University of Liège - Faculty of Applied Sciences

Academic Year 2015-2016

EVALUATION DE LA QUALITE DE FORMULATIONS ELEMENT FINI CLASSIQUES PAR LA TECHNIQUE FE^2

Graduation Studies conducted for obtaining the Master's degree in Civil Engineering by
Hélène MORCH

Jury: L. Duchêne, F. Collin, R. Boman, C. Henrard

Remerciements

Je tiens à remercier les personnes qui m'ont aidé dans la réalisation de mon travail de fin d'études.

En premier lieu, je remercie M. Duchêne, mon promoteur de Travail de Fin d'Etudes, qui m'a accompagnée dans mon travail et m'a aidée à avancer.

Ensuite, je remercie M. Henrard, ingénieur chez SAMTECH, qui m'a fourni des résultats complémentaires pour enrichir mon travail de fin d'études.

Abstract

Evaluation of the quality of classic finite elements by the FE² method

MORCH Hélène, Civil Engineering, Academic Year 2015-2016

Locking phenomena in classic finite element analysis are a well-known problem and many solutions have been developed over the years to reduce or suppress their undesirable effects. The most common types of locking are shear locking, that can occur with slender elements, and volumetric locking, that can occur when the material is quasi-incompressible (with a Poisson's ratio close to 0.5, for instance). The goal of this study is to assess the performance of finite elements created using the FE² technique. This technique consists in creating a sub-mesh on each element and was developed to help model materials with complex behaviours. In this study, however, the FE² technique is used in the elastic domain with homogeneous isotropic materials.

To conduct this study, a finite element program was created using MATLAB to solve two dimensional elastic problems. This program was used to create the equivalent stiffness matrices of the FE² elements, and then to test these elements.

Finally, the elements created were tested for each type of locking mentioned above. The test on shear locking (a cantilever beam subject to bending) revealed that even though the FE² method showed some improvement compared to classic fully integrated elements, shear locking still appeared and was non-negligible. To test the appearance of volumetric locking, a cylinder subjected to internal pressure was studied for different values of the Poisson's ratio. The results on volumetric locking, on the other hand, are very promising as the new elements developed showed little to no locking. The results on volumetric locking were also compared to results obtained with industrial elements, and showed better or equivalent performances.

Les phénomènes de blocage (« locking », en anglais) dans la méthode des éléments finis sont un problème bien connu et diverses solutions ont été développées au fil des années pour réduire ou éliminer leurs effets. Les types de blocages les plus courants sont le blocage en cisaillement, qui peut se produire sur des éléments élancés, et le blocage volumétrique qui se produit lorsque le matériau étudié est quasi-incompressible (coefficient de Poisson proche de 0,5, par exemple). Le but de cette étude est de déterminer la performance d'éléments finis créés en utilisant le méthode FE². Cette méthode consiste à créer un sous-maillage sur chaque élément. Elle a été développée pour modéliser des matériaux au comportement complexe. Dans cette étude, cependant, la méthode FE² est utilisée dans le domaine élastique avec des matériaux homogènes isotropes.

Pour réaliser cette étude, un programme éléments finis a été développé dans MATLAB pour résoudre des problèmes en deux dimensions dans le domaine élastique. Ce programme sert aussi à créer les matrices de raideur des éléments FE² et à tester ces éléments.

Finalement, les éléments créés sont testés contre les types de blocage mentionnés ci-dessus. Le test sur le blocage en cisaillement (poutre cantilever en flexion) montre que la méthode FE² apporte une amélioration par rapport à des éléments classiques complètement intégrés, mais le blocage apparaît et n'est pas négligeable. Pour tester l'apparition du blocage volumétrique, nous avons étudié un cylindre soumis à une pression interne radiale pour différentes valeurs du coefficient de Poisson. Les résultats sur ce type de blocage sont très encourageants, puisque les éléments FE² sont peu voire pas sensibles au phénomène. En comparaison avec des éléments industriels, les éléments FE² offrent des performances équivalentes, voire meilleures sur le blocage volumétrique.

Table of contents

Introduction 6
1 Finite element analysis 7
1.1 Mechanics of materials for elastic solid objects 7
1.2 Finite element method 8
1.2.1 Discretisation 8
1.2.2 Virtual work principle 9
1.2.3 Parametric system 10
1.2.4 Numerical integration 12
1.2.5 Convergence 13
1.3 Locking problems in classic finite elements 14
1.3.1 Shear locking 14
1.3.2 Volumetric locking 15
2 Methodology 16
2.1 Finite Element Program 16
2.1.1 Meshing 16
2.1.2 Resolution 17
2.1.2.1 Computation of the elementary stiffness matrices and load vector 17
2.1.2.2 Assembly process 18
2.1.2.3 Boundary conditions 19
2.1.3 Numerical tests 21
2.1.3.1 Rigid modes 21
2.1.3.2 Constant strain test 23
2.1.3.3 Shear test 25
2.1.3.4 Patch test 26
2.2 FE² elements 28
2.2.1 Method of construction 28
2.2.1.1 Equivalent stiffness matrix 28
2.2.1.2 Deformation of the element 29
2.2.1.3 Reactions at the nodes 32
2.2.1.4 Equivalent reactions at the vertices 34
2.2.1.5 Size of the displacement 36

2.2.1.6 Size of the sub-mesh 38
2.2.2 Rotation and size of the element 39
2.2.2.1 Homothetic transformation 39
2.2.2.2 Rotation 40
2.2.3 Tests 43
2.2.3.1 Rigid modes 43
2.2.3.2 Constant strain test 45
2.2.3.3 Shear test 46
2.2.3.4 Patch test 46
3 Performance of the FE² element on locking problems 49
3.1 Performance on shear locking 49
3.1.1 Description of the test 49
3.1.1.1 Meshing 50
3.1.1.2 Geometry and size of mesh 51
3.1.2 Results 52
3.1.3 Convergence 54
3.1.4 Comparison with other types of elements 56
3.1.4.1 Description of the problem 56
3.1.4.2 Load vector 57
3.1.4.3 Results 59
3.3 Performance on volumetric locking 62
3.3.1 Description of the test 62
3.3.1.1 Geometry and material properties 62
3.3.1.2 Meshing 63
3.3.1.3 Load vector 64
3.3.2 Element 66
3.3.3 Results 67
3.3.5 Comparison with an industrial element 70
Conclusion 72
References 74
1 Appendix 1: READ_MESH.m ..... **Erreur ! Signet non défini.**

2 Appendix 2: FE² stiffness matrices for the patch test ..... **Erreur ! Signet non défini.**

3 Appendix 3: FE² stiffness matrices of the rectangular elements**Erreur ! Signet non défini.**

3.1 Matrices for the cantilever beam with unitary tip displacement**Erreur ! Signet non défini.**

3.2 Matrices for the cantilever beam with sinusoidal moment loading**Erreur ! Signet non défini.**

Introduction

The finite element method is widely used in various domains, and more particularly in the field of mechanics of materials, to solve complex problems that cannot be solved through manual calculations. However, the finite element method is not infallible.

There are specific cases for which the results given by the classic finite element method are erroneous. This can be caused by a particular behaviour of the material, such as incompressibility for instance, or by a specificity in the geometry of the problem (slender elements for example). These problems are known as locking phenomena, due to the fact that the elements usually show an over-stiff behaviour and fail to reproduce the correct deformation of the solid object.

Solutions to these problems have been developed, which often tackle one specific locking phenomenon. The goal of this study is to assess the performance of the Finite Element square (FE^2) method against locking, compared to the classic elements.

The Finite Element square (FE^2) method is a recently developed method used to render with more precision the behaviour of certain materials. It is used mostly for materials with heterogeneous properties, as a way of obtaining homogeneous properties over each element. In this purpose, each element is refined into a sub-mesh to establish the material's behaviour on a smaller scale.

The point of this study, however, is to test a different utility of this method. Here, the FE^2 method is used to create new sorts of elements and test their performance against locking. The problems studied are limited to elastic two-dimensional problems. The study focuses particularly on the problems of shear locking encountered with slender elements, and of volumetric locking that happens with incompressible materials.

This study aims at comparing the performance of the FE^2 technique compared to the classic finite elements.

In a first part, the theory of the classic finite element method is reminded, and the problems of locking studied in this work are explained. The second part of this work focuses on the method put in place, first to develop a finite element program, and then to create the new elements using the FE^2 method. Finally, the results obtained with the new elements against shear locking and volumetric locking are presented.

1 Finite element analysis

The finite element method is a technique that allows the numerical resolution of differential equation(s) on a given domain with boundary conditions. It can be used to determine the behaviour of solid objects under a given set of loads or displacements. The general principle of the finite element method is to divide the studied domain into sub-domains called elements.

This work focuses exclusively on two-dimensional linear elastic problems. The equations used to determine the behaviour of the solid object are the equations from mechanics of materials reminded hereafter.

1.1 Mechanics of materials for elastic solid objects

In the elastic domain, the behaviour of a two-dimensional isotropic solid object can be described by Hooke's law:

$$\{\sigma\} = [C]\{\varepsilon\} \qquad [1.1.1]$$

With:

- $\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$ is the stress vector;
- $\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}$ is the strain vector;
- $[C] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} for\ the\ plane\ strain\ state$

  $or\ [C] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} for\ the\ plane\ stress\ state.$

For a solid object subjected to a body force $\langle F \rangle = \langle F_x\ F_y \rangle$ and to a surface force $\langle T \rangle = \langle T_x\ T_y \rangle$, the equations of equilibrium can be written as follows:

$$\langle \partial\sigma \rangle + \langle F \rangle = 0, where\ \{\partial\sigma\} = \begin{Bmatrix} \frac{\partial\sigma_x}{\partial x} + \frac{\partial\tau_{xy}}{\partial y} \\ \frac{\partial\sigma_y}{\partial y} + \frac{\partial\tau_{xy}}{\partial x} \end{Bmatrix} \qquad [1.1.2]$$

$$\langle T \rangle - \langle \sigma_n \rangle = 0, where\ \{\sigma_n\} = \begin{Bmatrix} \sigma_x n_x + \tau_{xy} n_y \\ \sigma_y n_y + \tau_{xy} n_x \end{Bmatrix} \qquad [1.1.3]$$

To use the finite element method, these equilibria have to be expressed through a weak formulation.

The weak formulation is obtained through the weighted residuals method.

Given a virtual displacement field $\langle \delta u^* \rangle$, the weighted residual corresponding to the equilibrium equations is expressed as follows:

$$WR = \int_V [\langle \partial\sigma \rangle + \langle F \rangle]\{\delta u^*\} dV + \int_A [\langle T \rangle - \langle \sigma_n \rangle]\{\delta u^*\}\, dA = 0 \qquad [1.1.4]$$

Where V is the volume of the domain and A is the contour of the domain.

Using an integration by parts, the principle of virtual works can be obtained:

$$W_{int} = W_{ext}$$

$$\int_V \langle \sigma \rangle \{\delta\varepsilon^*\}\, dV = \int_V \langle F \rangle \{\delta u^*\} dV + \int_A \langle T \rangle \{\delta u^*\}\, dA \qquad [1.1.5]$$

The virtual strain field can be expressed as a function of the virtual displacement:

$$\begin{Bmatrix} \delta\varepsilon_x^* = \dfrac{\partial \delta u_x^*}{\partial x} \\ \delta\varepsilon_y^* = \dfrac{\partial \delta u_y^*}{\partial y} \\ \delta\gamma_{xy}^* = \dfrac{\partial \delta u_y^*}{\partial x} + \dfrac{\partial \delta u_x^*}{\partial y} \end{Bmatrix}$$

1.2 Finite element method

The application of the finite element method necessitates the division of the solid object into sub-domains: the elements. This allows for a discrete application of the principle of virtual works on the solid object.

1.2.1 Discretisation

Each element is delimited by a group of nodes. The displacements U_x^i *and* U_y^i are computed at the nodes, as shown in Figure 1.2-1.

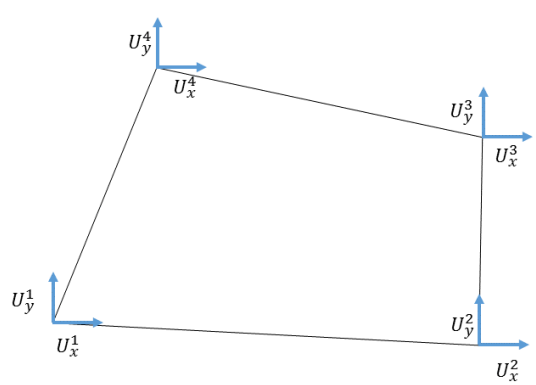


Figure 1.2-1 - Nodal displacements on a 4-node quadrangle

The strain field, displacement field, and stress field are expressed on the element through interpolation functions that allow a continuous expression of the field on each element based on the nodal values.

The discrete formulation of the displacement on an element is expressed as follows:

$$\begin{Bmatrix} u_x^*(x,y) \\ u_y^*(x,y) \end{Bmatrix} = \begin{bmatrix} h^1(x,y) & 0 & \cdots & h^N(x,y) & 0 \\ 0 & h^1(x,y) & & 0 & h^N(x,y) \end{bmatrix} \begin{Bmatrix} U_x^1 \\ U_y^1 \\ \vdots \\ U_x^N \\ U_y^N \end{Bmatrix}$$

$$\Leftrightarrow \{u^*(x,y)\} = [N(x,y)]\{q\} \qquad [1.2.1]$$

Where $h^i(x,y)$ are the shape functions. The shape functions are continuous functions of the two variables x and y and take the following value on a node j:

$$h^i(x_j, y_j) = \delta_{ij}$$

Where x_j *and* y_j are the coordinates of node j, and δ_{ij} is the Kronecker delta.

In the same way as the displacement field, the strain field and the stress field can be expressed using the interpolation functions:

$$\begin{Bmatrix} \epsilon_x^*(x,y) \\ \epsilon_y^*(x,y) \\ \gamma_{xy}^*(x,y) \end{Bmatrix} = \begin{bmatrix} \frac{\partial h^1}{\partial x} & 0 & & \frac{\partial h^N}{\partial x} & 0 \\ 0 & \frac{\partial h^1}{\partial y} & \cdots & 0 & \frac{\partial h^N}{\partial y} \\ \frac{\partial h^1}{\partial y} & \frac{\partial h^1}{\partial x} & & \frac{\partial h^N}{\partial y} & \frac{\partial h^N}{\partial x} \end{bmatrix} \begin{Bmatrix} U_x^1 \\ U_y^1 \\ \vdots \\ U_x^N \\ U_y^N \end{Bmatrix}$$

$$\Leftrightarrow \{\epsilon^*(x,y)\} = [B(x,y)]\{q\} \qquad [1.2.2]$$

Using Hooke's law: $\{\sigma^*(x,y)\} = [C]\{\epsilon^*(x,y)\} = [C][B(x,y)]\{q\}$

1.2.2 Virtual work principle

Using the discretised expressions, the virtual work principle can be rewritten in a discrete form.
For any infinitesimal virtual displacement field δu^* that fulfils the compatibility of the solid:

$$[1.1.5] \Leftrightarrow \sum_{El=1}^{N_{EL}} \left(\int_{V_{EL}} \langle \delta\varepsilon^* \rangle \{\sigma\}\, dV \right) = \sum_{EL=1}^{N_{EL}} \left(\int_{V_{EL}} \langle \delta u^* \rangle \{F\} dV + \int_{A_{EL}} \langle \delta u^* \rangle \{T\}\, dA \right)$$

Where:

- N_{EL} is the number of elements used to divide the domain;
- V_{EL} is the volume of the element;

- A_{EL} is the contour of the element.

$$[1.1.5] \Leftrightarrow \sum_{El=1}^{N_{EL}} \left(\int_{V_{EL}} \langle \delta q^* \rangle_{EL} [B(x,y)]^T [C][B(x,y)] \{q\}_{EL} \, dV \right)$$

$$= \sum_{EL=1}^{N_{EL}} \left(\int_{V_{EL}} \langle \delta q^* \rangle_{EL} [N(x,y)]^T \{F\} dV + \int_{A_{EL}} \langle \delta q^* \rangle_{EL} [N(x,y)]^T \{T\} \, dA \right)$$

$$\Leftrightarrow \sum_{El=1}^{N_{EL}} \left(\langle \delta q^* \rangle_{EL} \int_{V_{EL}} [B(x,y)]^T [C][B(x,y)] \, dV \{q\}_{EL} \right)$$

$$= \sum_{EL=1}^{N_{EL}} \left(\langle \delta q^* \rangle_{EL} \int_{V_{EL}} [N(x,y)]^T \{F\} dV + \langle \delta q^* \rangle_{EL} \int_{A_{EL}} [N(x,y)]^T \{T\} \, dA \right)$$

$$\Leftrightarrow \langle \delta q^* \rangle [K] \{q\} = \langle \delta q^* \rangle \{Q\} \qquad \forall \langle \delta q^* \rangle$$

$$\Leftrightarrow [K]\{q\} = \{Q\} \qquad \forall \langle \delta q^* \rangle \qquad [1.2.3]$$

The stiffness matrix $[K]$ and the vector of equivalent nodal loads $\{Q\}$ are obtained through assembling the element stiffness matrices $[K_{EL}]$ and load vectors $\{Q_{EL}\}$, with:

$$[K_{EL}] = \int_{V_{EL}} [B(x,y)]^T [C][B(x,y)] \, dV \qquad [1.2.4]$$

$$\{Q_{EL}\} = \int_{V_{EL}} [N(x,y)]^T \{F\} dV + \int_{A_{EL}} [N(x,y)]^T \{T\} \, dA \qquad [1.2.5]$$

The expression of $[K_{EL}]$ and $\{Q_{EL}\}$ depend on the interpolation functions $h^i(x,y)$.

1.2.3 Parametric system

In order to simplify the expression of the interpolation functions for an undefined geometry, the parametric representation of the element is used to define the interpolation functions, as shown in Figure 1.2-2.

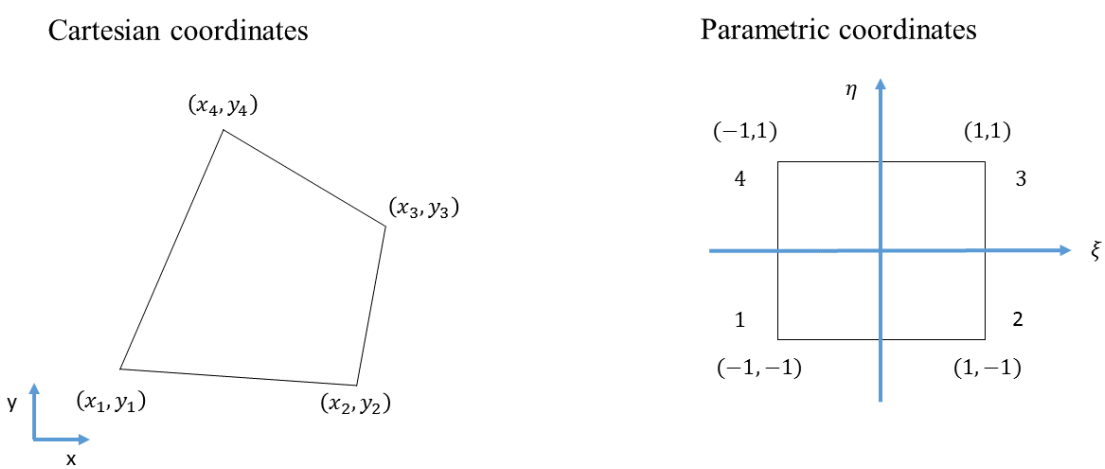


Figure 1.2-2 - Cartesian representation and parametric representation of a 4-node element

The interpolation functions can be written in the parametric system of coordinates:

$$\begin{cases} h^1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) \\ h^2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta) \\ h^3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta) \\ h^4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta) \end{cases} \qquad [1.2.6]$$

The expression of the Cartesian coordinates can be expressed as functions of the parametric coordinates ξ and η:

$$x = \sum_{i=1}^{4} h^i(\xi,\eta) X_i$$
$$y = \sum_{i=1}^{4} h^i(\xi,\eta) Y_i \qquad [1.2.7]$$

From [1.2.6] and [1.2.7], the Jacobian matrix can be determined to perform the transformation from the Cartesian system to the parametric system:

$$[J] = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dx}{d\eta} \\ \frac{dy}{d\xi} & \frac{dy}{d\eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{\partial h^i(\xi,\eta)}{\partial \xi} X_i & \sum_{i=1}^{4} \frac{\partial h^i(\xi,\eta)}{\partial \eta} X_i \\ \sum_{i=1}^{4} \frac{\partial h^i(\xi,\eta)}{\partial \xi} Y_i & \sum_{i=1}^{4} \frac{\partial h^i(\xi,\eta)}{\partial \eta} Y_i \end{bmatrix} \qquad [1.2.8]$$

An infinitesimal volume inside the element in the Cartesian space is defined as: $dV = e * dxdy$, where e is the thickness of the element (considered uniform in the initial configuration in a 2D analysis). In the parametric space, the volume element is: $dV = e * J(\xi,\eta) d\xi d\eta$, where $J(\xi,\eta) = \det[J(\xi,\eta)]$

Consequently, the expression of the stiffness matrix [1.2.4] and the equivalent load vector [1.2.5] can be written using the parametric system:

$$[K_{EL}] = \int_{V_{EL}} [B(x,y)]^T [C][B(x,y)]\, dV = e \int_{\Omega_{EL}} [B(x,y)]^T [C][B(x,y)]\, dxdy = e \iint_{-1}^{1} [B(\xi,\eta)]^T [C][B(\xi,\eta)] J(\xi,\eta)\, d\xi d\eta \qquad [1.2.9]$$

Where Ω_{EL} is the surface of the element.

$$\{Q_{EL}\} = \int_{V_{EL}} [N(x,y)]^T \{F\} dV + \int_{A_{EL}} [N(x,y)]^T \{T\}\, dA$$
$$= e \iint_{-1}^{1} [N(\xi,\eta)]^T \{F\} J(\xi,\eta) d\xi d\eta + e \int_{Contour} [N(\xi,\eta)]^T \{T\} J(\xi,\eta) d\xi d\eta \qquad [1.2.10]$$

Where:

$$[N(\xi,\eta)] = \begin{bmatrix} h^1(\xi,\eta) & 0 & \cdots & h^4(\xi,\eta) & 0 \\ 0 & h^1(\xi,\eta) & & 0 & h^4(\xi,\eta) \end{bmatrix} \qquad [1.2.11]$$

$$[B(\xi,\eta)] = \begin{bmatrix} \frac{\partial h^1(\xi,\eta)}{\partial x} & 0 & & \frac{\partial h^4(\xi,\eta)}{\partial x} & 0 \\ 0 & \frac{\partial h^1(\xi,\eta)}{\partial y} & \dots & 0 & \frac{\partial h^4(\xi,\eta)}{\partial y} \\ \frac{\partial h^1(\xi,\eta)}{\partial y} & \frac{\partial h^1(\xi,\eta)}{\partial x} & & \frac{\partial h^4(\xi,\eta)}{\partial y} & \frac{\partial h^4(\xi,\eta)}{\partial x} \end{bmatrix} \quad [1.2.12]$$

The partial derivative of the interpolation functions with respect to x or y are obtained via the Jacobian matrix:

$$\begin{Bmatrix} \frac{\partial h^i(\xi,\eta)}{\partial x} \\ \frac{\partial h^i(\xi,\eta)}{\partial y} \end{Bmatrix} = [J]^{-T} \begin{Bmatrix} \frac{\partial h^i(\xi,\eta)}{\partial \xi} \\ \frac{\partial h^i(\xi,\eta)}{\partial \eta} \end{Bmatrix} \quad [1.2.13]$$

1.2.4 Numerical integration

In order to implement the finite element method in a program, the stiffness matrices and load vectors have to be calculated through a numerical integration.

On a 4-node quadrangle, the stiffness matrix can be calculated using a Gaussian quadrature using 1 or 4 Gauss points (i.e. integration points) to calculate the integral from Equation [1.2.9].

The expression of the stiffness matrix is:

$$[K_{EL}] = e \sum_{IP=1}^{N_{IP}} [B(\xi_{IP},\eta_{IP})]^T [C][B(\xi_{IP},\eta_{IP})] J(\xi_{IP},\eta_{IP}) W_{IP} \quad [1.2.14]$$

Where (ξ_{IP},η_{IP}) are the coordinates of the Gauss points in the parametric system, and W_{IP} is its weight.

The coordinates of the Gauss points and their respective weight are given in Table 1.

Table 1 - Gauss points coordinates and weight

N_{IP}	IP	ξ_{IP}	η_{IP}	W_{IP}
1	1	0	0	4
4	1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1
	2	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1
	3	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1
	4	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1

It can be noted that for a 4-node quadrangle, the stiffness matrix calculated with a Gaussian quadrature using 4 points gives the same result as the fully integrated matrix, as mentioned in J-L. Batoz, G. Dhatt [1].

However, when only one Gaussian point is used, the element can develop cinematic modes, which means it can deform without any load applied. This happens because the stiffness matrix of the element is singular, therefore, there are vectors $\{q\}_{EL}$ different from the null vector that verify:

$$(\delta W_I)_{EL} =< \delta q >_{EL} [K_{EL}]\{q\}_{EL} = 0 \qquad [1.2.15]$$

This leads to a phenomenon called 'hourglassing', due to the shape the elements take (see Figure 1.2-3).

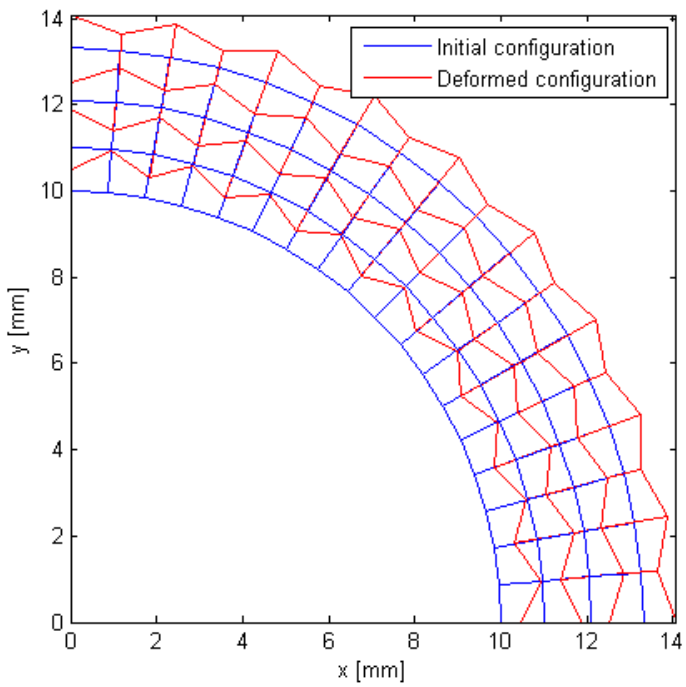


Figure 1.2-3 - Example of hourglassing on a tube loaded with an internal pressure

1.2.5 Convergence

For a given application of the finite element method, there are two ways of converging to a result closer to the theoretical result.

