الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة الشهيد الشيخ العربي التبسي – تبسة
Echahid Cheikh Larbi Tébessi University – Tebessa

Final dissertation

For obtaining the Academic Master's degree

Specialty: Mechanical Construction

Department: Mechanical Engineering

Title

Programming of 2D triangular finite element to model the mechanical behavior of plane structure and validation using simulation software

Presented by:

BOUCHEGRA Chaima

In front of the jury:

President	*Dr. YOUNES Ramdane*	*MCA*
Supervisor	*Pr. DEGHBOUDJ Samir*	*Professor*
Examiner	*Dr. LEMITA Nour El Houda*	*MCB*

Date of defense:03/06/2025

Abstract

In the field of mechanical engineering, analyzing the behavior of plane structures is a fundamental task that engineers constantly face. However, traditional methods for solving these problems are often slow, cumbersome, and error-prone. To address these challenges, the finite element method provides an efficient and reliable solution by dividing complex structures into smaller elements, facilitating to analysis process. This study focuses on the development of a special program based on the two-dimensional triangular element (T3) to model the mechanical behavior of plane structures. It calculates basic mechanical properties such as displacements, stress, and strain, making it a powerful tool for analyzing the behavior of structures. To ensure the accuracy of the obtained results, they were verified by comparing them with simulation results extracted from well-known software. This study aims to contribute to the field of mechanical constructions by providing an effective solution for performing accurate structural analyses.

Key words: Finite element method, Mechanical behavior, Plane structures, Structural analysis, 2D triangular element, T3 element.

ملخص

في مجال الهندسة الميكانيكية، يعد تحليل سلوك الهياكل المستوية من المهام الأساسية التي يواجهها المهندسون بشكل مستمر. إلَّا أن الطرق التقليدية لحل هذه المشكلات غالبًا ما تكون بطيئة ، مرهقة، و معرضة للأخطاء. و لمواجهة هذه التحديات، توفر طريقة العناصر المحدودة حلًّا فعالًا و موثوقًا من خلال تقسيم الهياكل المعقدة إلى عناصر أصغر مما يسهل عملية التحليل. تركز هذه الدراسة على تطوير برنامج خاص يعتمد على العنصر المثلثي ثنائي الأبعاد (T3) *لنمذجة السلوك الميكانيكي للهياكل المستوية، حيث يقوم بحساب الخصائص الميكانيكية المهمة كالإنتقال، الإجهاد، و التشوه، مما يجعله أداة قوية لتحليل سلوك الهياكل. و لضمان دقة النتائج الحاصلة، تم التحقق منها من خلال مقارنتها مع نتائج المحاكاة المستخرجة من برامجيات معروفة. تهدف هذه الدراسة إلى تقديم مساهمة في مجال الإنشاءات الميكانيكية من خلال توفير حل فعال لإجراء تحليلات هيكلية دقيقة.*

الكلمات المفتاحية: طريقة العناصر المحدودة، السلوك الميكانيكي، الهياكل المستوية، تحليل الهياكل، عنصر مثلثي ثنائي الأبعاد، *العنصر* T3.

Acknowledgment

First and foremost, I would like to thank Allah, whose grace and guidance have made the completion of this work.

I extend my sincere thanks and gratitude to my esteemed supervisor, Pr. DEGHBOUDJ Samir for his trust in me and for the support and guidance he provided throughout the preparation of this dissertation. His valuable advice and constant guidance were a source of pride and honor for me.

I would also like to express my deep gratitude and appreciation to the esteemed members of the jury Dr. YOUNES Ramdane and Dr. LEMITA Nour El Houda for agreeing to discuss and evaluate my dissertation.

Finally, I can only extend my deepest gratitude and affection to my Brothers and Friends, who have been a constant support to me throughout this journey. My heartfelt thanks go to my beloved mother, who has been a source of compassion and prayer, and my companion in every difficult moment, her presence in my life has been a great blessing. I also extend my feelings of pride and gratitude to my Father, who was always been a source of strength and inspiration in my life, his unlimited support, great confidence in me, and encouraging words were the primary motivation to continue and overcome all difficulties.

Dedication

To my beloved parents, the foundation of my strength

And the reason for where I am today,

To my brothers, thank you for

standing by my side,

To all my friends, whose presence has been

special in my life.

To everyone who has had a positive impact

On my journey.

I dedicate this humble work to you.

Thank you all.

List of Content

Abstract ..... *I*

ملخص ..... *II*

Acknowledgment ..... *III*

Dedication ..... *IV*

List of Content ..... *V*

List of Abbreviations ..... *VIII*

List of Tables ..... *IX*

List of Figures ..... *XI*

General Introduction *1*

Chapter1 Generalities and bibliographic synthesis *2*

1.1 Introduction 2

1.2 Finite element method 2

1.2.1 Historical evolution 2

1.2.2 Definition 3

1.3 Classification of FEM Models 3

1.3.1 One-Dimensional Finite Element Modeling (1D FEM) 3

1.3.2 Two-Dimensional Finite Element Modeling (2D FEM) 4

1.3.3 Three-Dimensional Finite Element Modeling (3D FEM) 4

1.4 Steps of the finite element method 5

1.5 Benefits of the Finite Element Method 5

1.6 Plates 6

1.6.1 Definition 6

1.6.2 Types of plates 6

1.6.3 Field of application 8

1.7 Plane structures analysis using FEM 8

1.8 Conclusion 8

Chapter2 Two dimensional triangulat finite element *9*

2.1 Introduction 9

2.2 Classification of two-dimensional finite element 9
2.3 Types of 2D triangular elements 9
2.4 Mathematical formulation of 2D triangular finite element 10
2.5 Triangular element T3 10
2.5.1 Linear Shape Function 11
2.5.2 Plane stress and Plane strain 13
2.5.3 Strain-displacement relation 16
2.5.4 Stress-Strain Relationship 17
2.5.5 Integration over a triangular element 17
2.5.6 Elementary Stiffness Matrix 18
2.6 Example calculation of triangular element T3 20
2.7 Conclusion 30

Chapter3 Programming and validation 31

3.1 Programming Languages 31
3.1.1 Overview of programing languages 31
3.1.2 How programming languages evolved 31
3.1.3 Development Tools 32
3.1.4 Basic concepts in programming 32
3.1.5 Development of Programming Paradigms 32
3.1.6 Programming Language Used in Numerical and Engineering Analysis *32*
3.2 Chosen Programming Language (MATLAB) 33
3.2.1 Definition and its historical background 33
3.2.2 MATLAB Desktop 34
3.2.3 Types of .m File and Their Applications 35
3.3 Programming using the finite element method 37
3.3.1 General structure of the finite element program 41
3.3.2 Program for Analyzing Plane Structures 37
3.4 Comparing program results with analytical solution results 41
3.5 Validation results using ABAQUS software 44
3.5.1 Overview of ABAQUS 44
3.5.2 ABAQUS program components 44
3.5.3 ABAQUS Unit System 44
3.5.4 Comparing program results with ABAQUS results 45
3.6 Validation result using RDM6 software 58
3.6.1 Overview of RDM6 58
3.6.2 Main modules in RDM6 59

3.6.3 Comparing programming results with RDM6 results 59
3.7 Conclusion 66
General Conclusion *68*
References *69*
Annexes *72*

List of Abbreviations

FEM	Finite Element Method
FEA	Finite Element Analysis
1D	One dimensional
2D	Two dimensional
3D	Three dimensional
DOF	Degree of Freedom
T3	Three node Triangular element
T6	Six node Triangular element
CST	Constant Strain Triangle
GUI	Graphical User Interface
OOP	Object-Oriented Paradigms
SVN	Subversion
IDEs	Integrated Development Environments

List of Tables

Table 2.1 Nodal coordinates of the structure..... 21

Table 2.2 Element connectivity..... 21

Table 2.3 Boundary conditions applied to the case study..... 26

Table 2.4 Results obtained from calculating the displacements in the system..... 29

Table 2.5 The basic results that relate to each element of the case studied..... 29

Table 3.1 Forces and displacements results obtained by manual calculation for T3 element..... 42

Table 3.2 Stress and strain results obtained by manual calculation for T3 element..... 43

Table 3.3 Results of displacements and forces obtained from the program for T3 element..... 43

Table 3.4 Results of stresses and strains obtained from the program for T3 element..... 43

Table 3.5 Relative error between manual solution and program output for forces and displacements..... 43

Table 3.6 Relative error between manual solution and program output for stresses and strains. 43

Table 3.7 The system of units used in ABAQUS..... 45

Table 3.8 Results of displacements and forces computed by the program for a thin plate..... 46

Table 3.9 Results of stresses and strains computed by the program for a thin plate..... 46

Table 3.10 Displacements and forces results for the first case extracted by ABAQUS software50

Table 3.11 Stresses and strains results for the first case extracted by ABAQUS software..... 51

Table 3.12 Relative error of displacements and forces for the first case between the results of the ABAQUS software and those calculated by the developed program..... 51

Table 3.13 Relative error of strains and stresses for the first case between the results of the ABAQUS software and those calculated by the developed program..... 51

Table 3.14 Results of displacements and forces for the second case computed by the program. 52

Table 3.15 Results of stresses and strains for the second case computed by the program..... 53

Table 3.16 Displacements and forces results for the second case provided by ABAQUS software 55

Table 3.17 Stresses and strains results for the second case provided by ABAQUS software..... 56

Table 3.18 Relative error of displacements and forces for second case between the results of the ABAQUS software and those calculated by the developed program..... 57

Table 3.19 Relative error of stresses and strains for second case between the results of the ABAQUS software and those calculated by the developed program..... 58

Table 3.20 Main modules in RDM6..... 59

Table 3.21 Results extracted from the developed program for the plate..... 60

Table 3.22 Results of structural analysis of the model using RDM6 software..... 65

Table 3.23 Relative error between the results of the proposed program and those extracted from the RDM6 software for the studied structures..... 66

List of Figures

Fig. 1.1 Discretization of a structure into nodes and elements (tooth of a gear wheel)..... 3

Fig. 1.2 Examples of 1D finite element: a) Axial deformation of a bar; b) Bending behavior of a beam..... 3

Fig. 1.3 Examples of 2D finite element: a) 6-node Triangular element; b) Quadrilateral element with 4 nodes in the natural coordinate; c) Triangular mesh for a geometric domain; d) Simple three-node triangular element..... 4

Fig. 1.4 Examples of 3D finite element: a) Tetrahedral element with 1o nodes; b) Hexahedral element with 20 nodes..... 4

Fig. 1.5 Description of a plate..... 6

Fig. 1.6 Deflection and rotation of the plate under the influence of loads according to Kirchhoff plate theory..... 7

Fig. 1.7 Representation of displacement and rotation according to Mindlin plate theory..... 7

Fig. 2.1 First-order triangular element(T3) and its degrees of freedom..... 11

Fig. 2.2 Positions of shape functions..... 11

Fig. 2.3 Representation of the stress field in a plane stress state..... 14

Fig. 2.4 Representation of the strain field in a plane strain state..... 15

Fig. 2.5 Uniformly distributed load in a triangular element..... 20

Fig. 2.6 Example of the calculation of a triangular element T3..... 21

Fig. 2.7 Discretization of the triangular elements..... 22

Fig. 3.1 The most popular programming languages in various fields..... 31

Fig. 3.2 MATLAB logo..... 33

Fig. 3.3 The MATLAB Desktop..... 35

Fig. 3.4 Example of solving a system of linear equations using MATLAB..... 36

Fig. 3.5 Example of plot a graph using function file..... 36

Fig. 3.6 Flowchart of the analysis program process sequence..... 38

Fig. 3.7 Initializing the workspace..... 38

Fig. 3.8 Material's mechanical properties..... 39

Fig. 3.9 Nodal coordinates..... 39

Fig. 3.10 Total number of elements and node connectivity..... 39

Fig. 3.11 Stiffness matrix for a three-node triangular element..... 40

Fig. 3.12 Input of boundary conditions..... 40

Fig. 3.13 Force vector..... 41

Fig. 3.14 Gaussian elimination method.....41
Fig. 3.15 Solving the system of equation.....41
Fig. 3.16 Computing strain and stress for each element.....42
Fig. 3.17 Thin plate with concentrated force.....46
Fig. 3.18 2D part definition of the plate.....47
Fig. 3.19 Properties of the material.....47
Fig. 3.20 Create section.....48
Fig. 3.21 Preparing the analysis step.....48
Fig. 3.22 Applying loads and boundary condition.....49
Fig. 3.23 Meshing the plate into triangular elements.....49
Fig. 3.24 Job completed successfully.....49
Fig. 3.25 Viewing the results.....50
Fig. 3.26 Create the model.....53
Fig. 3.27 Boundary conditions and applied forces.....54
Fig. 3.28 Divide the mesh into 30 linear triangular elements.....54
Fig. 3.29 Viewing the results.....55
Fig. 3.30 Geometric model of a 2D flat plate.....60
Fig. 3.31 Geometric model of a plate.....61
Fig. 3.32 Element mesh using block mesh technique.....61
Fig. 3.33 Definition of study type.....62
Fig. 3.34 Properties of the material used.....62
Fig. 3.35 Applying load and boundary conditions.....63
Fig. 3.36 Viewing the results.....63
Fig. 3.37 Exporting the report results.....64

General Introduction

Mechanical engineering is an important field in the field of engineering, due to its critical importance in the design, analysis, and maintenance of various systems and structures, whether in mechanics, civil engineering, transportation, aviation, or other fields.

With the advancement of technology and the complexity of structures and models, it has become necessary to develop numerical tools that accurately analyze these structures to understand their behavior. One of the most prominent of these tools is the Finite Element Method, which is among the most widely used methods for analyzing complex engineering models, especially those related to mechanical structures. However, this is not complete without a programming environment to solve complex systems that describe the behavior of materials under loads.

Hence, the idea of this dissertation came about, which aims to program a two dimensional (2D) numerical model using the three node triangular element (T3) to analyze the mechanical behavior of plane structures using MATLAB language. To ensure the reliability of this proposed program, verification was carried out using ABAQUS software, known for its accurate results, then to the RDM6 software.

This dissertation is divided into three chapters;

The first chapter covers the general principals of the finite element method, starting with its historical development and basic concepts, as well as its application areas, up to the types of elements. It also includes the possible types of analysis and explains the use of each according to the type of structure being studied.

The second chapter focuses on the mathematical derivations related to the T3 element, including shape functions, the stiffness matrix, the expression of stresses and strains, and everything related to the theoretical formulation.

The third chapter introduces the developed program, explaining each phase of the calculation, from input data to results. It also includes a detailed comparison between the results of the proposed program and traditional analytic results, as well as with ABAQUS results and finally with RDM6 to determine the accuracy and efficiency of the developed program. Finally, the results are then presented in the work.

In conclusion, a summary of the most important results achieved will be presented, highlighting the contribution of this dissertation to mechanical construction, and addressing future development prospects that suggest expanding the scope of the study.

Chapter 1
Generalities and bibliographic synthesis

1.1 Introduction

When studying engineering problems, mathematical models based on differential equations are used to describe them, but finding accurate solutions to these equations is often complex or impossible in some cases, so we turn to other alternatives such as practical experiments, but they may be expensive and time-consuming. With the progress made in the field of computing, many numerical methods have been developed, the most prominent of which is the finite element method (FEM), which relies on dividing the engineering field into small parts, so that the solution within each element is approximated using simple mathematical functions, after which the elements are collected to obtain an approximate solution to the problem, which makes this method effective in analyzing complex structures and solving various engineering problems accurately and efficiently [1].

1.2 Finite element method

1.2.1 Historical evolution

The finite element method emerged from successive developments in engineering and mathematics, starting with the concept of structural symmetry substitution introduced by Hrenikoff in1941[2], which replaced continuous structures with discrete meshes to simplify solutions. McHenry (1943) [3] and Newmark (1947) [4] made further improvements. In the mid-1950s, Argyris (1955) [5] and Turner et al. (1956) [6] led the development of direct continuity elements, which became the cornerstone of numerical modeling. Mathematically, the method was based on the variational methods developed by Rayleigh (1870) [7] and Ritz (1909) [8], as well as the weighted residual methods founded by Gauss (1795) [9] and further developed by Galerkin (1915) [10] and Biezeno-Koch (1923) [11]. The finite difference method introduced by Richardson (1910) [12], Liebmann (1918) [13], and Southwell (1946) [14] also played a role in the development of numerical solutions. However, the major shift towards the present form of the FEM came with the introduction of partially continuous empirical functions by Courant (1943) [15], which divided continuous domains into discrete parts, later enhanced by the work of Prager-Synge (1947) [16] and Zienkiewicz (1964) [17]. Finally, the variational finite difference method of Varga (1962) [18] contributed to improving the accuracy of numerical calculations, making the FEM one of the most important tools used in the analysis of many engineering and physical problems today [19].

1.2.2 Definition

The finite element method (FEM) is a numerical analysis technique widely used to study and solve complex engineering and physical problems, but especially in structural mechanics. This method is based on dividing the domain into small domains known as elements and connected to each other at points known as nodes (Fig. 1.1). Through this division, we can search for the detailed formula for each element and transform differential equations into a set of algebraic equations that can be solved numerically. The finite element method is used in the analysis of stresses and deformations in materials and structures, as well as heat transfer, fluid dynamics, and electromagnetism [20].

