1^{(13)} \end{bmatrix}$$

Thus:

$$[G_N] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \frac{1}{2A} \begin{bmatrix} x_2^{(23)} & x_1^{(31)} \\ x_2^{(31)} & x_1^{(13)} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2^{(23)} & x_1^{(32)} \\ x_2^{(31)} & x_1^{(13)} \\ x_2^{(12)} & x_1^{(21)} \end{bmatrix}$$

We can now rewrite $\underline{\underline{\nabla u}} = [G_N]^T[U_e]$

$$\underline{\underline{\nabla u}} = \frac{1}{2A} \begin{bmatrix} x_2^{(23)} & x_2^{(31)} & x_2^{(12)} \\ x_2^{(32)} & x_1^{(13)} & x_1^{(21)} \end{bmatrix} \begin{bmatrix} u_1^{(1)} & u_2^{(1)} \\ u_1^{(2)} & u_2^{(2)} \\ u_1^{(3)} & u_2^{(3)} \end{bmatrix}$$

$$= \begin{bmatrix} x_2^{(23)}u_1^{(1)} + x_2^{(31)}u_1^{(2)} + x_2^{(12)}u_1^{(3)} & x_2^{(23)}u_2^{(1)} + x_2^{(31)}u_2^{(2)} + x_2^{(12)}u_2^{(3)} \\ x_1^{(32)}u_1^{(1)} + x_1^{(13)}u_1^{(2)} + x_1^{(21)}u_1^{(3)} & x_1^{(32)}u_2^{(1)} + x_1^{(13)}u_2^{(2)} + x_1^{(21)}u_2^{(3)} \end{bmatrix}$$

Knowing that $\underline{\underline{\epsilon}}(\underline{u}) = \frac{1}{2}(\underline{\underline{\nabla u}} + \underline{\underline{\nabla u}}^T)$, we get:

$$\epsilon_{11} = \frac{1}{2A}(x_2^{(23)}u_1^{(1)} + x_2^{(31)}u_1^{(2)} + x_2^{(12)}u_1^{(3)})$$

$$\epsilon_{22} = \frac{1}{2A}(x_1^{(32)}u_2^{(1)} + x_1^{(13)}u_2^{(2)} + x_1^{(21)}u_2^{(3)})$$

$$2\epsilon_{12} = \frac{1}{2A}(x_2^{(23)}u_2^{(1)} + x_2^{(31)}u_2^{(2)} + x_2^{(12)}u_2^{(3)} + x_1^{(32)}u_1^{(1)} + x_1^{(13)}u_1^{(2)} + x_1^{(21)}u_1^{(3)})$$

Thus, by identification, we finally have:

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} = \frac{1}{2\mathrm{A}} \begin{bmatrix} x_2^{(23)} & 0 & x_2^{(31)} & 0 & x_2^{(12)} & 0 \\ 0 & x_1^{(32)} & 0 & x_1^{(13)} & 0 & x_1^{(21)} \\ x_1^{(32)} & x_2^{(23)} & x_1^{(13)} & x_2^{(31)} & x_1^{(21)} & x_2^{(12)} \end{bmatrix} \cdot \begin{Bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \\ u_2^{(3)} \end{Bmatrix}$$

2. Matrices B_e (2) and (3)

For the element 2, let's use the the matrix connec to identify the joints :

$$\text{connec}^{(2)} = \begin{bmatrix} 5 & 3 & 2 \end{bmatrix}$$

We have :

$$B_e^{(2)} = \frac{1}{2\mathrm{A}} \begin{bmatrix} x_2^{(32)} & 0 & x_2^{(25)} & 0 & x_2^{(53)} & 0 \\ 0 & x_1^{(23)} & 0 & x_1^{(52)} & 0 & x_1^{(35)} \\ x_1^{(23)} & x_2^{(32)} & x_1^{(52)} & x_2^{(25)} & x_1^{(35)} & x_2^{(53)} \end{bmatrix}$$

We can use the coordinate matrix to find associated coordinates :

$$\text{coord}^{(2)} = \begin{bmatrix} L/2 & H/2 \\ L & 0 \\ L & H/2 \end{bmatrix} = \begin{bmatrix} x_1^{(5)} & x_2^{(5)} \\ x_1^{(3)} & x_2^{(3)} \\ x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$