The first method is to increase the order of the interpolation functions, while keeping the same mesh. To represent higher order shape functions, the number of nodes on each element has to be increased as well (second order polynomial shape functions require 3 nodes per edge, third order require 4 nodes per edge, etc.), which leads to longer computation times and new developments.

The second method consists in refining the mesh, by using smaller elements. This also increases the time of calculation.

1.3 Locking problems in classic finite elements

Different locking phenomena can occur in a standard finite element analysis. When locking happens, the displacements computed by the analysis may be smaller than those expected (that can be obtained from an analytical analysis). This is due to an excess of stiffness in the element which usually appears when the stiffness matrix is computed with a full integration.

This work will focus on the two most frequent locking problems that can be observed in finite elements: shear locking and volumetric locking.

1.3.1 Shear locking

In a 2-dimension analysis, shear locking can occur when using 4-nodes quadrangles subjected to bending.

The interpolation functions of a 4-node quadrangle are linear. Therefore, the displacement field on each element will also be linear. However, when subjected to a pure bending moment, the element should assume a curved shape as shown in Figure 1.3-1**,** where the cross sections form a 90° angle with the neutral axis. Because of the shape the element actually assumes, the cross sections are not at 90° to the neutral axis (see Figure 1.3-2).

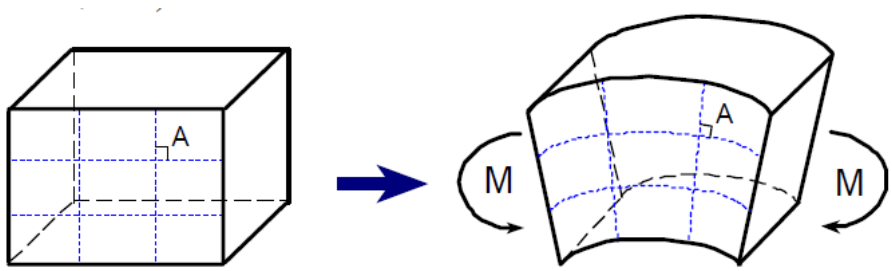


Figure 1.3-1 - Shape change of the material block under the moment in the ideal situation - From E. Qiuli Sun [2]

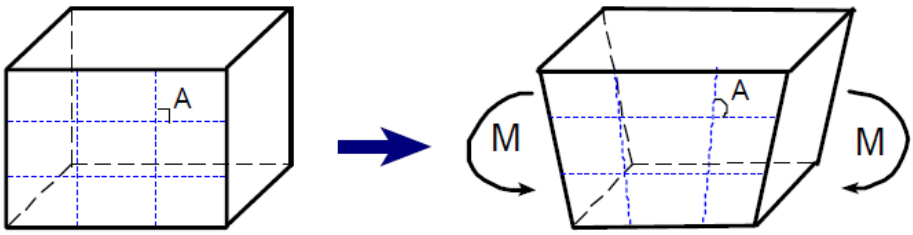


Figure 1.3-2 - Shape change of the fully integrated first order element under the moment - from E. Qiuli Sun [2]

Because the element is unable of reproducing this curved shape, it develops an artificial shear stress to allow the cross sections not to remain perpendicular to the neutral axis. Because of that, the strain energy of the element generates shear deformation instead of bending deformation. This results in displacements smaller than the theoretical displacements under pure bending.

A possible solution to that problem would be to use higher order elements, such as 8-nodes quadrangles, with parabolic interpolation functions, which allow a correct representation of the displacement field on each element. However, the use of this type of element would increase the computation time.

Another solution is to calculate the stiffness matrix using only one Gauss point. This decreases the element stiffness and can give better results in some cases, however the use of a one-point Gaussian quadrature usually gives less accurate results and can lead to hourglassing.

1.3.2 Volumetric locking

In the elastic domain, volumetric locking is encountered with quasi-incompressible materials (with a Poisson's ratio close to 0.5) in plane strain state.

This is due to the fact that the shape functions of the element are unable to approximate a volume preserving strain field, as explained in A.F. Bower, [5].

For a 2D analysis, Hooke's law can be written as follows for the plane strain state (see [1.1.1]):

$$\{\sigma\} = [C]\{\varepsilon\}$$

Where $[C] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$

When the Poisson's ratio is close to 0.5, the term $\frac{E}{(1+\nu)(1-2\nu)}$ becomes close to $+\infty$, which leads to an increase in the stiffness of the element.

2 Methodology

2.1 Finite Element Program

The first step of this work was the creation of a code that could be used to carry out a two-dimensional finite element analysis.

The code, written using the MATLAB software, contains several scripts that are necessary to carry out the analysis:

- the MAIN script contains the different parameters (Young's modulus, Poisson's ratio, number of Gauss points, loads, fixed nodes, displacements…);
- from the MAIN script, a second script is used to read or to generate the mesh of the domain;
- the EF_2D script generates the global stiffness matrix and the global vector of equivalent loads;
- the PLOT and RESULTS scripts generate a graphical representation of the displacement and of the stress in the domain.

2.1.1 Meshing

The software GMSH [3] was used in order to generate the mesh.

The geometry of the studied object is defined from points, oriented lines, and oriented surfaces. Once the geometry is defined, the mesh can be generated. By default, GMSH generates meshes using triangular elements. To obtain a mesh with quadrilateral elements, the triangular mesh is recombine using the "Blossom" recombination algorithm. This algorithm is the default recombination algorithm.

The mesh can then be refined using different global mesh size factors.

GMSH does not allow the direct input of boundary conditions such as supports, but nodes, lines, and surfaces can be sorted by assessing them different physical groups.

GMSH generates a ".msh" file containing the node numbers and coordinates, the element numbers, their type (linear, triangular, quadrilateral, …), their physical group(s), and the nodes they connect.

In the program, the script "READ_MESH.m" was created to transform the data from the .msh file into matrices that can be used with MATLAB. This script was created from an already existing script internal to the university that has been written for another project. It is presented in Appendix 1.

The first part of the script creates a matrix containing the numbers of the nodes and their coordinates.

The second part is used to define boundary conditions from the physical groups. The following convention was used to define different boundary conditions: the fixations are defined using a two-digit number "XY". The first digit, "X", corresponds to the type of element and take the following values: 1 for a single node, 2 for a line (containing several

nodes). The second digit "Y" corresponds to the direction in which the node is fixed. Table 2 contains the signification of the different values of "Y".

Table 2 - Physical groups for node fixation

Digit Y	Signification
1	Node fixed in the horizontal direction
2	Node fixed in the vertical direction
3	Node fixed in both directions

Finally, a matrix containing all the elements is generated from the .msh file.

2.1.2 Resolution

As explained in 1.1., the objective of the finite element program is to find a solution to equation [1.2.3].

Once the mesh is determined, the global stiffness matrix and the global load vector must be generated from the elementary stiffness matrices, the boundary conditions, and the loads imposed on the system.

2.1.2.1 Computation of the elementary stiffness matrices and load vector

The elementary stiffness matrices are computed using a Gaussian quadrature, using equation [1.2.14].

The code allows to choose between a numerical integration with 1 Gauss point or 4 Gauss points. This is done with the parameter "NIP" that has to be defined in the MAIN script.

From this value, the parametric coordinates of the Gauss points are defined:

```
if NIP==1
    WIP=[4];
    ksiIP=[0];
    etaIP=[0];
elseif NIP==4
    WIP=[1 1 1 1];
    ksiIP=[-1/(sqrt(3)) -1/sqrt(3) 1/sqrt(3) 1/sqrt(3)];
    etaIP=[-1/(sqrt(3)) 1/sqrt(3) -1/sqrt(3) 1/sqrt(3)];
end
```

Once this is defined, the elementary stiffness matrix can be computed for every element, using [1.2.14]. The load vector, however, is directly input as a global vector with the data in the MAIN script.

```
% Parametric coordinates of the element nodes
KSI=[-1 1 1 -1];
ETA=[-1 -1 1 1];
for i=1:Nel
    Kel=zeros(2*Nnode,2*Nnode);   %Initialisation   of   the   elementary
stiffness matrix
    % Cartesian coordinates of the element
```

```
    for j=1:Nnode
        X(j)=nodes(elem(i,j),2);
        Y(j)=nodes(elem(i,j),3);
    end
    %% Computation of the elementary stiffness matrix
        for j=1:NIP
            eta=etaIP(j);
            ksi=ksiIP(j);
            %Computation of the Jacobian matrix and determinant
            J11=(1-eta)*(X(2)-X(1))+(1+eta)*(X(3)-X(4));
            J12=(1-ksi)*(X(4)-X(1))+(1+ksi)*(X(3)-X(2));
            J21=(1-eta)*(Y(2)-Y(1))+(1+eta)*(Y(3)-Y(4));
            J22=(1-ksi)*(Y(4)-Y(1))+(1+ksi)*(Y(3)-Y(2));
            J=1/4*[J11 J12 ; J21 J22];
            detJ=det(J);
            %Computation of matrices B and N
            dH=zeros(2,Nnode);
            for k=1:Nnode
                h=1/4*(1+KSI(k)*ksi)*(1+ETA(k)*eta);
                dh=1/4*[KSI(k)*(1+ETA(k)*eta) ; ETA(k)*(1+ksi*KSI(k))];
                dH(:,k)=transpose(J)\dh;
                B(1,2*k-1,NIP*(i-1)+j)=dH(1,k);
                B(2,k*2,NIP*(i-1)+j)=dH(2,k);
                B(3,2*k-1,NIP*(i-1)+j)=dH(2,k);
                B(3,k*2,NIP*(i-1)+j)=dH(1,k);
                N(1,2*k-1,NIP*(i-1)+j)=h;
                N(2,2*k,NIP*(i-1)+j)=h;
            end
            Kel=Kel+transpose(B(:,:,NIP*(i-1)+j))*C*B(:,:,NIP*(i-
            1)+j)*detJ*WIP(j);
        end
```

2.1.2.2 Assembly process

Once the elementary matrix has been computed, its components must be added to the global stiffness matrix through the assembly process.

The elements $K_{el}(i,j), \begin{cases} i = 2f \text{ or } 2f-1 \\ j = 2g \text{ or } 2g-1 \end{cases}$ represents the relation between local nodes g and f, where $(g,f) \in [\![1,4]\!]^2$. This contribution must be added to the global stiffness matrix, using the global numbering of the nodes:

```
for i=1:Nel
…     % Computation of the elementary stiffness matrix
% Assembly process
    for f=1:Nnode
        a=elem(i,f);            % Global number of local node f
        for g=1:Nnode
            b=elem(i,g);        % Global number of local node g
            Kgl(2*a-1,2*b-1)=Kgl(2*a-1,2*b-1)+Kel(2*f-1,2*g-1);
            Kgl(2*a-1,2*b)=Kgl(2*a-1,2*b)+Kel(2*f-1,2*g);
            Kgl(2*a,2*b-1)=Kgl(2*a,2*b-1)+Kel(2*f,2*g-1);
            Kgl(2*a,2*b)=Kgl(2*a,2*b)+Kel(2*f,2*g);
        end
    end
end
```

2.1.2.3 Boundary conditions

The boundary conditions include fixed nodes and imposed displacements.

The fixations are defined in the 3-column matrix `fixed_DOF`. The first column contains the node number. The second column takes the value 1 if the node is fixed in the horizontal direction, 0 if it is not. The third column takes the value 1 if the node is fixed in the vertical direction, 0 if it is not.

When a node is fixed in one direction, the global stiffness matrix and the global load vector have to be modified. For instance, if a node k is fixed in the horizontal direction, the equation $U_{2k-1} = 0$ has to be included in the system described by Equation [1.2.3]. This is done by changing the stiffness matrix and the load vector as follows:

$$\begin{bmatrix} K_{1,1} & \cdots & 0 & K_{1,2k} & \cdots & K_{1,2N} \\ K_{2,1} & \cdots & 0 & K_{2,2k} & \cdots & K_{2,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ K_{2k,1} & \cdots & 0 & K_{2k,2k} & \cdots & K_{2k,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N-1,1} & \cdots & 0 & K_{2N-1,2k} & \cdots & K_{2N-1,2N} \\ K_{2N,1} & \cdots & 0 & K_{2N,2k} & \cdots & K_{2N,2N} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{2k-1} \\ U_{2k} \\ \vdots \\ U_{2N-1} \\ U_{2N} \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ 0 \\ F_{2k} \\ \vdots \\ F_{2N-1} \\ F_{2N} \end{Bmatrix}$$

This is computed as follows in the program:

```
for i=1:size(fixed_DOF,1)
    k=fixed_DOF(i,1);            % Global number of the node
    if fixed_DOF(i,2)==1
        Qgl(2*k-1)=0;
        for j=1:2*Nnode_tot
            Kgl(2*k-1,j)=0;
            Kgl(j,2*k-1)=0;
        end
        Kgl(2*k-1,2*k-1)=1;
    end
    if fixed_DOF(i,3)==1
        Qgl(2*k)=0;
        for j=1:2*Nnode_tot
            Kgl(2*k,j)=0;
            Kgl(j,2*k)=0;
        end
        Kgl(2*k,2*k)=1;
    end
end
```

The imposed displacements are defined by the matrix `dep_imp`. The first column contains the node number. The second column contains the value of the horizontal displacement (0 if no horizontal displacement is imposed), and the third column contains the value of the vertical displacement.

As well as for the fixations, imposing a horizontal displacement d on a node k is equivalent to including the equation $U_{2k-1} = d$, which is done by changing the stiffness matrix and the load vector as follows:

$$\begin{bmatrix} K_{1,1} & \cdots & K_{1,2k-1} & K_{1,2k} & \cdots & K_{1,2N} \\ K_{2,1} & \cdots & K_{2,2k-1} & K_{2,2k} & \cdots & K_{2,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ K_{2k,1} & \cdots & K_{2k,2k-1} & K_{2k,2k} & \cdots & K_{2k,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N-1,1} & \cdots & K_{2N-1,2k-1} & K_{2N-1,2k} & \cdots & K_{2N-1,2N} \\ K_{2N,1} & \cdots & K_{2N,2k-1} & K_{2N,2k} & \cdots & K_{2N,2N} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{2k-1} \\ U_{2k} \\ \vdots \\ U_{2N-1} \\ U_{2N} \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ d \\ F_{2k} \\ \vdots \\ F_{2N-1} \\ F_{2N} \end{Bmatrix}$$

In the program:

```
for i=1:size(dep_imp,1)
    k=dep_imp(i,1);         % Global node number
%% Modification of Kgl and Qgl to impose the displacement
    if dep_imp(i,2)~=0
        for j=1:2*Nnode_tot
            Qgl(j)=Qgl(j)-Kgl(j,2*k-1)*dep_imp(i,2);
            Kgl(2*k-1,j)=0;
            Kgl(j,2*k-1)=0;
        end
        Kgl(2*k-1,2*k-1)=1;
        Qgl(2*k-1)=dep_imp(i,2);
    end
    if dep_imp(i,3)~=0
        for j=1:2*Nnode_tot
            Qgl(j)=Qgl(j)-Kgl(j,2*k)*dep_imp(i,3);
            Kgl(2*k,j)=0;
            Kgl(j,2*k)=0;
        end
        Kgl(2*k,2*k)=1;
        Qgl(2*k)=dep_imp(i,3);
    end
end
```

Remark: This technique to impose fixed degrees of freedom or to impose displacements is not as efficient as the static condensation. Indeed, it does not permit to reduce the size of the system of equations for known DOF values. However, this method can be easily adapted for the construction of the FE² method. Additionally, the goal of this study is not the computational efficiency of the technique.

2.1.3 Numerical tests

Several tests were performed to validate the finite element program and ensure the right behaviour of the element under simple cases of stress.

The two first tests (rigid modes and constant strain) are necessary to assess the consistency criteria. A shear test is also performed on the element to ensure a correct behaviour under shear stress. Finally, a patch test was made to verify the behaviour of the elements and the assembly process, and therefore the proper functioning of the program.

2.1.3.1 Rigid modes

The rigid modes are the modes corresponding to a state with no deformation. In a two dimensional case, there are 3 rigid modes:

- Rigid translation along x axis
- Rigid translation along y axis
- Rigid rotation around z axis

To verify that the element is capable of representing a state where there is no deformation, each of these transformations has to be applied to the element represented in Figure 2.1-1.

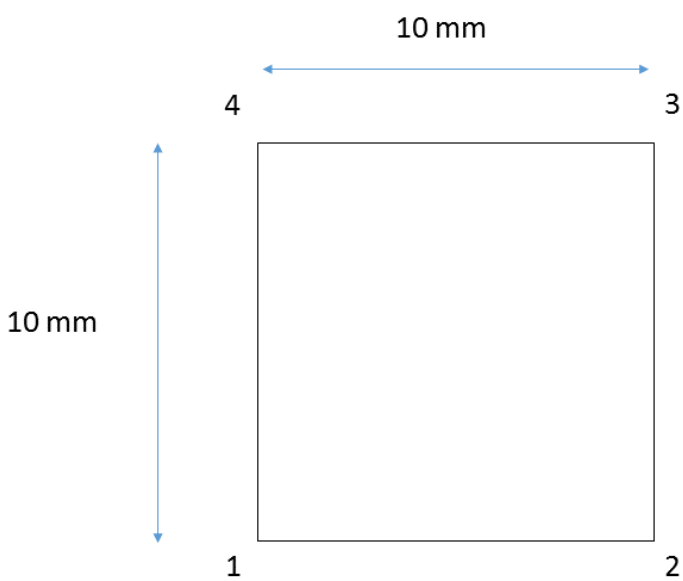


Figure 2.1-1 - Element tested for the rigid modes

For the first rigid mode (translation along x axis), a unitary displacement $U_x = 1$ is applied on node 1, and nodes 1 and 2 are fixed in the vertical direction.

For the second rigid mode (translation along y axis), a unitary displacement $U_y = 1$ is applied on nodes 1 and 2, and node 1 is fixed in the horizontal direction.

The third rigid mode (rotation around z axis) is defined by a direction vector $\{u\} = \begin{Bmatrix} -y \\ x \end{Bmatrix}$. To test this mode, a horizontal displacement $U_x = -0.1y_1$ and a vertical displacement $U_y = 0.1x_1$ are applied on node 1, and a vertical displacement $U_y = 0.1x_2$ is applied on node 2 (where (x_i, y_i) are the coordinates of node i in the initial configuration).

The results of these tests are the same using 1 or 4 Gauss nodes for the integration of the stiffness matrix and are shown in Figure 2.1-2 to Figure 2.1-4 below.

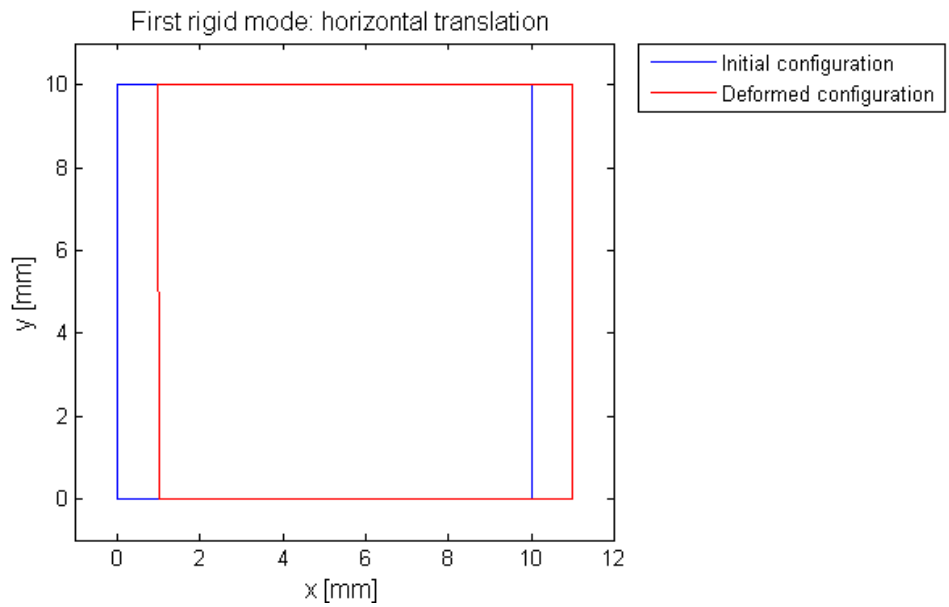


Figure 2.1-2 - Element displacement under the first rigid mode

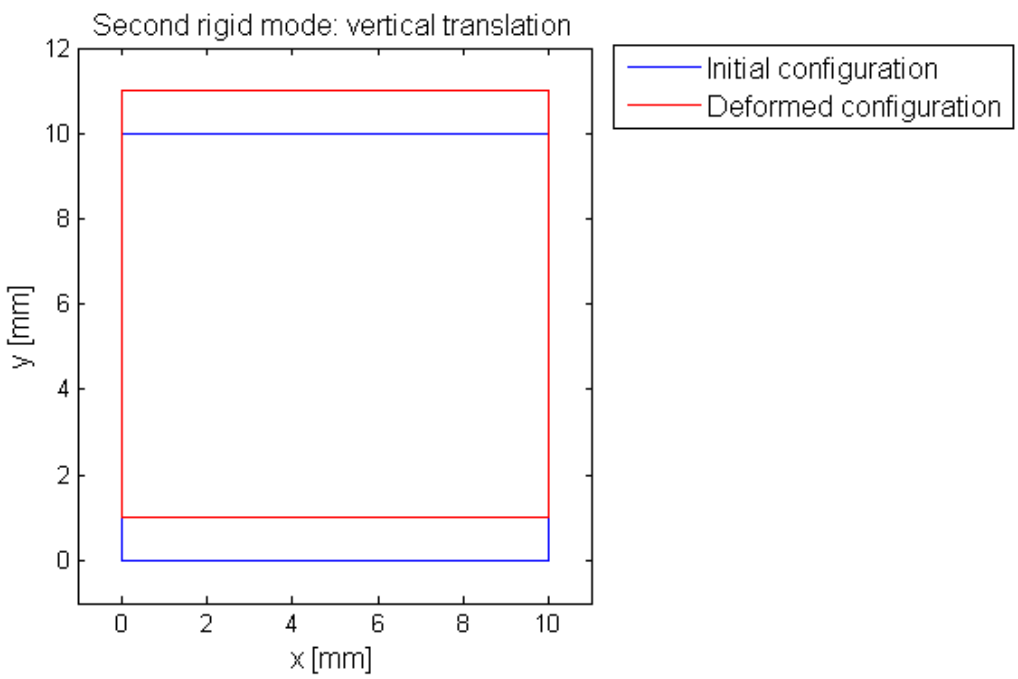


Figure 2.1-3 – Element displacement under the second rigid mode

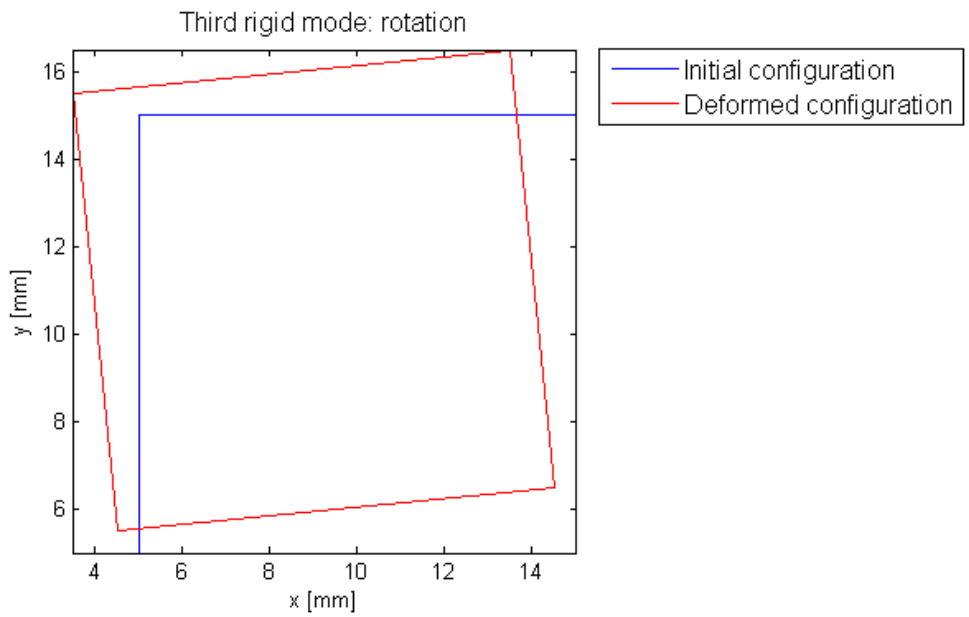


Figure 2.1-4 – Element displacement under the third rigid mode

2.1.3.2 Constant strain test

The constant strain test is performed on a single element. The element is a 10x10 mm² square with a theoretical thickness of 1 mm. The material properties are those of steel ($E = 210000\ MPa;\ \nu = 0.3$). The element is considered to be in a plane stress state.

A traction force P=100 kN is applied on nodes 2 and 3, as represented in Figure 2.1-5, so as to create a constant stress (and therefore a constant strain) in the element.

Nodes 1 and 2 are fixed in the vertical direction, and nodes 1 and 4 are fixed in the horizontal direction.

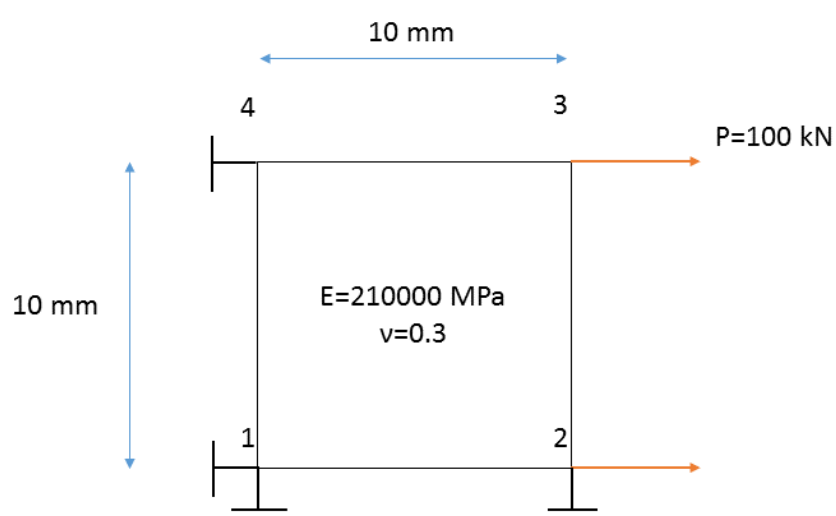


Figure 2.1-5 – Constant strain test on a square element

The theoretical stress, strain, and displacement can be calculated easily from the 2-D mechanics of materials:

- The tensile horizontal stress and the vertical stress are $\sigma_x = \frac{2P}{ht} = 20000\ MPa\ and\ \sigma_y = 0\ MPa$, where h is the height of the element and t its thickness;
- The horizontal and vertical strains are $\epsilon_x = \frac{\sigma_x}{E} = 9.52 * 10^{-2}\ and\ \epsilon_y = -\nu\epsilon_x = -2.86 * 10^{-2}$;
- The horizontal and vertical displacements of node 3 are $d_x = \epsilon_x * w = 0.952\ mm\ and\ d_y = -0.286\ mm$, where w is the width of the element;

The numerical results, using 1 or 4 Gauss points are presented in Table 3:

Table 3 - Stress and strain in the element under constant strain

	1 Gauss point	**4 Gauss points**
σ_x	20000 MPa	20000 MPa
σ_y	9e-13 MPa	2e-12 MPa
τ_{xy}	1e-12 MPa	8e-13 MPa
ϵ_x	0.0952	0.0952
ϵ_y	-0.0286	-0.0286
γ_{xy}	1e-17	1e-17

The results are the same in both cases and are identical to the theoretical results.