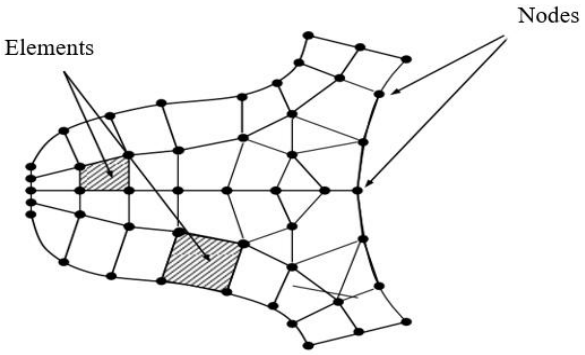


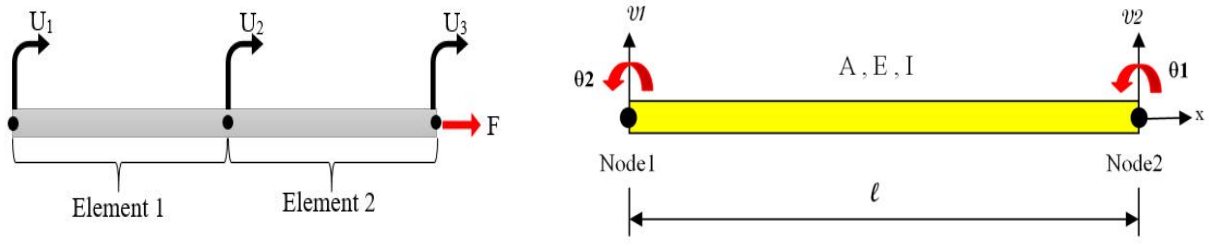
Fig. 1.1 Discretization of a structure into nodes and elements (tooth of a gear wheel) [21]

1.3 Classification of FEM Models

Finite element method (FEM) models are categorized based on their dimensionality, with determines the complexity of the problem and the type of elements used. The three primary classifications are:

1.3.1 One-Dimensional Finite Element Modeling (1D FEM)

1D elements are used when the structure is extended along a single axis like bar and beam.



a) b)

Fig. 1.2 Examples of 1D finite element: a) Axial deformation of a bar; b) Bending behavior of a beam

1.3.2 Two-Dimensional Finite Element Modeling (2D FEM)

Applies to thin structures like plates and shells, using triangular (T3, T6) and quadrilateral (Q4, Q8) elements with two translational DOF per node.

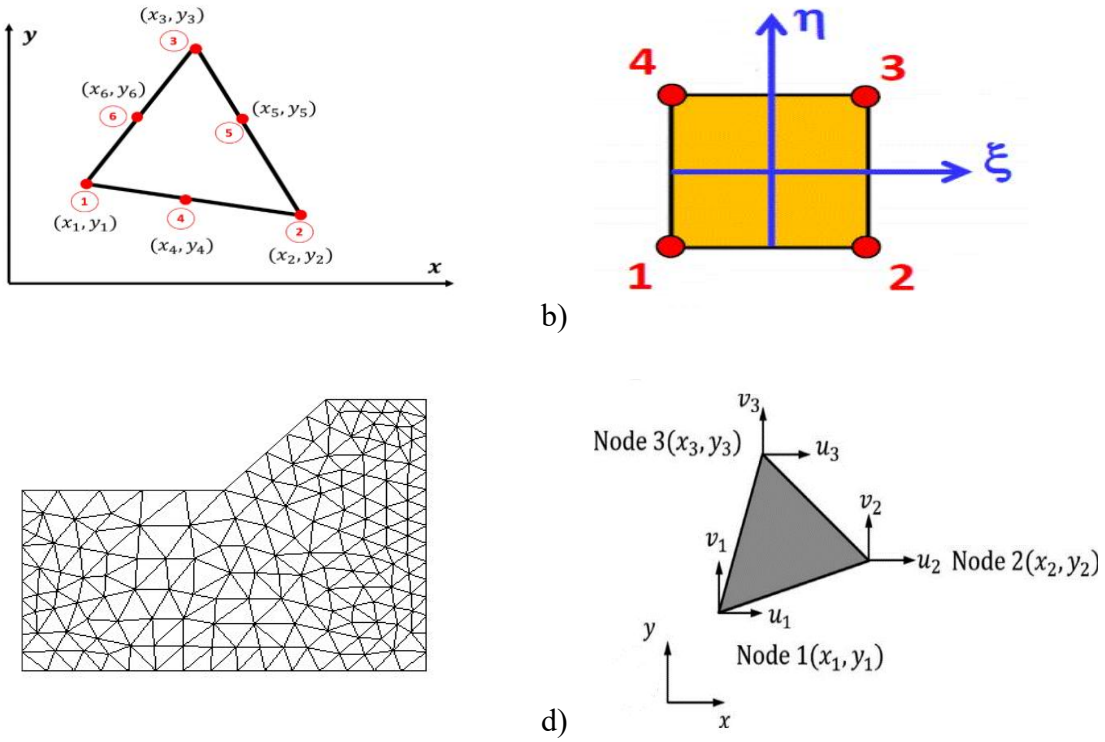


Fig. 1.3 Examples of 2D finite element: a) Six-node triangular element; b) Quadrilateral element with 4 nodes in the natural coordinate; c) Triangular mesh for a geometric domain; d) Simple three-node triangular element

1.3.3 Three-Dimensional Finite Element Modeling (3D FEM)

3D FEM is for solid structures, using tetrahedral (T4, T10) and hexahedral (H8, H20) elements, with three translational DOF per node.

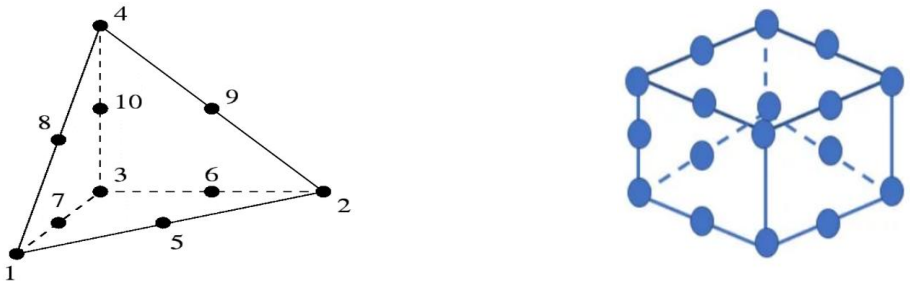


Fig. 1.4 Examples of 3D finite element: a) Tetrahedral element with 10 nodes; b) hexahedral element with 20 nodes

1.4 Steps of the finite element method

To solve any physical or engineering problem using the finite element method, we generally follow the following sequential steps:

- Dividing the domain: This is the first and important step in the analysis by dividing the domain into elements that are connected to each other by nodes in order to obtain more accurate results.
- Element stiffness matrix: Writing the element stiffness matrix for each element in the domain, so that it accurately represents the mechanical properties of the individual elements.
- Global stiffness matrix: All the individual element stiffness matrices are collected into one global matrix that expresses the behavior of the entire system and is used to solve the equation later.
- Applying boundary conditions: Applying the boundary conditions applied to the system, fixed supports and loads on the network nodes.
- Solving the global system: This is done by calculating the unknown variables, by solving a set of linear algebraic equations, using the following relationship [K]{U} = {F}.

1.5 Benefits of the Finite Element Method

The finite element method (FEM) is an effective numerical technique widely used in engineering, physics, and applied mathematics, enabling the accurate and efficient analysis of complex systems. Here is some key advantage for FEM:

- It can be used in various engineering disciplines, such as mechanical, civil and aeronautical engineering and applied in various fields, including structural analysis, fluid dynamics, heat transfer, electromagnetic, and biomechanics.
- The method can impose all kinds of boundary conditions, such as constrained displacements, distributed loads and point loads.
- FEM offers stable and reliable solutions for problems involving linear and non-linear equations.
- This method can analyze composite materials, transformable materials and even materials whose properties change with time and direction, and it provides numerical solutions that are close to the actual physical behavior of mechanical structures and systems.
- The results can be improved by reducing the mesh size or using higher resolution elements, without the need to reformulate the basic equations.
- It can accurately model and analyze complex and large structures, which is difficult using traditional analysis methods.

1.6 Plates

1.6.1 Definition

Plates are flat structural elements, but they are not like any ordinary thing; they are characterized by a small thickness compared to their other dimensions, which are length and width. They are designed to bear vertical loads on their surface, which leads to a complex and exciting distribution of stresses between bending and compression [22]. Understanding the mathematical modeling of plates is an essential part of structural analysis, especially when using methods such as the FEM, where plates are considered two-dimensional (2D) elements subject to plate bending theories.

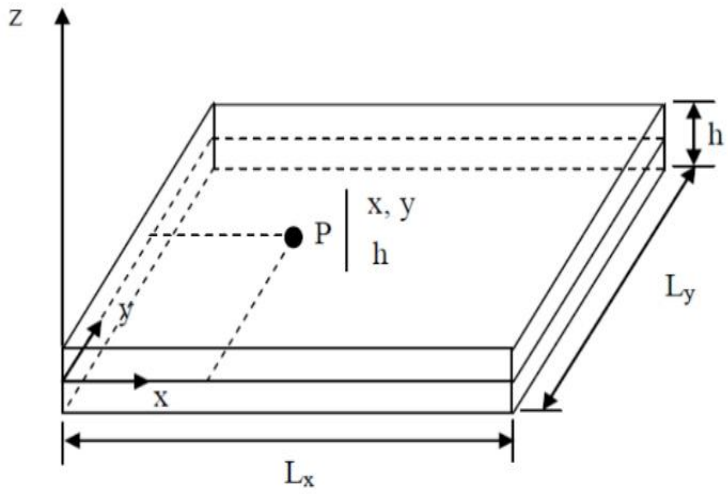


Fig. 1.5 Description of a plate

1.6.2 Types of plates

We can classify plates based on their thickness and mechanical behavior, the most important of which are:

- Thin plates: These plates are treated using the Kirchhoff-Love theory, which assumes that deformation in transverse shear can be neglected. We often find these plates used in structures where the thickness is small compared to the longitudinal dimension, such as the steel and concrete plates that we encounter daily [23].

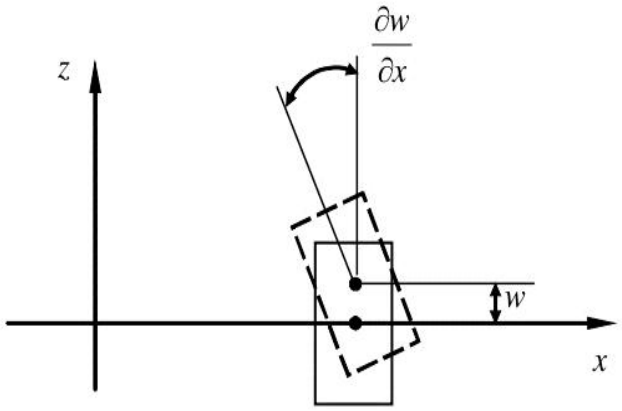


Fig. 1.6 Deflection and rotation of the plate under the influence of loads according to Kirchhoff plate theory.

- Thick plates: They are dealt with by the Mindlin-Reissner theory, which takes into account the effects related to transverse shear deformation. Therefore, they are most suitable for structures with moderate thicknesses relative to longitudinal dimensions, such as heavy mechanical parts that require strength and reliability [23].

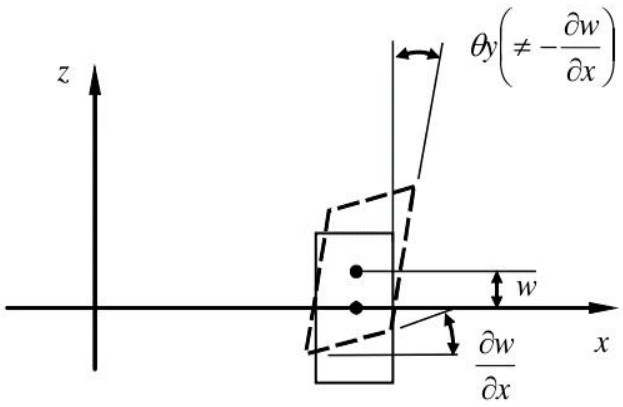


Fig. 1.7 Representation of displacement and rotation according to Mindlin plate theory

There are other types of plates that are classified based on their material properties, and they are:

- Isotropic plates: They are characterized by having the same mechanical properties in all directions, such as plates made of steel or aluminum, which ensure strength and reliability.
- Orthotropic plates: Here we are talking about materials that have different properties in different directions, and this is common in composite structures and laminated materials, which show complexity and beauty in their designs.

1.6.3 Field of application

Plates play a fundamental role in many sectors, the most important of which are structural and mechanical engineering, due to their efficiency in resisting bending and plane forces. Among the most common applications of these plates are:

- Civil engineering: They can be used in reinforced concrete slabs, bridge decks, and load-bearing walls, to ensure the structural integrity of buildings and facilities.
- Mechanical and aeronautical engineering: Design of aircraft wings, aircraft structures, and car body panels, where plates theories are used to achieve the best balance between weight and strength.
- Marine structures: Plates are used in ship structures and marine facilities to resist hydrodynamic and mechanical loads.
- They can also be used in industrial applications: in the design of pressure vessels, heat exchangers, and mechanical parts that depend on resisting high loads.

Finally, we conclude that by incorporating plates into the finite element analysis (FEA) framework, engineers can predict the distribution of stresses, deformations, and failure mechanisms, making them an integral part of modern mechanical and structural analysis.

1.7 Plane structures analysis using FEM

Plane structures are engineering systems studied in two dimensions, where one dimension is negligible compared to the others. Their analysis is classified as follows:

- Plane stress analysis: Used when one dimension is very thin, such as thin metal sheets or structural walls.
- Plane strain analysis: Used when the structure is very long in one dimension, such as concrete dams or thick walls [24].

FEM is widely used in plane structure analysis as it provides high accuracy in stress and deformation calculations, allowing for more efficient and responsive structural designs [25].

1.8 Conclusion

The finite element method (FEM) is a very important tool in structural analysis, as it provides accuracy and efficiency in modeling complex engineering problems. In this chapter, we discussed the most important concepts of this method, which are historical development, and basic principles. The role of the FEM in engineering continues to expand, making it essential for advanced simulation. In the next chapter, we will discuss the mathematical formulation of two dimensional (2D) triangular finite elements for a plane structure.

Chapter 2
Two dimensional triangular finite element

2.1 Introduction

Engineering and applied sciences rely heavily on modeling structures and physical phenomena to analyze and predict their behavior. Numerical methods especially the finite element method FEM play a fundamental role in this modeling as a powerful numerical tool used to analyze complex engineering problems. In two-dimensional analysis (2D FEM), triangular elements, whether three-node or six-node triangular elements, are widely used compared to quadrilateral elements due to their ability to adapt to irregular shapes and the ease of dividing the domain into a network of automatic triangles, making them a preferred choice for simulating mechanical behavior in various engineering applications. These elements rely on shape functions that allow the estimation of displacements and stresses within each element. The triangular element T3 provides a linear approximation of displacement, while the triangular element T6 provides higher accuracy using quadratic shape functions, which improves the results of numerical analysis [25-26].

This chapter aims to introduce the mathematical foundations for 2D triangular finite elements. This numerical formulation will help in understanding how triangulation finite elements can be applied in the analysis and design of engineering structures to ensure efficient and accurate prediction of their mechanical behavior.

2.2 Classification of two-dimensional finite element

Two-dimensional elements are classified according to their geometric shape and the number of nodes that define them. The most common elements are triangular elements and quadrilateral elements. Triangular elements are useful for dividing irregular domains due to their flexibility in representing complex geometric shapes, while quadrilateral elements are used in regular shapes and provide better accuracy for these shapes. However, they pose challenges in meshing complex geometric shapes, which makes triangular elements preferred and more used in cases where flexibility is in meshing. However, in this study, we will focus only on two-dimensional triangular elements, specifically on the three-node element (T3), due to their importance in analyzing the behavior of plane structures using the finite element method.

2.3 Types of 2D triangular elements

Triangular elements are further classified based on the number of nodes per element:

- **Three-node linear triangular element (T3)**
  - The simplest form of a two-dimensional finite triangular element, containing three nodes at the vertices of the triangle.

- The shape functions are linear, resulting in low accuracy compared to higher-order elements.
- Computationally simple.
- Used in irregularly distributed networks due to its flexibility in representing complex geometric shapes.

- **Six-node quadratic triangular element (T6)**
  - A triangular element containing 6 nodes: 3 at the vertices and 3 at the midpoints of the sides.
  - The shape functions are quadratic, providing high accuracy in the results.
  - More efficient in approximation compared to T3, but requires larger and more complex calculations.
  - Reduces the need for a large number of elements compared to T3.

The selection between T3 and T6 elements depends on the accuracy requirements and computational limitations of the analysis. If accuracy is not very important but the goal is to simplify the calculations, T3 is used. If the required accuracy is high, T6 is preferred because it provides a better approximation at a higher computational cost.

2.4 Mathematical formulation of 2D triangular finite element

The two-dimensional finite element formulation relies on the fundamental principles of continuum mechanics and numerical approximation, where the displacement field within each triangular element is expressed using shape functions, which specify how nodal displacements affect the motion of any point within the element. To develop a robust mathematical framework, these key steps must be considered:

1. The shape functions
2. Strain-displacement relationship
3. The element stiffness matrix
4. Global system assembly

2.5 Triangular element T3

A triangular element T3 is called a finite triangular element that has straight sides with a node at each end (Fig. 2.1). A T3 element has 6 degrees of freedom (DOF), with 2 DOF per node.

$$\{\delta\}^T = \langle u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3 \rangle \tag{2.1}$$

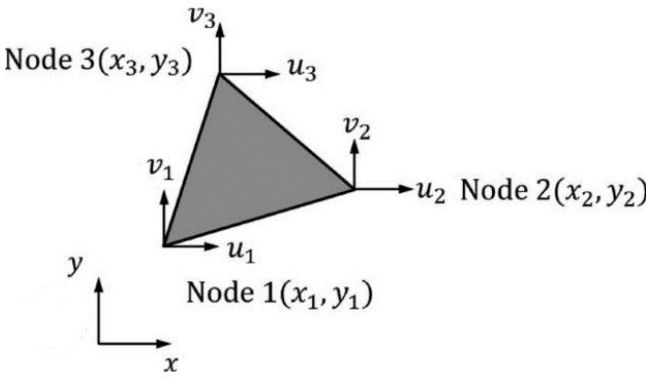


Fig. 2.1 First-order triangular element(T3) and its degrees of freedom

2.5.1 Linear shape function

The shape functions define how the displacement varies inside an element based on the nodal values (Fig. 2.2).