With $A = \frac{4}{LH}$, we finally obtain :

$$\left[Be^{(2)}\right] = \begin{bmatrix} -2/L & 0 & 0 & 0 & 2/L & 0 \\ 0 & 0 & 0 & -2/H & 0 & 2/H \\ 0 & -2/L & -2/H & 0 & 2/H & 2/L \end{bmatrix}$$

Using the same method we get for (3) :

$$\left[Be^{(3)}\right] = \begin{bmatrix} 0 & 0 & -2/L & 0 & 2/L & 0 \\ 0 & 2/H & 0 & -2/H & 0 & 0 \\ 2/H & 0 & -2/H & -2/L & 0 & 2/L \end{bmatrix}$$

Numerical application 5.1

Define in `python` the following objects:

- The matrix `G_N`.
- The function `B_e(a)` returning the matrix at a given point of the reference element

5.2 Elementary stiffness matrix

1. Stiffness matrix $[K_e^{(2)}]$ (i.e. Case A 3)

   In the general case exposed in 3, we have the following local formulation :

   $$a(u,v) = \int_\Omega \underline{\underline{\epsilon}}(\underline{u}) : [A] : \underline{\underline{\epsilon}}(\underline{u}) dV$$

   with the gradient matrix $[B_e(\underline{a})]$:

   $$\{\varepsilon_e(\underline{x})\} = [B_e(\underline{a})]\,\{U_e\} \quad \forall \underline{x} \in E_e$$

   Then we have :

   $$\begin{aligned} a(u,v) &= \{V_e\}^t \left(\int_\Omega [B_e(\underline{a})] : [A] : [B_e(\underline{a})]\, dV\right) \{U_e\} \\ &= \{V_e\}^t\, [K_e]\, \{U_e\} \end{aligned}$$

   By changing the integral domain into referential element :

   $$[K_e] = \int_{\Delta_e} [B_e(\underline{a})] : [A] : [B_e(\underline{a})]\, J(\underline{a}) d\Delta_e$$

   In the problem, the value of Jacobian is the same for every element ($\frac{HL}{4}$), and $[B_e(\underline{a})]$ doesn't depend on $\underline{a}$, therefore $[K_e]$ is the same for every element. As there is no variable $\underline{a}$ in the integration, the integration amounts to multiply the internal term by the surface of referential element.

   Subsequently :

   $$\left[K_e^{(2)}\right] = \left[Be^{(2)}\right] : [A] : \left[Be^{(2)}\right] \times \frac{HL}{4} \times \frac{1}{2}$$

2. Dimension

   The dimension of $[K_e]$ is $(\mathbf{6} \times \mathbf{6})$.

Numerical application 5.2

Define in `python` the following objects:

- A function `K_e(i,...)` providing the element stiffness matrix for the i-th element.

5.3 Elementary right hand member

1. Volume nodal force

   The general form of $\{F_v^e\}$ is:

$$\{F_v^e\} = \int_{\Delta_e} [N_e(\underline{a})]^t \cdot \{f_v\}\, J_e(\underline{a}) d\Delta_e$$

   In the weak formulation, external volume forces are :

$$\int_{\Omega_e} \underline{f_v} \cdot \underline{v} d\Omega_e$$

   In finite element: $\underline{v} = [N_e(\underline{a}(\underline{x}))] \cdot \{V_e\}$, in which $\{V_e\}$ is the virtual displacement vector. Then the previous term becomes:

$$\begin{aligned}\int_{\Omega_e} \underline{f_v} \cdot \underline{v} d\Omega_e &= \int_{\Omega_e} \{V_e\}^t \cdot [N_e(\underline{a}(\underline{x}))]^t \cdot \{f_v\}\, d\Omega_e \\ &= \{V_e\}^t \int_{\Delta_e} \cdot [N_e(\underline{a})]^t \cdot \{f_v\}\, J_e(\underline{a}) d\Delta_e \\ &= \{V_e\}^t \cdot \{F_v^e\}\end{aligned}$$

   In this problem, the external volume force is the gravity : $\{f_v\} = \begin{bmatrix} 0 \\ -\rho g \end{bmatrix}$. Because $[N_e(\underline{a})]$, $\{f_v\}$ and $J_e(\underline{a})$ are the same for every element, the volume nodal force is also the same for each element. Not lose generality, we calculate for the element (8):

$$\begin{aligned}\{F_v^e\} &= \int_{\Delta_e} [N_e(\underline{a})]^t \cdot \{f_v\}\, J_e(\underline{a}) d\Delta_e \\ &= \int_{\Delta_e} \begin{bmatrix} a_1 & 0 \\ 0 & a_1 \\ a_2 & 0 \\ 0 & a_2 \\ 1-a_1-a_2 & 0 \\ 0 & 1-a_1-a_2 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -\rho g \end{bmatrix} \frac{HL}{4} d\Delta_e \\ &= -\frac{\rho g H L}{4} \int_{\Delta_e} \begin{bmatrix} 0 \\ a_1 \\ 0 \\ a_2 \\ 0 \\ 1-a_1-a_2 \end{bmatrix} d\Delta_e\end{aligned}$$