However, the representation of the deformed element (see Figure 2.1-6 and Figure 2.1-7) reveals that the numerical integration using 1 Gauss point gives erroneous results in terms of displacement. This can be explained mathematically by the fact that the global stiffness matrix is singular (non-invertible). Physically, this corresponds to a cinematic mode of the element, where it can deform freely regardless of the stress and strain at the integration point. To prevent this problem, another degree of freedom should be suppressed by adding a support, or the mechanism should be removed by assembling multiple elements.

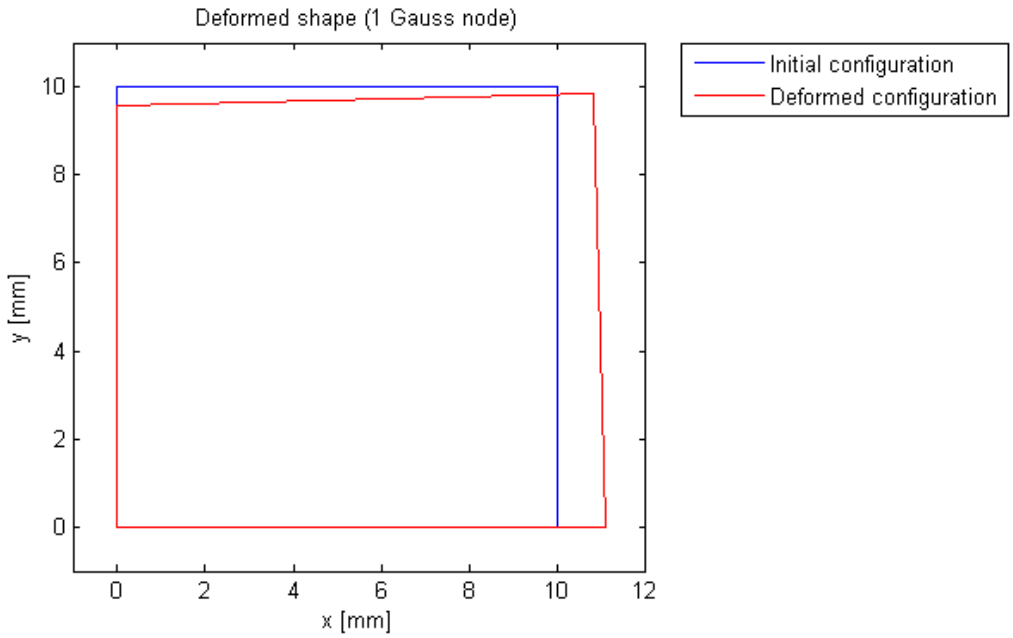


Figure 2.1-6 - Deformed shape under constant stress - Reduced integration using 1 Gauss point

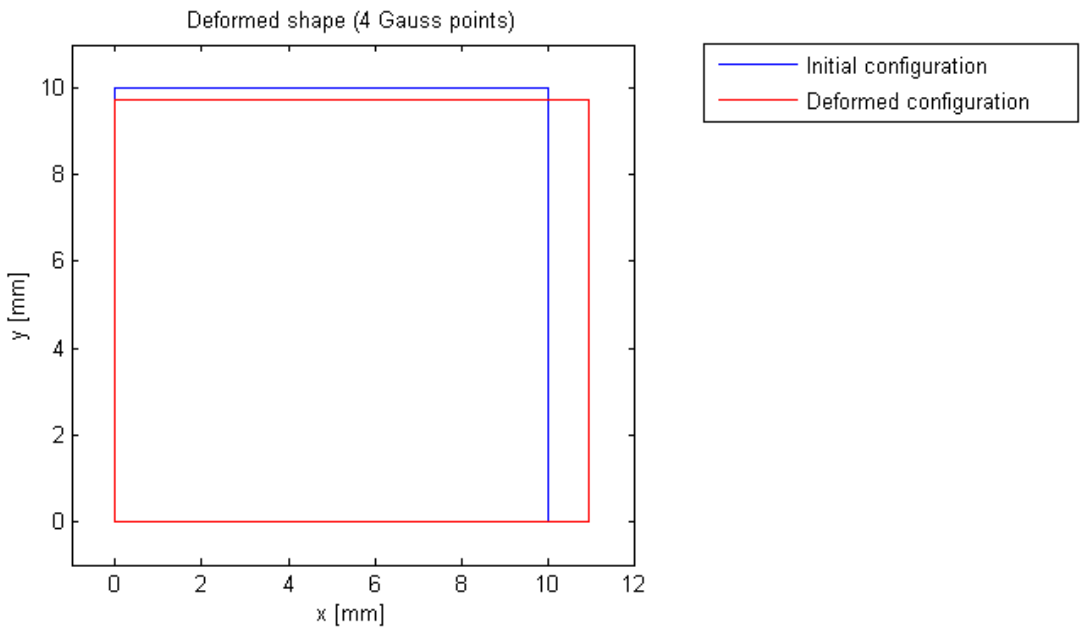


Figure 2.1-7 - Deformed shape under constant stress - Fully integrated element

2.1.3.3 Shear test

A shear test is conducted on the same element as the one used for the constant strain test (same geometry, same material properties), to verify that the element can reproduce the behaviour of a solid object subjected to shear.

The element is subjected to a horizontal displacement $U_x = \frac{L_x}{10}$ on both nodes 3 and 4. Nodes 1 and 2 are fixed. The deformed shape of the element is represented in Figure 2.1-8 (the deformed shape is the same for the fully integrated element and the 1 Gauss point element).

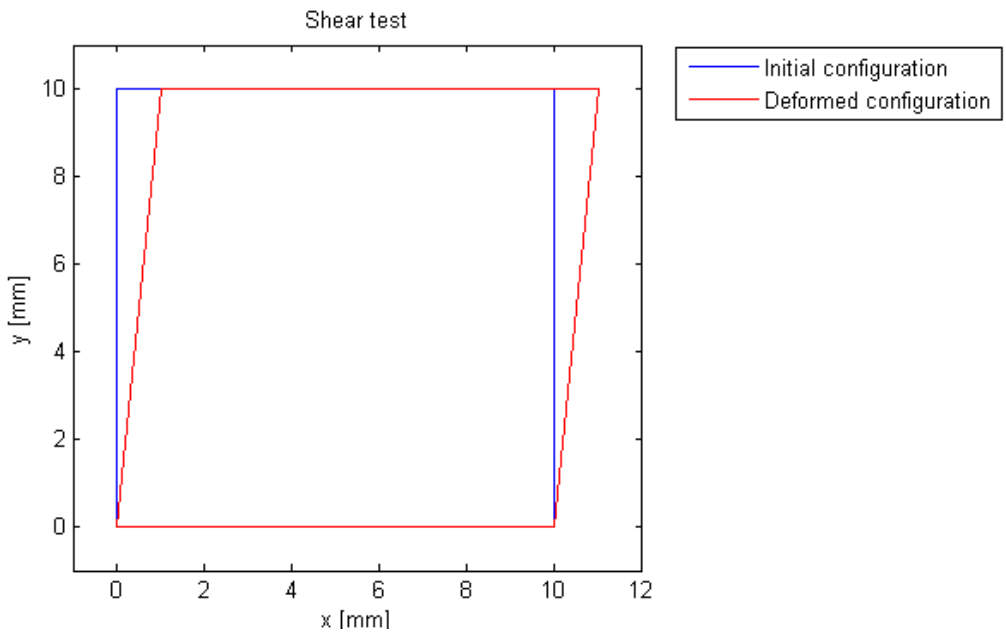


Figure 2.1-8 - Deformed shape of the element under shear stress

The stresses obtained through the numerical computation are then compared to the theoretical shear stress.

Analytically, the shear stress is:

$$\tau = G * \gamma = \frac{E}{2 * (1 + \nu)} * \frac{\Delta L}{L} = \frac{210000}{2 * 1.3} * \frac{1}{10} = 8077\ MPa$$

The finite element analysis gives the following stress vector:

$$\sigma = \begin{Bmatrix} 0 \\ 0 \\ 8077 \end{Bmatrix}$$

The program is therefore functional for shear stress.

2.1.3.4 Patch test

A patch test is performed in order to verify that the assembly process is correctly computed in the program.

The patch test is similar to the constant strain test, but instead of a single element, four elements are used as shown in Figure 2.1-9. Nodes 2, 4, 6, and 8 are placed in the middle of each side of the square. The coordinates of node 5 are (x=3; y=6) (node 1 is the origin of the system).

Again, a constant stress $\sigma = 20000\ MPa$ is applied on the right side of the solid object. The equivalent nodal loads are represented in Figure 2.1-9.

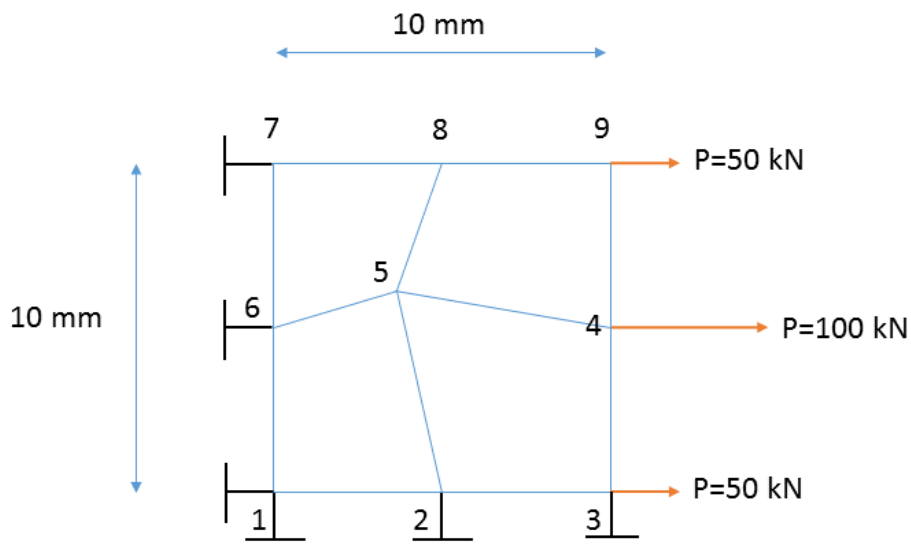


Figure 2.1-9 - Patch test

The numerical results are shown in Table 4.

Table 4 - Patch test - Displacement values

	U_x			U_y		
Node	1 Gauss point	4 Gauss points	Analytical result	1 Gauss point	4 Gauss points	Analytical result
1	0	0	0	0	0	0
2	0,476	0,476	0,476	0	0	0
3	0,952	0,952	0,952	0	0	0
4	0,952	0,952	0,952	-0,143	-0,143	-0,143
5	0,285	0,286	0,286	-0,171	-0,171	-0,171
6	0	0	0	-0,143	-0,143	-0,143
7	0	0	0	-0,286	-0,286	-0,286
8	0,476	0,476	0,476	-0,286	-0,286	-0,286
9	0,952	0,952	0,952	-0,286	-0,286	-0,286

The stress and strains computed numerically also correspond to the theoretical results. The patch-test is successful, which means the assembly process in the program works properly.

It can also be noted that the reduced integration (1 Gauss point) gives the same result as the fully integrated element (4 Gauss points), contrarily to the constant strain test with one element.

The different tests performed show that the elements are consistent and that the program is working.

2.2 FE² elements

The FE² technique consists in meshing the domain on two different scales. A primary mesh is done on the domain, which could then be used in a classic finite element analysis. Once this primary mesh is defined, a secondary mesh is applied to each element, as shown in Figure 2.2-1.

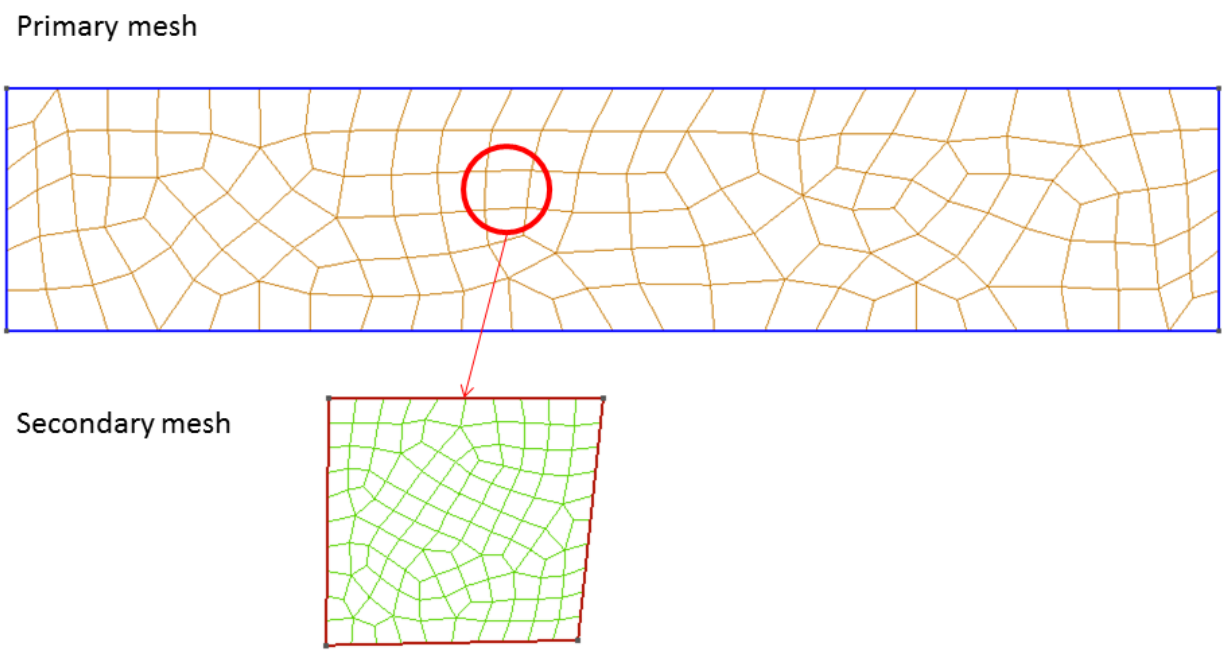


Figure 2.2-1 - Schematic representation of the FE² method

Using this secondary mesh, an equivalent stiffness matrix is created for each element of the primary mesh. The process used to create these equivalent matrices is explained hereafter in 2.2.1. The elements created with this method are then tested to verify their consistency and to assess their accuracy.

2.2.1 Method of construction

2.2.1.1 Equivalent stiffness matrix

In a finite element analysis, the elements are represented by their stiffness matrices. The stiffness matrix represents the relation between the nodal displacements and the forces required to create these displacements.

For a 4-node element in 2-D, the stiffness matrix is an 8-by-8 matrix. Physically, the element K_{ij} of the matrix is the force needed on DOF i to create a unitary displacement on DOF j, the other remaining DOF's being fixed.

This can be understood using equation [1.2.3]:

$$[1.2.3] \Leftrightarrow [K]\{q\} = \{Q\}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & K_{12} & \dots & K_{18} \\ K_{21} & K_{22} & \dots & K_{28} \\ \vdots & \vdots & \vdots & \vdots \\ K_{81} & K_{82} & \dots & K_{88} \end{bmatrix} \begin{Bmatrix} U_{x1} \\ U_{y1} \\ \vdots \\ U_{x4} \\ U_{y4} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ F_{x4} \\ F_{y4} \end{Bmatrix}$$

In the case where only one of the nodal displacements is different from zero, for instance $U_{x1} = 1$ and $\begin{cases} U_{xi} = 0 \ \ \forall i \neq 1 \\ U_{yi} = 0 \ \ \forall i \end{cases}$, equation [1.2.3] becomes:

$$\begin{Bmatrix} K_{11} * U_{x1} \\ K_{21} * U_{x1} \\ \vdots \\ K_{71} * U_{x1} \\ K_{81} * U_{x1} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ F_{x4} \\ F_{y4} \end{Bmatrix} \Leftrightarrow \begin{Bmatrix} K_{11} \\ K_{21} \\ \vdots \\ K_{71} \\ K_{81} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ F_{x4} \\ F_{y4} \end{Bmatrix} \qquad [2.2.1]$$

This can be generalised, for any $k, l \in [\![1\,;4]\!]$:

- $K_{2k,2l}$ is the vertical force on node k to create a unitary vertical displacement on node l;
- $K_{2k-1,2l}$ is the horizontal force on node k to create a unitary vertical displacement on node l;
- $K_{2k,2l-1}$ is the vertical force on node k to create a unitary horizontal displacement on node l;
- $K_{2k-1,2l-1}$ is the horizontal force on node k to create a unitary horizontal displacement on node l;

This definition of the stiffness is going to be used to compute the equivalent stiffness matrix of the FE² element.

The general idea is to impose a horizontal or vertical displacement on one of the vertex nodes of the meshed element, and calculate the equivalent vertical and horizontal forces on each vertex node.

2.2.1.2 Deformation of the element

The FE² element has to be able to reproduce the behaviour of the classic element. Therefore, the global deformation of the meshed element subjected to a displacement at one node has to be the same as the deformation of the classic element.

As it can be seen in Figure 2.2-2, this is not the case if the same boundary conditions are applied to both elements. In this example, a 1-by-1 mm square element is considered. A horizontal displacement is applied at node 1 and node 1 is fixed in the vertical direction, while the three other vertex nodes (2, 3, and 4) are fixed in both directions.

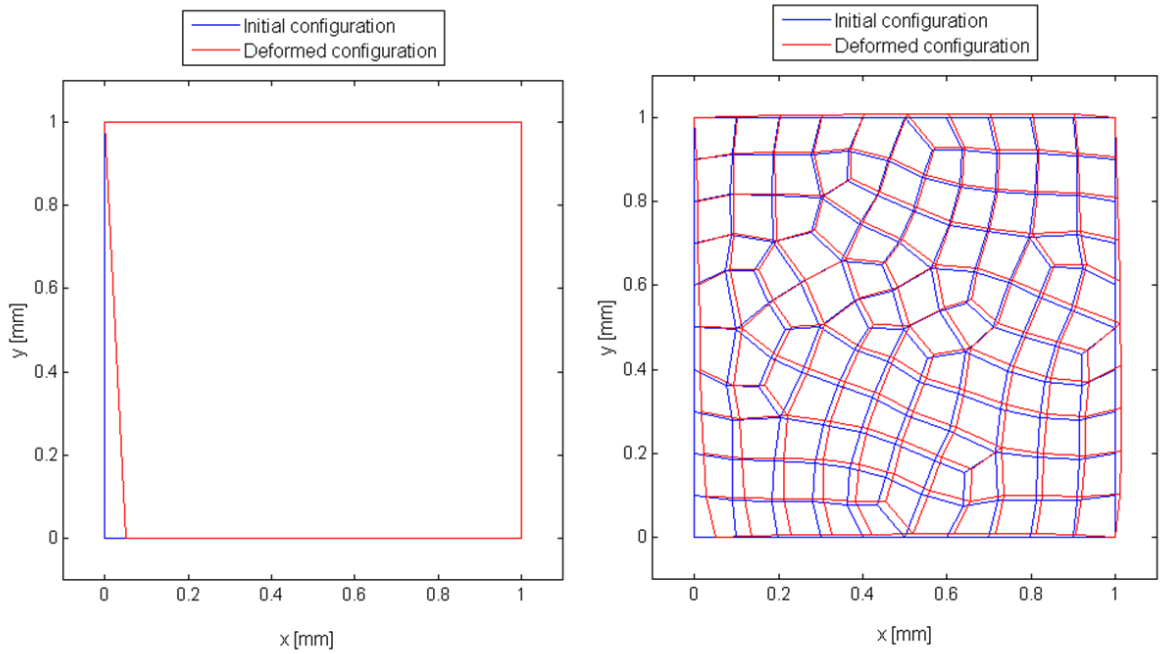


Figure 2.2-2 - Deformation of the classic element (left) and of the FE² element (right) with the same boundary conditions

In order to obtain the same deformation for both elements, the nodes on the sides of the FE² element have to be kept aligned.

For a square element, the sides that are unaffected by the displacement can be fixed in the horizontal or vertical direction (e.g. top and right sides in Figure 2.2-2). However, for a different geometry, or for the side of the square that does not stay parallel to one of the axes, specific boundary conditions have to be applied (e.g. left side in Figure 2.2-2).

In the finite element program, the global stiffness matrix and the equivalent load vector have to be modified to take in account these boundary conditions.

The element is modelled as shown in Figure 2.2-3. The nodes in the vertices are hinged (except for the node that is subjected to a displacement), and the nodes on the sides are on elastic supports in the direction perpendicular to the side.

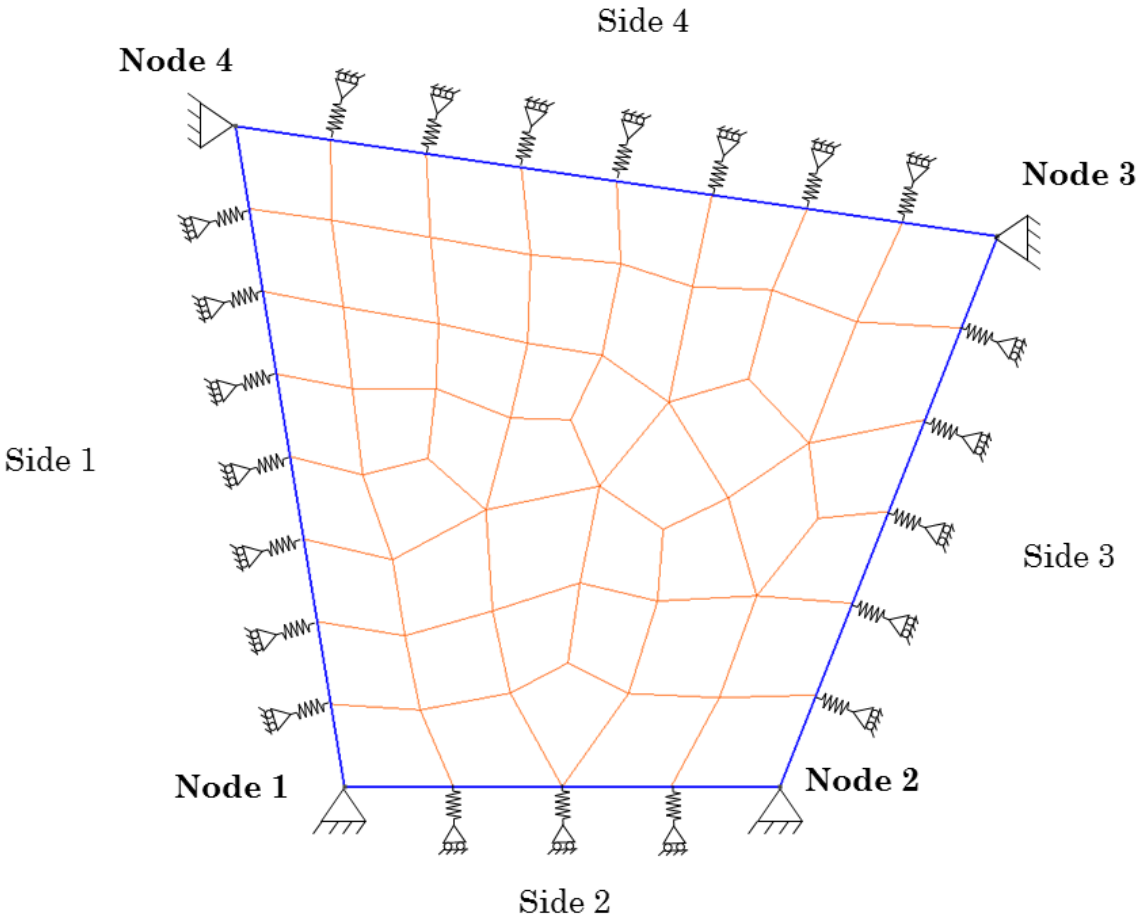


Figure 2.2-3 - Model of the FE² element

This model is valid as long as the nodal displacement is small compared to the size of the element. Indeed, if the displacement at a given node is big in comparison to the size, then the direction of the displacement of the nodes will not be quasi-perpendicular to the side, and therefore, the model will not be able to represent the behaviour of the element correctly (see section 2.2.1.5 for the choice of the displacement's value).

Let (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) be the coordinates of the four vertex nodes in the initial configuration. Given a horizontal displacement d at node 1 (see Figure 2.2-3), the coordinates (x, y) of the nodes belonging to side 1 in the deformed configuration must verify the following equation:

$$y = ax + b \qquad [2.2.2]$$

With $\begin{cases} a = \frac{y_4 - y_1}{x_4 - x_1 - d} \\ b = y_4 - ax_4 \end{cases}$

Let (U_{xi}, U_{yi}) the horizontal and vertical displacements of the considered node, and (x_i, y_i) its coordinates in the initial configuration:

$$\begin{cases} x = x_i + U_{xi} \\ y = y_i + U_{yi} \end{cases}$$

Therefore:

$$[2.2.2] \Rightarrow y_i + U_{yi} = ax_i + aU_{xi} + b$$
$$\Rightarrow U_{yi} = \alpha U_{xi} + \beta$$

$$\Leftrightarrow U_{xi} - \frac{1}{\alpha} U_{yi} = -\frac{\beta}{\alpha}$$

With: $\begin{cases} \alpha = a \\ \beta = ax_i - y_i + b \end{cases}$

To input this condition in the finite element program, the global stiffness matrix and the equivalent load vector have to be modified as follows:

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,2i-1} & K_{1,2i} & \cdots & K_{1,2N-1} & K_{1,2N} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,2i-1} & K_{2,2i} & \cdots & K_{2,2N-1} & K_{2,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2i-1,1} & K_{2i-1,2} & \cdots & K_{2i-1,2i-1} & K_{2i-1,2i} & \cdots & K_{2i-1,2N-1} & K_{2i-1,2N} \\ K_{2i,1} & K_{2i,2} & \cdots & K_{2i,2i-1} & K_{2i,2i} & \cdots & K_{2i,2N-1} & K_{2i,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N-1,1} & K_{2N-1,2} & \cdots & K_{2N-1,2i-1} & K_{2N-1,2i} & \cdots & K_{2N-1,2N-1} & K_{2N-1,2N} \\ K_{2N,1} & K_{2N,2} & \cdots & K_{2N,2i-1} & K_{2N,2i} & \cdots & K_{2N,2N-1} & K_{2N,2N} \end{bmatrix} \begin{Bmatrix} U_{x1} \\ U_{y1} \\ \vdots \\ U_{xi} \\ U_{yi} \\ \vdots \\ U_{xN} \\ U_{yN} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ F_{xi} \\ F_{yi} \\ \vdots \\ F_{xN} \\ F_{yN} \end{Bmatrix}$$

Becomes:

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,2i-1} & K_{1,2i} & \cdots & K_{1,2N-1} & K_{1,2N} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,2i-1} & K_{2,2i} & \cdots & K_{2,2N-1} & K_{2,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1/\alpha & \cdots & 0 & 0 \\ K_{2i,1} & K_{2i,2} & \cdots & K_{2i,2i-1} & K_{2i,2i} & \cdots & K_{2i,2N-1} & K_{2i,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N-1,1} & K_{2N-1,2} & \cdots & K_{2N-1,2i-1} & K_{2N-1,2i} & \cdots & K_{2N-1,2N-1} & K_{2N-1,2N} \\ K_{2N,1} & K_{2N,2} & \cdots & K_{2N,2i-1} & K_{2N,2i} & \cdots & K_{2N,2N-1} & K_{2N,2N} \end{bmatrix} \begin{Bmatrix} U_{x1} \\ U_{y1} \\ \vdots \\ U_{xi} \\ U_{yi} \\ \vdots \\ U_{xN} \\ U_{yN} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ -\beta/\alpha \\ F_{yi} \\ \vdots \\ F_{xN} \\ F_{yN} \end{Bmatrix}$$

Where N is the total number of nodes in the mesh.