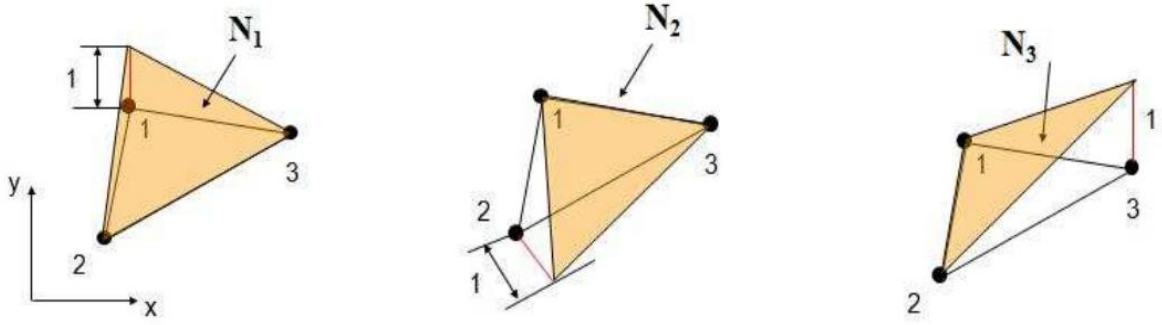


Fig. 2.2 Positions of shape functions

The chosen approximation function for this element is in the form:

$$\begin{cases} u(x,y) = a_1 + a_2 x + a_3 y \\ v(x,y) = a_4 + a_5 x + a_6 y \end{cases} \tag{2.2}$$

The function *u* and *v* describe the displacement of any point inside the triangle.

In matrix form, equation (2.2) becomes:

$$\begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{Bmatrix} \tag{2.3}$$

To determine the coefficients a_i ,we substitute the values of the coordinates $(x_i, y_i)$of nodes1,2 and 3:

$$\begin{cases} u(x_1,y_1) = u_1 = a_1 + a_2\,x_1 + a_3\,y_1 \\ v(x_1,y_1) = v_1 = a_4 + a_5\,x_1 + a_6\,y_1 \\ u(x_2,y_2) = u_2 = a_1 + a_2\,x_2 + a_3\,y_2 \\ v(x_2,y_2) = v_2 = a_4 + a_5\,x_2 + a_6\,y_2 \\ u(x_3,y_3) = u_3 = a_1 + a_2\,x_3 + a_3\,y_3 \\ v(x_3,y_3) = v_3 = a_4 + a_5\,x_3 + a_6\,y_3 \end{cases} \tag{2.4}$$

If we consider the equation of u :

$$\underbrace{\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}}_{\{u\}} = \underbrace{\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}}_{[X]} \underbrace{\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix}}_{\{a_i\}} \tag{2.5}$$

The values a_i are continuous by inversion of this system:

$$\{a\} = [X]^{-1}\{u\} \tag{2.6}$$

To invert $[X]$ we can use the cofactor method :

$$[X]^{-1} = \frac{1}{2A}\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \tag{2.7}$$

With:

$$2A = det\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{2.8}$$

2A: The area of triangular, calculated from the determinant of $[X]$

$$2A = x_1(y_2 - y_3) + x_2\,(y_3 - y_1) + x_3\,(y_1 - y_2) \tag{2.9}$$

With:

$$\begin{cases} \alpha_1 = x_2\,y_3 - x_3\,y_2 \;;\; \beta_1 = y_2 - y_3 \;\;;\; \gamma_1 = x_3 - x_2 \\ \alpha_2 = x_3 y_1 - x_1 y_3 \;\;;\; \beta_2 = y_3 - y_1 \;;\; \gamma_2 = x_1 - x_3 \\ \alpha_3 = x_1\,y_2 - x_2\,y_1 \;;\; \beta_3 = y_1 - y_2 \;\;;\; \gamma_3 = x_2 - x_1 \end{cases} \tag{2.10}$$

Once the α_i are determined, we can write the following relation:

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2A}\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{2.11}$$

We can extend this result:

$$\begin{Bmatrix} a_4 \\ a_5 \\ a_6 \end{Bmatrix} = \frac{1}{2A}\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}\begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \tag{2.12}$$

We can thus develop a general expression of the function $u(x,y)$ of displacement ($v(x,y)$ are the same approach) as a function of the variables x, y and α_i, β_i, γ_i and the nodal displacements u_1, u_2, and u_3

$$\{u\} = [1 \quad x \quad y]\begin{Bmatrix}a_1\\a_2\\a_3\end{Bmatrix} \tag{2.13}$$

If we substitute (2.11) into (2.13), we obtain:

$$\{u\} = \frac{1}{2A}[1 \quad x \quad y]\begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\\ \beta_1 & \beta_2 & \beta_3\\ \gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}\begin{Bmatrix}u_1\\u_2\\u_3\end{Bmatrix} \tag{2.14}$$

After development:

$$u(x,y) = \frac{1}{2A}[1 \quad x \quad y]\begin{bmatrix}\alpha_1u_1 + \alpha_2u_2 + \alpha_3u_3\\ \beta_1u_1 + \beta_2u_2 + \beta_3u_3\\ \gamma_1u_1 + \gamma_2u_2 + \gamma_3u_3\end{bmatrix} \tag{2.15}$$

$$u(x,y) = \frac{1}{2A}\left\{\underbrace{(\alpha_1 + \beta_1x + \gamma_1y)}_{N_1(x,\,y)}u_1 + \underbrace{(\alpha_2 + \beta_2x + \gamma_2y)}_{N_2(x,\,y)}u_2 + \underbrace{(\alpha_3 + \beta_3x + \gamma_3y)}_{N_3(x,\,y)}u_3\right\} \tag{2.16}$$

By analogy:

$$v(x,y) = \frac{1}{2A}\left\{\underbrace{(\alpha_1 + \beta_1x + \gamma_1y)}_{N_1(x,\,y)}v_1 + \underbrace{(\alpha_2 + \beta_2x + \gamma_2y)}_{N_2(x,\,y)}v_2 + \underbrace{(\alpha_3 + \beta_3x + \gamma_3y)}_{N_3(x,\,y)}v_3\right\} \tag{2.17}$$

We thus define the function of form:

$$\begin{cases}N_1(x,y) = \dfrac{1}{2A}(\alpha_1 + \beta_1x + \gamma_1y)\\ N_2(x,y) = \dfrac{1}{2A}(\alpha_2 + \beta_2x + \gamma_2y)\\ N_3(x,y) = \dfrac{1}{2A}(\alpha_3 + \beta_3x + \gamma_3y)\end{cases} \tag{2.18}$$

In equations (2.16) and (2.17) are recite as follows:

$$\begin{Bmatrix}u(x,y)\\v(x,y)\end{Bmatrix} = \begin{Bmatrix}N_1u_1 + N_2u_2 + N_3u_3\\ N_1v_1 + N_2v_2 + N_3v_3\end{Bmatrix} \tag{2.19}$$

Or:

$$\begin{Bmatrix}u(x,y)\\v(x,y)\end{Bmatrix} = \underbrace{\begin{bmatrix}N_1 & 0 & N_2 & 0 & N_3 & 0\\ 0 & N_1 & 0 & N_2 & 0 & N_3\end{bmatrix}}_{[N]}\begin{Bmatrix}u_1\\v_1\\u_2\\v_2\\u_3\\v_3\end{Bmatrix} \tag{2.20}$$

Finally, in matrix form:

$$\{u\} = [N]\{\delta\} \tag{2.21}$$

2.5.2 Plane stress and Plane strain

To simplify a 3D problem to a 2D problem, there are two approaches, the first is based on the plane stress hypothesis and the second on the plane strain hypothesis.

2.5.2.1 Plane stress state

This state exists in the case of thin structures when there are no forces parallel to the thickness. The components of the stress tensor $\sigma_{zz}, \sigma_{xz}, \sigma_{yz}$ are in this case very small and can be neglected. The stress field in the planes becomes x and y, as shown in Fig. 2.3.

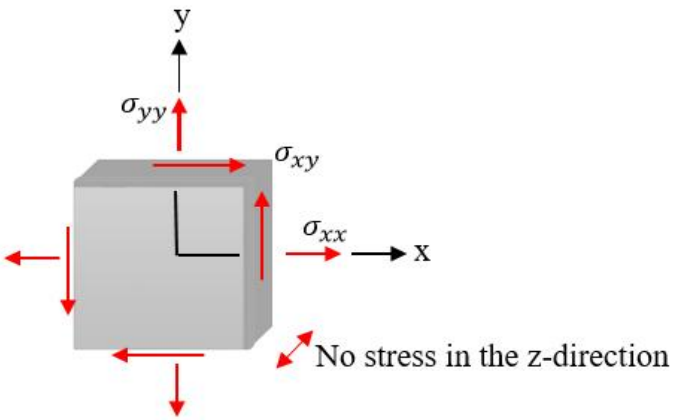


Fig. 2.3 Representation of the stress field in a plane stress state

The stress vector becomes:

$$\{\sigma\}^T = [\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{xy}] \tag{2.22}$$

For a stress state:

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \tag{2.23}$$

Hook's law allows us to write:

$$\varepsilon_{zz} = \frac{-\nu}{E}(\sigma_{xx} + \sigma_{xx}) \tag{2.24}$$

Where:

E: Young's modulus

ν*:* Poisson's ratio

And $$\varepsilon_{xz} = \varepsilon_{yz} = 0 \tag{2.25}$$

In the (x, y) plane the strain vector is given by the relation:

$$\{\varepsilon\}^T = [\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{xy}] \tag{2.26}$$

ε_{xx} and ε_{yy} are the normal strains in the x and y directions, and ε_{xy} is the shear strain corresponding to the shear stress σ_{xy} .

The total strain consists of two components: mechanical strain resulting from external forces, and thermal strain resulting from temperature changes.

Mechanical strain:

$$\{\varepsilon_m\}^T = [\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{xy}] \tag{2.27}$$

And thermal strain:

$$\{\varepsilon_t\}^T = [\alpha\delta T \quad \alpha\delta T \quad 0] \tag{2.28}$$

Where:

α: Coefficient of thermal expansion

δT: Change in temperature

Hook's law which links stresses to strain, is written as:

$$\{\sigma\} = [C]\{\varepsilon_m + \varepsilon_t\} \tag{2.29}$$

With:

$$[C] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{2.30}$$

2.5.2.2 Plane strain state

Solid of constant cross-section with a very large dimension compared to the other two (Fig. 2.4). If the solid is stressed in the (x, y) plane, the strain ε_{zz} is very small and can be neglected for the case of mechanical deformation.

$$\{\varepsilon\}^T = [\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{xy}] \tag{2.31}$$

And

$$\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0 \tag{2.32}$$

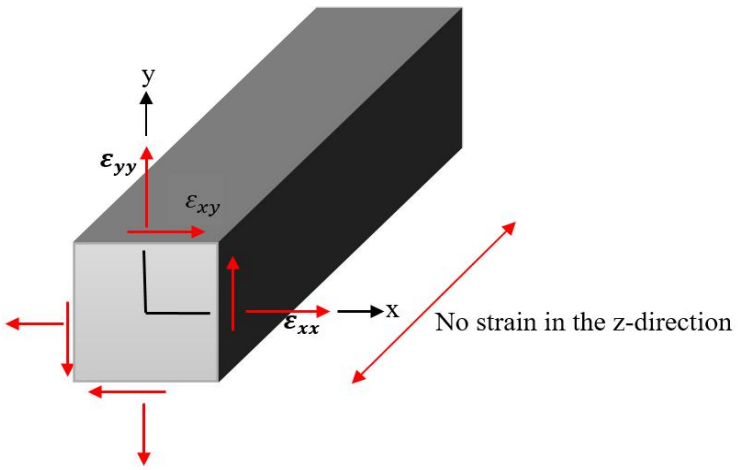


Fig. 2.4 Representation of the strain field in a plane strain state

Hook's law allows us to write:

$$\sigma_{zz} = \nu(\sigma_{xx} + \sigma_{yy}) \tag{2.33}$$

The matrix [D] becomes in the case:

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{2.34}$$

In addition:

$$2\varepsilon_{ij} = \gamma_{ij} \tag{2.35}$$

2.5.3 Strain-displacement relation

The strain-displacement relationship is a fundamental step in the FEM, as it allows the nodal displacements to be linked to the strains within the element via a mathematical matrix known as the [B] matrix, which is later used to calculate the stiffness and strain matrix. The strain vector associated with a two-dimensional (2D) state, is given as follows:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} \tag{2.36}$$

If we substitute the relation (2.2) into (2.36), we obtain:

$$\begin{cases} \varepsilon_{xx} = \frac{\partial u}{\partial x} = a_2 \\ \varepsilon_{yy} = \frac{\partial v}{\partial y} = a_6 \\ \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = a_3 + a_5 \end{cases} \tag{2.37}$$

From relation (2.37), we notice the strain in the element is constant. The element is thus named constant strain triangle (CST).

If we use the relation (2.19):

$$\begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = \begin{Bmatrix} N_1 u_1 + N_2 u_2 + N_3 u_3 \\ N_1 v_1 + N_2 v_2 + N_3 v_3 \end{Bmatrix}$$

$$\frac{\partial u}{\partial x} = \frac{\partial}{\partial x}(N_1(x,y)u_1 + N_2(x,y)u_2 + N_3(x,y)u_3) \tag{2.38}$$

If we use the relation (2.18), we obtain:

$$\frac{\partial N_1}{\partial x} = \frac{1}{2A}\frac{\partial}{\partial x}(\alpha_1 + \beta_1 x + \gamma_1 y) = \frac{\beta_1}{2A} \tag{2.39}$$

Likewise:

$$\frac{\partial N_2}{\partial x} = \frac{\beta_2}{2A} \quad and \quad \frac{\partial N_3}{\partial x} = \frac{\beta_3}{2A} \tag{2.40}$$

So:

$$\frac{\partial u}{\partial x} = \frac{1}{2A}(\beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3) \tag{2.41}$$

Similarly:

$$\frac{\partial v}{\partial y} = \frac{1}{2A}(\gamma_1 v_1 + \gamma_2 v_2 + \gamma_3 v_3) \tag{2.42}$$

And:

$$\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{1}{2A}(\gamma_1 u_1 + \beta_1 v_1 + \gamma_2 u_2 + \beta_2 v_2 + \gamma_3 u_3 + \beta_3 v_3) \tag{2.43}$$

If we substitute the relation (2.41), (2.42) and (2.43) in the relation (2.36) we obtain:

$$\{\varepsilon\} = \frac{1}{2A}\begin{bmatrix} \beta_1 & 0 & \beta_2 & 0 & \beta_3 & 0 \\ 0 & \gamma_1 & 0 & \gamma_2 & 0 & \gamma_3 \\ \gamma_1 & \beta_1 & \gamma_2 & \beta_2 & \gamma_3 & \beta_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \tag{2.44}$$

In matrix form:

$$\{\varepsilon\} = [B]\{u_n\} \tag{2.45}$$

With

$$[B] = \partial[N] \tag{2.46}$$

The matrix [B] called gradient matrix which links the strains to the displacements is independent of the variables x and y.

2.5.4 Stress-Strain Relationship

The stress-strain relationship is used to understand how a material responds to applied loads, and is a fundamental step in any structural analysis. In general, this relationship is written as follows:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = [C] \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \tag{2.47}$$

If we substitute the relation (2.45) into equation (2.47), we obtain:

$$\{\sigma\} = [C][B]\{u_n\} \tag{2.48}$$

2.5.5 Integration over a triangular element

The following formulas can be used to calculate integrals over the side or area of a triangular element [27]:

- Integrals over length

$$\int_l L_i^{\alpha} L_j^{\beta} dl = \frac{\alpha!\,\beta!}{(\alpha+\beta+1)!} l_{ij} \tag{2.49}$$

Where: dl represents a length between nodes i and j.

- Integrals over area

$$\int_A L_i^{\alpha} L_j^{\beta} L_k^{\gamma} dA = \frac{\alpha!\,\beta!\,\gamma!}{(\alpha+\beta+\gamma+2)!} 2A \tag{2.50}$$

Where: dA represents area of elements.

Since the shape functions of a triangular element $N_i(x, y)$ are defined in terms of the coordinates x and y, they can be replaced by the area coordinates $L_i(x, y)$ to perform the integration.

2.5.6 Elementary Stiffness Matrix

The stiffness matrix is derived in the finite element method based on the principle of the internal strain energy stored in the element when subjected to displacement. This energy represents to work done by the internal forces and is given by the equation:

$$U = \frac{1}{2}\int_V \varepsilon^T \sigma dV \tag{2.51}$$

Substituting equations (2.45) and (2.48) into equation (2.51), we find

$$U = \frac{1}{2}\{u_n\}^T \left(\int_A [B]^T [C][B].t.dA\right)\{u_n\} = \frac{1}{2}\{u_n\}^T [K]\{u_n\} \tag{2.52}$$

We conclude that the elemental stiffness matrix for a 2D element is given by the following relation:

$$[K]^e = \int_V [B]^T [C]\,[B] dV \tag{2.53}$$

With:

[B]: The strain-displacement matrix given by the relation (2.44)

[C]: the stress-strain matrix given by the relation (2.30)

Since [B] and[C] are constant inside a linear triangular element, the integral simplifies to:

$$[K]^e = [B]^T [C][B] A.t \tag{2.54}$$

t: Thickness of the element

A: Surface area of the element

The elementary force vector is given by the following relation, written in the plane:

$$\{f\}^e = \underbrace{\int_v [B]^T[C]\,[\varepsilon_T]dv}_{I} + \underbrace{\int_V [N]^T \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} dV}_{II} + \underbrace{\int_\Gamma [N]^T\{q\}\,d\Gamma}_{III} \tag{2.55}$$

I: Thermal forces

II: Body force

III: Distributed load

[N]: Is given by relation (2.22). the first term of equation (2.55) is easily evaluated because all its components are constant:

$$\int_v [B]^T[C]\,[\varepsilon_T]dv = [B]^T[C][\varepsilon_T]A.t \tag{2.56}$$

This term (I) exists if and only if it corresponds to a thermal load (expansion) due to a temperature variation.