   We use Gauss-Hammer method for calculating this integration. The terms to integrate are of first order, one point of integration is used, $\underline{a}_g = (\frac{1}{3},\ \frac{1}{3})$ and the related wight is $w_g = \frac{1}{2}$, with $\int_{\Delta_e} f(a_1, a_2) d\Delta_e = \sum w_g f(a_{g1}, a_{g2})$.

$$\int_{\Delta_e} a_1 d\Delta_e = w_g a_{g1} = \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$$
$$\int_{\Delta_e} a_2 d\Delta_e = w_g a_{g2} = \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$$
$$\int_{\Delta_e} (1 - a_1 - a_2) d\Delta_e = w_g (1 - a_{g1} - a_{g2}) = \frac{1}{2}(1 - \frac{1}{3} - \frac{1}{3}) = \frac{1}{6}$$

Finally the volume nodal force for one element is:

$$\{F_v^e\} = -\frac{\rho g H L}{24} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

We represent theses nodal forces on the mesh:

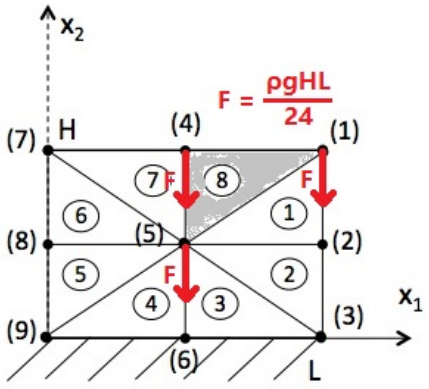


Figure 7: Volume nodal force for element 8

2. Surface nodal force [1]

   The general form of $\{F_s^e\}$ is:

$$\{F_s^e\} = \int_{\partial\Delta_e} \left[\hat{N}_e(\underline{b})\right]^t \cdot \left\{F^d(\underline{x}(\underline{b}))\right\} \hat{J}_e(\underline{b}) dS_{\Delta_e}$$

   $\underline{b}$ being the variable of reference for surface ($\underline{x} = \sum_{nodes\ on\ \partial\Omega_e} \hat{N}_i(\underline{b})\underline{x}^{(i)}$).

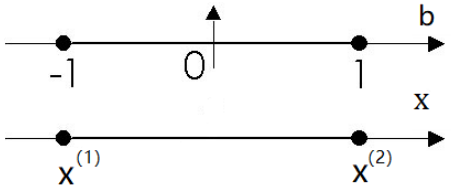


Figure 8: 1D reference mesh

In this problem Ω is a 2D domain, and so the associated surface is a 1D line. We here use a notation with middle of the line centered in 0 as you can see in the figure (8). The shape functions then are:

$$\hat{N}_1(b) = \frac{1-b}{2} \quad \hat{N}_2(b) = \frac{1+b}{2}$$

In the real element:

$$x = \hat{N}_1(b)x^{(1)} + \hat{N}_2(b)x^{(2)} \tag{1}$$

x here is x_1 or x_2 in the 2D element, depending on the surface.

The only effort on surface is the pressure load $\underline{P} = -P\underline{e_2}$ on $x_2 = H$, $x_1 \in [0, L]$. For element (8), the pressure load acts on the edge between node (4) and (1), the x in 1 is x_1, $x^{(1)}$ and $x^{(2)}$ are respectively $L/2$ and L.