This change is made for every node i on side 1. A similar process is done for each side of the quadrangle.

2.2.1.3 Reactions at the nodes

So far, the modifications made in the global stiffness matrix and the equivalent load vector force the side nodes to stay on a determined line. However, this is not sufficient to represent the behaviour of the element as modelled in Figure 2.2-3.

In order to fit the chosen model, the reactions R_k at the nodes situated on the sides of the quadrangle have to be perpendicular to the side, as it should be for a movable support.

The reaction at a given node i has a horizontal component R_{2i-1} and a vertical component R_{2i}.

These reactions can be calculated with the following expression:

$$R_k = \sum_{j=1}^{2N} K_{kj} U_j - F_k \qquad [2.2.3]$$

Where K_{kj} are the elements of the global stiffness before modification, and F_k are the elements of the equivalent load vector before modification.

The total reaction at the node i can be expressed as:

$$\vec{R}_{tot,i} = R_{2i-1}\vec{x} + R_{2i}\vec{y} \qquad [2.2.4]$$

Let a be the slope of the side on which node i is situated. A direction vector for the considered side is $\{\vec{u}\} = \begin{Bmatrix} 1 \\ a \end{Bmatrix}$

$\vec{R}_{tot,i}$ is orthogonal to $\vec{u}$ if and only if:

$$\vec{R}_{tot,i}.\vec{u} = 0 \;\Rightarrow R_{2i-1} + aR_{2i} = 0 \qquad [2.2.5]$$

Using [2.2.3] and [2.2.5]:

$$\sum_{j=1}^{2N} (K_{2i-1,j} + aK_{2i,j})U_j - (F_{xi} + aF_{yi}) = 0$$

This equation is then used to modify the global stiffness matrix and the load vector:

$$\begin{bmatrix} K_{1,1} & \cdots & K_{1,2i-1} & K_{1,2i} & \cdots & K_{1,2N} \\ K_{2,1} & \cdots & K_{2,2i-1} & K_{2,2i} & \cdots & K_{2,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & -\dfrac{1}{\alpha} & \cdots & 0 \\ K_{2i-1,1} + aK_{2i,1} & \cdots & K_{2i-1,2i-1} + aK_{2i,2i-1} & K_{2i-1,2i} + aK_{2i,2i} & \cdots & K_{2i-1,2N} + aK_{2i,2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N-1,1} & \cdots & K_{2N-1,2i-1} & K_{2N-1,2i} & \cdots & K_{2N-1,2N} \\ K_{2N,1} & \cdots & K_{2N,2i-1} & K_{2N,2i} & \cdots & K_{2N,2N} \end{bmatrix} \begin{Bmatrix} U_{x1} \\ U_{y1} \\ \vdots \\ U_{xi} \\ U_{yi} \\ \vdots \\ U_{xN} \\ U_{yN} \end{Bmatrix}$$

$$= \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ -\dfrac{\beta}{\alpha} \\ F_{xi} + aF_{yi} \\ \vdots \\ F_{xN} \\ F_{yN} \end{Bmatrix}$$

This modification is applied for each of the side nodes. Once this is done, the system can be resolved and the nodal displacements obtained.

To compute this in the program, each side is defined as a vector containing the numbers of the nodes belonging to this side. These vectors are contained in an array L.

It is important to note that the modifications of the stiffness matrix and the load vector must only be applied if the side considered is not vertical or horizontal in the deformed configuration (otherwise, $\alpha = 0 \; or \; \alpha = \infty$). Therefore, for a square element on which a horizontal displacement is applied at node 1, only side 1 will have to be taken into account.

This is done in the program by defining an interval (`interval` in the program) containing the numbers of the side(s) that have to be taken into account.

The following is an excerpt of the code that corresponds to the process described previously in 2.2.1.2 and 2.2.1.3:

```
for f=interval
    alpha=a(f);     %slope of side f
    for i=2:length(L{1,f})-1    % loop on the internal nodes of side f
        k=L{1,f}(i);            % global number of i-th node of side f
        % Coordinates of node k in the original configuration
        y0=nodes(k,3);
        x0=nodes(k,2);
        beta=alpha*x0+b(f)-y0;
        % Modification of the stiffness matrix and of the load vector
        for j=1:2*Nnode_tot
            Kgl(2*k-1,j)=0;
            Kgl(2*k,j)=KglOrigin(2*k-1,j)+alpha*KglOrigin(2*k,j);
        end
        Kgl(2*k-1,2*k-1)=1;
        Kgl(2*k-1,2*k)=-1/alpha;
        Qgl(2*k-1)=-beta/alpha;
        Qgl(2*k)=QglOrigin(2*k-1)+alpha*QglOrigin(2*k);
    end
end
```

2.2.1.4 Equivalent reactions at the vertices

From the displacement computed, the reactions can be calculated at each side node.

Each of these reactions is orthogonal to the side, except for the reactions at the vertices. These reactions have to be decomposed into two vectors, each perpendicular to one of the sides.

Let a_i, a_j be the respective slopes of the two sides i and j that intersect at the vertex node k. The direction vectors orthogonal to these two sides are:

$$\{\overrightarrow{u_i}\} = \begin{Bmatrix} -a_i \\ 1 \end{Bmatrix} \; and \; \{\overrightarrow{u_j}\} = \begin{Bmatrix} -a_j \\ 1 \end{Bmatrix}$$

N.B.: if the side is vertical, $a_i = \pm\infty$, and the orthogonal vector $\{\overrightarrow{u_i}\}$ is $\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$

The reaction force at node k is (see [2.2.4]):

$$\vec{R}_{tot,k} = \begin{Bmatrix} R_{2k-1} \\ R_{2k} \end{Bmatrix}$$

Decomposing this reaction force is equivalent to finding α and β such that:

$$\begin{Bmatrix} R_{2k-1} \\ R_{2k} \end{Bmatrix} = \alpha \begin{Bmatrix} -a_i \\ 1 \end{Bmatrix} + \beta \begin{Bmatrix} -a_j \\ 1 \end{Bmatrix}$$

$$\Rightarrow \begin{cases} R_{2k-1} = -\alpha a_i - \beta a_j \\ R_{2k} = \alpha + \beta \end{cases} \Rightarrow \begin{cases} \alpha = R_{2k} - \beta \\ \beta = -\dfrac{R_{2k-1} + a_i R_{2k}}{a_j - a_i} \end{cases}$$

At the vertex node, the reaction force is now expressed as a vector $\alpha \begin{Bmatrix} -a_i \\ 1 \end{Bmatrix}$ perpendicular to side i, and a vector $\beta \begin{Bmatrix} -a_j \\ 1 \end{Bmatrix}$ perpendicular to side j.

To evaluate the equivalent force at the vertices, each side can now be considered as a simply supported beam subjected to vertical nodal forces $p_k = \pm\|\overrightarrow{R_{tot,k}}\|$ as shown in Figure 2.2-4. The support reactions on this beam correspond to the vertex forces.

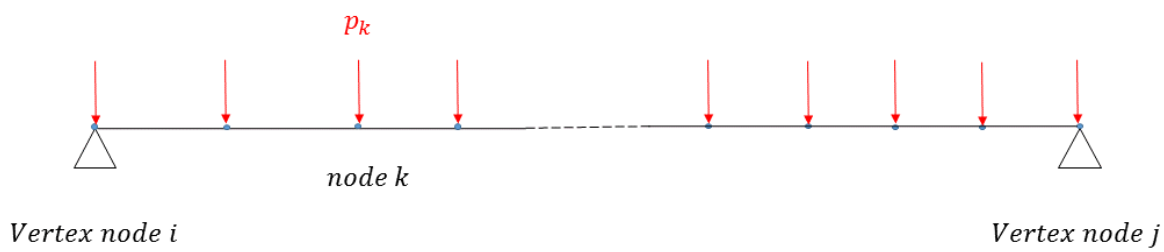


Figure 2.2-4 - Model of the side of the element for the calculation of the equivalent vertex forces

To simplify the calculations, a secondary coordinate system $(\overrightarrow{x_i}, \overrightarrow{y_i})$ is created for each side as shown in Figure 2.2-5.

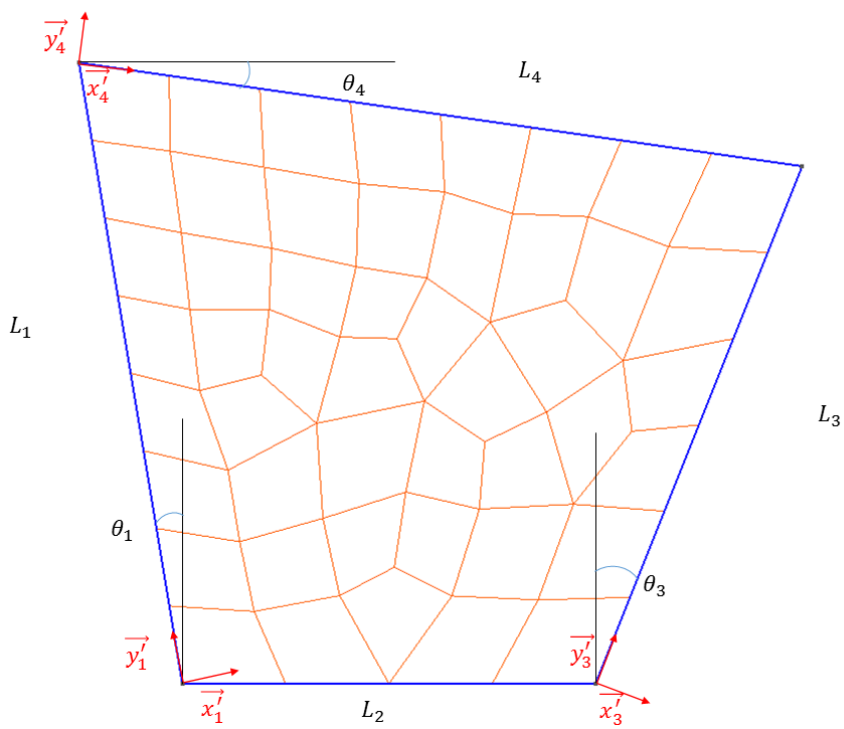


Figure 2.2-5 – Secondary coordinate systems

For instance, on line 1 (L1), the equivalent vertex forces are then obtained by calculating the support reactions of each equivalent beam through an isostatic analysis:

Let A and B be the two supports (vertex nodes 1 and 4), R_A, R_B the support reactions:

$$\begin{cases} R_A + R_B + \sum_{k \in L_1} p_k = 0 \\ \sum M_A = R_B l_1 + \sum_{k \in L_1} p_k y'_{1k} = 0 \end{cases}$$

$$\begin{cases} R_A = -\sum_{k \in L_1} p_k - R_B \\ R_B = -\frac{1}{l_1} \sum_{k \in L_1} p_k y'_{1k} \end{cases}$$

Where l_1 is the length of side 1, and y'_{1k} is the distance of node k to node 1.

The same process is applied for each side. Each vertex node therefore receives two forces (one from each of the sides it intersects) that can be added and expressed in the Cartesian coordinate system.

2.2.1.5 Size of the displacement

As mentioned in 2.2.1.2, the displacement d must be small for the model to be valid.

However, if the displacement is too small, the term $-\frac{1}{\alpha}$,that correspond to the inverse of the slope of the side in the deformed configuration, becomes close to zero if the side was originally vertical (infinite slope) and close to infinity if the side was originally horizontal (zero slope).

This can cause the global stiffness matrix to be singular, and therefore, can lead to errors when solving the system to find the displacements of the nodes.

Consequently, a compromise has to be made to find a value of displacement small enough to fit the model and large enough to avoid making the global stiffness matrix singular.

In order to find that value, the calculation of the FE² stiffness matrix of a square element was performed for different values of the displacement d imposed at the vertices. The element is a 1x1 mm² square. The sub-mesh contains 247 elements.

The global stiffness matrix becomes singular for a displacement $d = 10^{-9}$. Therefore, the range of values for d becomes $[10^{-8}; 10^{-1}]$.

Figure 2.2-6 shows the evolution of $K_{11}/K_{11,ref}$ for different values of the displacement, where $K_{11,ref}$ is the value of the first element of the equivalent stiffness matrix for $d = 10^{-8}$.

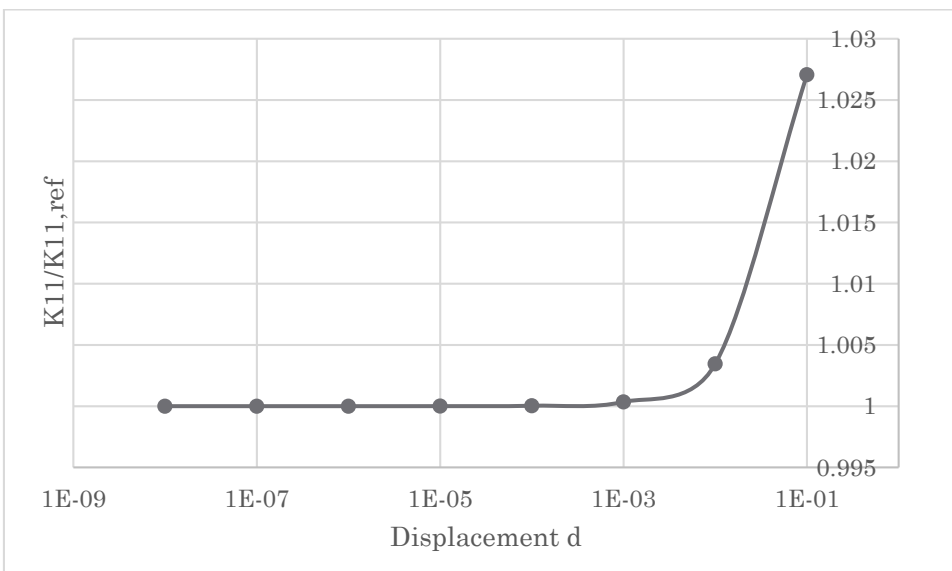


Figure 2.2-6 – Evolution of the ratio between element K11 of the FE² stiffness matrix and a reference value of K11 for different values of displacement d

Similar curves can be obtained for other elements of the stiffness matrix K_{ij}. It can be observed on the graph from Figure 2.2-6 that an asymptote is reached for values of d below 10^{-4}.

Considering the element was a 1x1 square, it can be concluded that the value of the displacement d should be kept between $h/10^4$ and $h/10^8$, where h is the characteristic size of the element.

The minimum value $h/10^8$ will be used in the rest of the study.

2.2.1.6 Size of the sub-mesh

To generate the equivalent stiffness matrix of the FE² element, the size of the sub mesh has to be chosen.

The number of elements will be used to characterize the size of the sub mesh.

The chosen size of the sub-mesh has to be as small as possible to limit the time of computation, but it must be big enough to bring a difference compared to a classic element.

In order to choose the size of the sub-mesh, the evolution of the stiffness matrix elements with the number of elements used was observed for a square element.

Only the first line of the stiffness matrix was studied here, since the square element is doubly symmetrical.

Figure 2.2-7 represents the evolution of the ratio between the elements of the equivalent stiffness matrix (FE²) and the elements of the fully integrated stiffness matrix.

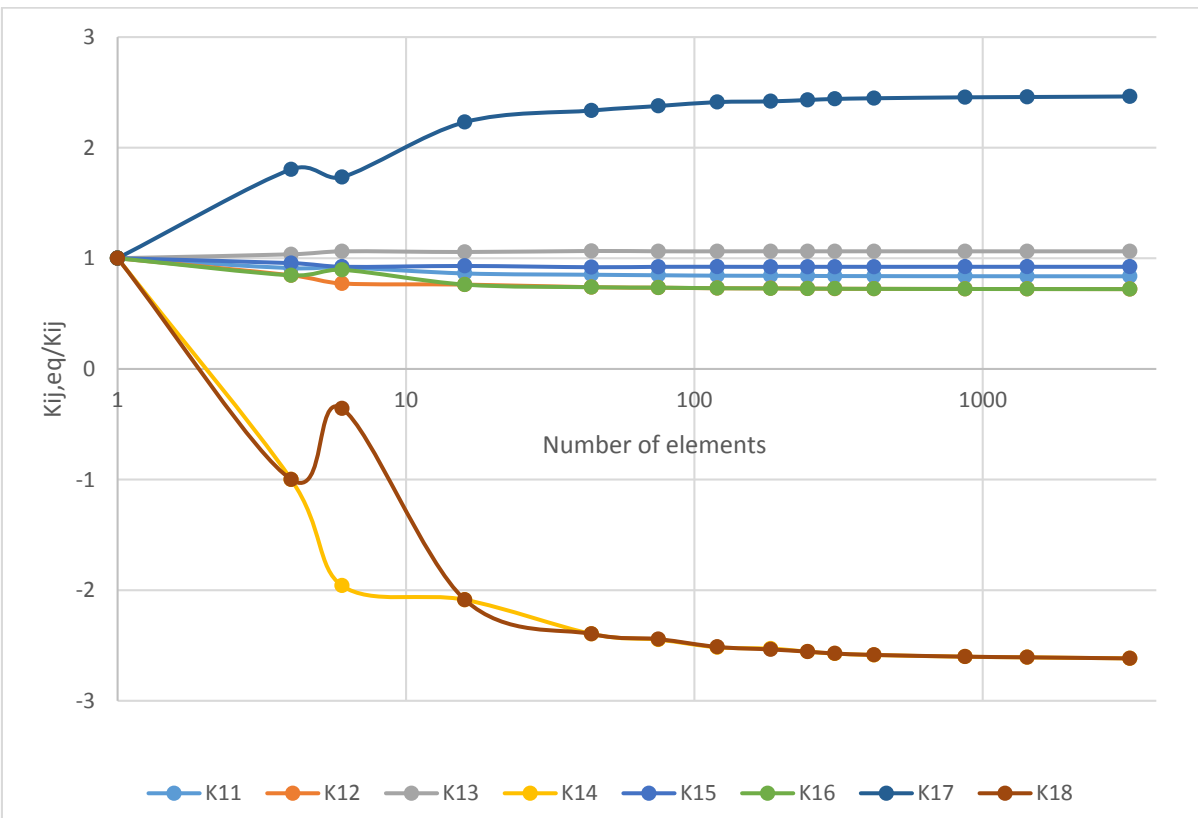


Figure 2.2-7 – Evolution of the stiffness with the number of elements used in the sub mesh

It can be observed that, as expected, all the curves start with a ratio of 1 when only one element is used. Some of the elements ($K_{11}, K_{12}, K_{13}, K_{15}, K_{16}$) stay close to the value of the fully integrated matrix (ratio close to 1), while others (K_{14}, K_{17}, K_{18}) show an important variation in comparison to the fully integrated matrix.

The non-uniform variation in the curve between 1 and 10 elements can be explained by the fact that a uniform mesh was used with 4 and 16 elements, whereas the rest of the meshes were random.

In all cases, the curve approaches a horizontal asymptote. In order to choose a minimal value for the number of elements in the sub-mesh, a criterion of $\pm 1\%$ of the asymptotic value was used. It was assumed that the asymptotic values were reached with the maximum tested number of elements ($N_{el} = 3300$).

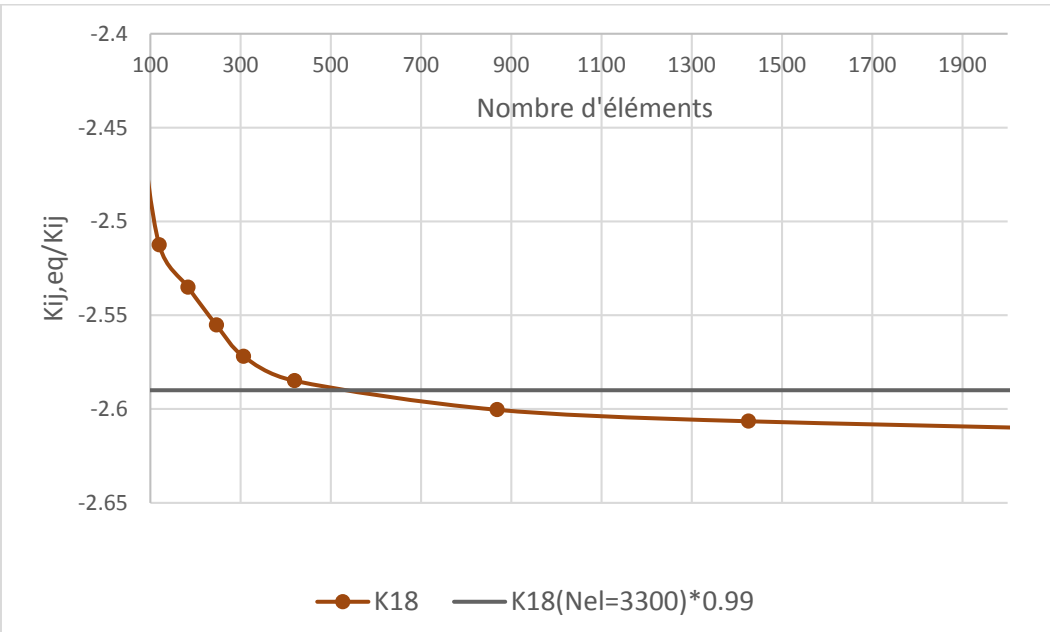


Figure 2.2-8 - Choice of a minimum number of elements for the sub-mesh

Figure 2.2-8 shows that in order to stay within 1% of the value of K_{18} obtained with 3300 elements, the sub mesh must contain at least 550 elements. A similar conclusion would be drawn for the other components of the matrix.

2.2.2 Rotation and size of the element

The process needed to generate the equivalent stiffness matrix of the element increase with N, N being the number of elements in the sub-mesh.

Therefore, given the FE² stiffness matrix of an element, it is interesting to find ways of obtaining the stiffness matrix of the same element after a simple transformation, such as a homothetic transformation or a rotation.

2.2.2.1 Homothetic transformation

The first thing that was observed from the results of the FE² element calculation was that the stiffness matrix is invariant through homothetic transformation. This property of the element can be useful to refine a regular mesh, without changing the aspect ratio of the elements.

This can be demonstrated for a classic element:

Let $(x_1, y_1) = (0,0)$ *and* $(x_i, y_i), i \in [\![2; 4]\!]$ be the coordinates of a 4-node quadrangle, and $[K_{ref}]$ its stiffness matrix.

A homothetic transformation with a scale factor λ and a centre (0,0) is applied to the element. The coordinates of the new element are:

$$\begin{cases} node\ 1: (0,0) \\ node\ 2: (\lambda x_2, \lambda y_2) \\ node\ 3: (\lambda x_3, \lambda y_3) \\ node\ 4: (\lambda x_4, \lambda y_4) \end{cases}$$

The Jacobian matrix for the parametric system of the new element is:

$$[J] = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dx}{d\eta} \\ \frac{dy}{d\xi} & \frac{dy}{d\eta} \end{bmatrix} = \begin{bmatrix} \lambda J_{ref1,1} & \lambda J_{ref1,2} \\ \lambda J_{ref2,1} & \lambda J_{ref2,2} \end{bmatrix} = \lambda [J_{ref}]$$

And the Jacobian determinant is: $J = \lambda^2 J_{ref}$

From [1.2.10] and [1.2.11], the matrix $[B]$ of the new element can be expressed:

$$[B] = \frac{1}{\lambda}[B_{ref}]$$

From [1.2.9], the stiffness matrix can be calculated:

$$[K] = e \iint_{-1}^{1} [B(\xi,\eta)]^T [C][B(\xi,\eta)] J(\xi,\eta)\, d\xi d\eta$$

$$= e \iint_{-1}^{1} \frac{1}{\lambda}[B_{ref}(\xi,\eta)]^T [C] \frac{1}{\lambda}[B_{ref}(\xi,\eta)] \lambda^2 J_{ref}(\xi,\eta)\, d\xi d\eta$$

$$[K] = [K_{ref}]$$

This was also observed in the stiffness matrices of the FE² elements that were computed with the program for square elements of different sizes.

2.2.2.2 Rotation

Contrarily to the homothetic transformation, a rotation of the element will lead to a modification of the stiffness matrix. This type of transformation can be useful when meshing an axisymmetric object.

However, this change can be analytically determined using the definition of the stiffness matrix established in 2.2.1.1.

The following calculations are valid for a rotation of an angle $\theta \in \left[0; \frac{\pi}{2}\right]$, measured positive clockwise, as represented in Figure 2.2-9.

Two coordinate systems are used: the original Cartesian coordinate system $(\vec{x}, \vec{y})$, and the rotated coordinate system $(\overrightarrow{x'}, \overrightarrow{y'})$. The rotated element in $(\overrightarrow{x'}, \overrightarrow{y'})$ is equivalent to the original element in $(\vec{x}, \vec{y})$.