The second term (II) in relation (2.55) becomes:

$$\int_V [N]^T \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} dV = \int_A [N]^T \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} t.dA = t \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} \int_A \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} dA \tag{2.57}$$

Individual integrals over the area including the shape functions are evaluated using the formulas for integration over a triangle. Applying relation (2.50), we find the following:

$$\int_{A_e} N_1 dA = \int_{A_e} {N_1}^1 {N_2}^0 {N_3}^0 dA = \frac{1!\,0!\,0!}{(1+0+0+2)!} 2A_e = \frac{1}{3}A_e \tag{2.58}$$

$$\int_{A_e} N_2 dA = \int_{A_e} {N_1}^0 {N_2}^1 {N_3}^0 dA = \frac{0!\,1!\,0!}{(0+1+0+2)!} 2A_e = \frac{1}{3}A_e \tag{2.59}$$

$$\int_{A_e} N_3 dA = \int_{A_e} {N_1}^0 {N_2}^0 {N_3}^1 dA = \frac{0!\,0!\,1!}{(0+0+1+2)!} 2A_e = \frac{1}{3}A_e \tag{2.60}$$

Substituting in relation (2.57), we get:

$$\int_V [N]^T \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} dV = \frac{A_e.t}{3} \begin{Bmatrix} f_x \\ f_y \\ f_x \\ f_y \\ f_x \\ f_y \end{Bmatrix} \tag{2.61}$$

Assume that the element shown in Fig. 2.5 is subjected to a uniformly distribution load of magnitude "q", applied perpendicular to side 2-3 and at an angle θ to the overall x-axis. Therefore, the components of this load are written as follows:

$$\{q\} = \begin{Bmatrix} -qcos\theta \\ -qsin\theta \end{Bmatrix} \tag{2.62}$$

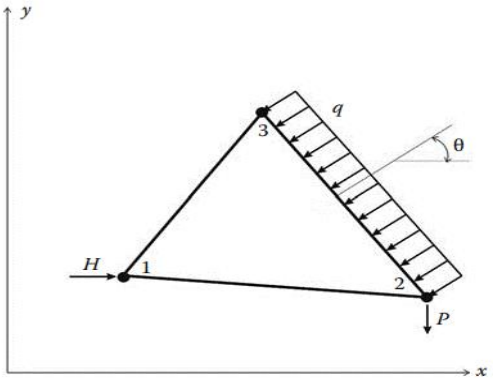


Fig. 2.5 Uniformly distributed load in a triangular element [27]

So the third term (III) of equation (2.55) is evaluated as:

$$\int_{\Gamma} [N]^T \{q\}\, d\Gamma = \int_{L_{2-3}} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \begin{Bmatrix} -qcos\theta \\ -qsin\theta \end{Bmatrix} t.\, dL = t \int_{L_{2-3}} \begin{Bmatrix} 0 \\ 0 \\ -N_2 qcos\theta \\ -N_2 qsin\theta \\ -N_3 qcos\theta \\ -N_3 qsin\theta \end{Bmatrix} dl \tag{2.63}$$

The integrals along a triangle are evaluated using the integration formula over a side of triangle given in equations (2.49). Applying this formula, equation (2.63) becomes:

$$\int_{\Gamma} [N]^T \{q\}\, d\Gamma = t \begin{Bmatrix} 0 \\ 0 \\ -qcos\theta L_{2-3}/2 \\ -qsin\theta L_{2-3}/2 \\ -qcos\theta L_{2-3}/2 \\ -qsin\theta L_{2-3}/2 \end{Bmatrix} \tag{2.64}$$

2.6 Example calculation of triangular element T3

This example aims to study the behavior of a two-dimensional plane plate subjected to concentrated forces (Fig. 2.6), using two-dimensional finite elements of the triangular type with three-node (T3). The analysis is based on the use of the plane stress assumption, while neglecting body force compared to the applied concentrated forces. We will analyze the system by determining the unknown nodal displacements and the reaction at support, the element stresses and strains.

Given: E= 210×10^3 MPa, ν= 0.25, t=10 mm, P=1 KN

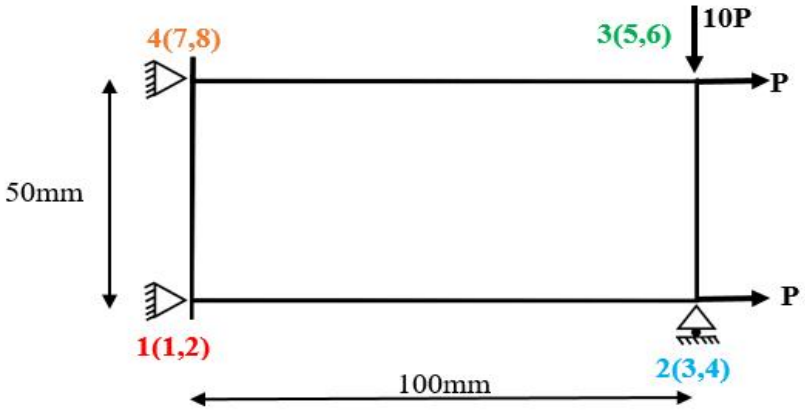


Fig. 2.6 Example of the calculation of a triangular element T3

The steps to follow to solve this problem are:

1. Properties of the structure: The structural characteristics will be summarized in the following tables Table 2.1 and Table 2.2.

Table 2.1 Nodal coordinates of the structure

Node	*x (mm)*	*y (mm)*
1	*0*	*0*
2	*100*	*0*
3	*100*	*50*
4	*0*	*50*

Table 2.2 Element connectivity

Elements Number	*Nodes*		
	1	*2*	*3*
1	*1*	*2*	*3*
2	*1*	*3*	*4*

2. Discretize of the structure: The plate is divided into two triangular finite elements with 3 nodes, where each node has 2 degrees of freedom, 1(1,2), 2(3,4), 3(5,6) and 4(7,8), as well as two displacements per node as shown in Fig. 2.7.

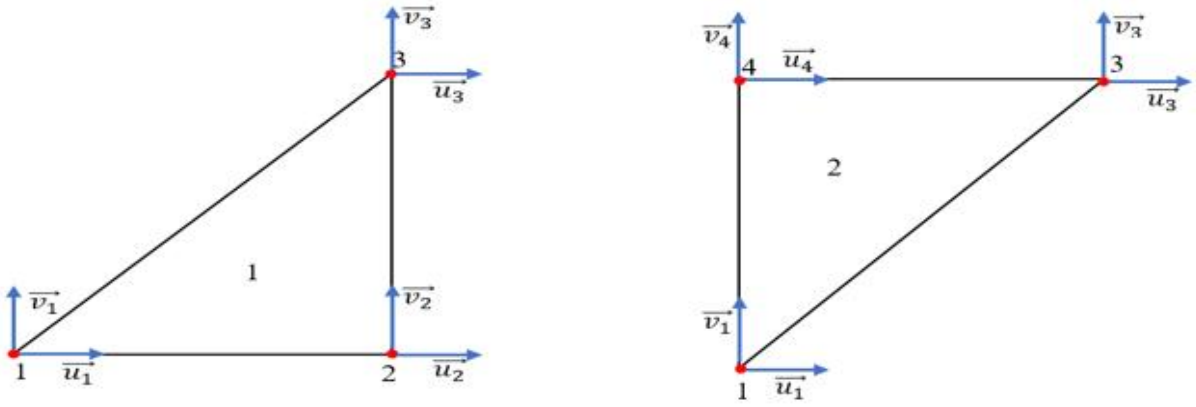


Fig. 2.7 Discretization of the triangular elements

3. Shape function: The shape function N_1, N_2 and N_3 are given by the following relations:

$$N_1(x,y) = \frac{1}{2A}(\alpha_1 + \beta_1 x + \gamma_1 y)$$
$$N_2(x,y) = \frac{1}{2A}(\alpha_2 + \beta_2 x + \gamma_2 y)$$
$$N_3(x,y) = \frac{1}{2A}(\alpha_3 + \beta_3 x + \gamma_3 y)$$

- **Element 1:** to find the shape function we do the following calculations:

- Calculate the area of triangle:

$$2A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 100 & 0 \\ 1 & 100 & 50 \end{vmatrix} = 5000\ mm^2$$

$$A = \frac{5000}{2} = 2500\ mm^2$$

- The coefficients:

$\alpha_1 = x_2\, y_3 - x_3\, y_2 = 5000$ $\quad$ $\beta_1 = y_2 - y_3 = -50$ $\quad$ $\gamma_1 = x_3 - x_2 = 0$

$\alpha_2 = x_3 y_1 - x_1 y_3 = 0$ $\quad$ $\beta_2 = y_3 - y_1 = 50$ $\quad$ $\gamma_2 = x_1 - x_3 = -100$

$\alpha_3 = x_1\, y_2 - x_2\, y_1 = 0$ $\quad$ $\beta_3 = y_1 - y_2 = 0$ $\quad$ $\gamma_3 = x_2 - x_1 = 100$

So:

$$N_1(x,y) = \frac{1}{2A}(\alpha_1 + \beta_1 x + \gamma_1 y) = \frac{1}{5000}(5000 - 50x) = 1 - \frac{1}{100}x$$

$$N_2(x,y) = \frac{1}{2A}(\alpha_2 + \beta_2 x + \gamma_2 y) = \frac{1}{5000}(50x - 100y) = \frac{1}{100}x - \frac{1}{50}y$$

$$N_3(x,y) = \frac{1}{2A}(\alpha_3 + \beta_3 x + \gamma_3 y) = \frac{1}{5000}(100y) = \frac{1}{50}y$$

- **Element 2:**

- Area of triangle:

$$2A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 100 & 50 \\ 1 & 0 & 50 \end{vmatrix} = 5000\ mm^2$$

$$A = \frac{5000}{2} = 2500\ mm^2$$

- The coefficients: The coefficients are obtained as follows:

$\alpha_1 = x_2\, y_3 - x_3\, y_2 = 5000$ $\quad$ $\beta_1 = y_2 - y_3 = 0$ $\quad$ $\gamma_1 = x_3 - x_2 = -100$

$\alpha_2 = x_3 y_1 - x_1 y_3 = 0$ $\quad$ $\beta_2 = y_3 - y_1 = 50$ $\quad$ $\gamma_2 = x_1 - x_3 = 0$

$\alpha_3 = x_1\, y_2 - x_2\, y_1 = 0$ $\quad$ $\beta_3 = y_1 - y_2 = -50$ $\quad$ $\gamma_3 = x_2 - x_1 = 100$

So:

$$N_1(x,y) = \frac{1}{2A}(\alpha_1 + \beta_1 x + \gamma_1 y) = \frac{1}{5000}(5000 - 100y) = 1 - \frac{1}{50}y$$

$$N_2(x,y) = \frac{1}{2A}(\alpha_2 + \beta_2 x + \gamma_2 y) = \frac{1}{5000}(50x) = \frac{1}{100}x$$

$$N_3(x,y) = \frac{1}{2A}(\alpha_3 + \beta_3 x + \gamma_3 y) = \frac{1}{5000}(-50x + 100y) = -\frac{1}{100}x + \frac{1}{50}y$$

4. Element stiffness matrices: Calculate the stiffness matrix for each triangular element (T3) using the finite element method (FEM).

Element 1: Between nodes 1(1,2), 2(3,4) and 3(5,6), with strain-displacement matrix [B], stress-strain matrix [C], area A, and thickness t.

$$[K]^{(1)} = \left[B^{(1)}\right]^T [C][B]^{(1)} A.t$$

$$[B]^{(1)} = \frac{1}{2A}\begin{bmatrix} \beta_1 & 0 & \beta_2 & 0 & \beta_3 & 0 \\ 0 & \gamma_1 & 0 & \gamma_2 & 0 & \gamma_3 \\ \gamma_1 & \beta_1 & \gamma_2 & \beta_2 & \gamma_3 & \beta_3 \end{bmatrix} = \frac{1}{100}\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \\ 0 & -1 & -2 & 1 & 2 & 0 \end{bmatrix}$$

For plane stress:

$$[C] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{210 \times 10^3}{1 - 0.25^2}\begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & \frac{0.75}{2} \end{bmatrix} = 56 \times 10^3 \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}$$

So:

$$[K]^{(1)} = \frac{1}{100} \times \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & -2 \\ 0 & -2 & 1 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix} \times 56 \times 10^3 \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \times \frac{1}{100}\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \\ 0 & -1 & -2 & 1 & 2 & 0 \end{bmatrix} \times 2500 \times 10$$

And from it we find:

$$[K]^{(1)} = 10^4 \times \begin{bmatrix} 56 & 0 & -56 & 28 & 0 & -28 \\ 0 & 21 & 42 & -21 & -42 & 0 \\ -56 & 42 & 140 & -70 & -84 & 28 \\ 28 & -21 & -70 & 245 & 42 & -224 \\ 0 & -42 & -84 & 42 & 84 & 0 \\ -28 & 0 & 28 & -224 & 0 & 224 \end{bmatrix}$$

Element 2: Between nodes 1(1,2), 3(5,6) and 4(7,8), with strain-displacement matrix [B], stress-strain matrix [C], area A, and thickness t.

$$[K]^{(2)} = \left[B^{(2)}\right]^T [C][B]^{(2)} A.t$$

$$[B]^{(2)} = \frac{1}{2A}\begin{bmatrix} \beta_1 & 0 & \beta_2 & 0 & \beta_3 & 0 \\ 0 & \gamma_1 & 0 & \gamma_2 & 0 & \gamma_3 \\ \gamma_1 & \beta_1 & \gamma_2 & \beta_2 & \gamma_3 & \beta_3 \end{bmatrix} = \frac{1}{100}\begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & 0 & 0 & 1 & 2 & -1 \end{bmatrix}$$

Substituting into the equation, we find:

$$[K]^{(2)} = \frac{1}{100}\begin{bmatrix} 0 & 0 & -2 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 2 \\ 0 & 2 & -1 \end{bmatrix} 56 \times 10^3 \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \frac{1}{100}\begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & 0 & 0 & 1 & 2 & -1 \end{bmatrix} 2500 \times 10$$

$$[K]^{(2)} = 10^4 \times \begin{bmatrix} 84 & 0 & 0 & -42 & -84 & 42 \\ 0 & 224 & -28 & 0 & 28 & -224 \\ 0 & -28 & 84 & 0 & -56 & 28 \\ -42 & 0 & 0 & 21 & 42 & -21 \\ -84 & 28 & -56 & 42 & 140 & -70 \\ 42 & -224 & 28 & -21 & -70 & 245 \end{bmatrix}$$

5. Assemble the global stiffness matrix: After calculating the stiffness matrix for each element, we combine them into a single matrix known as the global stiffness matrix (GSM).

$$Dim\ (GSM) = (nddl_{/n} * nn) \times (nddl_{/n} * nn) = (2 \times 4) \times (2 \times 4) = 8 \times 8$$

After assembly and calculating, we find the final expression for the global stiffness matrix as follows:

$$[K] = 10^4 \begin{bmatrix} 140 & 0 & -56 & 28 & 0 & -70 & -84 & 42 \\ 0 & 245 & 42 & -21 & -70 & 0 & 28 & -224 \\ -56 & 42 & 140 & -70 & -84 & 28 & 0 & 0 \\ 28 & -21 & -70 & 245 & 42 & -224 & 0 & 0 \\ 0 & -70 & -84 & 42 & 140 & 0 & -56 & 28 \\ -70 & 0 & 28 & -224 & 0 & 245 & 42 & -21 \\ -84 & 28 & 0 & 0 & -56 & 42 & 140 & -70 \\ 42 & -224 & 0 & 0 & 28 & -21 & -70 & 245 \end{bmatrix}$$

6. Force vector in each element: In this study, it was assumed that the concentrated force (P) is applied to node 2 in x-direction, and also (P,10P) at node 3. As for the rest of the nodes, they are free.

- **Element 1:** between nodes 1, 2 and 3, concentrated force at nodes 2 and 3, $F_{2x} = P$, $F_{3x} = \frac{P}{2}$ and $F_{3y} = -5P$

$$\{f\}^{(1)} = \begin{Bmatrix} 0 \\ 0 \\ P \\ 0 \\ \frac{P}{2} \\ -5P \end{Bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

- **Element 2:** between nodes 1, 3 and 4, concentrated force at nodes 3, $F_{3x} = \frac{P}{2}$ and $F_{3y} = -5P$

$$\{f\}^{(2)} = \begin{Bmatrix} 0 \\ 0 \\ \frac{P}{2} \\ -5P \\ 0 \\ 0 \end{Bmatrix} \begin{matrix} 1 \\ 2 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix}$$

7. Global force vector: We collect all the obtained force vectors to obtain the global force vector, taking into account the degrees of freedom.

$$\{F\} = \begin{Bmatrix} 0 \\ 0 \\ P \\ 0 \\ P \\ -10P \\ 0 \\ 0 \end{Bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix}$$

8. Application of boundary conditions: The boundary conditions applied in the study are shown in **Table 2.3**.

Table 2.3 Boundary conditions applied to the case study

Nodes	*Boundary conditions*	*Displacements*	*Forces*
Node 1	*Double supports*	$u_1=0$ *and* $v_1=0$	$R_{1x}=?$ *and* $R_{1y}=?$
Node 2	*Simple supports*	$u_2=?$ *and* $v_2=0$	$F_{2x}=P$ *and* $R_{2y}=?$
Node 3	*Concentrated forces*	$u_3=?$ *and* $v_3=?$	$F_{3x}=P$ *and* $F_{3y}= -10P$
Node 4	*Double supports*	$u_4=0$ *and* $v_4=0$	$R_{4x}=?$ *and* $R_{4y}=?$

9. Formation of the global system: solving the global system to determine the unknown variables, using the relationship [K]{U} ={F}

$$10^4\begin{bmatrix}140 & 0 & -56 & 28 & 0 & -70 & -84 & 42\\ 0 & 245 & 42 & -21 & -70 & 0 & 28 & -224\\ -56 & 42 & 140 & -70 & -84 & 28 & 0 & 0\\ 28 & -21 & -70 & 245 & 42 & -224 & 0 & 0\\ 0 & -70 & -84 & 42 & 140 & 0 & -56 & 28\\ -70 & 0 & 28 & -224 & 0 & 245 & 42 & -21\\ -84 & 28 & 0 & 0 & -56 & 42 & 140 & -70\\ 42 & -224 & 0 & 0 & 28 & -21 & -70 & 245\end{bmatrix}\cdot\begin{Bmatrix}0\\0\\u_2\\0\\u_3\\v_3\\0\\0\end{Bmatrix}=\begin{Bmatrix}0\\0\\P=1000\\0\\P=1000\\P=-10000\\0\\0\end{Bmatrix}$$

10. Solution: To calculate the unknown displacements we eliminate columns and rows with zero displacement 1, 2, 4, 7, and 8, the system becomes as follows:

$$10^4\begin{bmatrix}140 & -84 & 28\\ -84 & 140 & 0\\ 28 & 0 & 245\end{bmatrix}\begin{Bmatrix}u_2\\u_3\\v_3\end{Bmatrix}=\begin{Bmatrix}1000\\1000\\-10000\end{Bmatrix}$$