The Jacobien value is:

$$\hat{J}_e(b) = \frac{dx}{db} = -\frac{x^{(1)}}{2} + \frac{x^{(2)}}{2} = \frac{L}{4}$$

Application to the general form of $\{F_s^e\}$:

$$\begin{aligned}
\{F_s^e\} &= \int_{\partial\Delta_e} \left[\hat{N}_e(b)\right]^t \cdot \left\{F^d\right\} \hat{J}_e(b) dS_{\Delta_e} \\
&= \int_{-1}^{1} \begin{bmatrix} \frac{1-b}{2} & 0 \\ 0 & \frac{1-b}{2} \\ \frac{1+b}{2} & 0 \\ 0 & \frac{1+b}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -P \end{bmatrix} \frac{L}{4} db \\
&= -\frac{PL}{4} \int_{-1}^{1} \begin{bmatrix} 0 \\ \frac{1-b}{2} \\ 0 \\ \frac{1+b}{2} \end{bmatrix} db
\end{aligned}$$

To integrate, we use one point Gauss-Legendre method as the terms to integrate are first order. $\int_{-1}^{1} f(b)d\Delta_e = \sum w_g f(b_g)$, with $b_g = 0$, $w_g = 2$.

$$\begin{aligned}
\int_{-1}^{1} \frac{1-b}{2} d\Delta_e &= 2 \cdot \frac{1-0}{2} = 1 \\
\int_{-1}^{1} \frac{1+b}{2} d\Delta_e &= 2 \cdot \frac{1+0}{2} = 1
\end{aligned}$$

Finally we have:

$$\left\{F_s^e \text{ (8)}\right\} = -\frac{PL}{4} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

We represent theses nodal forces on the mesh:

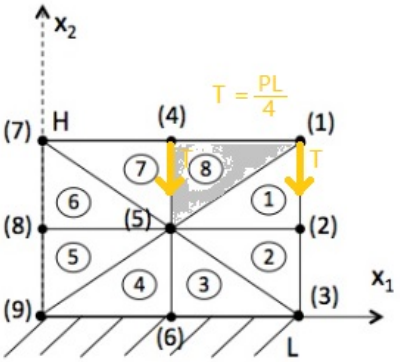


Figure 9: Surface nodal force for element 8

Numerical application 5.3

Define in `python` the required corresponding objects for the numerical computation.

6 Assembly

The system to solve in finite element modelling has the final following form :

$$[K]\{U\} = \{F\} \text{ soit } K_{IJ}U_J = F_I$$

where $[K]$ stands for the stiffness matrix (or conductivity matrix in thermal problems), $\{U\}$ is the dof vector, and $\{F\}$ is obviously the right hand member of the equation.

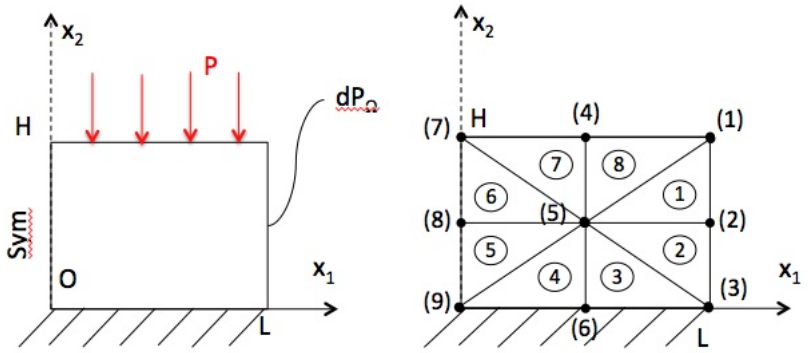


Figure 10: Schematic representation of the structure and its load.

1. Dimensions of $[K]$ and $\{F\}$ before taking into account boundary conditions

   We have a mesh composed of 9 nodes in 2D, so the dimensions of K and F are :

   $$\begin{cases} [K] = (N_N.D) \times (N_N.D) = 18 \times 18 \\ \{F\} = (N_N.D) \times 1 = 18 \times 1 \end{cases}$$

   These dimensions are deduce from the matrices and vectors used to get to K and F, which derive themselves from the mesh's number of nodes and dimension.

2. Geometric support $\Omega^{(i)}$ of each of the 9 nodes (i) of the structure

   It's a feature that will allow us to have more hindsight on the values taken by K matrix.

   Indeed, it's a feature listing the elements a node is attached to. Thereby, most of the nodes are bounded to two elements while the fifth node is bounded to all the elements of mesh [10].