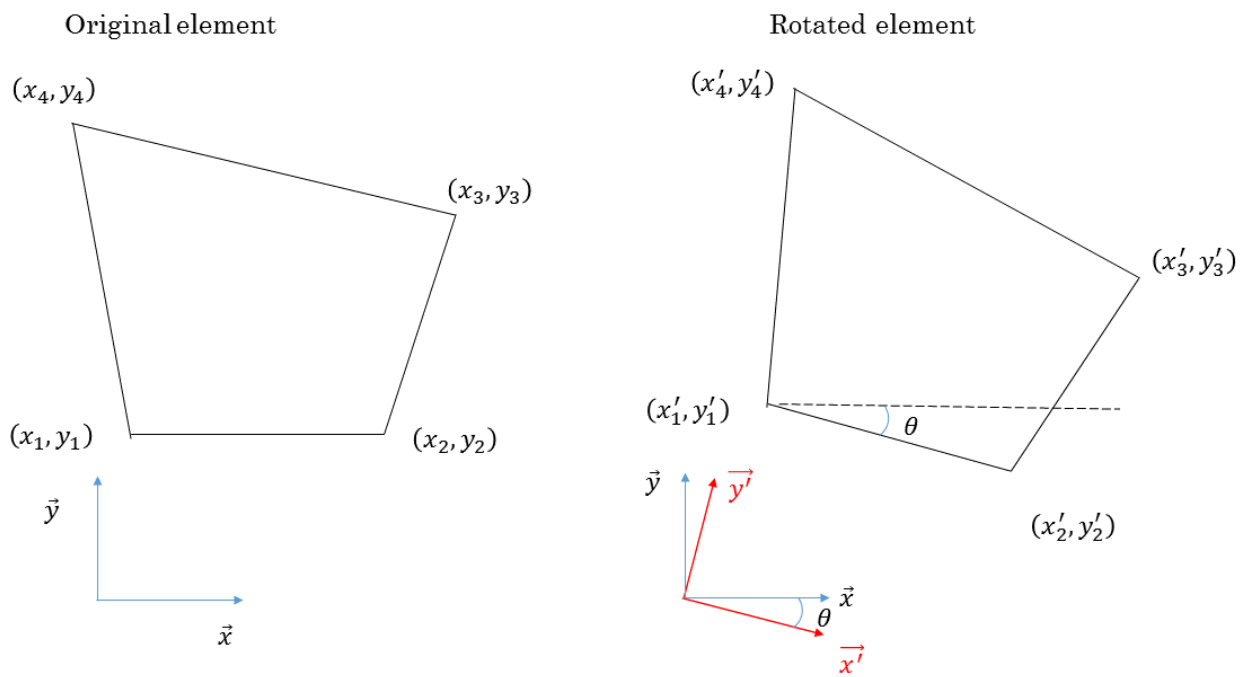


Figure 2.2-9 - Rotation of the element

Let $[K]$ the stiffness matrix of the base element, and $[K^*]$ the stiffness matrix of the rotated element.

A horizontal unitary displacement $1 * \vec{x}$ on a node i ($i \in [\![1; 4]\!]$) in the rotated configuration can be decomposed into two displacements:

$$\begin{cases} d'_x \overrightarrow{x'} = cos\theta \overrightarrow{x'} \\ d'_y \overrightarrow{y'} = sin\theta \overrightarrow{y'} \end{cases}$$

In the $(\overrightarrow{x'}, \overrightarrow{y'})$ coordinate system, the equivalent forces on a node j necessary to produce these displacements on node i are:

$$\begin{cases} F_{x'j} = d'_x K_{2j-1,2i-1} + d'_y K_{2j-1,2i} = cos\theta K_{2j-1,2i-1} + sin\theta K_{2j-1,2i} \\ F_{y'j} = d'_x K_{2j,2i-1} + d'_y K_{2j,2i} = cos\theta K_{2j,2i-1} + sin\theta\ K_{2j,2i} \end{cases} \quad [2.2.6]$$

These forces can be expressed in the original coordinate system $(\vec{x}, \vec{y})$, as shown in Figure 2.2-10:

$$\begin{cases} F_{xj} = cos\theta F_{x'j} + sin\theta F_{y'j} \\ F_{yj} = cos\theta F_{y'j} - sin\theta\ F_{x'j} \end{cases} \quad [2.2.7]$$

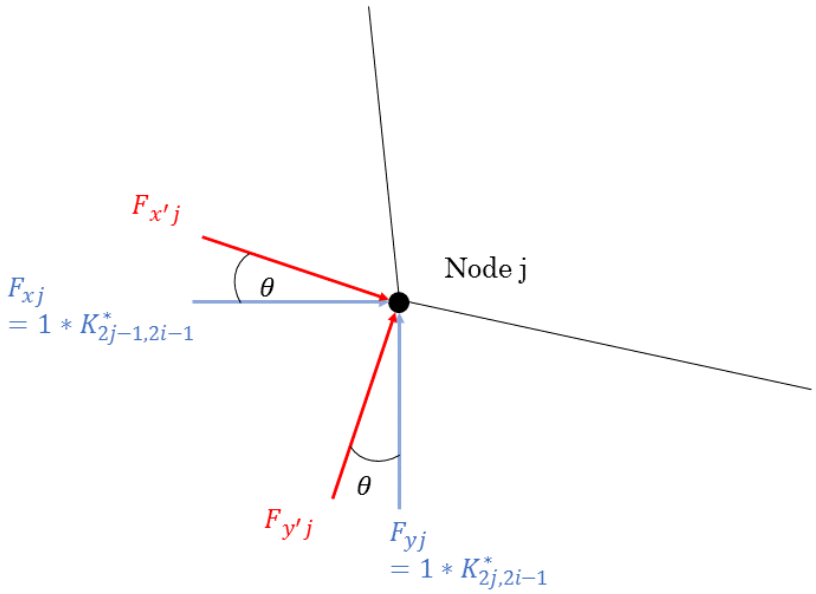


Figure 2.2-10 - Forces on node j to induce a unitary horizontal displacement on node i

From [2.2.6] and [2.2.7] the expression of the elements $K^*_{2j,2i-1}$ and $K^*_{2j-1,2i-1}$ can be obtained:

$$\begin{cases} K^*_{2j-1,2i-1} = cos\theta(cos\theta K_{2j-1,2i-1} + sin\theta K_{2j-1,2i}) + sin\theta(cos\theta K_{2j,2i-1} + sin\theta\ K_{2j,2i}) \\ K^*_{2j,2i-1} = cos\theta(cos\theta K_{2j,2i-1} + sin\theta\ K_{2j,2i}) - sin\theta\ (cos\theta K_{2j-1,2i-1} + sin\theta K_{2j-1,2i}) \end{cases} \quad [2.2.8]$$

Similarly, a vertical unitary displacement $1 * \vec{y}$ on a node $i\ (i \in [\![1;4]\!])$ in the rotated configuration can be decomposed into two displacements:

$$\begin{cases} d'_x\overrightarrow{x'} = -sin\theta\overrightarrow{x'} \\ d'_y\overrightarrow{y'} = cos\theta\overrightarrow{y'} \end{cases}$$

In the $(\overrightarrow{x'}, \overrightarrow{y'})$ coordinate system, the equivalent forces on a node j necessary to produce these displacements on node i are:

$$\begin{cases} F_{x'j} = d'_x K_{2j-1,2i-1} + d'_y K_{2j-1,2i} = -sin\theta K_{2j-1,2i-1} + cos\theta K_{2j-1,2i} \\ F_{y'j} = d'_x K_{2j,2i-1} + d'_y K_{2j,2i} = -sin\theta K_{2j,2i-1} + cos\theta\ K_{2j,2i} \end{cases} \quad [2.2.9]$$

Again, these forces are expressed in the original coordinate system to obtain the expressions of the elements $K^*_{2j,2i}$ and $K^*_{2j-1,2i}$:

$$\begin{cases} K^*_{2j-1,2i} = cos\theta(-sin\theta K_{2j-1,2i-1} + cos\theta K_{2j-1,2i}) + sin\theta(-sin\theta K_{2j,2i-1} + cos\theta\ K_{2j,2i}) \\ K^*_{2j,2i} = cos\theta(-sin\theta K_{2j,2i-1} + cos\theta\ K_{2j,2i}) - sin\theta\ (-sin\theta K_{2j-1,2i-1} + cos\theta K_{2j-1,2i}) \end{cases} \quad [2.2.10]$$

Using [2.2.8] and [2.2.10], the stiffness matrix of the rotated element can be determined from the stiffness matrix of the original element.

2.2.3 Tests

The same tests as those that were performed on classic elements in part 2.1.3 were performed on the FE² element, to verify its consistency.

For the three first tests, the element tested is a square element with the material properties of steel (E=210 000 MPa and ν=0.3).

The stiffness matrix of this element, obtained through the process described in 2.2.1, is shown in Table 5. The values of the elements of the equivalent stiffness matrix are displayed in black, and the difference between their value and the elements of the classic fully integrated stiffness matrix $\frac{K_{classic}(i,j)-K_{FE^2}(i,j)}{K_{classic}(i,j)}$ are displayed in red. The sub-mesh contained 594 elements and the displacement d at the vertices was 10^{-8}.

Table 5 - Stiffness matrix of the square FE² element

K_{i1}		K_{i2}		K_{i3}		K_{i4}	
87118	16.1%	27145	27.6%	-67443	-6.3%	7471	359.0%
27145	27.6%	87107	16.1%	-7481	359.3%	28278	-145.1%
-67443	-6.3%	-7481	359.3%	87105	16.1%	-27134	27.6%
7471	359.0%	28278	-145.1%	-27134	27.6%	87107	16.1%
-47942	7.7%	-27134	27.6%	28279	-145.1%	-7481	359.3%
-27146	27.6%	-47941	7.7%	7472	359.0%	-67443	-6.3%
28266	-145.0%	7471	359.0%	-47942	7.7%	27145	27.6%
-7469	358.9%	-67443	-6.3%	27143	27.6%	-47941	7.7%
K_{i5}		K_{i6}		K_{i7}		K_{i8}	
-47942	7.7%	-27146	27.6%	28266	-145.0%	-7469	358.9%
-27134	27.6%	-47941	7.7%	7471	359.0%	-67443	-6.3%
28279	-145.1%	7472	359.0%	-47942	7.7%	27143	27.6%
-7481	359.3%	-67443	-6.3%	27145	27.6%	-47941	7.7%
87105	16.1%	27143	27.6%	-67443	-6.3%	7472	359.0%
27143	27.6%	87118	16.1%	-7469	358.9%	28267	-145.0%
-67443	-6.3%	-7469	358.9%	87118	16.1%	-27146	27.6%
7472	359.0%	28267	-145.0%	-27146	27.6%	87118	16.1%

2.2.3.1 Rigid modes

The deformation of the three rigid mode tests (see 2.1.3.1) are presented hereafter in Figure 2.2-11, Figure 2.2-12, and Figure 2.2-13.

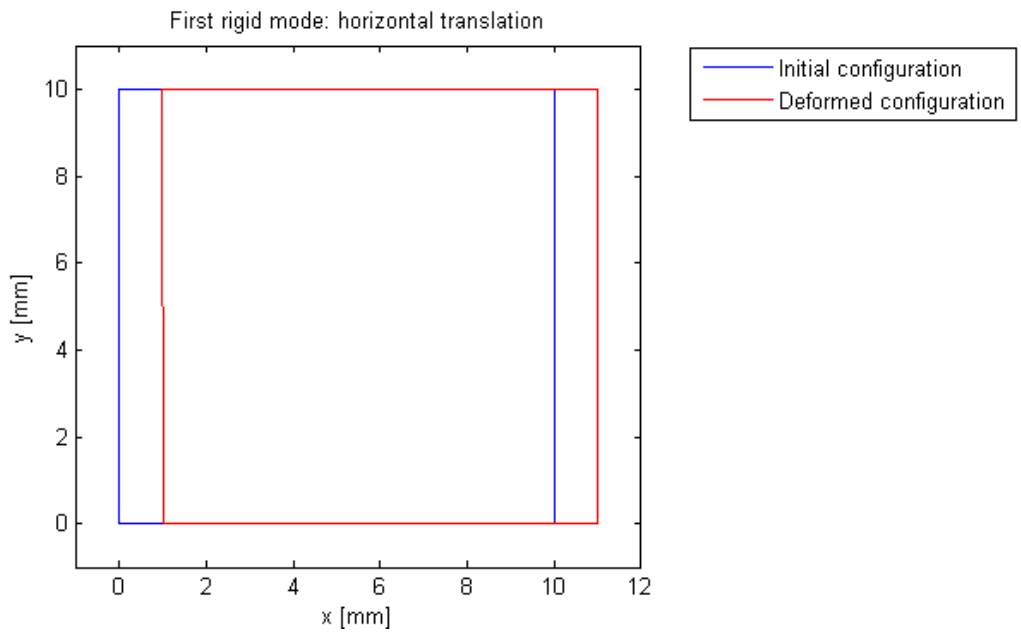


Figure 2.2-11 - Displacement of the FE² element under the first rigid mode (horizontal translation)

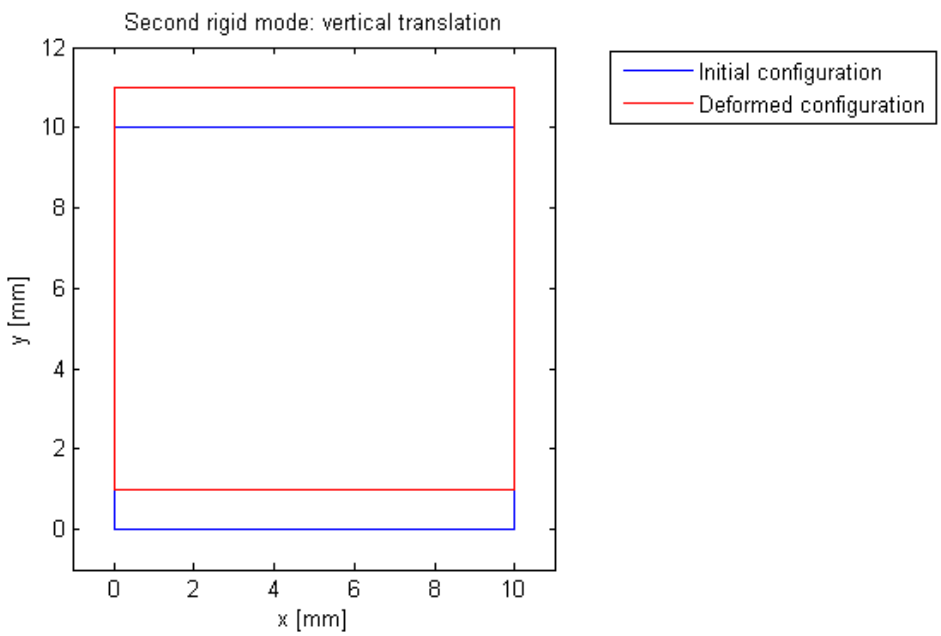


Figure 2.2-12 - Displacement of the FE² element under the second rigid mode (vertical translation)

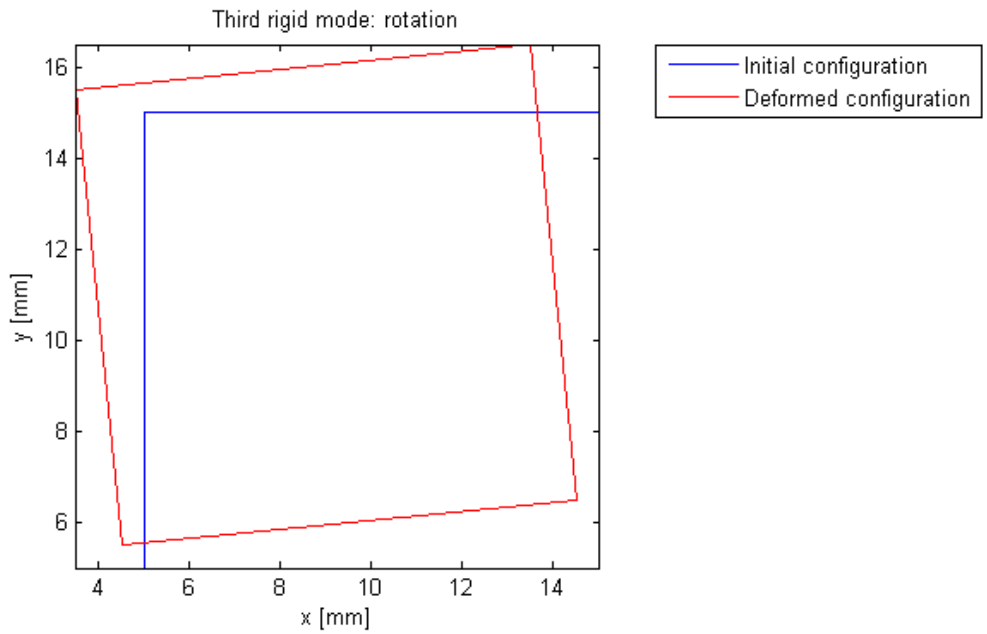


Figure 2.2-13 - Displacement of the FE² element under the third rigid mode (rotation)

2.2.3.2 Constant strain test

The constant strain test described in 2.1.3.2 is now performed on the FE² element.

The deformed shape is shown in Figure 2.2-14. The numerical horizontal and vertical displacements at node 3 are the same as the analytical displacement.

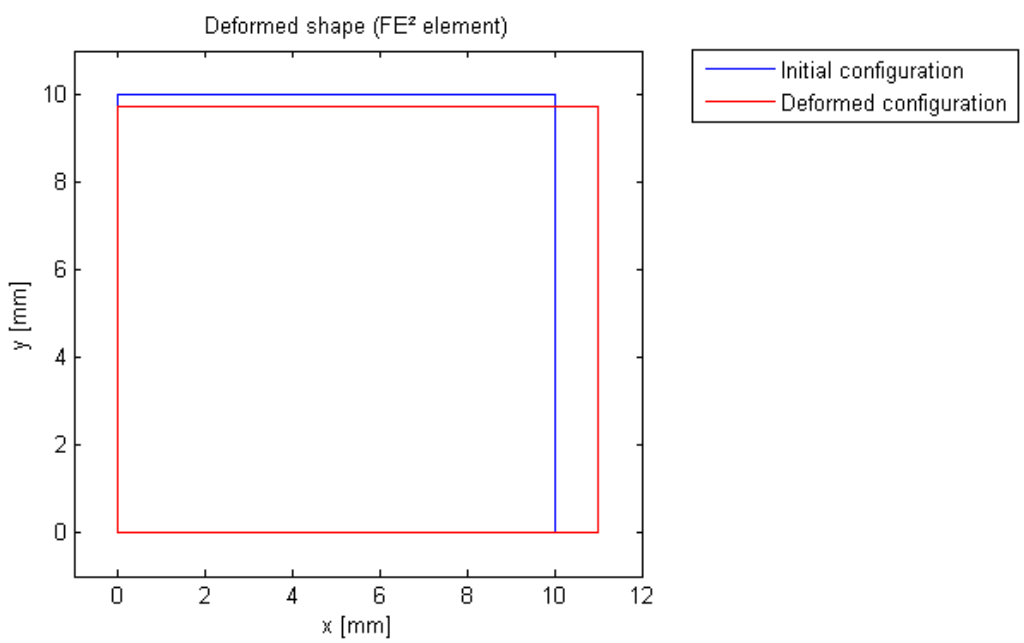


Figure 2.2-14 - Deformed shape of the FE² element under constant stress

The numerical results for the FE² element are presented in the following table, along with the results for the fully integrated classic element:

	FE²	**Fully integrated**
σ_x	20000 MPa	20000 MPa
σ_y	-9e-6 MPa	2e-12 MPa
τ_{xy}	-4e-6 MPa	8e-13 MPa
ϵ_x	0.0952	0.0952
ϵ_y	-0.0286	-0.0286
γ_{xy}	-5e-11	1e-17

The vertical stress and the shear stress, despite still being negligible, are much higher when using the FE² element. This shows that the FE² element is slightly less accurate than the classic element on a simple case. This can be related to the fact that the elements of the stiffness matrix are not perfectly accurate (see Table 5): all diagonal elements should be equal considering the element is a square. The irregularity comes from the mesh of the element. Nevertheless, the results can still be considered satisfying in term of consistency.

2.2.3.3 Shear test

The shear test (see 2.1.3.3) is now performed on the FE² element.

The stress vector obtained is the same as in 2.1.3.3:

$$\sigma = \begin{Bmatrix} 0 \\ 0 \\ 8077 \end{Bmatrix}$$

The FE² element can represent shear stress correctly.

2.2.3.4 Patch test

The patch test is performed with four trapezoidal elements. The geometry used for this test is shown in Figure 2.2-15. The four elements have similar geometry, which allows to compute all their equivalent stiffness matrices from the equivalent stiffness matrix of the first element. These matrices are displayed in Appendix 2.

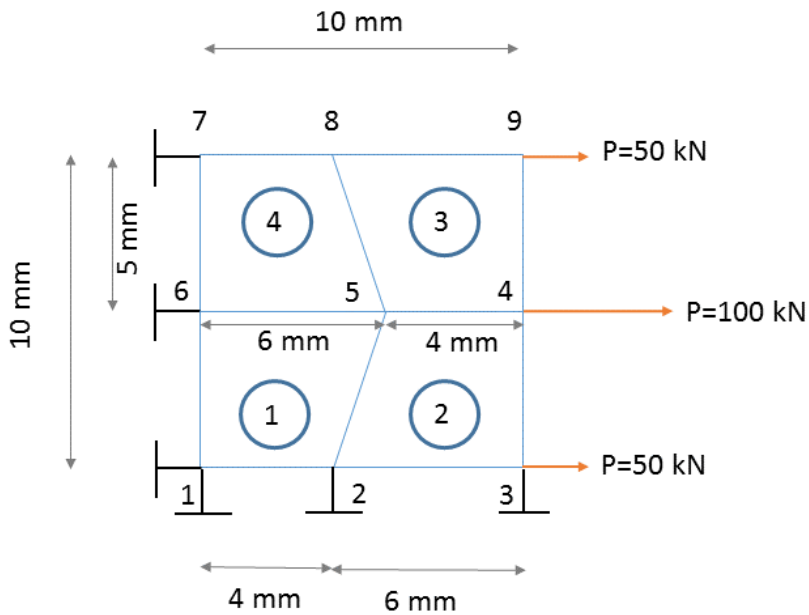


Figure 2.2-15 - Patch test with trapezoidal elements

The deformed shape of the elements is shown in Figure 2.2-16

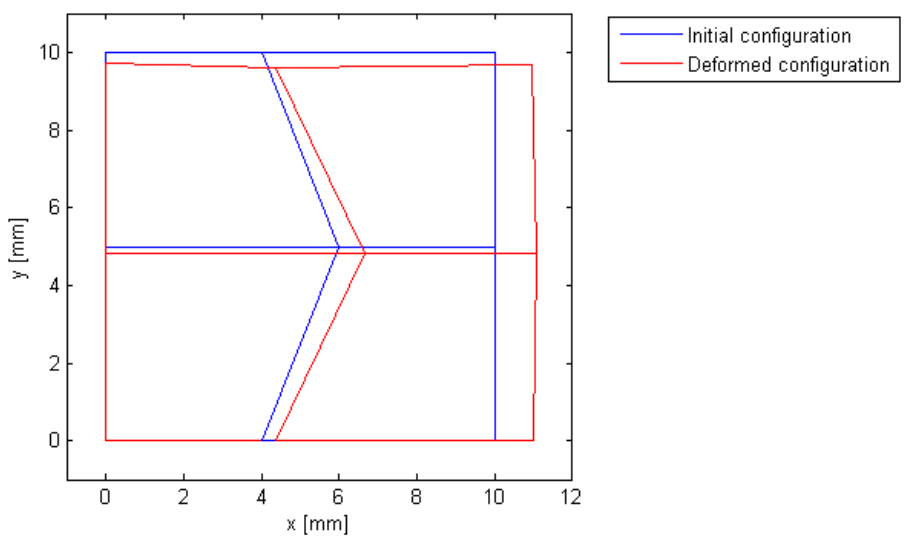


Figure 2.2-16 - Deformed shape of the FE² elements

The deformed shape does not correspond to the expected deformation of the square plate. The error on the displacement at each node is quantified in Table 6.

Table 6 - Patch test results with FE² elements

Node	x	y	$U_{x,th}$	$U_{x,num}$	Error %	$U_{y,th}$	$U_{y,num}$	Error %
1	0	0	0	0		0	0	
2	4	0	0.3810	0.3634	**4.61%**	0	0	
3	10	0	0.9524	0.9869	**3.63%**	0	0	
4	10	5	0.9524	1.0769	**13.07%**	-0.1429	-0.1740	**21.81%**
5	6	5	0.5714	0.6676	**16.82%**	-0.1429	-0.1880	**31.57%**
6	0	5	0	0		-0.1429	-0.1728	**20.99%**
7	0	10	0	0		-0.2857	-0.2956	**3.45%**
8	4	10	0.3810	0.3653	**4.10%**	-0.2857	-0.4065	**42.29%**
9	10	10	0.9524	0.9696	**1.81%**	-0.2857	-0.3284	**14.95%**

The error on the horizontal displacement goes from 1.81% to 16.82%, whereas the error on the vertical displacement goes from 3.45% to 42.29%. Therefore, the FE² trapezoidal element is not fit to represent the behaviour of the solid object.

The square and rectangular elements perform correctly under the simple tests presented previously. However, when the geometry of the element contains a side that is not vertical or horizontal, the FE² element does not perform well. This may be due to the model used to compute the deformation of the sub-mesh under a displacement at one of the vertices.

Nevertheless, square and rectangular elements, and shapes that are close to squares perform well. For the rest of the study, meshes with these types of elements will be used.

3 Performance of the FE² element on locking problems

3.1 Performance on shear locking

The first type of locking for which the FE² element is tested is shear locking. As mentioned in 1.3.1, this type of locking appears, among other cases, on solid objects subjected to bending.

3.1.1 Description of the test

The problem studied in this test is a cantilever beam subjected to a unitary vertical tip displacement, as shown in Figure 3.1-1.

The beam is a steel beam, with a Young's modulus $E = 210000\ MPa$ and a Poisson's ratio $\nu = 0.3$.

The object is considered to be in plane stress state.

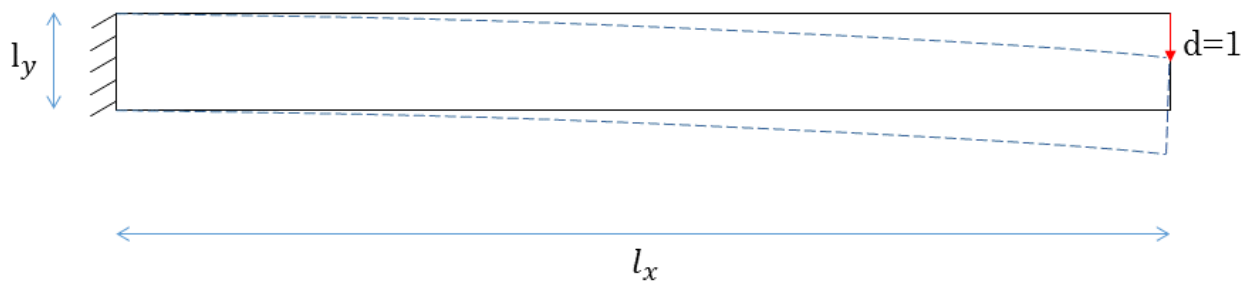


Figure 3.1-1 – Cantilever beam with unitary tip displacement

The length of the beam l_x is fixed to 200 mm. The width b of the beam is 1 mm. The thickness varies to study the effect of shear locking for different values of the slenderness l_x/l_y.