We use the Gaussian elimination method to solve a system of equations with three unknowns:

- **Step 1:** Simplify the system by dividing both sides by 10^4 to make the calculations easier.

$$\begin{bmatrix}140 & -84 & 28\\ -84 & 140 & 0\\ 28 & 0 & 245\end{bmatrix}\begin{Bmatrix}u_2\\u_3\\v_3\end{Bmatrix}=\begin{Bmatrix}0.1\\0.1\\-1\end{Bmatrix}$$

- **Step 2:** Create the augmented matrix

$$\left(\begin{array}{ccc|c}140 & -84 & 28 & 0.1\\ -84 & 140 & 0 & 0.1\\ 28 & 0 & 245 & -1\end{array}\right)$$

- **Step 3:** Make the numbers under the diagonal of the matrix equal to zero

- Eliminate element (2,1): We calculate the coefficient

$$c_{21} = \frac{-84}{140} = -0.6$$

We use the following relationship to find the new row:

$$r_2 = r_2 + 0.6r_1 = \begin{cases} -84 + 0.6 \times 140 = 0 \\ 140 + 0.6 \times (-84) = 89.6 \\ 0 + 0.6 \times 28 = 16.8 \\ 0.1 + 0.6 \times 0.1 = 0.16 \end{cases}$$

New row: $[0 \quad 89.6 \quad 16.8 \quad 0.16]$

- Eliminate element (3,1): We calculate the coefficient

$$c_{31} = \frac{28}{140} = 0.2$$

We use the following relationship to find the new row:

$$r_3 = r_3 - 0.2r_1 = \begin{cases} 28 - 0.2 \times 140 = 0 \\ 0 - 0.2 \times (-84) = 16.8 \\ 245 - 0.2 \times 28 = 239.4 \\ -1 - 0.2 \times 0{,}1 = -1.02 \end{cases}$$

New row: $[0 \quad 16.8 \quad 239.4 \quad -1.02]$

- Elimination element (3,2): We calculate the coefficient

$$c_{32} = \frac{16.8}{89.6} = 0.1875$$

We use the following relationship to find the new row:

$$r_3 = r_3 - 0.1875.\, r_2 = \begin{cases} 16.8 - 0.1875 \times 89.6 = 0 \\ 239.4 - 0.1875 \times 16.8 = 236.25 \\ -1.02 - 0.1875 \times 0.16 = -1.05 \end{cases}$$

New row: $[0 \quad 0 \quad 236.25 \quad -1.05]$

- **Step 4:** Put the new rows in place and the system becomes:

$$\begin{bmatrix} 140 & -84 & 28 \\ 0 & 89.6 & 16.8 \\ 0 & 0 & 236.25 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} 0.1 \\ 0.16 \\ -1.05 \end{Bmatrix}$$

Finally, we find:

$$v_3 = \frac{-1.05}{236.25}$$

$$u_3 = \frac{0.16 - 16.8v_3}{89.6} = \frac{55.44}{21168}$$

$$u_2 = \frac{0.1 + 84u_3 - 28v_3}{140} = \frac{7}{2205}$$

The results are as follows:

$$u_2 = 3.175 \times 10^{-3}\ mm$$
$$u_3 = 2.619 \times 10^{-3}\ mm$$
$$v_3 = -4.444 \times 10^{-3}\ mm$$

To calculate the reactions at the supports, we consider equations 1, 2, 4, 7 and 8, and replace the displacement u_2, u_3 and v_3 with their previously calculated values:

$$R_{1x} = 10^4 \times [(-56)u_2 + (-70)v_3] = 1333\ N$$
$$R_{1y} = 10^4 \times [42u_2 + (-70)u_3] = -500\ N$$
$$R_{2y} = 10^4 \times [(-70)u_2 + 42u_3 + (-224)v_3] = 8833\ N$$
$$R_{4x} = 10^4 \times [(-56)u_3 + 42v_3] = -3333\ N$$
$$R_{4y} = 10^6 \times [28u_3 + (-21)v_3] = 1667\ N$$

11. Calculating strains element: The strains in element 1 and 2 are calculated from the relation:

$$\{\varepsilon\}^e = [B]^e\{u_n\}^e$$

- For element 1: The nodal displacement vector given by:

$$\{u\}^{(1)} = \langle 0 \quad 0 \quad u_2 \quad 0 \quad u_3 \quad v_3 \rangle$$

So:

$$\{\varepsilon\}^{(1)} = [B]^{(1)}\{u_n\}^{(1)} = \frac{1}{100}\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \\ 0 & -1 & -2 & 1 & 2 & 0 \end{bmatrix}\begin{Bmatrix} 0 \\ 0 \\ u_2 \\ 0 \\ u_3 \\ v_3 \end{Bmatrix}$$

After solving, we obtained the following results:

$$\{\varepsilon\}^{(1)} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} 3.175 \times 10^{-5} \\ -8.889 \times 10^{-5} \\ -1.111 \times 10^{-5} \end{Bmatrix}$$

- For element 2: The nodal displacement vector given by:

$$\{u\}^{(2)} = \langle 0 \quad 0 \quad u_3 \quad v_3 \quad 0 \quad 0 \rangle$$

Therefore:

$$\{\varepsilon\}^{(2)} = [B]^{(2)}\{u_n\}^{(2)} = \frac{1}{100}\begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & 0 & 0 & 1 & 2 & -1 \end{bmatrix}\begin{Bmatrix} 0 \\ 0 \\ u_3 \\ v_3 \\ 0 \\ 0 \end{Bmatrix}$$

After calculation, we find:

$$\{\varepsilon\}^{(2)} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} 2.619 \times 10^{-5} \\ 0 \\ -4.444 \times 10^{-5} \end{Bmatrix}$$

12. Element stresses: To calculate the element stress for each element, we use the following relationship between stress and strain: $\{\sigma\}^e = [C]\{\varepsilon\}^e$

When we substitute it into the equation for element 1, we find as follows:

$$\{\sigma\}^{(1)} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = 56 \times 10^3 \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}\begin{Bmatrix} 3.175 \times 10^{-5} \\ -8.889 \times 10^{-5} \\ -1.111 \times 10^{-5} \end{Bmatrix} = \begin{Bmatrix} 2.133 \\ -18.133 \\ -0.933 \end{Bmatrix} MPa$$

And from the element 2 we get:

$$\{\sigma\}^{(2)} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = 56 \times 10^3 \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 1{,}5 \end{bmatrix}\begin{Bmatrix} 2.619 \times 10^{-5} \\ 0 \\ -4.444 \times 10^{-5} \end{Bmatrix} = \begin{Bmatrix} 5.866 \\ 1.466 \\ -3.733 \end{Bmatrix} MPa$$

After calculating the nodal displacements and element strains, as well as the element stresses, we will extract their results in Table 2.4 and Table 2.5

Table 2.4 Results obtained from calculating the displacements in the system

Nodes	*Displacements u (mm)*	*Displacements v (mm)*	*Forces /x-axis*	*Forces /y-axis*
Node 1	$u_1=0$	$v_1=0$	$R_1=1333\ N$	$R_1=-500\ N$
Node 2	$u_2=3.175\times10^{-3}$	$v_2=0$	$R_2=1000\ N$	$R_2=8833\ N$
Node 3	$u_3=2.619\times10^{-3}$	$v_3=-4.444\times10^{-3}$	$R_3=1000\ N$	$R_3=-10000\ N$
Node 4	$u_4=0$	$v_4=0$	$R_4=-3333\ N$	$R_4=1667\ N$

Table 2.5 The basic results that relate to each element of the case studied

Elements	*Element strains*	*Element stresses (MPa)*
Element 1	$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} 3.175 \times 10^{-5} \\ -8.889 \times 10^{-5} \\ -1.111 \times 10^{-5} \end{Bmatrix}$	$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \begin{Bmatrix} 2.133 \\ -18.133 \\ -0.933 \end{Bmatrix}$
Element 2	$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} 2.619 \times 10^{-5} \\ 0 \\ -4.444 \times 10^{-5} \end{Bmatrix}$	$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \begin{Bmatrix} 5.866 \\ 1.466 \\ -3.733 \end{Bmatrix}$

2.7 Conclusion

In this chapter, we discussed the mathematical foundations of the two-dimensional finite element method using triangular elements, with emphasis on the derivation of shape functions, stiffness matrix, and stress-strain relationships. The plane stress and plan strain cases are analyzed and the equations associated with them are derived. Finally, a numerical example is presented to illustrate the application of the methodology. This analysis lays the foundation for the next chapter, where the model will be implemented programmatically in MATLAB and its accuracy verified by comparison with the results of commercial software ABAQUS and educational software RDM6.

Chapter 3
Programming and validation

3.1 Programming languages

3.1.1 Overview of programing languages

Programming languages play a fundamental role in controlling systems and machines as a set of instructions that include symbols and rules, enabling programmers to give commands to computers for execution. Although the scope of programming languages is very broad, they can be classified into high-level languages and low-level languages. High-level languages resemble human language, are easier to use, and are abstract, allowing code to be written in an organized and readable manner, such as MATLAB, Java, Python, and C++. Low-level languages, such as machine language and assembly language, are closer to the native computer language, highly efficient and complex, making them difficult to understand and read [28].



Fig. 3.1 The most popular programming languages in various fields

3.1.2 How programming languages evolved

Programming languages have evolved significantly since their inception. Konrad Zuse introduced the first high-level programming language in 1946, called Plankalkül, but it was never used in practice. In 1957, John Backus led a team at the International Business Machines (IBM) to develop Fortran, which became the first high-level programming language widely used in scientific computing. C followed between 1969 and 1973, developed by Dennis Ritchie and the basis of modern operating systems. In the 1980s, C++, developed by Bjarne Stroustrup, emerged, followed by Java in 1995, providing a platform-independent operating system. In recent years, Python (1991) and Julia (2012) have gained significant popularity in numerical analysis [29]. Despite the emergence of several languages, MATLAB, developed in the 1970s, remains a key tool in engineering applications, particularly in finite element analysis (FEA).

3.1.3 Development tools

Development tools are essential components of software development, testing, and implementation. These tools vary depending on the needs of developers. Text editors such as Notepad++ and Visual Studio Code provide a simple interface for writing code. Integrated Development Environments (IDEs) such as Visual Studio and Eclipse provide advanced features such as debugging, project management, and converting code into executable programs [30]. Compilers such as C, C++, Java and many others convert high-level programming language code into executable code. Interpreters like Python, Ruby and MATLAB provide a mechanism for executing code directly without the need for prior compilation. Finally, version control systems such as SVN and Git track code changes and ensure good coordination among developers [31].

3.1.4 Basic concepts in programming

Concepts in programming are the foundation for building any program, as they contain a set of elements represented by variables, function, conditional statement, data types and looping. Variables are memory locations that store values such as numbers and strings. Functions are independents units of code that perform specific tasks and can be called at any time within the program to reduce repetition. Conditional statement such as if and else allow the program to make decisions based on specific conditions. Looping is performed through commands such as for and while to execute code multiple times, allowing the programmer to perform repeated operations on a set of data or repeat specific tasks [32]. Finally, data types help classifies what information a variable can hold and what can be done with it, such as booleans, strings, integer and float [29].

3.1.5 Development of programming paradigms

Programming languages have undergone significant developments to meet the needs of different computational paradigms. The procedural paradigm began with the sequential execution of instructions using variables to store data. Despite its widespread popularity, it faces numerous challenges, such as limited support for parallel computing. Other paradigms have emerged to address these challenges, including the functional paradigm based on mathematical functions, and object-oriented paradigms (OOP), which allows developers to write flexible and reusable programs for objects to interact with each other via messages [33].

3.1.6 Programming language used in numerical and engineering analysis

Programming languages are essential tools for building software for numerical analysis, and they vary in performance and library support. Fortran is widely used in advanced engineering programs such as ABAQUS and ANSYS due to its high performance and efficiency in numerical calculations. C and C++ have high memory management performance and support for

object-oriented programming, making them suitable for performing large calculations. MATLAB provides an integrated environment that facilitates the implementation of mathematical operations, differential equations, and numerical analysis, and provides powerful tools for graphing and visualizing results, making it ideal for academic research and development. However, in recent years, Python has become an increasingly popular choice thanks to its scientific libraries such as NumPy and FEniCS, enhancing its use in modeling, numerical analysis, and simulation [34-35].

3.2 Chosen programming language (MATLAB)

3.2.1 Definition and its historical background

MATLAB (short for MATrix LABoratory) is a technical computing environment and high-level programming languages, specifically developed for efficient and rapid engineering and scientific calculations.

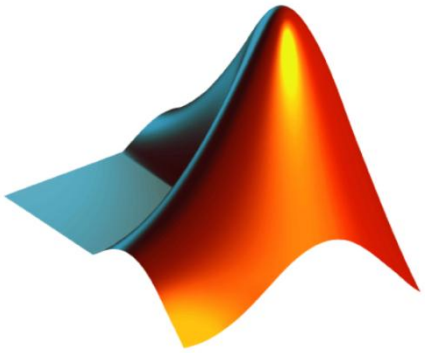


Fig. 3.2 MATLAB logo

It was developed in the late 1970s by Cleve Moler, a mathematics professor at the University of New Mexico, as an easy way for his students to use the LINPACK and EISPACK libraries for linear algebra calculations without having to use Fortran. Later, in 1984, Moler founded MathWorks, along with John Little and Steve Bangert, which led to the transformation of MATLAB into a widely used commercial product [36].

MATLAB is characterized by its reliance on procedural programming, with its support for object-oriented programming (OOP) techniques, with makes it suitable for developing advanced analytical programs. It also includes a large library of predefined functions, exceeding 1000 functions in the basic version, allowing users to solve many complex problems such as differential equations, matrix analysis, and numerical processing.

In addition, MATLAB provides additional specialized toolboxes covering diverse fields such as control, signal processing, physical modeling, and artificial intelligence, expanding its application capabilities in various branches of engineering and science. MATLAB also includes

an integrated development environment (IDE) consisting of a text editor, debugging tools, and interactive graphics modules for plotting and analyzing data [37].

Thus, we can say that this language is an essential tool in scientific research and engineering and industrial education, thanks to its high ability to simplify technical tasks and also to provide accurate and reliable solutions in record time.

3.2.2 MATLAB Desktop

MATLAB Desktop is the graphical user interface (GUI) that appears when you run MATLAB (Fig. 3.3). It contains a set of tools that help users develop and analyze code. Each tool has a specific function, such as executing commands, editing scripts, and viewing variables and files. The most important of these tools are [37] :

1. Command Window: This is an essential part of MATLAB Desktop, where you can enter programming commands and view the results immediately. It also allows you to perform calculations, create plots, and execute functions or scripts.
2. Toolbar: This is a bar at the top of the screen containing icons for functions and tools. The icons are organized into different tabs, and each tab contains a set of tools related to a specific function, such as file management tools, plotting tools, or mathematical analysis tools.
3. Command History Window: This window displays previously used commands. It can be accessed by clicking the up arrow within the command window.
4. Document Window: A dedicated window for viewing MATLAB files (such as scripts and .m function files). This window allows the user to write, edit, and execute code. It also contains debugging tools for the code as it runs.
5. Figure Window: A window used to display graphs resulting from sensitive operations. The user can modify these graphs within this window, such as changing colors, titles, and axes labels.
6. Workspace: Displays all variables stored in the workspace. The user can view the variables and their values, which helps manage data while running code.
7. Current Folder: A window that displays all files and folders in the current directory. This window allows you to manage, open, or delete files. When you select any file from the browser, the file details will appear in the Details window.
8. Help Browser: A tool for getting help with functions and functions in MATLAB. Access it by clicking the «Help » button in the toolbar and provides helpful explanations.

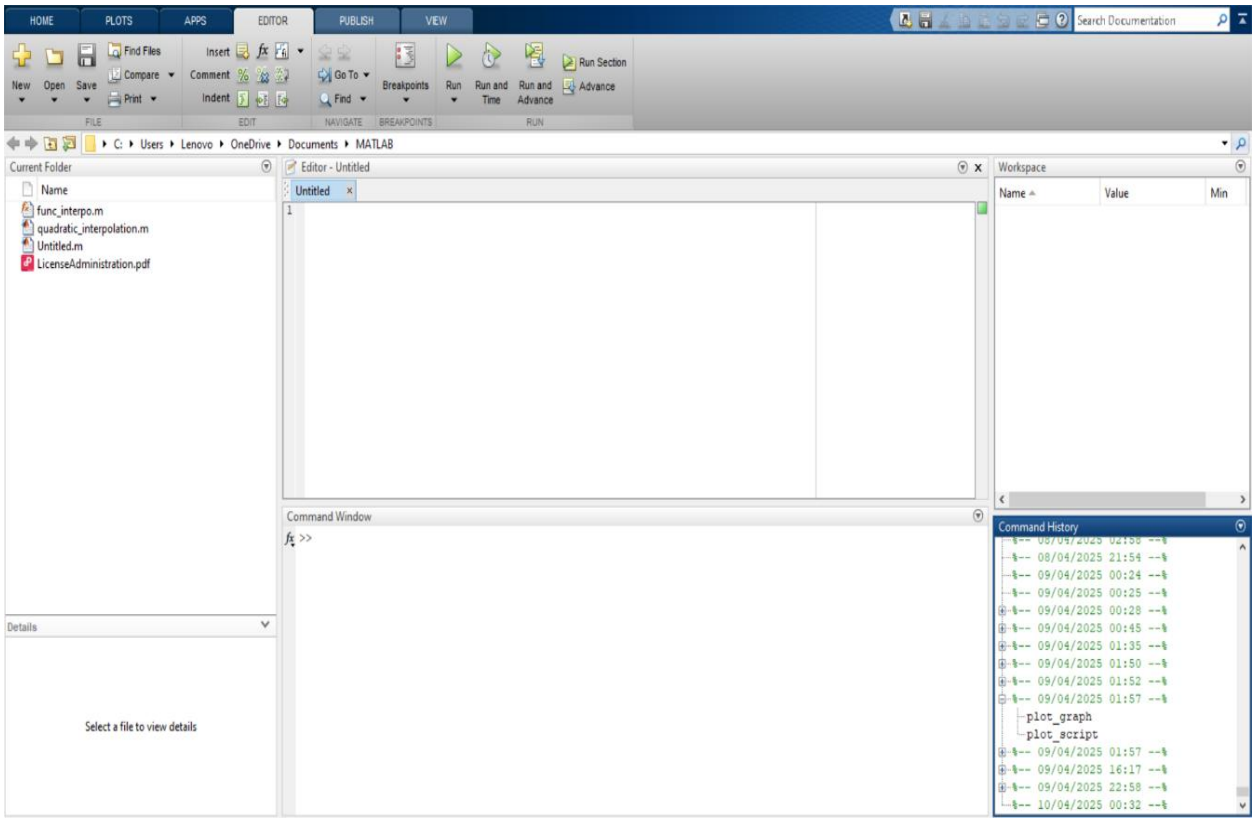


Fig. 3.3 The MATLAB Desktop

3.2.3 Types of .m File and Their Applications

In MATLAB, .m files are essential files used to write scripts and functions, execute commands, solve mathematical problems, and manipulate data. There are two main types of .m files: script files and function files.