$\Omega^{(i)}$	geometric support
$\Omega^{(1)}$	$\Omega^{\textcircled{1}} \cup \Omega^{\textcircled{8}}$
$\Omega^{(2)}$	$\Omega^{\textcircled{1}} \cup \Omega^{\textcircled{2}}$
$\Omega^{(3)}$	$\Omega^{\textcircled{2}} \cup \Omega^{\textcircled{3}}$
$\Omega^{(4)}$	$\Omega^{\textcircled{7}} \cup \Omega^{\textcircled{8}}$
$\Omega^{(5)}$	Ω
$\Omega^{(6)}$	$\Omega^{\textcircled{3}} \cup \Omega^{\textcircled{4}}$
$\Omega^{(7)}$	$\Omega^{\textcircled{6}} \cup \Omega^{\textcircled{7}}$
$\Omega^{(8)}$	$\Omega^{\textcircled{5}} \cup \Omega^{\textcircled{6}}$
$\Omega^{(9)}$	$\Omega^{\textcircled{4}} \cup \Omega^{\textcircled{5}}$

3. Position of terms equal to zero in $[K]$

For each node (i), every dof which does not have a common element with node (j) has its corresponding term equal to zero in $[K]$, $i, j \in [1, ..., 9]$.

Here below is a graph showing non-zero values :

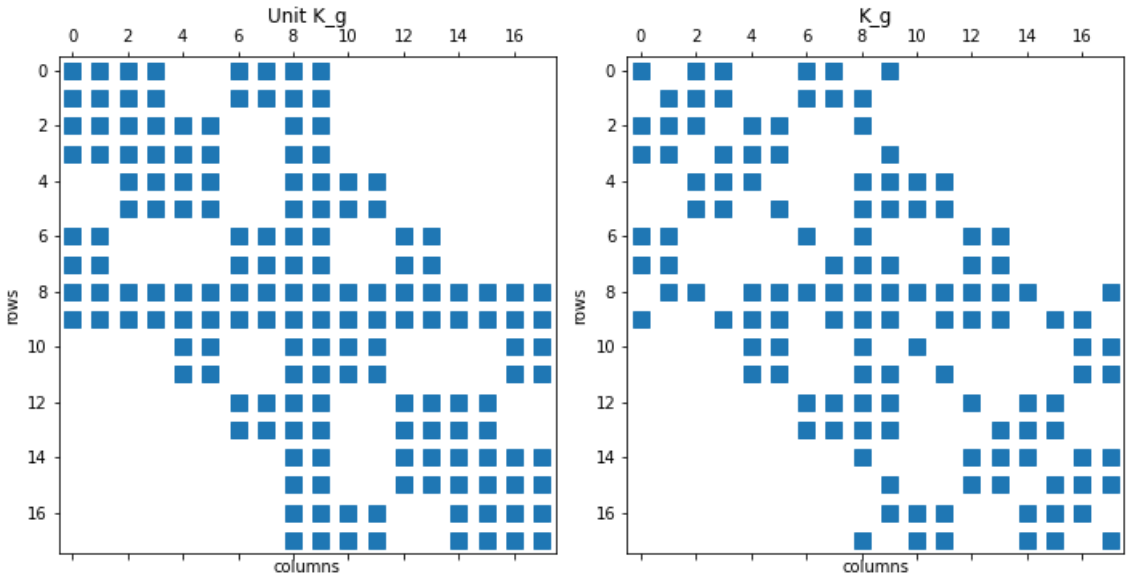


Figure 11: $[K]$.

Note : to have a better understanding of the difference between left and right figure you might have to look at the code.

In fact, while the left figure represents the global stiffness matrix for a unit K_e, the right one represents our current global stiffness matrix. In the code, this K_{unit} is used to verify the disposition of elementary values into global matrix. This allows for instance to make sure of the position in global matrix as well as how much components we have in a specific case.

Moreover, as you can see the two matrices differ. It's due to the fact that some of the values added in global matrix will cancel out in real case (while they can't when we only have 1 as a value in K_{unit}). Thereby, it confirms the use of this unit global matrix K_{unit}

To summarize : white part correspond to zero values, and blue squares to non zero values.

4. Term K_{55} of the stiffness matrix

From above figures, we have a better understanding of where does the elementary component go and how much of them are added in each case of global matrix. However, one might want to make sure the values in K are the one expected from theory.

We can then determinate the term K_{55} of the stiffness matrix by using parametric representation (i.e.shape functions) and it will afterwards be compared to the value from the code.