Timoshenko's beam theory is used to determine the theoretical reaction force at the tip of the beam.

For a static cantilever beam, the expression of the tip deflection as a function of the point load P at the tip is:

$$w = \frac{Pl_x}{\kappa AG} + \frac{Pl_x^3}{3EI} \qquad [3.1.1]$$

Where:

- κ is the Timoshenko shear coefficient, $\kappa = 5/6$ for a rectangular section;
- $A = l_y b$ is the cross section area;
- $G = \frac{E}{2(1+\nu)}$ is the shear modulus;
- $I = \frac{bl_y^3}{12}$ is the moment of inertia.

From [3.1.1], it can be observed that when the slenderness of the beam becomes larger ($l_x \gg l_y$), the term $\frac{Pl_x}{\kappa AG} \propto \frac{l_x}{l_y}$ becomes negligible compared to $\frac{Pl_x^3}{3EI} \propto \left(\frac{l_x}{l_y}\right)^3$ and the expression of the deflection becomes $w = \frac{Pl_x^3}{3EI}$, as predicted by the Euler-Bernoulli theory. The term $\frac{Pl_x}{\kappa AG}$ corresponds to the contribution of the shear stress to the deflection. For large values of the slenderness, the beam has a pure bending behaviour.

The theoretical equivalent point load at the tip of the beam for a unitary displacement is:

$$P_{th} = \frac{1}{\frac{l_x}{\kappa AG} + \frac{l_x^3}{3EI}} \qquad [3.1.2]$$

This force will be compared to the numerical forces estimated in the different tests.

3.1.1.1 Meshing

The test described previously was performed using different values of slenderness and different meshes.

The meshes used are regular meshes using only square or rectangular elements. These meshes were generated through the script Square_mesh.m. The input parameters for this script are the length of the beam `lx`, the thickness of the beam `ly`, the height of the elements `hy` and the length of the elements `hx`.

The parameters of the mesh are then defined:

```
% Number of nodes on the length and on the height
Nnodex=lx/hx+1;
Nnodey=ly/hy+1;
% Number of elements on the length and on the height
Nelx=lx/hx;
Nely=ly/hy;
Nnode_tot=Nnodex*Nnodey;     % Total n° of nodes
Nel=Nelx*Nely;               % Total n° of elements
nodes=zeros(Nnode_tot,3);    % Initialisation of the nodes matrix
elem=zeros(Nel,Nnode);       % Initialisation of the element matrix
fixed_DOF=zeros(Nnodey,3);   % Definition of the fixed nodes
```

Afterwards, the nodes and the elements are defined as shown in Figure 3.1-2:

```
% Definition of the nodes
for i=1:Nnodex
    for j=1:Nnodey
        nodes(Nnodex*(j-1)+i,:)=[Nnodex*(j-1)+i (i-1)*hx (j-1)*hy];
        if i==1
            fixed_DOF(j,:)=[Nnodex*(j-1)+i 1 1];
        end
    end
end
% Definition of the elements
for i=1:Nelx
    for j=1:Nely
        nel=Nelx*(j-1)+i;
        n1=Nnodex*(j-1)+i;
```

```
            elem(nel,:)=[n1 n1+1 n1+Nnodex+1 n1+Nnodex];
        end
    end
```

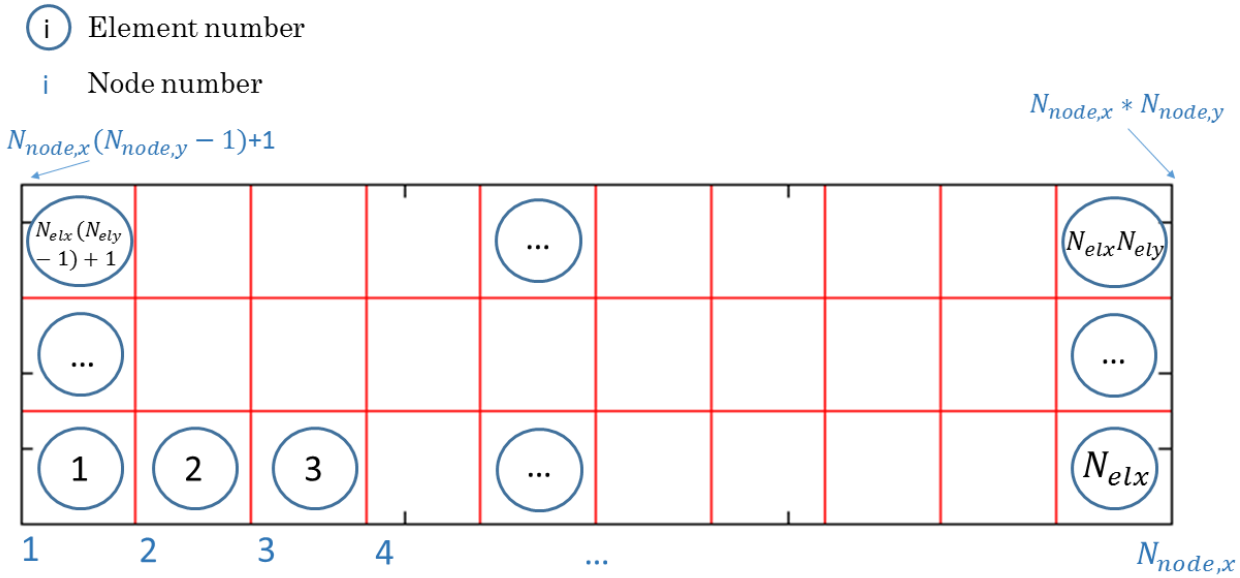


Figure 3.1-2 - Regular mesh using rectangular elements for the beam

3.1.1.2 Geometry and size of mesh

In this first test, the goal was to compare how close the numeric results were to the analytical results for different values of the slenderness ratio l_x/l_y.

As mentioned before, the value of the length is fixed at $l_x = 200\ mm$. The values of l_y shown in Table 7 were tested:

Table 7 - Values of thickness and slenderness for the cantilever beam

$\boldsymbol{l_y}$	40	30	20	10	8	5
$\boldsymbol{l_x/l_y}$	5	6.667	10	20	25	40
$\boldsymbol{h_y}$	10	7.5	5	2.5	2	1.25

The numbers of elements along the height and along the length of the element are kept constant. This means that as the slenderness of the beam increases, the length to thickness ratio of the element increases as well.

The mesh used consists of 4 elements along the height of the beam, and 40 along the length. Therefore, the length of the elements keeps a constant value $h_x = 5\ mm$, whereas the height of the element h_y takes the values displayed in Table 7.

The equivalent stiffness matrices of the 6 elements defined from Table 7 were computed using the process described in 2.2.1. These matrices were computed using a value of displacement $d = 10^{-8}$ and with a sub-mesh containing approximately 550 elements. The stiffness matrices of the elements in the sub-mesh were computed using 4 Gauss points for

the Gaussian quadrature. The matrices are displayed in **Erreur ! Source du renvoi introuvable.**.

3.1.2 Results

The program computes the equivalent point load at the tip P_{num} necessary to create a unitary vertical displacement at the tip of the cantilever:

```
P=-QglOrigin(2*Nnode_tot);
for k=1:2*Nnode_tot
    P=P+KglOrigin(2*Nnode_tot,k)*U(k);
end
```

Where `QglOrigin` is the load vector before taking into account the boundary conditions, and `KglOrigin` is the global stiffness matrix before taking into account the boundary conditions. `U` is the vector containing the displacements computed from the finite element analysis.

This numerical load is compared to the theoretical P_{th} calculated using Equation [3.1.2].

The results obtained with the two classic elements (reduced integration with one Gauss point and full integration), and with the FE² element are shown in Figure 3.1-3.

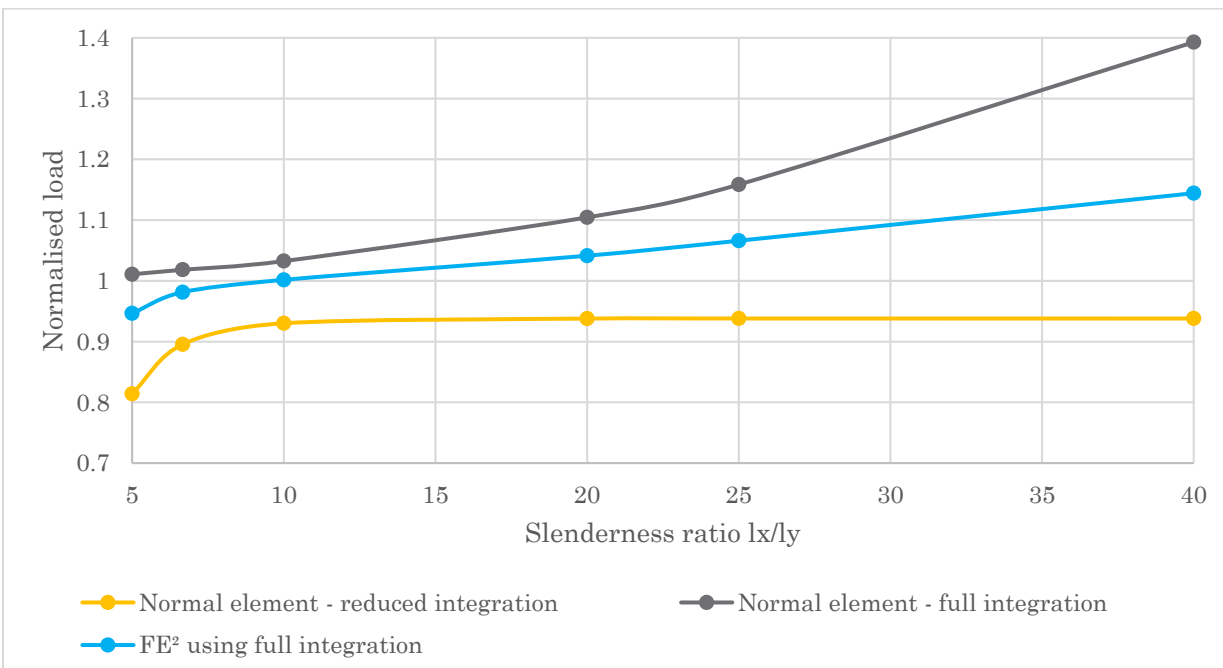


Figure 3.1-3 - Normalised load P_{num}/P_{th} for different slenderness ratios

The grey curve represents the evolution of the normalised load P_{num}/P_{th} computed using classic fully integrated elements. The shear locking phenomenon appears very clearly, as the normalised load becomes superior to 1 when the slenderness ratio increases. A higher numerical load means that a bigger force is necessary to create a unitary displacement, which corresponds to the locking phenomenon where the deflection is smaller than expected.

The yellow curve shows the results using a reduced integration with 1 Gauss point on the elements. It can be observed that the value of the normalised load stays below 1, which means that the element is less stiff than in theory. For slenderness ratios above 10, the numerical load stays at a value of approximately $0.94P_{th}$. However, for lower slenderness ratios, the element does not perform well: a phenomenon of hourglassing appears for a slenderness ratio of 5, as shown in Figure 3.1-4.

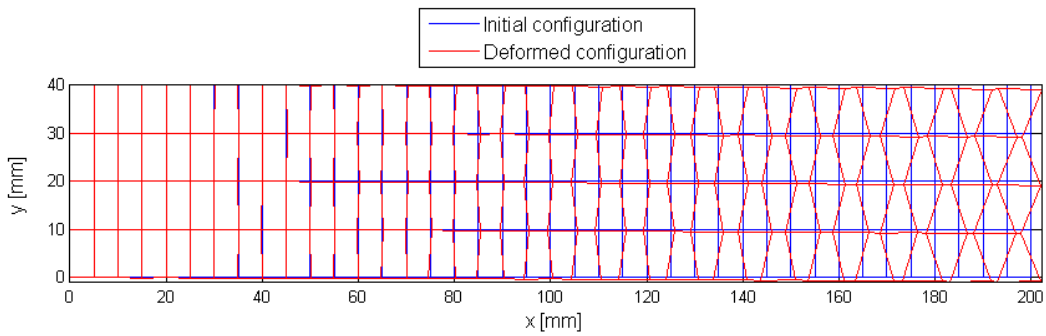


Figure 3.1-4 - Deflection of the beam for a slenderness ratio of 5 using a reduced integration with 1 Gauss point

Finally, the blue curve shows the performance of the FE² elements. The FE² element shows good results for slenderness ratios below 20, as the numerical load stays within a 5% margin of error of the theoretical load. However, as for the fully integrated element, the locking phenomenon appears when the slenderness ratio increases. Nevertheless, the FE² element performs better than the classic fully integrated element.

Considering that a reduced integration gives better results than a full integration against shear locking, the FE² equivalent stiffness matrix was also computed using a reduced integration on the sub-mesh. This was done only for the most critical case, i.e. when the slenderness ratio is equal to 40. The element is a 1.25x5 mm² rectangle.

Table 8 and show the comparison between the FE² stiffness matrices computed using fully integrated elements, and using a reduced integration with 1 Gauss point. The columns K_{ij} contain the value of the FE² stiffness matrix using fully integrated elements, and the percentages in red show the difference between the two matrices:

$$\frac{K_{ij,full\ integration} - K_{ij,reduced\ integration}}{K_{ij,full\ integration}}$$

Table 8 - Comparison between the FE² stiffness matrix computed with fully integrated elements or reduced integration in the sub-mesh

$K_{i1,full\ integration}$		$K_{i2,full\ integration}$		$K_{i3,full\ integration}$		$K_{i4,full\ integration}$	
43486	1.3%	20937	0.3%	-14426	0.0%	13678	-0.5%
20937	0.3%	301846	0.0%	-13679	-0.6%	159692	0.0%
-14426	0.0%	-13679	-0.6%	43441	1.4%	-20936	0.4%
13678	-0.5%	159692	0.0%	-20936	0.4%	301846	0.0%
-14421	0.0%	-20936	0.4%	-14595	4.1%	-13679	-0.6%
-20943	0.3%	-161507	0.0%	13683	-0.5%	-300031	0.0%
-14640	3.7%	13678	-0.5%	-14421	0.0%	20937	0.3%
-13672	-0.5%	-300031	0.0%	20933	0.3%	-161507	0.0%

$K_{i5,full\ integration}$		$K_{i6,full\ integration}$		$K_{i7,full\ integration}$		$K_{i8,full\ integration}$	
-14421	0.0%	-20943	0.3%	-14640	3.7%	-13672	-0.5%
-20936	0.4%	-161507	0.0%	13678	-0.5%	-300031	0.0%
-14595	4.1%	13683	-0.5%	-14421	0.0%	20933	0.3%
-13679	-0.6%	-300031	0.0%	20937	0.3%	-161507	0.0%
43441	1.4%	20933	0.3%	-14426	0.0%	13683	-0.5%
20933	0.3%	301847	0.0%	-13672	-0.5%	159692	0.0%
-14426	0.0%	-13672	-0.5%	43486	1.3%	-20943	0.3%
13683	-0.5%	159692	0.0%	-20943	0.3%	301847	0.0%

It can be observed that the difference between the two matrices is very small, with a maximum of around 4% difference on some of the elements.

The results on the performance against shear locking are almost the same, the FE² element computed with reduced integration performs slightly better but the difference is negligible:

	P_{num}/P_{th}
FE² using full integration on the sub-mesh	1.144
FE² using reduced integration on the sub-mesh	1.140

It can be concluded that for shear locking, the use of fully integrated elements or reduced integration on the elements of the sub-mesh has no influence on the result.

3.1.3 Convergence

The results presented previously were obtained using the same number of elements for each case.

Another interesting observation is the convergence of the results for the different elements.

The same geometry as before is kept for the beam. In this test, however, the slenderness ratio is kept constant and the number of elements varies.

The study is conducted on a cantilever beam with a slenderness ratio of 25, where shear locking is observable for fully integrated elements and FE² elements. The thickness of the beam is $l_y = 8\,mm$.

The theoretical load necessary to create a unitary tip displacement is (from [3.1.2]):

$$P_{th} = \frac{1}{\frac{l_x}{\kappa AG} + \frac{l_x^3}{3EI}} = \frac{1}{\frac{200}{\frac{5}{6} * 8 * 80770} + \frac{200^3}{3 * 210000 * \frac{8^3}{12}}} = 3{,}36\ N$$

The mesh is done with square elements. The different mesh sizes used are shown in Table 9:

Table 9 - Mesh used to study the convergence of the FE² element

$h_x = h_y$	8	4	2	1.6	1	0.8	0.5
Number of elements on the height	1	2	4	5	8	10	16
Number of elements on the length	25	50	100	125	200	250	400
N_{el}	25	100	400	625	1600	2500	6400

Figure 3.1-5 represents the evolution of the normalised load for different number of elements in the mesh. As expected, the results converge for all the elements, however the FE² element seems to converge more rapidly to the theoretical result than the classic elements.

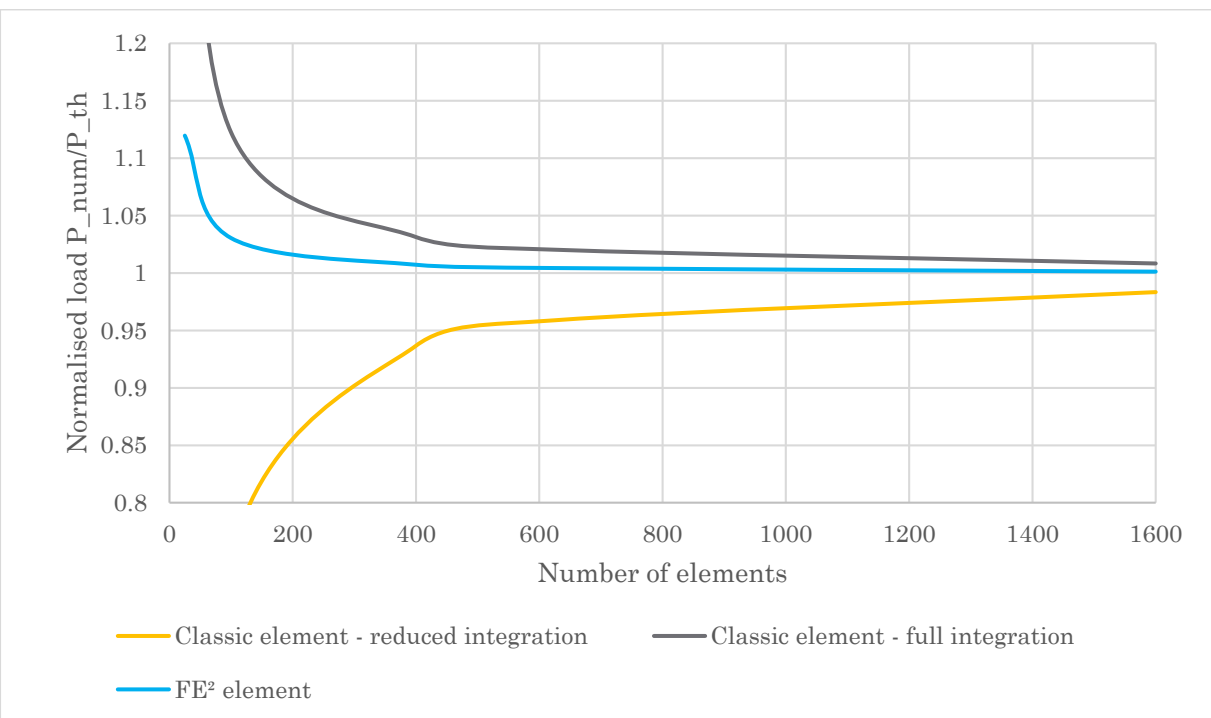


Figure 3.1-5 - Convergence of the results

In order to see if the order of convergence is the same for all the elements, the logarithm of the relative error on the numerical load $\epsilon = \frac{|P_{num} - P_{th}|}{P_{th}}$ was plotted as a function of $\ln\left(\frac{1}{h}\right)$, where h is the size of the element ($h = h_x = h_y$) in Figure 3.1-6.

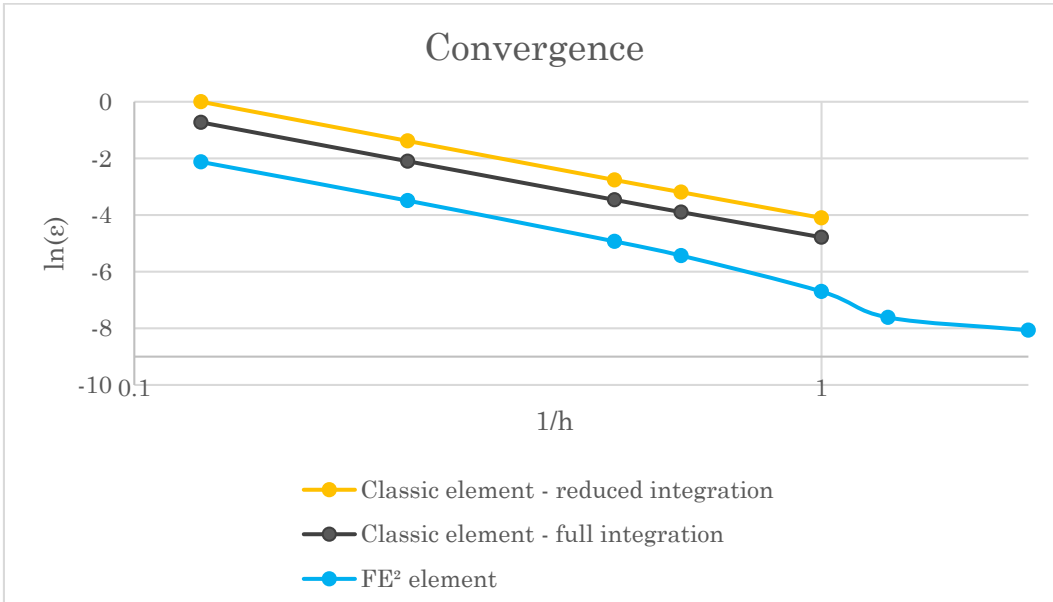


Figure 3.1-6 - Evolution of the error with the size of the elements

For both classic elements, the evolution of $\ln(\epsilon)$ is linear with $\ln\left(\frac{1}{h}\right)$. For the FE² element, the evolution seems to be linear as well for $h < 1$. The non-linearity observed for $h > 1$ seems to be a numerical problem and not the actual tendency of the curve. This might be due to memory limitations in MATLAB.

Note: the values of $\ln(\epsilon)$ for $\frac{1}{h} = 1{,}25\ and\ 2$ were not computed by the program for the classic element, as the direct computation was too heavy in terms of memory (matrices B, N, and J have to be computed for each element).

3.1.4 Comparison with other types of elements

So far, the FE² element has only been compared to classic elements, which are not always performant in terms of stability (hourglassing with the reduced integration) or in terms of locking (over-stiffness of the fully integrated elements).

3.1.4.1 Description of the problem

A similar test as the one described before was made by Echter and Bischoff [4] to evaluate the performance of NURBS (Non Uniform Rational B-Splines) finite elements.

In this test, referred to as the Timoshenko beam, a cantilever beam is subjected to a sinusoidal moment $m(x) = 5\sin\left(\frac{x}{5}\right)$, as shown in Figure 3.1-7. The beam is subjected to pure bending, which makes it easier to evidence the occurrence of a parasite shear stress.

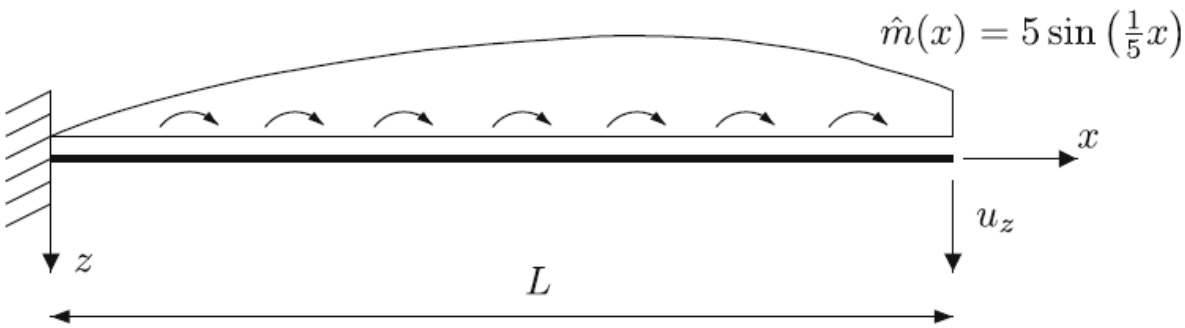


Figure 3.1-7 - Timoshenko beam with sinusoidal moment loading, from [4]

The length of the beam is $L = 10\ mm$, the width is still $b = 1$. The material has the following properties: $E = 10\ 000\ MPa$ and $\nu = 0$.

As for the previous test, the thickness is varied to test different values of the slenderness ratio.

The beam is loaded with the moment loading $\hat{m}(x)$ and the error on the deflection is computed. The load is scaled by h^3 ($\hat{m}(x) = 5h^3 \sin\left(\frac{x}{5}\right)$ to make the displacement independent of the beam thickness. The exact solution for the deflection of the beam is given in [4]:

$$w_{ex}(x) = \frac{3}{200}x^2\cos(2) + \frac{3}{4}\cos\left(\frac{x}{5}\right) - \frac{3}{4}$$

3.1.4.2 Load vector

The load vector F_ponct has to be implemented directly in the program. The elements of this vector are nodal loads expressed in Newtons. In order to impose the sinusoidal moment loading to the beam, equivalent nodal loads must be calculated.

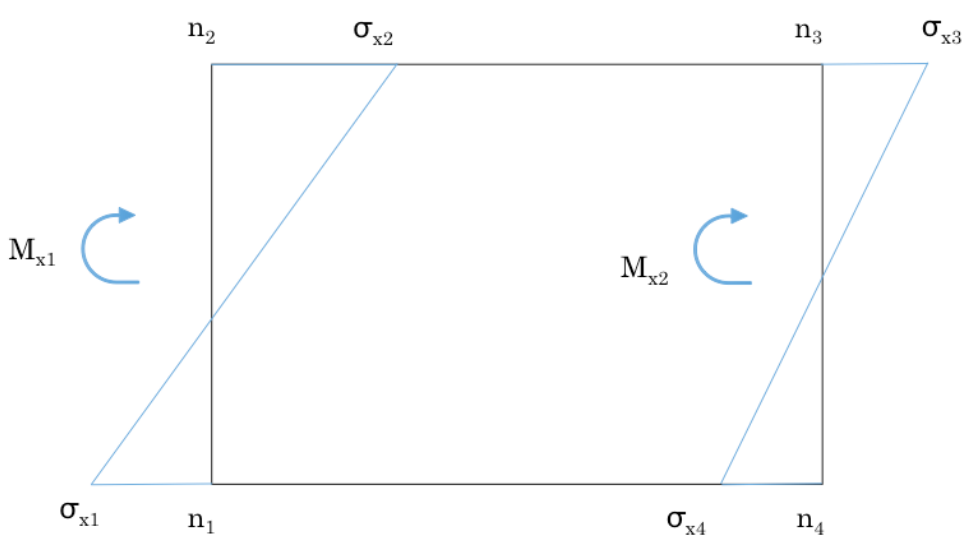


Figure 3.1-8 - Equivalent moment and stress on the sides of the element

The equivalent bending moments M_{x1} and M_{x2} on the lateral sides are calculated using the force method on n_2n_3 and n_1n_4 (see Figure 3.1-9):

$$X_1 f_{11} + f_{1P} = 0$$

Where f_{11} is the displacement at the right support for a unit value of the force X_1 (see Figure 3.1-9), and f_{1P} is the displacement at the right support under the axial force $N_P(x)$ created by the moment loading $\widehat{m}(x)$. Here, the goal is to determine the value of the support reaction X_1 which corresponds to the stress σ_{x4} if the bar studied is n_1n_4.