Script files contain a series of MATLAB commands that are executed in the order they are written. They are used to perform a series of operations without requiring input or output (Fig. 3.4). They are particularly useful for automating repetitive tasks or performing step-by-step analyses.

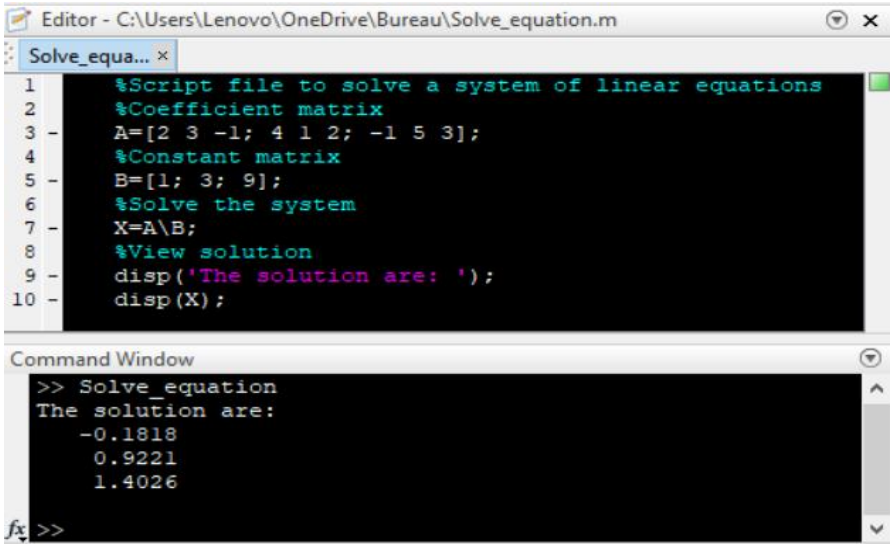


Fig. 3.4 Example of solving a system of linear equations using MATLAB

Function files are more advanced and contain functions that accept inputs and return outputs (Fig. 3.5). They allow for greater flexibility by allowing the same function to be reused in different contexts and with different inputs. They are essential for modular programming because they enable users to create reusable blocks of code that can be called with different parameters.

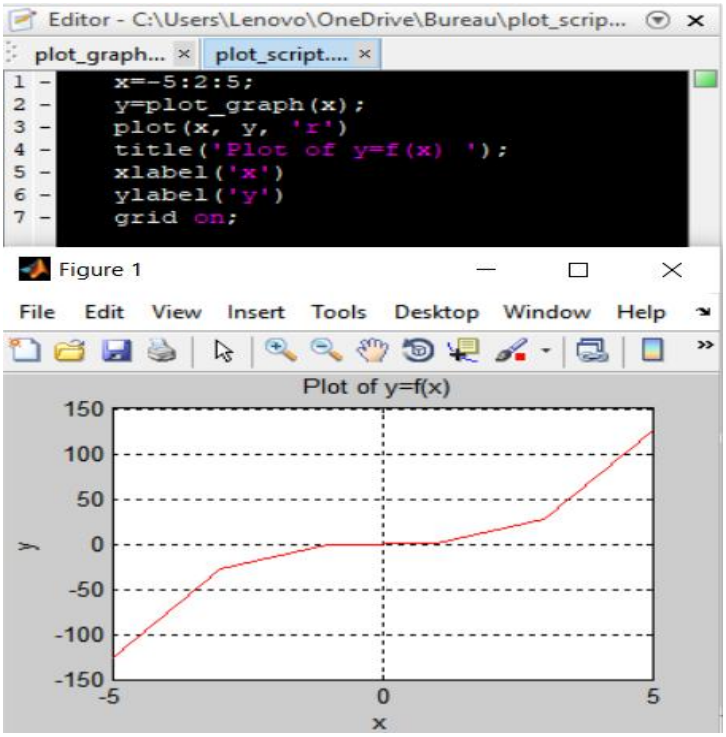


Fig. 3.5 Example of plot a graph using function file

Both types of .m files are important in MATLAB programming because they are used in a wide range of applications, from basic calculations and data processing to advanced numerical

simulations and graphical visualizations. By leveraging these files, MATLAB users can write efficient, maintainable, and reusable code to solve a wide variety of complex problems [38].

3.3 Programming using the finite element method

In this program, a two-dimensional numerical model was developed using the finite element method (FEM) to analyze the mechanical behavior of plane structures. This model is based on dividing the problem under study into triangular elements connected by nodes. The physical behavior of these elements is then represented on the network formed by the nodes. This program includes a sequential process, starting with defining the geometric data and material properties, then calculating the stiffness matrices for each element, and then assembling them to form the overall system of the structure. After applying the boundary conditions, the system is solved to obtain the nodal displacements, which are subsequently used to calculate the stresses and strains within each element.

3.3.2 Program for Analyzing Plane Structures

In this section, a program is presented for analyzing the mechanical behavior of plane structures using the finite element method, adopting the three-node triangular element as the basic model for the subdivision, developed using MATLAB. This program aims to accurately simulate the distribution of displacements, strains, and stresses within plane structures subject to specific loads and boundary conditions. It is designed to be an effective tool that enables the study of the response of plane structures with high elasticity and acceptable accuracy.

To ensure the clarity of the output and its non-interference with previous data, the program begins by initializing the work environment by using the **clc** commands to clear the command window, **clear all** to delete all variables from the workspace, and **close all** to close all open graphical windows (Fig. 3.7).

```
clc;
clear all;
close all;
```

Fig. 3.7 Initializing the workspace

Phase 1: input

In this initial phase, the user is prompted to enter the material's mechanical properties such as Young's modulus "E", Poisson's ratio "ν" and thickness "t", which are essential for calculating the elasticity matrix later. Next, select the type of stress condition: plane stress or plane strain, which significantly affects the construction of the elasticity matrix, where this information is stored for later use (Fig. 3.8).

```
E =input('Young''s Modulus E(MPa): ');
nu =input('Poisson''s ratio nu: ');
t =input('thickness t(mm):');
disp('Select stress condition:');
disp('1 - Plane Stress');
disp('2 - Plane Strain');
stress_case = input('Enter 1 or 2: ');
```

Fig. 3.8 Material's mechanical properties

In this part of the program, the basic geometric data for the model is entered, namely the number of nodes and the coordinates of each node in the (x, y) plane. This is a fundamental step in building a finite element model, as the actual locations of the nodes that represent the points of contact between the elements (Fig. 3.9).

```
num_nodes=input('Number of nodes:');
disp('Coordinates for each node [x y]:');
node_coords=zeros(num_nodes, 2);
for i=1:num_nodes
    node_coords(i,:)=input(['Node ', num2str(i), ' :']);
end
```

Fig. 3.9 Nodal coordinates

After entering the nodal coordinates, the next step is to enter the total number of elements and identify the nodes associated with each element. Each element is represented by a triangle composed of three nodes, which are identified by entering the node numbers that comprise the element. This information is critical for assembling the global stiffness matrix (Fig. 3.10).

```
num_elements=input('Number of elements:');
disp('Node connectivity for each element [ n1 n2 n3]:');
element_connectivity=zeros(num_elements,3);
for e=1:num_elements
    element_connectivity(e,:)=input(['Element ', num2str(e), ' :']);
end
```

Fig. 3.10 Total number of elements and node connectivity

- The num2str function is used to convert numeric data into a string using a specific format. It is one of the basic functions used when displaying numbers as text or when combining them with other text.

Phase 2: Calculating and assembly of elemental and global stiffness matrix

The stiffness matrix is a fundamental component of finite element numerical analysis, as it is the basis for constructing the system of equations that is solved to obtain the results. The stiffness matrix is first calculated for each element of the model [k_e], and these matrices are then combined to form the global stiffness matrix for the model [K]. the following program illustrates how to calculate it for three-node triangular elements(T3), based on input data such as node

coordinates, mechanical properties “E, v”, and thickness “t”. the process begins by calculating the area “A”, then constructing a strain-displacement matrix [B] that includes the derivatives of the shape function, as well as material property matrix [D], which is determined based on the stress state (plane stress or plane strain). The stiffness matrix is then calculated for each element and combined at the appropriate locations within the global matrix using the node degrees of freedom. The following is the practical part responsible for this process (Fig. 3.11):

```
K_global=zeros(2*num_nodes,2*num_nodes);
for e=1:num_elements
    nodes=element_connectivity(e, :);
    coords=node_coords(nodes, :);
    x1=coords(1,1);     y1=coords(1,2);
    x2=coords(2,1);     y2=coords(2,2);
    x3=coords(3,1);     y3=coords(3,2);
    A=0.5*det([1 x1 y1;1 x2 y2;1 x3 y3]);
    B=(1/(2*A))*[y2-y3    0         y3-y1    0        y1-y2     0;
                 0        x3-x2     0        x1-x3    0         x2-x1;
                 x3-x2    y2-y3     x1-x3    y3-y1    x2-x1     y1-y2];
    if stress_case==1
    D=(E/(1-nu^2))*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
    else
    D=(E/((1+nu)*(1-(2*nu))))*[1-nu nu 0;nu 1-nu 0;0 0 (1-(2*nu))/2];
    end
    Ke=t*A*B'*D*B;
    dof = reshape([2*nodes-1; 2*nodes], 1, []);
    K_global(dof,dof)=K_global(dof,dof)+Ke;
end
```

Fig. 3.11 Stiffness matrix for a three-node triangular element

Phase 3: Boundary conditions

In this part of the program, boundary conditions related to known displacements of certain degrees of freedom (DOFs) are entered to ensure the stability of the mathematical model. The user enters the number of boundary conditions, then enters both the number of degrees of freedom corresponding to the displacement and the displacement value, which are stored in a special array called BC. Each row contains the number of freedom in the first column and the displacement value in the second column (Fig. 3.12).

```
num_BC=input('How many boundary condition are known? ');
BC=zeros(num_BC,2);
for i=1:num_BC
    BC(i,:)=input(['BC ', num2str(i), ' :']);
end
```

Fig. 3.12 Input of boundary conditions

Phase 4: Input of known forces

We input the degrees of freedom corresponding to the nodes where the loads are applied, as well as the numerical values of those as follows (Fig. 3.13):

```
add_load=input('are there any concentrated forces?(Yes/No):', 's');
if strcmpi(add_load, 'Yes')
  num_node=input('number of Nodes were applied the load: ');
  for i=1:num_node
      node_num=input('Node number:');
      fx=input('force in x_direction:');
      fy=input('force in y_direction:');
      F(2*node_num-1)=fx;
      F(2*node_num)=fy;
  end
end
```

Fig. 3.13 Force vector

Phase 5: Gaussian elimination

The gaussian elimination method simplifies the by returning the elements below the main diagonal to zero to arrive at an upper triangular shape, as shown in (Fig. 3.14):

```
function U_solution = gaussian_elimination(K_new, F_new)
n = length(F_new);
Ab = [K_new F_new];
for i = 1:n-1
    [~, m] = max(abs(Ab(i:n,i)));
    m = m + i - 1;
    Ab([i m],:) = Ab([m i],:);
    for j = i+1:n
        Ab(j,:) = Ab(j,:) - Ab(j,i)/Ab(i,i) * Ab(i,:);
    end
end
U_solution = zeros(n,1);
U_solution(n) = Ab(n,end)/Ab(n,n);
for i = n-1:-1:1
    U_solution(i) = (Ab(i,end) - Ab(i,i+1:n)*U_solution(i+1:n)) / Ab(i,i);
end
end
```

Fig. 3.14 Gaussian elimination method

Phase 6: Solving the global system

After completing the gaussian elimination process, the unknown displacements are then calculated. These values are then substituted into the original system to calculate the unknown forces (Fig. 3.15).

```
U=zeros(2*num_nodes,1);
free_dof=setdiff(1:2*num_nodes,BC(:,1));
F_new = F(free_dof);
K_new = K_global(free_dof, free_dof);
U_solution = gaussian_elimination(K_new, F_new);
U(free_dof) = U_solution;

F=K_global*U;
```

Fig. 3.15 Solving the system of equation

- setdiff is used to extract elements in a set that do not belong to another set, it is used in the program to identify unconstrained nodes in a numerical model.

Phase 7: Computation of strains and stresses

At this stage, the strain and stress within each element are calculated as shown in Fig. 3.16:

```
stress=zeros(num_elements,3);
strain=zeros(num_elements,3);
for e=1:num_elements
    nodes=element_connectivity(e,:);
    coords=node_coords(nodes, :);
    x1=coords(1,1);      y1=coords(1,2);
    x2=coords(2,1);      y2=coords(2,2);
    x3=coords(3,1);      y3=coords(3,2);
    A=0.5*det([1 x1 y1;1 x2 y2;1 x3 y3]);
    B=(1/(2*A))*[y2-y3    0         y3-y1    0        y1-y2    0;
                 0        x3-x2     0        x1-x3    0        x2-x1;
                 x3-x2    y2-y3     x1-x3    y3-y1    x2-x1    y1-y2];
    U_element=U(reshape([2*nodes-1; 2*nodes], 1, [ ]));
    strain(e,:)=B*U_element;
    stress(e,:)=D*strain(e,:)';
end
```

Fig. 3.16 Computing strain and stress for each element

3.3.1 General structure of the finite element program

The general structure of finite element analysis consists of several main stages, each of which plays a specific role in solving engineering problems. The program is designed to model the physical behavior of structures to provide accurate numerical results. These include:

1. Input data: This step contains the necessary input data, including the structure's geometric properties, material properties, boundary conditions, and element definitions. These data must be entered accurately, as they will affect the results.
2. Calculating and assembling the stiffness matrix for the elements: The program calculates the stiffness matrix for each element based on the element's material and geometry properties, and then assembles them to form the general stiffness matrix.
3. Applying boundary conditions: In this stage, the program applies the boundary conditions to the general stiffness matrix. This stage is essential for determining how the structure responds to external forces and ensuring realistic results.
4. Solving the system: After establishing the general stiffness matrix and applying the boundary conditions, the program solves the system of linear equations to find the unknown displacements at the nodes. Numerical techniques, such as Gaussian elimination or iterative solutions, are used to obtain the solution, and computing the reactions at supports.
5. Output: Finally, the program outputs the results in a suitable format, including the stiffness matrix, displacements, reactions at the supports, and the stress and strain distribution.

The steps can be represented in (Fig. 3.6):

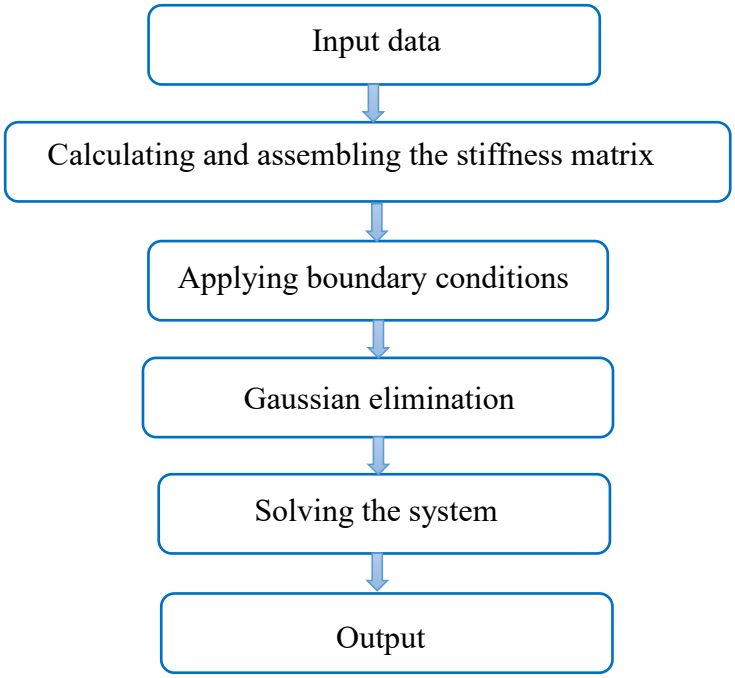


Fig. 3.6 Flowchart of the analysis program process sequence

3.4 Comparing program results with analytical solution results

In order to verify the accuracy and efficiency of the developed program in representing the mechanical behavior of planar structures, it is necessary to compare the results obtained from the code developed in MATLAB, and manual calculations using the basic principles of the finite element method. Through our comparison, we evaluate the degree of agreement and convergence between the results of these methods and verify the validity of the numerical model and its ability to simulate the expected physical responses.