To this aim the shape of gradient matrices $\left[Be^{(i)}\right]$ for elements (2) and (3) will be useful :

$$\left[Be^{(2)}\right] = \begin{bmatrix} -2/L & 0 & 0 & 0 & 2/L & 0 \\ 0 & 0 & 0 & -2/H & 0 & 2/H \\ 0 & -2/L & -2/H & 0 & 2/H & 2/L \end{bmatrix}$$

$$\left[Be^{(3)}\right] = \begin{bmatrix} 0 & 0 & -2/L & 0 & 2/L & 0 \\ 0 & 2/H & 0 & -2/H & 0 & 0 \\ 2/H & 0 & -2/H & -2/L & 0 & 2/L \end{bmatrix}$$

By definition we know that K matrix is defined as follows :

$$K_{IJ} = \int_{\Omega(m)\cap\Omega(n)} {}^{T}\left\{\underline{\underline{\varepsilon}}(\tilde{N}_m(\underline{x})\underline{e_i})\right\}[A]\left\{\underline{\underline{\varepsilon}}(\tilde{N}_n(\underline{x})\underline{e_j})\right\} dV$$

Yet, the elements 2 and 3 are connected by node 3, and we already know the shape function of this node, which we recall here is :

$$\tilde{N}_3(\underline{x}) = \begin{cases} \frac{2}{L}(x_1 - \frac{L}{2}) & \underline{x} \in (3) \\ \frac{2}{H}(\frac{H}{2} - x_2) & \underline{x} \in (2) \\ 0 \quad else\ where \end{cases}$$

Thanks to this we can calculate K_{55}, using relation $\{\varepsilon_e(\underline{x})\} = [B_e(\underline{a})]\{U_e\}$

and so :

$$K_{55} = \int_{\Omega_2} {}^{T}\begin{Bmatrix} 0 \\ 0 \\ -(\frac{2}{H})^2(-x_2 + \frac{H}{2}) \end{Bmatrix}[A]\begin{Bmatrix} 0 \\ 0 \\ -(\frac{2}{H})^2(-x_2 + \frac{H}{2}) \end{Bmatrix} d\Omega_2 + \int_{\Omega_3} {}^{T}\begin{Bmatrix} (\frac{2}{L})^2(x_1 - \frac{L}{2}) \\ 0 \\ 0 \end{Bmatrix}[A]\begin{Bmatrix} (\frac{2}{L})^2(x_1 - \frac{L}{2}) \\ 0 \\ 0 \end{Bmatrix} d\Omega_3 \quad (2)$$

After integration and knowing A matrix from elasticity relations, we have :

$$K_{55} = \frac{1}{2}(\frac{\mu L}{H} + \frac{(\lambda + 2\mu)H}{L})$$

5. Degree of freedom

$$\begin{bmatrix} K_{11} & \cdots & K_{1\ i-1} & 0 & K_{1\ i+1} & \cdots & K_{1\ N} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ K_{i-1\ 1} & \cdots & K_{i-1\ i-1} & 0 & K_{i-1\ i+1} & \cdots & K_{i-1\ N} \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ K_{i+1\ 1} & \cdots & K_{i+1\ i-1} & 0 & K_{i+1\ i+1} & \cdots & K_{i+1\ N} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ K_{N\ 1} & \cdots & K_{N\ i-1} & 0 & K_{N\ i+1} & \cdots & K_{N\ N} \end{bmatrix} \begin{Bmatrix} U_1 \\ \vdots \\ U_{i-1} \\ U_i \\ U_{i+1} \\ \vdots \\ U_N \end{Bmatrix} = \begin{Bmatrix} F_1 - K_{1\ i}U^d \\ \vdots \\ F_{i-1} - K_{i-1\ i}U^d \\ U^d \\ F_{i+1} - K_{i+1\ i}U^d \\ \vdots \\ F_N - K_{N\ i}U^d \end{Bmatrix}$$

We have 18 theoretical dof, but as mentioned before one must fix some to get unique displacement. In our case and as we consider no restoring force, we have embedded connection to the bottom and symmetry condition to the left ; and so we have now $(18 - 6) = 12$ dof left.

This number of dof amounts to say that 6 movements will be blocked and then that only 12 displacements of the nodes are possible.

6. Kinematical boundary conditions

To find a solution to our problem, we have to impose kinematical boudary conditions. Otherwise, the problem present an infinite number of solutions (namely within a rigid body).

Using the unit diagonal method, we can impose displacements by redefying parts of the matrix K and vector F. This will eventually reduce This method is made like so :

- We look at the positions where a displacement $U_i = U^d$ is imposed
- We then put $F_i = U^D$ to the corresponding value
- It leads $K = 0$ to make the matrix product correct
- Except for $K_{ii} = 1$
- Finally, we substract to F_k the product $K_{ki}U^D$

Numerical application 6.1

In `python` :

- Assemble the K and F.
- Modify K and F to account for the Dirichlet boundary conditions.