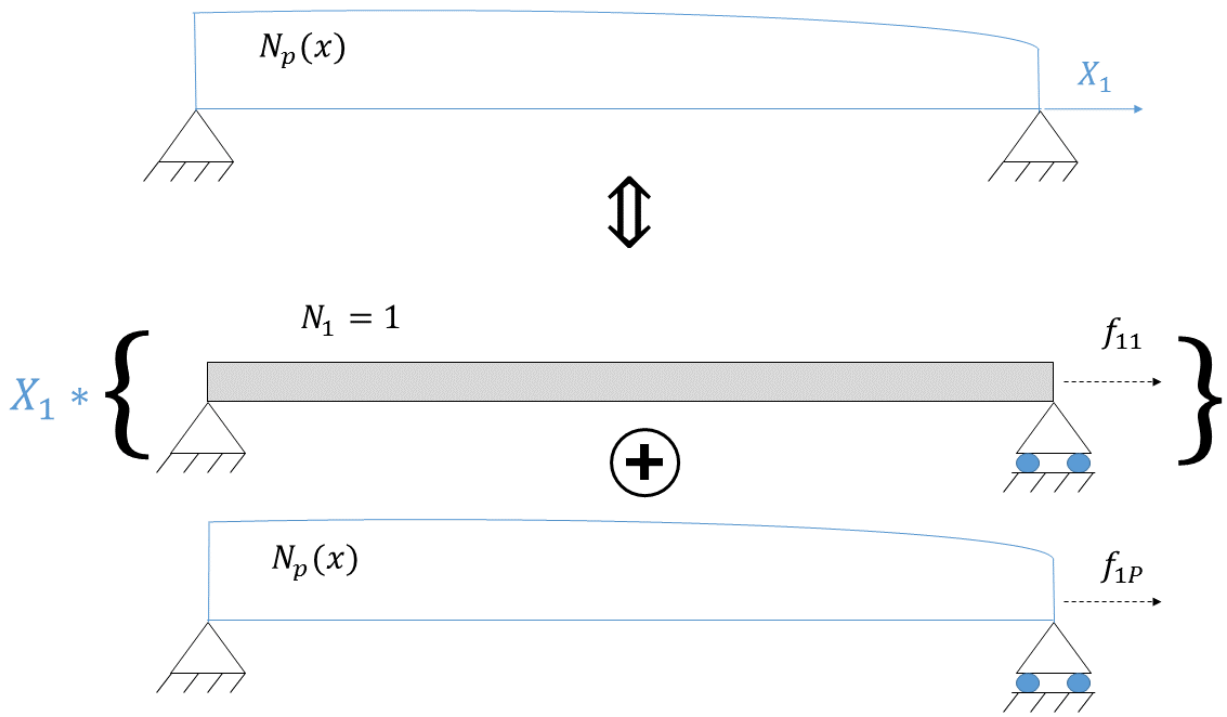


Figure 3.1-9 - Force method to determine the equivalent stress at the nodes

With, for n_1n_4:

- $f_{11} = \int_{n_1}^{n_4} \frac{N_1^2}{EA} dx = \frac{h_x}{EA}$;
- $f_{1P} = \int_{n_1}^{n_4} \frac{N_1 N_P}{EA} dx$;
- $N_P(x) = -\int_{n_1}^{n_4} \frac{\widehat{m}(x)}{I}\left(y_{n4} - \frac{l_y}{2}\right) dx - \int_{n_1}^{x} \frac{\widehat{m}(x)}{I}\left(y_{n4} - \frac{l_y}{2}\right) dx = -25\frac{\left(y_{n4} - \frac{l_y}{2}\right)}{I}\left(\cos\left(\frac{x_{n1}+h_x}{5}\right) - \cos\left(\frac{x}{5}\right)\right)$, where y_{n4} is measured from the bottom of the element;
- $X_1 = -\frac{f_{1P}}{f_{11}} = \frac{\left(y_{n4} - \frac{l_y}{2}\right)}{I} M_{x2}$, the expression of M_{x2} is given in the code below;
- $M_{x1} = \int_{n_1}^{n_4} \widehat{m}(x) dx - M_{x2}$

```
%% Sinusoidal moment loading
F_ponct=zeros(2*Nnode_tot,1);   % Initialisation of the load vector
I=1/12;     % Moment of inertia
for i=1:Nnodex-1
    for j=1:Nnodey-1
       n1=i+(j-1)*Nnodex;
       n2=i+j*Nnodex;
       n3=n2+1;
       n4=n1+1;
       Mx2=25*(-cos(hx/5+nodes(n1,2)/5)+5/hx*(sin(hx/5+nodes(n1,2)/5)-
sin(nodes(n1,2)/5)));
       Mx1=25*(-cos((nodes(n1,2)+hx)/5)+cos(nodes(n1,2)/5))-Mx2;
       sig1=Mx1/I*(nodes(n1,3)-ly/2);   %Stress at node 1 from Mx1
       sig2=Mx1/I*(nodes(n2,3)-ly/2);   %Stress at node 2 from Mx1
       P2=hy*((sig2-sig1)/3+sig1/2);
       P1=(sig1+sig2)/2*hy-P2;
       sig4=Mx2/I*(nodes(n1,3)-ly/2);   %Stress at node 4 from Mx2
       sig3=Mx2/I*(nodes(n2,3)-ly/2);   %Stress at node 3 from Mx2
       P3=hy*((sig3-sig4)/3+sig4/2);
       P4=(sig3+sig4)/2*hy-P3;
       F_ponct(2*n1-1)=F_ponct(2*n1-1)+P1;
       F_ponct(2*n2-1)=F_ponct(2*n2-1)+P2;
       F_ponct(2*n3-1)=F_ponct(2*n3-1)+P3;
       F_ponct(2*n4-1)=F_ponct(2*n4-1)+P4;
    end
end
```

3.1.4.3 Results

For this test, a mesh with 4 elements on the height and 20 elements along the length is used, and the values of the thickness tested are shown in Table 10:

Table 10 - Beam thickness, slenderness ratio, and size of the elements tested

h	4	3	2	1	0.8	0.5
L/h	2.5	3.333	5	10	12.5	20
Element size $h_x * h_y$	0.5*1	0.5*0.75	0.5*0.5	0.5*0.25	0.5*0.2	0.5*0.125

The FE² stiffness matrices of the 6 different elements were computed and are shown in **Erreur ! Source du renvoi introuvable.**.

Figure 3.1-10 shows the evolution of the normalised tip displacement $\frac{w_{num}(L)}{w_{ex}(L)}$ for different values of the slenderness ratio. The results are similar to those observed in the previous test in 3.1.2: the displacement using classic elements with reduced integration is constant but stays above the exact solution (The numerical displacement is 7% higher than the exact solution). The classic fully integrated elements and the FE² elements are both subject to shear locking, but the FE² element performs better than the classic element. For a slenderness ratio of 20, the FE² element leads to a 13% difference with the exact solution, whereas the classic fully integrated leads to 33% difference.

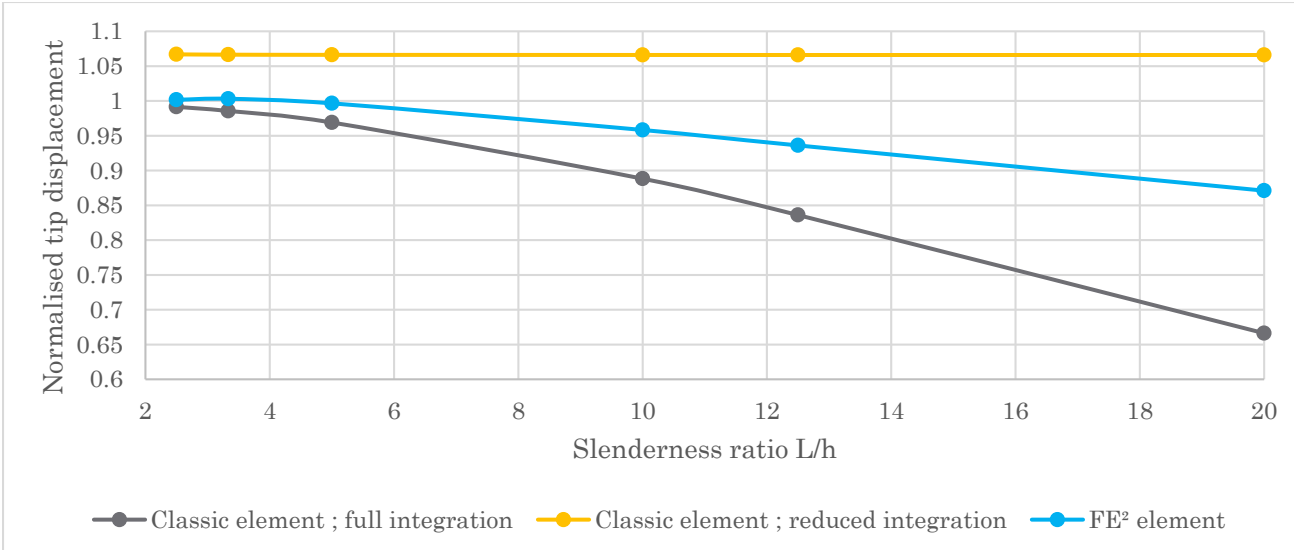


Figure 3.1-10 – Normalised tip displacement $w_{num}(L)/w_{ex}(L)$ under the sinusoidal moment loading

To compare the results of the FE² element with the NURBS elements used by Echter and Bischoff, the L2-norm was calculated and plotted in Figure 3.1-11.

The L2-norm is defined by:

$$\|e_u\|_{L2} = \left(\sum_{x_{nodes}} (w_{ex}(x) - w_{num}(x))^2 \right)^{\frac{1}{2}}$$

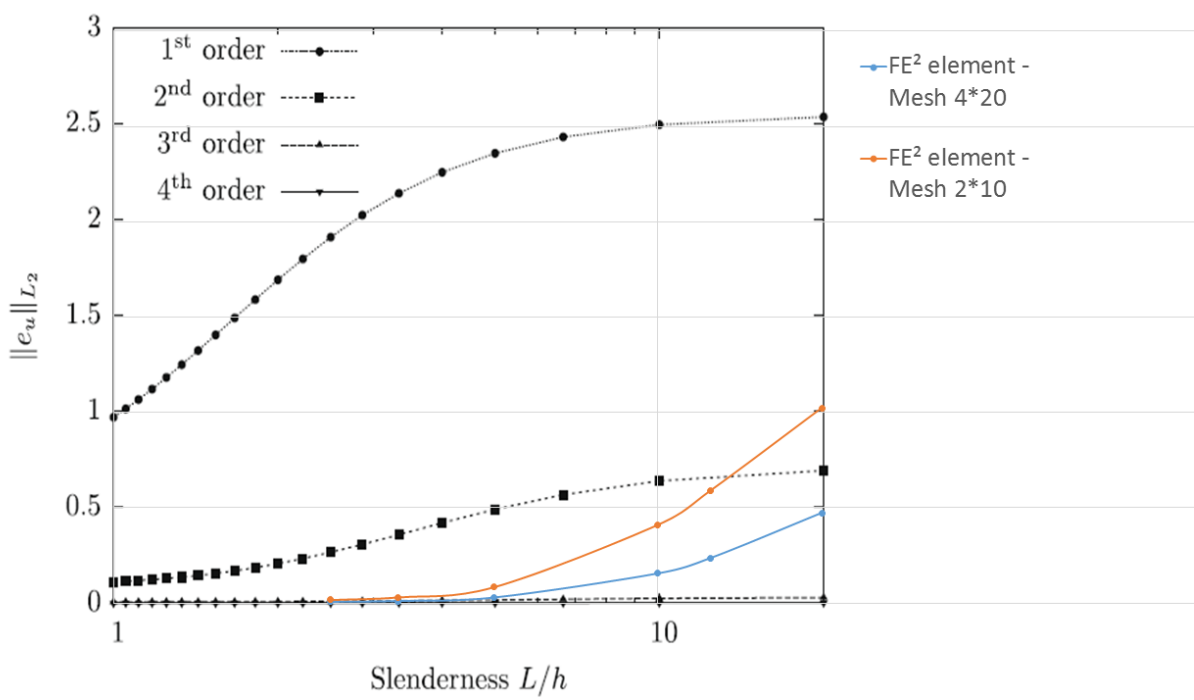


Figure 3.1-11 - Evolution of the L2-norm with the slenderness ratio, comparison with results of the NURBS elements from Echter and Bischoff, [4]

The results shown for the NURBS are given for a mesh of one element. Since the results would not have been comparable with only one FE^2 element, two meshes were tested, to compare the evolution of the L2-norm rather than its absolute value. It can be observed in Figure 3.1-11 that the evolution is similar for both meshes tested (red and blue curves). The FE^2 element seems to perform better than the 1st order and 2nd order NURBS elements for slenderness ratios below 10. However, it can be noted that the slope of the curve keeps increasing for the FE^2 elements, whereas the NURBS elements show an asymptotic behaviour.

To conclude on shear locking, it appears that the FE^2 element is not as efficient as other specific elements on shear locking. The FE^2 element performs better than a classic fully integrated element, and is more accurate and stable than the elements with reduced integration for small values of the slenderness ratio. However, specific elements such as NURBS elements offer better performances against shear locking.

3.3 Performance on volumetric locking

The second type of locking that the FE² elements are tested on is volumetric locking (see 1.3.2).

3.3.1 Description of the test

3.3.1.1 Geometry and material properties

In the elastic domain, volumetric locking appears for solid objects in plane strain state when the Poisson's ratio becomes close to 0.5.

This type of locking can be shown on a hollow cylinder subjected to an internal radial pressure, as shown in Figure 3.3-1.

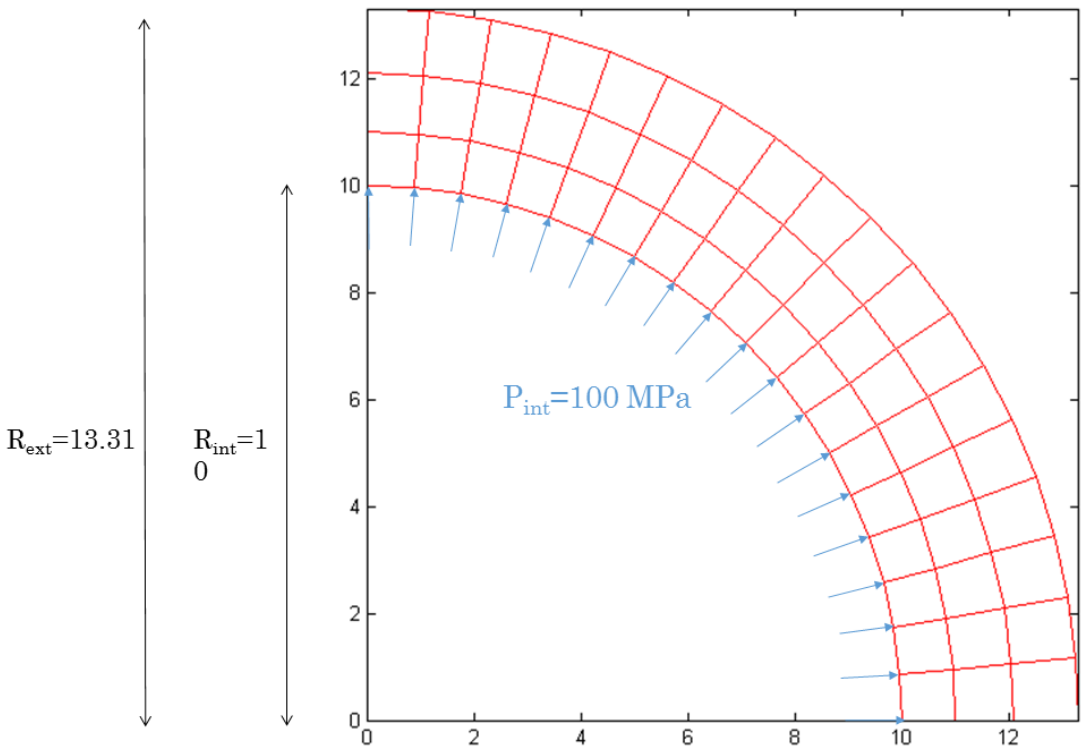


Figure 3.3-1 – Geometry of the problem

Since the cylinder is axisymmetric, only a quarter of it is studied, as represented in Figure 3.3-1. The horizontal border (y=0) is considered fixed in the vertical direction, and the vertical border (x=0) is considered fixed in the horizontal direction.

The internal radius of the tube is $R_{int} = 10\ mm$, and the external radius is $R_{ext} = 13.31\ mm$. The internal pressure is $P_{int} = 100\ MPa$.

The Young's modulus of the material is $E = 5000\ MPa$. The Poisson's ratio is variable to study the appearance of volumetric locking when it becomes close to 0.5.

Under the internal pressure, the tube deforms. The theoretical expression of the radial displacement is (from A.F. Bower, [5]):

$$u_{th}(r) = \frac{(1+\nu)R_{int}^2 R_{ext}^2}{\left(E\left(R_{ext}^2 - R_{int}^2\right)\right)}\left(\frac{(1-2\nu)P_{int}}{R_{ext}^2} * r + \frac{P_{int}}{r}\right)$$

Where r is the radius, and ν is the Poisson's ratio.

3.3.1.2 Meshing

The mesh of the cylinder is regular, as shown in Figure 3.3-1. All the elements are homothetic trapezoids.

In order to create this mesh, GMSH was used to create the nodes on the circular arcs shown in Figure 3.3-2. In GMSH, the arcs were defined and then divided in 18 segments to create the mesh shown in Figure 3.3-1.

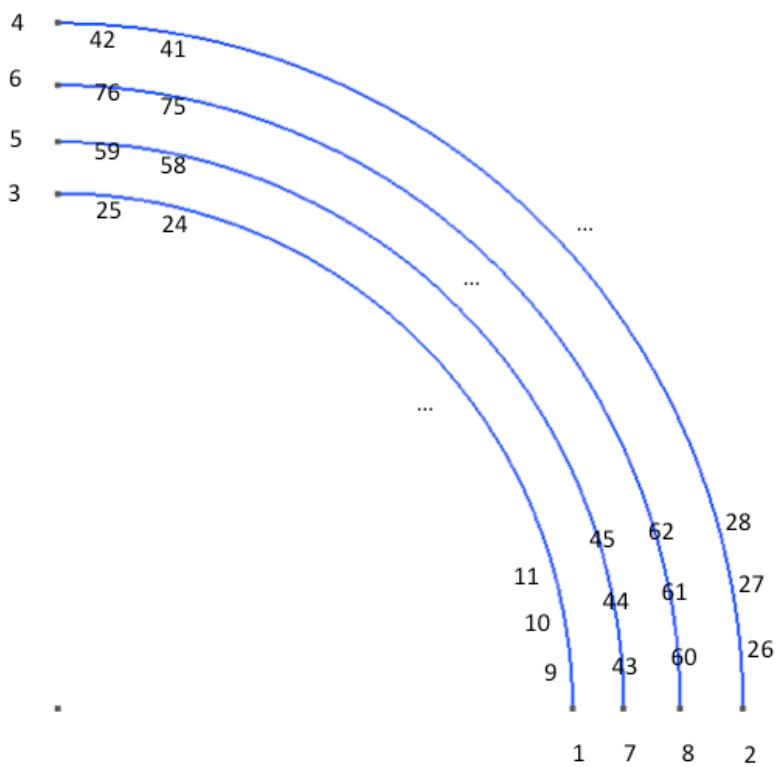


Figure 3.3-2 - Circular arcs for the creation of the mesh

In order to make the trapezoids homothetic, the distances h, kh, k^2h between those circular arcs were chosen so as to respect the proportions displayed in Figure 3.3-3.

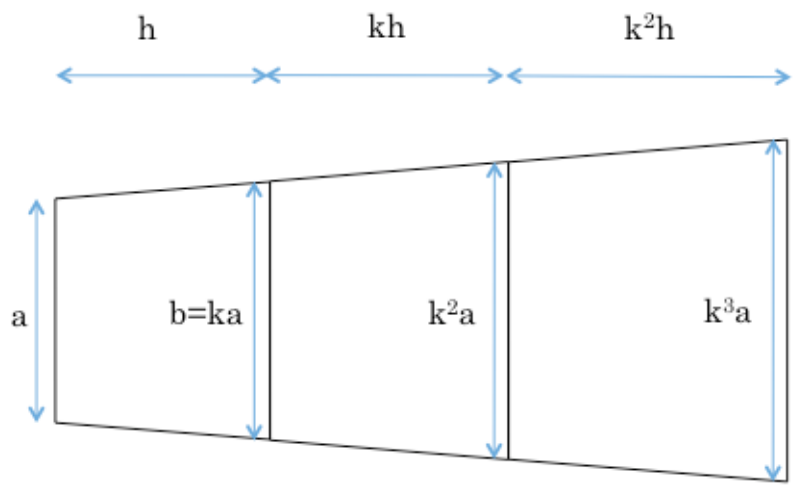


Figure 3.3-3 - Homothetic trapezoids

The chosen value of ratio k is 1.1, to keep the trapezoid's shape close to the shape of a rectangle. The height of the trapezoid h is 1 and the base a can be calculated as follows:

$$a = 2R_{int} \sin\left(\frac{1}{2}\left(\frac{\pi}{2}\frac{1}{18}\right)\right) \approx 0.872$$

Once the nodes are defined, they are imported in MATLAB using the script "READ_MESH.m" (see **Erreur ! Source du renvoi introuvable.**). The vector `paroi_int` contains the nodes of the internal side ($r = 10$), and will be used to define the load vector.

The elements are defined manually using the script written hereafter:

```
elem=zeros(54,4);
elem(1:6,:)=[9 1 7 43 ; 43 7 8 60 ; 60 8 2 26 ; 3 25 59 5 ; 5 59 76 6 ;
6 76 42 4];
for n=7:22
    elem(n,:)=[n+3 n+2 n+36 n+37];
end
for n=23:38
    elem(n,:)=[n+21 n+20 n+37 n+38];
end
for n=39:54
    elem(n,:)=[n+22 n+21 n-13 n-12];
end
Nel=length(elem);
```

3.3.1.3 Load vector

The load is an internal pressure `p_int=100` MPa. Equivalent forces in the horizontal and the vertical direction have to be computed to be used in the program.

In the program, the internal side of the tube is divided in small segments of length a. The contribution of the pressure on each of these segments is calculated at the nodes, projected onto the Cartesian axes, and then added to the load vector.

To simplify the calculation, the pressure on a segment is considered perpendicular to the segment, as shown in Figure 3.3-4. Each node, except the extreme ones, receives the contribution of two elements.

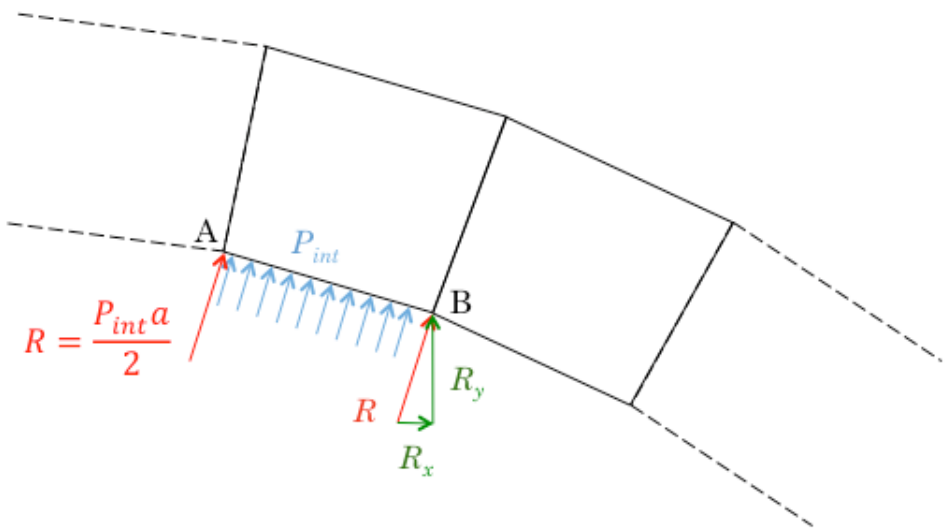


Figure 3.3-4 - Calculation of the equivalent nodal loads

The MATLAB script used to compute the load vector is written hereafter:

```
%% Load vector of the tube
F_ponct=zeros(2*Nnode_tot,1);
for i=1:length(paroi_int)
    nA=paroi_int(i,1);
    nB=paroi_int(i,2);
% Cartesian coordinates of the nodes
    xA=nodes(nA,2);
    yA=nodes(nA,3);
    xB=nodes(nB,2);
    yB=nodes(nB,3);
% Length of the linear element
    xl=abs(xA-xB);
    yl=abs(yA-yB);
    l=sqrt((xA-xB)^2+(yA-yB)^2);
% Equivalent radial point load at the nodes [N]
    R=p_int*l/2;
% Projection of the load on the horizontal and vertical axes
    Rx=R/sqrt(1+(xl/yl)^2);
    Ry=xl/yl*Rx;
% Load vector
    F_ponct(2*nA-1)=F_ponct(2*nA-1)+Rx;
    F_ponct(2*nA)=F_ponct(2*nA)+Ry;
    F_ponct(2*nB-1)=F_ponct(2*nB-1)+Rx;
    F_ponct(2*nB)=F_ponct(2*nB)+Ry;
end
```

3.3.2 Element

The geometry of the element has been described in 3.3.1.2. For each value of the Poisson's ratio, only one equivalent stiffness matrix has to be computed through the process described in 2.2.1. The mesh used contains 762 elements and is shown in Figure 3.3-5.