In this example, a plate subjected to certain loading and limit conditions is studies as described in Paragraph 2.6, where a comparison will be made between the results calculated using the manual solution (Table 3.1 and Table 3.2), and those obtained by the developed program (Table 3.3 and Table 3.4) by calculating the relative error using in relation (3.1):

$$e\% = \frac{\left|X_{program} - X_{manual}\right|}{X_{manual}} \times 100 \tag{3.1}$$

Table 3.1 Forces and displacement results obtained by manual calculation for T3 element

Nodes	*Displacements u (mm)*	*Displacements v (mm)*	*Forces /x-axis*	*Forces /y-axis*
Node 1	$u_1=0$	$v_1=0$	$R_1=1333\ N$	$R_1=-500\ N$
Node 2	$u_2=3.175\times10^{-3}$	$v_2=0$	$R_2=1000\ N$	$R_2=8833\ N$
Node 3	$u_3=2.619\times10^{-3}$	$v_3=-4.444\times10^{-3}$	$R_3=1000\ N$	$R_3=-10000\ N$
Node 4	$u_4=0$	$v_4=0$	$R_4=-3333\ N$	$R_4=1667\ N$

Table 3.2 Stress and strain results obtained by manual calculation for T3 element

Element	*strains*			*stresses (MPa)*		
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
1	3.175×10^{-5}	-8.889×10^{-5}	-1.111×10^{-5}	*2.133*	*-18.133*	*-0.933*
2	2.619×10^{-5}	*0*	-4.444×10^{-5}	*5.866*	*1.466*	*-3.733*

Table 3.3 Results of displacements and forces obtained from the program for T3 element

node	Ux	Uy	Fx	Fy
1	0.000e+00	0.000e+00	1.333e+03	-5.000e+02
2	3.175e-03	0.000e+00	1.000e+03	8.833e+03
3	2.619e-03	-4.444e-03	1.000e+03	-1.000e+04
4	0.000e+00	0.000e+00	-3.333e+03	1.667e+03

Table 3.4 Results of stresses and strains obtained from the program for T3 element

Elements	Exx	Eyy	Exy	Sxx	Syy	Sxy
1	3.175e-05	-8.889e-05	-1.111e-05	2.133e+00	-1.813e+01	-9.333e-01
2	2.619e-05	0.000e+00	-4.444e-05	5.867e+00	1.467e+00	-3.733e+00

To compare the results, we use the relative error to find out the convergence ration, and the calculated values will be shown in a Table 3.4 and Table 3.5:

Table 3.5 Relative error between manual solution and program output for forces and displacements

Node	*u (e%)*	*v (e%)*	*(e%) F/x-axis*	*(e%) F/y-axis*
1	*0*	*0*	*0*	*0*
2	*0*	*0*	*0*	*0*
3	*0*	*0*	*0*	*0*
4	*0*	*0*	*0*	*0*

Table 3.6 Relative error between manual solution and program output for stresses and strains

Element	*(e%) strains*			*(e%) stresses*		
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
1	*0*	*0*	*0*	*0.01*	*0.001*	*0.03*
2	*0*	*0*	*0*	*0.01*	*0.05*	*0.008*

Table 3.4 and table 3.5 represents the relative error between the manual calculation results and those obtained from the program for a plate consisting of 4 nodes and 2 triangular element of type T3, we observed that there is no error in the displacements, force and strain, but in the stress there are relative error in the two elements. In the first element in the x-direction it was 0.01%, in the y-direction it was 0.001%, and in the xy-shear component it was 0.03%, and for the second element in the x-direction it was 0.01%, in y-direction it was 0.05%, and in the xy-shear component it was

0.008%, if we add up all the errors we get a total error of 0.109%, the total errors are less than 1%, so we can say that the developed program is reliable in simulating the mechanical behavior.

3.5 Validation results using ABAQUS software

Validation the results is an essential step to ensure the accuracy of the program created in MATLAB to simulate the behavior of two-dimensional plane structure using FEM. The numerical results obtained from MATLAB will be compared to those obtained from a commercial simulation program.

In our study, ABAQUS was chosen for the validation process due to its high accuracy, widespread use in engineering application, and adoption in academia.

3.5.1 Overview of ABAQUS

ABAQUS it is an integrated suite of powerful and advanced engineering simulation software program based on the finite element method [39]. It was developed in 1978 by Dassault Système within the SIMULIA platform. It is widely used for its high efficiency in handling linear and nonlinear problems in various fields such as body mechanics, structural analysis, and heat transfer. It also allows users to simulate the physical response of structures under the influence of loads, temperatures, or shocks. Technically, this program provides accurate results including energy, forces at nodes, deformations, displacements, stresses, velocity and acceleration, and all the physical quantities necessary to understand and evaluate the behavior of engineering models in various fields [40].

3.5.2 ABAQUS program components

ABAQUS consists of three main products: Abaqus/CAE, Abaqus/Standard, Abaqus/Explicit [41]

- Abaqus/CAE (the complete environment for ABAQUS): An integrated graphical interface used for modeling, simulation, and displaying results.
- Abaqus/Standard: An implicit solver used to solve linear and nonlinear problems in static or quasi-static system.
- Abaqus/Explicit: An explicit solver used in dynamic and wave propagation analyses.

From here we can say that there is a connection between these components, starting from creating the model in ABAQUS/CAE and determining the type of simulation required, whether static or dynamic. Based on the type, the data is directed to the appropriate unit, ABAQUS/Standard or ABAQUS/Explicit, ending with implementing the numerical analysis and displaying the results within the graphical interface ABAQUS/CAE.

3.5.3 ABAQUS Unit System

Before defining any model in ABAQUS, you must choose an appropriate unit system that will be used throughout the modeling process. ABAQUS does not have a built-in unit system.

Therefore, all values must be entered in compatible units within a specific system because they will be used in the analysis results. The international system of units (SI) with millimeters (mm) is often used [42].

The system of units used in ABAQUS is shown in Table 3.7:

Table 3.7 The system of units used in ABAQUS

Quantity	*SI*	*SI(mm)*	*US Unit(ft)*	*US Unit(inch)*
Length	*m*	*mm*	*ft*	*in*
Force	*N*	*N*	*lbf*	*lbf*
Mass	*Kg*	*tonne (10^3 Kg)*	*slug*	*lbf s^2/in*
Time	*s*	*s*	*s*	*s*
Stress	*Pa (N/m^2)*	*MPa($N/mm^{2)}$*	*lbf /ft^2*	*psi(lbf/in^2)*
Energy	*J*	*mJ(10^{-3} J)*	*ft lbf*	*In lbf*
Density	*Kg/m^3*	*tonne/ m^3*	*slug/ft^3*	*lbf s^2/in^4*

3.5.4 Comparing program results with ABAQUS results

To ensure that the developed program provides accurate results, a comparison was made between its results and those provided by ABAQUS software. This comparison aims to ensure that the calculations performed by the developed program are close to those provided by the ABAQUS program. If we find that the results are close, this means that the developed program is accurate and reliable, which means it can be relied upon even in difficult and complex cases where it is difficult to resort to a manual solution, which is used only in simple cases.

To compare the results of the developed program with those obtained from ABAQUS, two different, complex cases were selected for simulation.

- **First case**

Given a thin plate fixed on the left and subjected to concentrated force, the plate is divided into 12 linear triangular elements and 11 nodes as shown in Fig. 3.17, with Young's modulus E=210 GPa, Poisson's ratio $\nu = 0.3$, thickness t=0.025 m, and force P=12.5 KN, and the stress type is plane stress type. We propose to determine the unknown displacements, reactions at nodes, stress and strain.

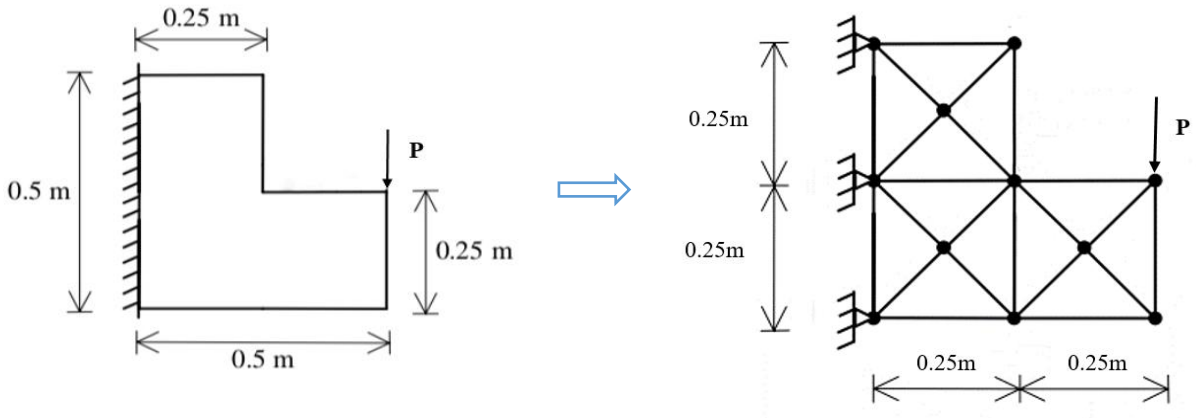


Fig. 3.17 Thin plate with concentrated force

By using the program for the 2D linear triangular element, and inputting the data for this case, we reached the results show in the Tables 3.8 and 3.9.

Table 3.8 Results of displacements and forces computed by the program for a thin plate

node	Ux	Uy	Fx	Fy
1	-3.0410e-03	-1.5423e-02	0.0000e+00	0.0000e+00
2	-1.1454e-02	-2.4841e-02	0.0000e+00	0.0000e+00
3	6.9878e-03	-2.8471e-02	0.0000e+00	-1.2500e+04
4	2.5912e-03	-5.1144e-03	0.0000e+00	0.0000e+00
5	-7.4936e-03	-7.2905e-03	0.0000e+00	0.0000e+00
6	-9.6366e-04	-1.8873e-03	0.0000e+00	0.0000e+00
7	0.0000e+00	0.0000e+00	1.4842e+04	1.7813e+03
8	0.0000e+00	0.0000e+00	-4.6849e+03	2.9133e+03
9	1.1337e-03	-2.3331e-03	0.0000e+00	0.0000e+00
10	1.1866e-03	-3.5419e-03	0.0000e+00	0.0000e+00
11	0.0000e+00	0.0000e+00	-1.0158e+04	7.8055e+03

Table 3.9 Results of stresses and strains computed by the program for a thin plate

Elements	Exx	Eyy	Exy	Sxx	Syy	Sxy
1	6.4631e-06	-1.4519e-05	-1.6095e-05	4.8630e-01	-2.9031e+00	-1.3000e+00
2	1.7587e-05	-1.0955e-05	-3.0783e-05	3.3000e+00	-1.3106e+00	-2.4863e+00
3	-1.5842e-05	5.1402e-06	-1.8741e-05	-3.3000e+00	8.9442e-02	-1.5137e+00
4	-2.9974e-05	1.4063e-05	-6.8969e-06	-5.9435e+00	1.1703e+00	-5.5706e-01
5	-4.7186e-06	8.7043e-06	-3.3429e-05	-4.8630e-01	1.6820e+00	-2.7000e+00
6	-7.7093e-06	0.0000e+00	-1.5099e-05	-1.7791e+00	-5.3372e-01	-1.2195e+00
7	-1.1900e-05	8.7043e-06	5.8180e-06	-2.1436e+00	1.1848e+00	4.6992e-01
8	4.7465e-06	4.4976e-06	-1.8491e-05	1.4067e+00	1.3665e+00	-1.4935e+00
9	1.0365e-05	-5.3591e-06	-2.3836e-06	2.0208e+00	-5.1915e-01	-1.9252e-01
10	9.0699e-06	0.0000e+00	-1.8665e-05	2.0931e+00	6.2792e-01	-1.5076e+00
11	6.0413e-06	6.2902e-06	-2.1578e-05	1.8296e+00	1.8698e+00	-1.7429e+00
12	1.0365e-05	1.7926e-06	-2.1752e-05	2.5160e+00	1.1312e+00	-1.7569e+00

To compare this results with those obtained from ABAQUS software. You must first follow these steps to perform the simulation process:

1. **Part creating**

- Select a 2D model of the “Deformable” type and choose “Shell” as the base feature.
- Draw a rectangle representing the dimensions of the plate, then perform a Partition operation to remove the square as shown in the Fig. 3.18.

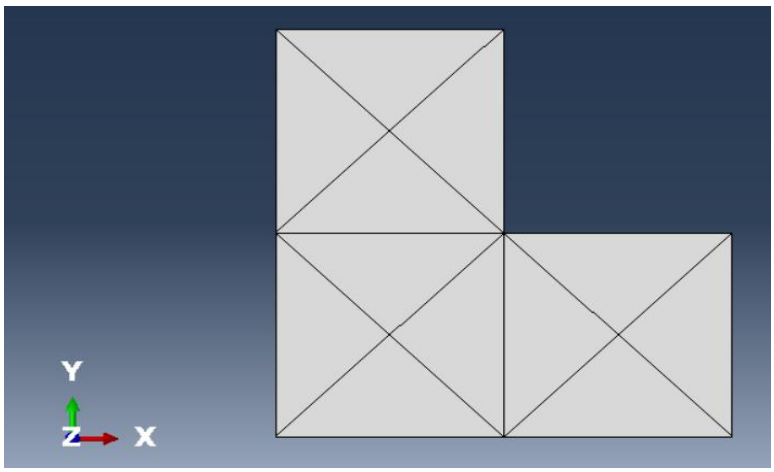


Fig. 3.18 2D part definition of the plate

2. **Material properties and section definition**

- Material properties: enter the material’s mechanical properties, such as Young’s modulus and Poisson’s ratio, to represent the material’s, etc. (Fig. 3.19).

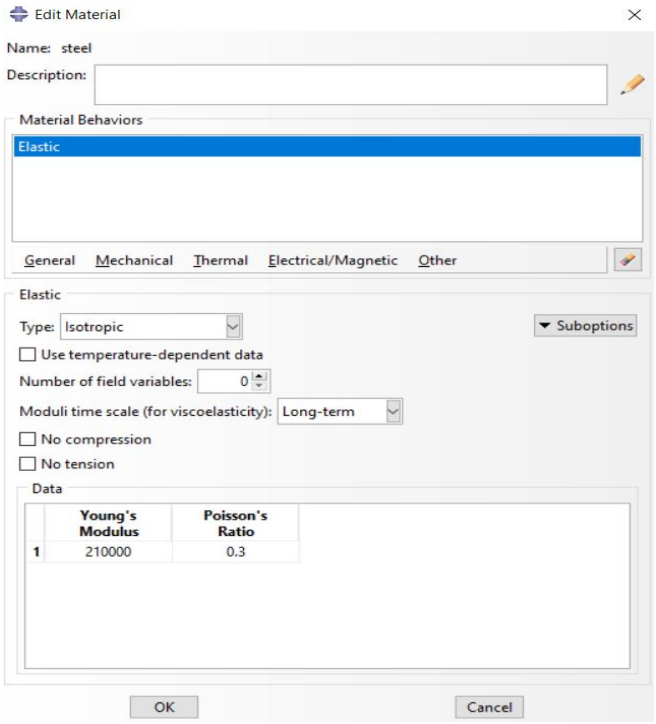


Fig. 3.19 Properties of the material

- Section definition: Creating a section involves selecting the appropriate category, specifying the type of plate used in the model, and its thickness (Fig. 3.20).

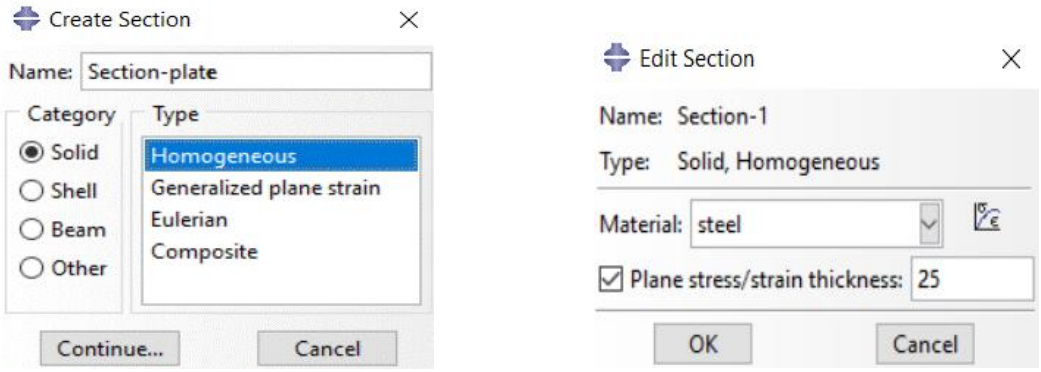


Fig. 3.20 Create section

3. Module assembly

The draw part is inserted into the assembly environment as a dependent instance, to maintain its associated with the original part and facilitate the modification of the mesh and properties.

4. Creating an analysis step

The analysis type is selected as Static, General, which is considered optimal for analyzing stresses and deformation in static conditions (Fig. 3.21).

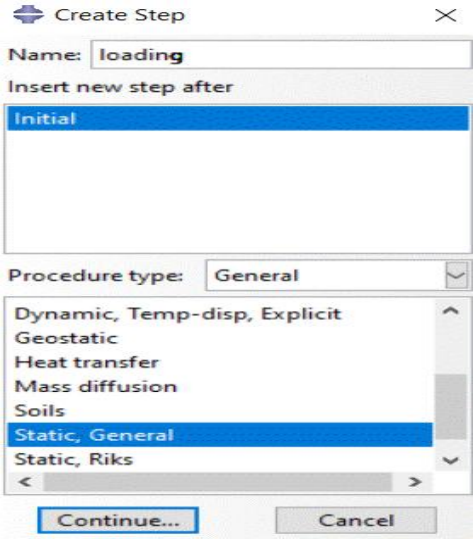


Fig. 3.21 Preparing the analysis step

5. Applying a boundary condition and a load to the model

- Applying load: add the forces applied to the node, specifying the direction and value of this force precisely (Fig. 3.22).

- Apply boundary conditions: fixed the left edge of the plate completely by applying appropriate boundary conditions that prevent movement on this edge (Fig. 3.22).

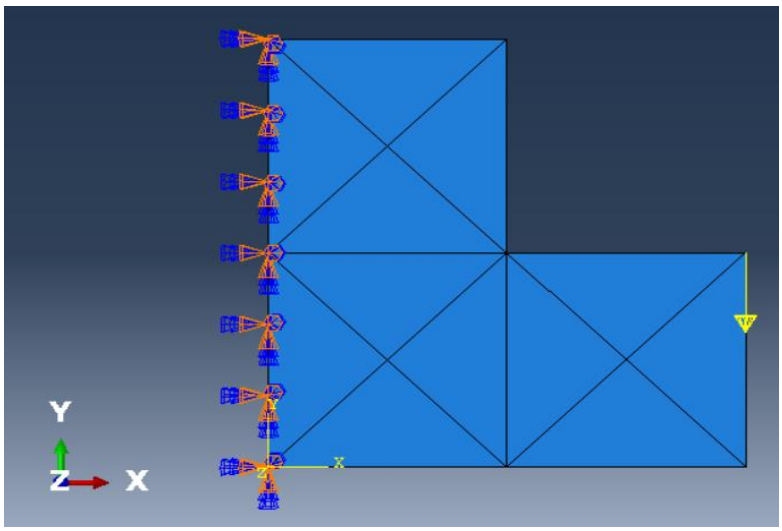


Fig. 3.22 Applying loads and boundary conditions

6. Meshing the model

The mesh was created by dividing the plate into triangular elements, with an appropriate element size determined before meshing and choose the geometric order “linear” (Fig. 3.23).