7 Solution

1. Numerical parameters

   We work on a structure made of steel

   - $\rho = 7500\ kg.m^{-3}$
   - $E = 210$ GPa
   - $\nu = 0.3$

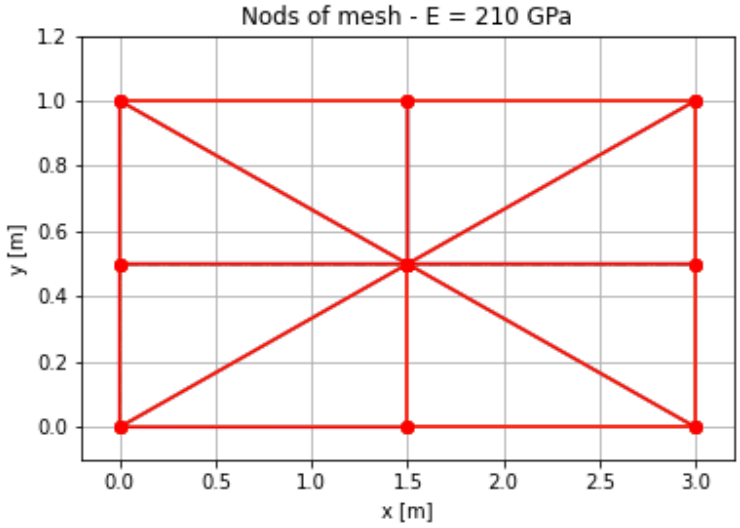


Figure 12: Displacement in physical case

Same force than associated line (i.e. red one) below.

As a recall, the problem is to study the deformation of a 3D structure subjected to a volume and surface force.

By geometric symmetry we have shown that we can reduce this problem to a 2D problem. And then create a mesh of this structure to observe the displacement of its nodes.

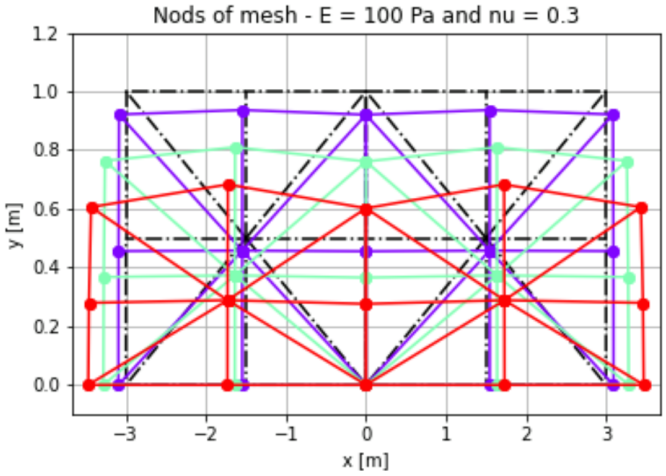
Thanks to this last figure 12 we can observe that a steel structure is not observably deformed. It seems logical because a steel structure is supposed to be rigid, and therefore to be able to withstand this kind of stress (i.e. its own weight and some pressure force).

We have compared two different order of forces, $F = 10^2$N and $F = 10^4$N. The displacements produced by these forces on the structure are respectively for a same node $U = 10^{-10}$m and $U = 10^{-8}$m. The displacement, even though if increasing by a factor 100, stays low, which is in agreement with the assumption of **small displacements**.

2. Solver demonstration

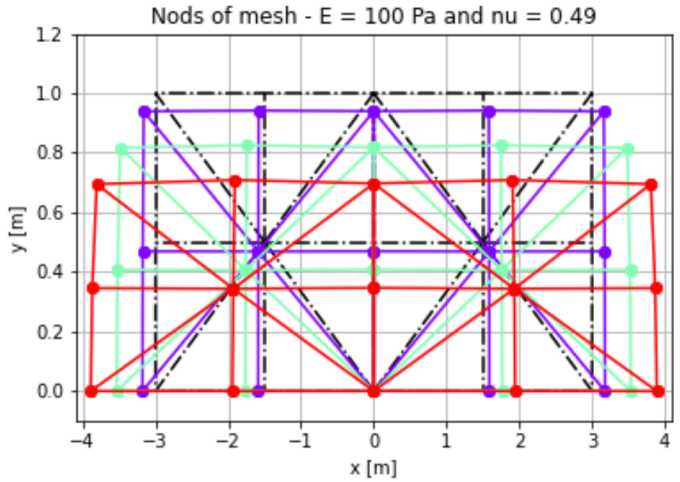
First and foremost, we recall that the studied mesh is in fact half of the initial structure, splitted for simplicity sake. We then have a symmetry in $x = 0$.