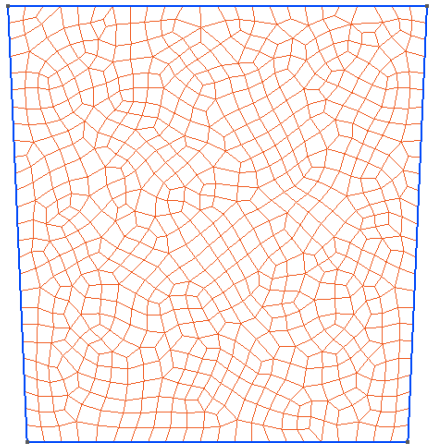


Figure 3.3-5 - Mesh of the trapezoid element

The elements are homothetic trapezoids, therefore, once the stiffness matrix of the basis element is computed, the stiffness matrix of the other elements can be obtained by simply applying the formulas from Equations [*2.2.8*] and [*2.2.10*]:

```
for i=1:Nel % Element number
% Coordinates of the nodes of the element
    for j=1:Nnode
        X(j)=nodes(elem(i,j),2);
        Y(j)=nodes(elem(i,j),3);
    end
v=atan((nodes(3,3)-nodes(25,3))/(nodes(25,2)-nodes(3,2)));
theta=atan((X(4)-X(1))/(Y(4)-Y(1)))+theta0;
    for j=1:2*Nnode
        for k=1:2*Nnode
            if mod(j,2)~=0
                if mod(k,2)==0
Kel(j,k)=cos(theta)*(cos(theta)*Kel_eq(j,k)+sin(theta)*Kel_eq(j+1,k))-
sin(theta)*(cos(theta)*Kel_eq(j,k-1)+sin(theta)*Kel_eq(j+1,k-1));
                else
Kel(j,k)=cos(theta)*(cos(theta)*Kel_eq(j,k)+sin(theta)*Kel_eq(j+1,k))+
sin(theta)*(cos(theta)*Kel_eq(j,k+1)+sin(theta)*Kel_eq(j+1,k+1));
                end
            elseif mod(k,2)==0
Kel(j,k)=cos(theta)*(cos(theta)*Kel_eq(j,k)-sin(theta)*Kel_eq(j,k-1))-
sin(theta)*(cos(theta)*Kel_eq(j-1,k)-sin(theta)*Kel_eq(j-1,k-1));
            else
Kel(j,k)=cos(theta)*(cos(theta)*Kel_eq(j,k)+sin(theta)*Kel_eq(j,k+1))-
sin(theta)*(cos(theta)*Kel_eq(j-1,k)+sin(theta)*Kel_eq(j-1,k+1));
            end
        end
    end
```

Note: the angle `theta0` is the angle of the left side of the trapezoid sown in Figure 3.3-5 with the vertical, since the stiffness matrix was computed for a trapezoid with horizontal bases.

3.3.3 Results

The test was realized for 12 different values of the Poisson's ratio which are displayed in Table 11 along with the theoretical displacement of the internal border ($r = 10$):

Table 11 - Values of the Poisson's ratio tested and the corresponding exact displacement

$\boldsymbol{\nu}$	0	0.1	0.2	0.3	0.4	0.45
$\boldsymbol{u(r = 10)}$ **[mm]**	0,7184	0,7332	0,7377	0,7318	0,7155	0,7034
$\boldsymbol{\nu}$	0.46	0.47	0.48	0.49	0.499	0.4999
$\boldsymbol{u(r = 10)}$ **[mm]**	0,7007	0,6979	0,6950	0,6920	0,6891	0,6889

For each value of the Poisson's ratio, the deformation of the cylinder is computed using the FE² stiffness matrix of the trapezoid elements. The numerical results obtained are compared to those obtained using classic fully integrated elements. Unlike what was done for the shear-locking problem, the reduced integration was not used in this case, because it provoked hourglassing, as shown in Figure 1.2-3.

However, two types of FE² elements were tested: the first one where the sub-mesh was done with fully integrated elements, and the second one where reduced integration was used in the sub-mesh.

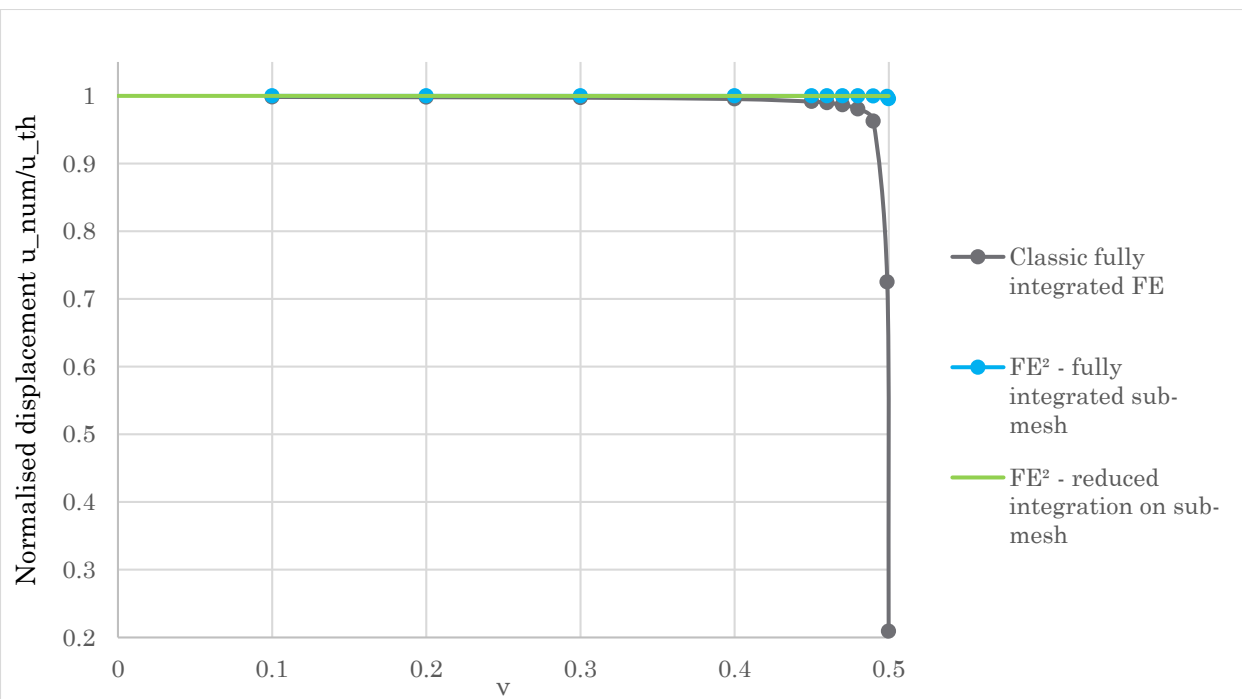


Figure 3.3-6 - Normalised internal displacement for different values of the Poisson's ratio

The normalised displacement $\frac{u_{num}}{u_{th}}$ is shown in Figure 3.3-6. The locking phenomenon appears very clearly for the classic fully integrated element. As the Poisson's ratio becomes

close to 0.5, the normalised displacement becomes exponentially smaller, with the numerical displacement reaching only 20% of the theoretical displacement for $\nu = 0.4999$.

The FE² element, however, gives good results, as the value of the normalised displacement stays close to 1 for both integrations tested.

Figure 3.3-7 shows a close-up of the percentage of error $\varepsilon = \frac{|u_{th} - u_{num}|}{u_{th}}$. Again, it appears that the FE² elements perform better than the classic element, even for values of the Poisson's ratio that are not close to 0.5.

Moreover, it can be observed that the FE² element with fully integrated elements in the sub-mesh displays a slight locking, with an error reaching 0.5% for $\nu = 0.4999$. However, the FE² with reduced integration shows no locking at all, and the error stays at a value close to 0 for any Poisson's ratio.

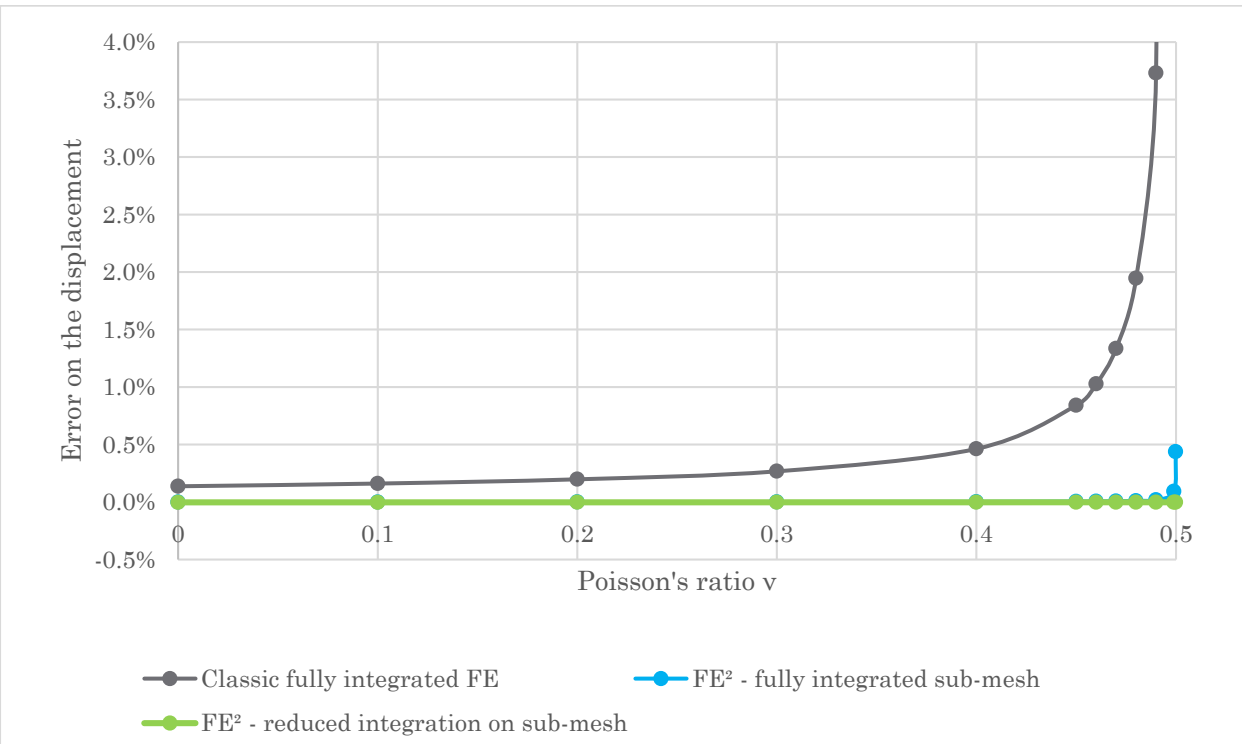


Figure 3.3-7 - Error on the displacement as a function of the Poisson's ratio

The difference between full integration and reduced integration appears even more clearly in Figure 3.3-8. This figure displays the error on the displacement for a Poisson's ratio $\nu = 0.4999$. The error was computed using four different elements:

- 762 fully integrated elements on the sub-mesh: $\epsilon = 0.438\%$;
- 355 fully integrated elements on the sub-mesh: $\epsilon = 0.708\%$;
- 762 elements numerically integrated using 1 Gauss point in the sub-mesh: $\epsilon = 0.002\%$;
- 355 elements numerically integrated using 1 Gauss point in the sub-mesh: $\epsilon = 0.0114\%$.

Once again, it appears that the reduced integration gives much better results, with an error lower than 0.02%. This can be explained by the fact that the volumetric locking is caused by the incompressibility of the material, which has to be verified on the whole element for a fully integrated element. The element interpolation functions are unable to approximate a volume-preserving strain field at all the integration points. Using reduced integration decreases the number of integration points at which the incompressibility constraint has to be met, which makes it possible for the interpolation functions to make the volumetric strain disappear.

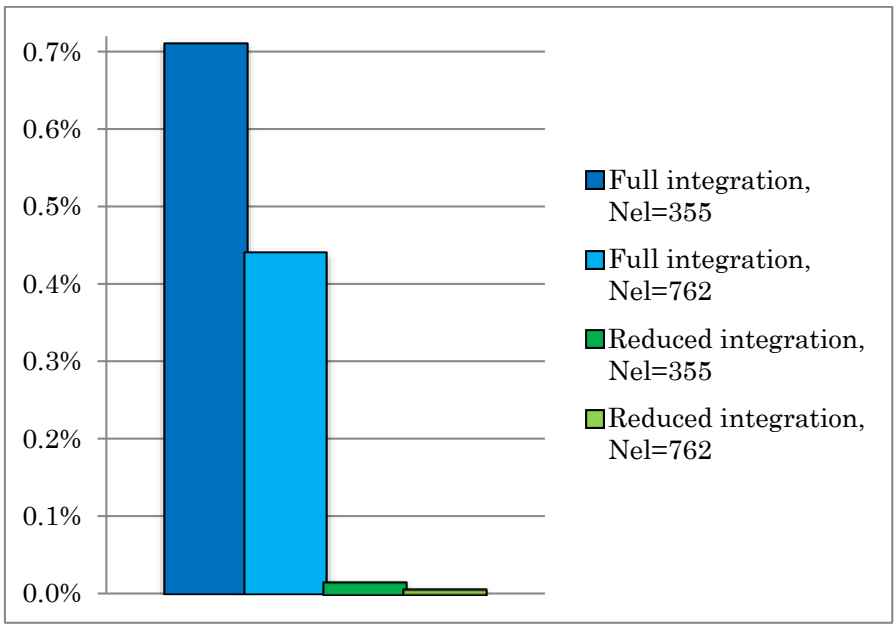


Figure 3.3-8 - Error on the displacement (%) for a Poisson's ratio of 0.4999 for different types of FE² – logarithmic scale

The results displayed in Figure 3.3-8 also show that increasing the size of the sub-mesh (i.e. increasing the number of elements in the sub-mesh) gives better performance on volumetric locking. With fully integrated elements, the error on the displacement is 1.6 times smaller when the number of elements is doubled in the sub-mesh. For the reduced integration, the absolute error is 5 times smaller when the size of the sub-mesh is doubled.

Nevertheless, the error is small (inferior to 1%), even with only 355 elements in the sub-mesh, which validates the choice of a minimum of 550 elements for the sub-mesh that was made in 2.2.1.6.

The use of a larger number of elements in the sub-mesh makes the element more deformable, and decreases the stiffness of the FE² element. Therefore, the element is less likely to be subject to locking.

3.3.5 Comparison with an industrial element

The FE² technique has proven very efficient against volumetric locking in comparison to classic elements. The test described earlier was also performed with industrial elements from the software SAMCEF.

In these elements, the problem of volumetric locking is tackled by using different integration schemes for the deviatoric and the volumetric parts of the constitutive law [6]. While the deviatoric part is integrated in a classic way, Lagrange multipliers are used for the volumetric part. The Lagrange multipliers connect the internal pressure to the change of volume. Two laws can be used, where K is the bulk modulus, p the internal pressure, and J the change of volume:

- Bulk1: the constraint is written $\ln(J) = p/K$;
- Bulk2, 22, 32: the constraint is written $J - 1 = p/K$.

Table 12 shows the number of Lagrange multipliers used for the different options of bulk behaviour. At degree 2, Bulk 1|2 correspond to bi or trilinear pressure fields, Bulk 22 corresponds to a linear pressure field, and Bulk 32 corresponds to a bilinear pressure field with additional displacement unknowns added at the centre of the element.

Table 12 - Number of Lagrange multipliers for the quadrangle, from [6]

Degree	Number of Lagrange multipliers Bulk1|2	Number of Lagrange multipliers Bulk22	Number of Lagrange multipliers Bulk32
1	1		
2	4	3	4

The cylinder was meshed using first-degree SAMCEF elements, and the error on the displacement is shown in Figure 3.3-9 for Bulk parameters 0 (no Lagrange multiplier) and 1.

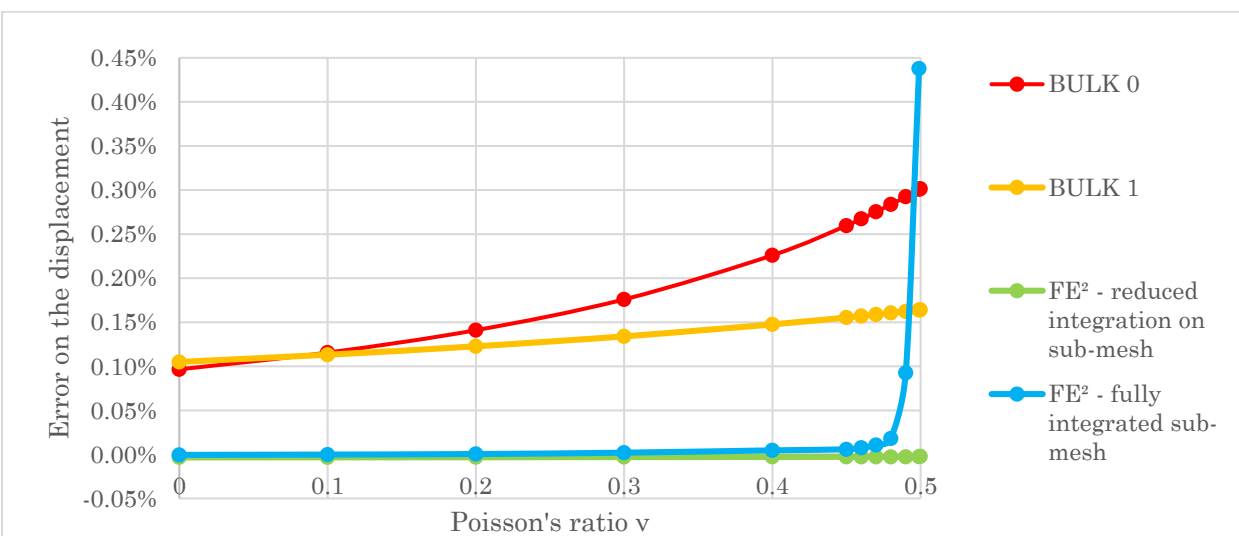


Figure 3.3-9 - Error on the displacement for the FE² elements and the SAMCEF element as a function of the Poisson's ratio

It can be observed that the use of the Lagrange multiplier improves the solution with regards to the volumetric locking. The evolution of the error with the Poisson's ratio is almost linear with the parameter Bulk1, contrarily to the evolution of the error with the FE^2 or the classic elements. Even though the FE^2 element with full integration gives better results than the SAMCEF elements for small values of the Poisson's ratio, it is much less stable when the material approaches incompressible behaviour. However, the FE^2 element with reduced integration shows better performance than both SAMCEF elements, with an error that virtually stays equal to zero.

The Bulk2 parameter and the 2nd degree SAMCEF elements were also tested. To limit the computations, the displacement was only computed for the most critical value of the Poisson's ratio: 0.4999. The results are displayed in Figure 3.3-10.

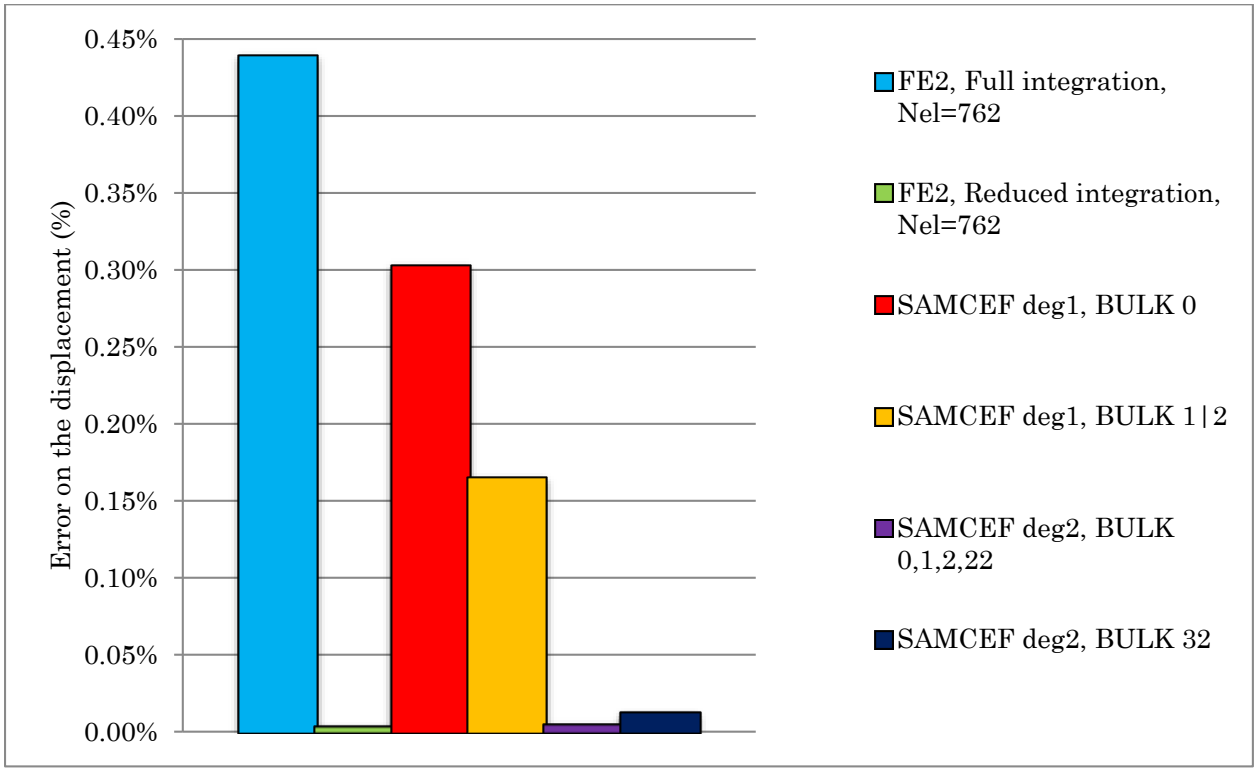


Figure 3.3-10 - Error on the displacement for a Poisson's ratio of 0.4999

Bulk 1 and 2 give the same result for the 1st degree element. The 2nd degree elements perform much better than the first degree elements, with an error below 0.01%. The FE^2 element with reduced integration on the sub-mesh performs as well as the second degree SAMCEF elements.

Conclusion

The objective of this study was to assess the performance of the FE^2 technique against the problems of locking that can occur with classic finite elements.

Firstly, a classic finite element program was developed. This program can be used to solve problems in the elastic domain, using different material parameters and different geometries. This program was validated by a set of tests to verify its proper functioning.

Secondly, new elements were created using the FE^2 technique and the finite element program mentioned before. These elements were created by calculating an equivalent stiffness matrix, based on the meshed element. The use of a mesh generator allowed to test different types and sizes of meshes for the FE^2 element. The change in the stiffness matrix after a transformation of the element (homothetic transformation, rotation) was determined through analytical considerations. Unfortunately, after testing these new elements, it appeared that elements containing at least one slanting side did not perform well and gave erroneous results. The elements with rectangular or quasi-rectangular shapes, however, showed a satisfying behaviour under the tests that were performed.

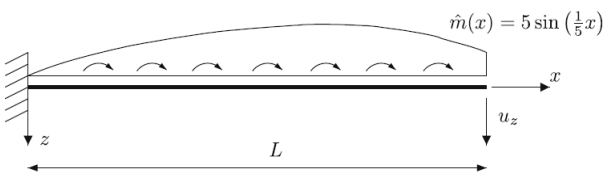
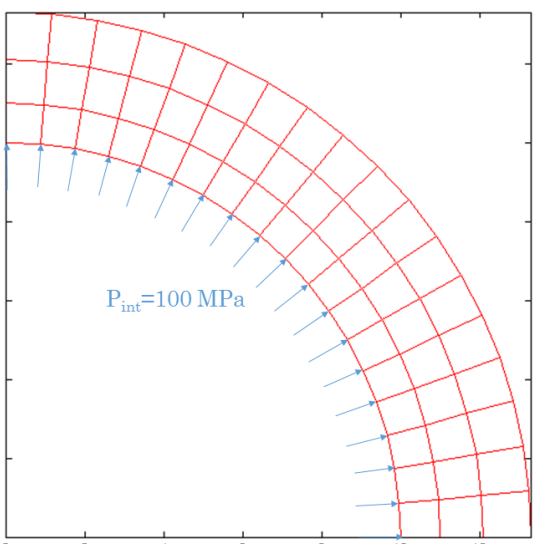
Finally, the new elements were used to test their performance on the problem of shear locking and of volumetric locking. The elements were tested and compared to classic elements on a Timoshenko beam with a variable thickness. Several meshes were tested and two types of loading were used. This revealed that the FE^2 element performed better than the classic fully integrated element, but not better than elements with reduced integration for high values of the slenderness ratio.

To test the elements against volumetric locking, we studied the deformation of a cylinder subjected to an internal pressure for different values of the Poisson's ratio. Two types of FE^2 elements were tested: one using fully integrated elements on the sub-mesh, others, using reduced integration in the sub-mesh. The full integration FE^2 elements lead to a very slight locking, whereas no locking was observed with the reduced-integration FE^2 elements. Moreover, when comparing the results to those obtained with industrial elements, the FE^2 elements with reduced integration performed as well as second degree elements against volumetric locking.

The results obtained are summarized in Table 13.

The results of this study, especially on volumetric locking, are very promising. However, for the moment, the calculation of the FE^2 stiffness matrix has to be computed individually in pre-processor for each geometry of element and material properties. Moreover, a functioning model for non-rectangular shapes still has to be developed, using elastic support elements on the edge nodes for instance. In order for the FE^2 elements to be used on more general problems, it would be essential to develop a method to generate the equivalent stiffness matrix from the material properties and the geometry of the element.

Table 13 - Summary of the results on locking problems

Test	Results	Compared to
Cantilever beam with unitary tip displacement	• Less than 5% error for slenderness ratios between 5 and 20 • Better performance than fully integrated elements • Reduced integration performs better for slenderness ratios above 25	Theory, classic elements
Cantilever beam subjected to sinusoidal moment loading 	• Appearance of the shear locking phenomenon • Non satisfactory behaviour compared to NURBS elements	NURBS elements [4]
Cylinder with radial internal pressure 	• Less than 0.5% error for a Poisson's ratio of 0.4999 compared to the theory • Error on the displacement almost 200 times smaller with FE² elements than with classic elements for a Poisson's ratio of 0.4999 • Negligible error (±0.002%) for FE² elements with reduced integration	Theory, classic elements
	• Performance equivalent to first degree SAMCEF elements when using full integration on the sub-mesh • Performance equivalent to second degree SAMCEF elements when using reduced integration on the sub-mesh	SAMCEF elements

References

- [1] J-L. Batoz, G. Dhatt, Modélisation de structures par elements finis, Volume 1: solides élastiques, Hermès, Paris, 1990.
- [2] E. Qiuli Sun, Shear Locking and Hourglassing in MSC Nastran, ABAQUS, and ANSYS. MSC Software Users Meeting, 2006
- [3] C. Geuzaine and J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, Volume 79, Issue 11, pages 1309-1331, 2009
- [4] R.Echter, M. Bischoff, Numerical efficiency, locking and unlocking of NURBS finite elements. Comput. Methods Appl. Mech. Engrg. 199 (2010) 374–382.
- [5] A. F. Bower, Applied mechanics of solids, CRC Press, 2009.
- [6] Samtech, SAMCEF, User Manual-17.0, 2015.