Thereby, as we now solved the problem, we will re-combine the two parts to conclude on our analysis. Here below are two graphs showing our overall structure :



Pressure force	Volume force	Color
0	0	- -
$-\frac{5L}{4}$	$-\frac{5g}{24}$	
$-\frac{15L}{4}$	$-\frac{15g}{24}$	
$-\frac{25L}{4}$	$-\frac{25g}{24}$	

Figure 13: Nodes of the mesh for different forces



Pressure force	Volume force	Color
0	0	- -
$-\frac{5L}{4}$	$-\frac{5g}{24}$	
$-\frac{15L}{4}$	$-\frac{15g}{24}$	
$-\frac{25L}{4}$	$-\frac{25g}{24}$	

Figure 14: Nodes of the mesh for different forces

To prove the well functioning of the code, we show strain undergone by a structure very malleable (i.e. $E = 100$, value much lower than physical value which is $E \approx 10^9$). To do so make it possible to see direct effect of the forces we consider.

We present here the mesh of the structure for 3 different imposed forces to have an idea of how the mesh moves when deformed. Moreover, even if we won't treat this case any

further, we display two cases : one for a soft material (i.e. $nu = 0.49$ - rubber) and one for a rigid material (i.e. $nu = 0.30$ - steel).

Firstly, we can easily see that the structure is crushed when we impose a load on the top of it, and also that the right side of the structure is moved to the right. Generally speaking, the mesh moves downwards and flattens under the effect of exterior forces.

Besides, when looking to the nodes where we imposed displacements, we can confirm they don't move and stay still along specified axis.

For what is up to the difference between the two materials, the comparison has the **only purpose** to **show** that **displacement** is being influenced by the material's characteristics and that the displacements showed here are to be **considered carefully**.

Therefore, the point is to look at the overall behavior of the material here, not to conclude on physical values.

As a conclusion, when looking to the overall shapes of both figures, one can see that the **more** the structure is **loaded**, the **greater** the **displacement**, and so is the deformation.

Because of the boundaries conditions, the left and the bottom side don't move, and then the structures is distorted in consequences.

That's what we observe on the figures 13 14

Numerical application 7.1

In `python` :

- Set specific values of the numerical parameters.
- Solve the problem.
- Visualize the solution.

8 Conclusion

As a conclusion, we have been building during this course a set of competences from FEM, which we used here in the case of a rectangular 2D mesh. We then derived equations from theory to then apply them numerically, thanks to a discretisation method, more precisely by using T3 elements (triangles with 3 nodes as reference element).

The process was made to constantly keep trace of the physical problem (BC, material characteristics, forces' intensity), while fitting numerical constraints (sparse matrices, non-unique solution, code optimisation through functions).

For what is up to this coding part, we have been building the most consistent code possible to enable for consideration of any BC as well as own user-defined forces. The graphs displayed here in the end have been a way for us to see how behaves the structure given the forces we apply to it. Doing a general code as we have has the purpose to facilitate future modifications. Therefore, one can add both areal and volumic forces to his convenience. We could similarly considering a restoring force to the bottom instead of an embedded connection.
In the end, the obtained displacement is in agreement with the direction and intensity one could expect given the material's characteristics.

This project was a very good way to understand the lectures through a direct application on both theoretical and numerical aspects. The theory behind the Finite element method being inter-dependant with this numerical aspect, to have had to build piece by piece an autonomous code has favored our keenness to look and understand the course for coding application.

to this work, we are for instance better prepared to continue our study of the FEM in our practical work.

References

[1] Centre des Matériaux. UMR CNRS 7633 Georges Cailletaud. Ecole des Mines de Paris. Finite element : Matrix formulation. http://mms2.ensmp.fr/ef_paris/technologie/transparents/e_ElemFormulation.pdf. Lecture's slides. Last consulted 23-dec-2020.

[2] Heiner Igel. Finite elements : *Basis functions*. https://www.geophysik.uni-muenchen.de/~igel/Lectures/NMG/08_finite_elements_basisfunctions.pdf, July 2010. Lecture's slides. Last consulted 23-dec-2020.