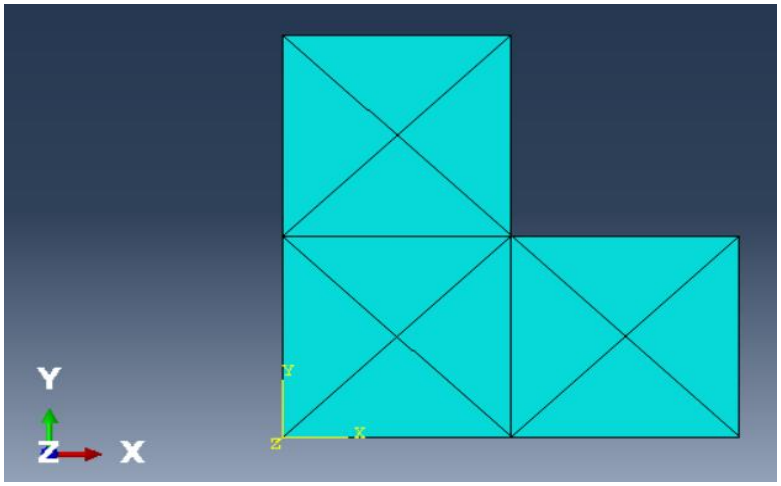


Fig. 3.23 Meshing the plate into triangular elements

7. Module Job

After completing all the model preparation stages, the numerical analysis stage comes. This process is carried out through the job module, where a task is created to analyze and define the model, then the calculations are launches. Upon successful completion, the program displays a message confirming the success of the task, as shown in the Fig. 3.24.



Fig. 3.24 Job completed successfully

8. Visualization

The results files in .obd format is opened using the visualization module in ABAQUS, which displays the model responses as color maps that illustrate the distribution of displacements, stresses, strains, and reactions at the supports. These maps explain the mechanical behavior of the system under study and are also a key reference in the simulation validation process.

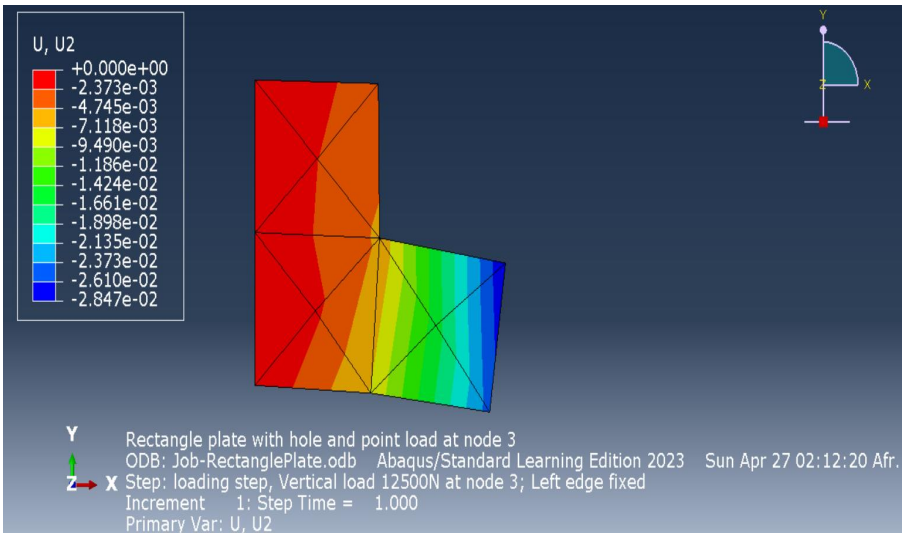


Fig. 3.25 Viewing the results

After performing the analysis in ABAQUS, we obtained numerical results that included nodal displacements, reactions at the supports, as well as stresses and strains. Table 3.10 and table 3.11 shows the extracted results, which will be used to compare the performance and theoretical results of the model.

Table 3.10 Displacements and forces results for the first case extracted by ABAQUS software

Node	RF1	RF2	U1	U2
1	0.	0.	-3.0410E-03	-15.423E-03
2	0.	0.	-11.454E-03	-24.841E-03
3	0	-12.500E+03	6.9878E-03	-28.471E-03
4	0.	0.	2.5912E-03	-5.1144E-03
5	0.	0.	-7.4936E-03	-7.2905E-03
6	0.	0.	-963.66E-06	-1.8873E-03
7	14.842E+03	1.7813E+03	0.	0.
8	-4.6849E+03	2.9133E+03	0.	0.
9	0.	0.	1.1337E-03	-2.3331E-03
10	0.	0.	1.1866E-03	-3.5419E-03
11	-10.158E+03	7.8055E+03	0.	0.

Table 3.11 Stresses and strains results for the first case extracted by ABAQUS software

Element	E11	E22	E12	S11	S22	S12
1	6.4631E-06	-14.519E-06	-16.095E-06	486.30E-03	-2.9031	-1.3000
2	17.587E-06	-10.955E-06	-30.783E-06	3.3000	-1.3106	-2.4863
3	-15.842E-06	5.1402E-06	-18.741E-06	-3.3000	89.442E-03	-1.5137
4	-29.974E-06	14.063E-06	-6.8969E-06	-5.9435	1.1703	-557.06E-03
5	-4.7186E-06	8.7043E-06	-33.429E-06	-486.30E-03	1.6820	-2.7000
6	-7.7093E-06	0.	-15.099E-06	-1.7791	-533.72E-03	-1.2195
7	-11.900E-06	8.7043E-06	5.8180E-06	-2.1436	1.1848	469.92E-03
8	4.7465E-06	4.4976E-06	-18.491E-06	1.4067	1.3665	-1.4935
9	10.365E-06	-5.3591E-06	-2.3836E-06	2.0208	-519.15E-03	-192.52E-03
10	9.0699E-06	0.	-18.665E-06	2.0931	627.92E-03	-1.5076
11	6.0413E-06	6.2902E-06	-21.578E-06	1.8296	1.8698	-1.7429
12	10.365E-06	1.7926E-06	-21.752E-06	2.5160	1.1312	-1.7569

To compare these results, we relied on the relative error expression (3.1), and we found the values as shown in the Tables 3.12 and 3.13.

Table 3.12 Relative error of displacements and forces for the first case between the results of the ABAQUS software and those calculated by the developed program

Nodes	*(e%) u*	*(e%) v*	*(e%) F/x-axis*	*(e%) F/y-axis*
Node 1	*0*	*0*	*0*	*0*
Node 2	*0*	*0*	*0*	*0*
Node 3	*0*	*0*	*0*	*0*
Node 4	*0*	*0*	*0*	*0*
Node 5	*0*	*0*	*0*	*0*
Node 6	*0*	*0*	*0*	*0*
Node 7	*0*	*0*	*0*	*0*
Node 8	*0*	*0*	*0*	*0*
Node 9	*0*	*0*	*0*	*0*
Node 10	*0*	*0*	*0*	*0*
Node 11	*0*	*0*	*0*	*0*

Table 3.13 Relative error of strains and stresses for the first case between the results of the ABAQUS software and those calculated by the developed program

Elements	*(e%) strains*			*(e%) stresses*		
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
Element 1	*0*	*0*	*0*	*0*	*0*	*0*
Element 2	*0*	*0*	*0*	*0*	*0*	*0*
Element 3	*0*	*0*	*0*	*0*	*0*	*0*
Element 4	*0*	*0*	*0*	*0*	*0*	*0*
Element 5	*0*	*0*	*0*	*0*	*0*	*0*
Element 6	*0*	*0*	*0*	*0*	*0*	*0*
Element 7	*0*	*0*	*0*	*0*	*0*	*0*
Element 8	*0*	*0*	*0*	*0*	*0*	*0*
Element 9	*0*	*0*	*0*	*0*	*0*	*0*
Element 10	*0*	*0*	*0*	*0*	*0*	*0*
Element 11	*0*	*0*	*0*	*0*	*0*	*0*
Element 12	*0*	*0*	*0*	*0*	*0*	*0*

Table 3.12 and table 3.13 represents the relative error between the ABAQUS results and those obtained from the program for a plate consisting of 11 nodes and 12 triangular element of type T3, we observed that there is no error in the displacements, force, stress, and strain. Since the error is zero, this indicates the accuracy of the results calculated by the developed program, and thus it can be relied upon to analyze such engineering models with high accuracy.

- The second case

The same plate shown in Paragraph 2.6 will be used to perform the comparison process and verify the accuracy of the program, by dividing it into 30 linear triangular elements.

By using the program for the 2D linear triangular element, and inputting the data for this case, we obtained the results show in the Tables 3.14 and 3.15.

Table 3.14 Results of displacements and forces for the second case computed by the program

node	Ux	Uy	Fx	Fy
1	0.0000e+00	0.0000e+00	6.3614e+02	-1.9301e+02
2	-5.4866e-04	-9.4254e-04	0.0000e+00	0.0000e+00
3	-9.1589e-04	-2.7674e-03	0.0000e+00	0.0000e+00
4	-7.5707e-04	-5.1231e-03	0.0000e+00	0.0000e+00
5	6.0919e-04	-6.5381e-03	0.0000e+00	0.0000e+00
6	4.8239e-03	0.0000e+00	1.0000e+03	9.1819e+03
7	4.5161e-04	9.6914e-05	0.0000e+00	0.0000e+00
8	4.1238e-04	-8.2125e-04	0.0000e+00	0.0000e+00
9	3.8693e-04	-2.6115e-03	0.0000e+00	0.0000e+00
10	2.9564e-04	-5.0362e-03	0.0000e+00	0.0000e+00
11	2.4813e-04	-7.3147e-03	0.0000e+00	0.0000e+00
12	1.2361e-03	-7.1284e-03	0.0000e+00	0.0000e+00
13	9.3952e-04	1.6920e-04	0.0000e+00	0.0000e+00
14	1.3020e-03	-7.3816e-04	0.0000e+00	0.0000e+00
15	1.7859e-03	-2.4867e-03	0.0000e+00	0.0000e+00
16	2.1072e-03	-4.9091e-03	0.0000e+00	0.0000e+00
17	2.0087e-03	-8.3280e-03	0.0000e+00	0.0000e+00
18	2.8545e-03	-1.3095e-02	0.0000e+00	0.0000e+00
19	0.0000e+00	0.0000e+00	-2.6361e+03	1.0111e+03
20	2.0045e-03	-8.0752e-04	0.0000e+00	0.0000e+00
21	3.5132e-03	-2.6210e-03	0.0000e+00	0.0000e+00
22	4.9936e-03	-5.1242e-03	0.0000e+00	0.0000e+00
23	6.7933e-03	-8.9737e-03	0.0000e+00	0.0000e+00
24	7.6265e-03	-1.9472e-02	1.0000e+03	-1.0000e+04

Table 3.15 Results of stresses and strains for the second case computed by the program

Elements	Exx	Eyy	Exy	Sxx	Syy	Sxy
1	-2.7433e-05	7.2774e-06	1.0535e-05	-5.7374e+00	9.3890e-02	8.8496e-01
2	-1.9617e-06	5.8148e-06	-1.8812e-05	-1.1378e-01	1.1927e+00	-1.5802e+00
3	-1.8361e-05	9.3569e-06	-1.3076e-05	-3.5890e+00	1.0677e+00	-1.0984e+00
4	-1.2724e-06	7.2774e-06	-3.1850e-05	1.2251e-01	1.5589e+00	-2.6754e+00
5	7.9408e-06	5.2098e-06	-5.4620e-05	2.0705e+00	1.6117e+00	-4.5881e+00
6	-4.5643e-06	9.3569e-06	-4.3070e-05	-4.9841e-01	1.8404e+00	-3.6178e+00
7	6.8313e-05	-4.6597e-05	-9.2414e-05	1.2693e+01	-6.6121e+00	-7.7628e+00
8	-2.3755e-06	5.2098e-06	-5.0760e-05	-2.4037e-01	1.0340e+00	-4.2638e+00
9	2.1073e-04	-4.2770e-04	1.1164e-04	2.3253e+01	-8.4005e+01	9.3775e+00
10	4.9397e-05	-4.6597e-05	-1.2349e-05	8.4556e+00	-7.6714e+00	-1.0374e+00
11	-1.9617e-06	4.9852e-06	7.4711e-06	-1.6024e-01	1.0068e+00	6.2757e-01
12	1.8125e-05	4.3370e-06	-1.6093e-05	4.3029e+00	1.9865e+00	-1.3518e+00
13	-1.2724e-06	7.4844e-06	-5.5716e-06	1.3410e-01	1.6053e+00	-4.6802e-01
14	2.4195e-05	4.9852e-06	-3.4050e-05	5.6988e+00	2.4716e+00	-2.8602e+00
15	-4.5643e-06	7.6282e-06	-1.2543e-05	-5.9521e-01	1.4531e+00	-1.0536e+00
16	1.6065e-05	7.4844e-06	-3.7179e-05	4.0178e+00	2.5762e+00	-3.1230e+00
17	-2.3755e-06	-6.0796e-05	-8.2865e-06	-3.9367e+00	-1.3751e+01	-6.9607e-01
18	-4.9252e-06	7.6282e-06	-6.2247e-05	-6.7606e-01	1.4329e+00	-5.2288e+00
19	4.9397e-05	-3.5797e-04	1.0642e-04	-8.9816e+00	-7.7420e+01	8.9391e+00
20	4.2287e-05	-6.0796e-05	-1.3270e-04	6.0676e+00	-1.1250e+01	-1.1147e+01
21	1.8125e-05	-4.1613e-06	-3.2216e-06	3.8270e+00	8.2878e-02	-2.7061e-01
22	1.0022e-04	-1.0152e-05	-9.6747e-05	2.1882e+01	3.3385e+00	-8.1268e+00
23	2.4195e-05	-8.0572e-06	1.6209e-05	4.9685e+00	-4.4989e-01	1.3616e+00
24	7.5438e-05	-4.1613e-06	-4.8529e-05	1.6665e+01	3.2924e+00	-4.0764e+00
25	1.6065e-05	-1.2907e-05	5.2066e-05	2.8759e+00	-1.9915e+00	4.3735e+00
26	7.4021e-05	-8.0572e-06	-2.1522e-05	1.6129e+01	2.3404e+00	-1.8079e+00
27	-4.9252e-06	-3.8744e-05	1.1613e-04	-3.2729e+00	-8.9545e+00	9.7550e+00
28	8.9983e-05	-1.2907e-05	-1.9290e-05	1.9433e+01	2.1478e+00	-1.6203e+00
29	4.2287e-05	-3.8267e-04	4.7990e-05	-1.1957e+01	-8.3350e+01	4.0311e+00
30	4.1662e-05	-3.8744e-05	-2.3786e-04	7.1626e+00	-6.3456e+00	-1.9981e+01

To compare these results, we will follow the same simulation methodology as in the first case, but the difference lies in the material properties, mesh dividing, geometry shape of the plate, applied forces, and boundary conditions.

- **Geometric shape:** A rectangular plate with dimensions of 100*50 mm^2 was used, as shown in Fig. 3.26.

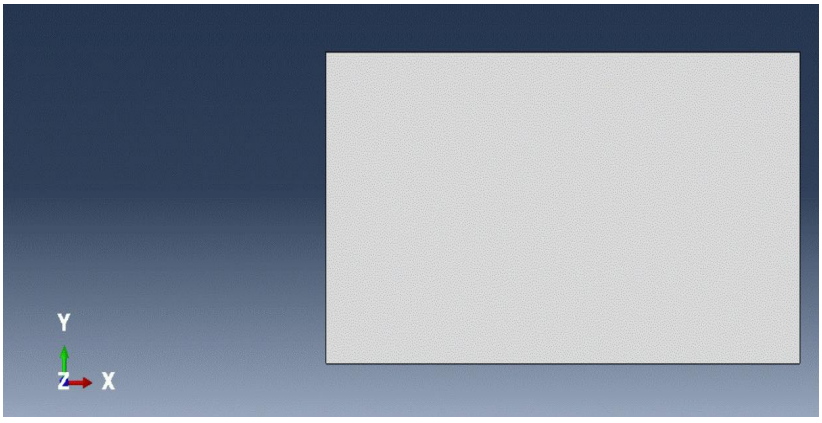


Fig. 3.26 Create the model

- **Materials properties:** The Poisson's ratio was set a value of 0.25, and the element thickness was 10 mm.
- **Applied forces and boundary conditions:** Boundary conditions were applied by adding a double support to the nodes on the right side, while on the other side, forces were applied as shown in the Fig. 3.27.

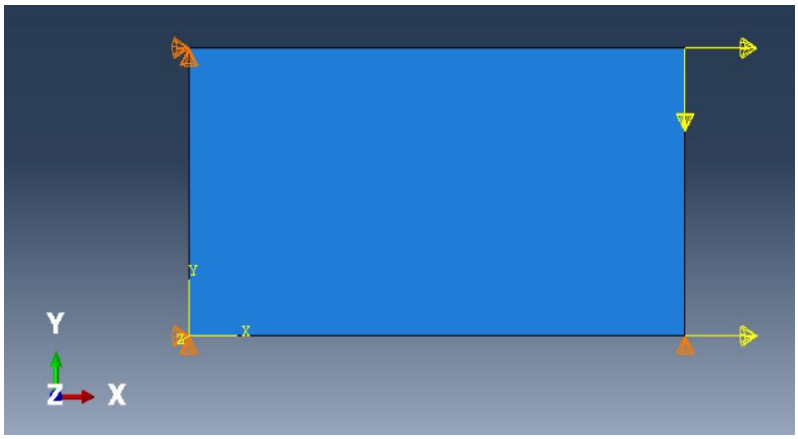


Fig. 3.27 Boundary conditions and applied forces

- **Mesh dividing:** The plate was divided into 30 linear triangular elements and 24 nodes to increase the accuracy of the simulation (Fig. 3.28).

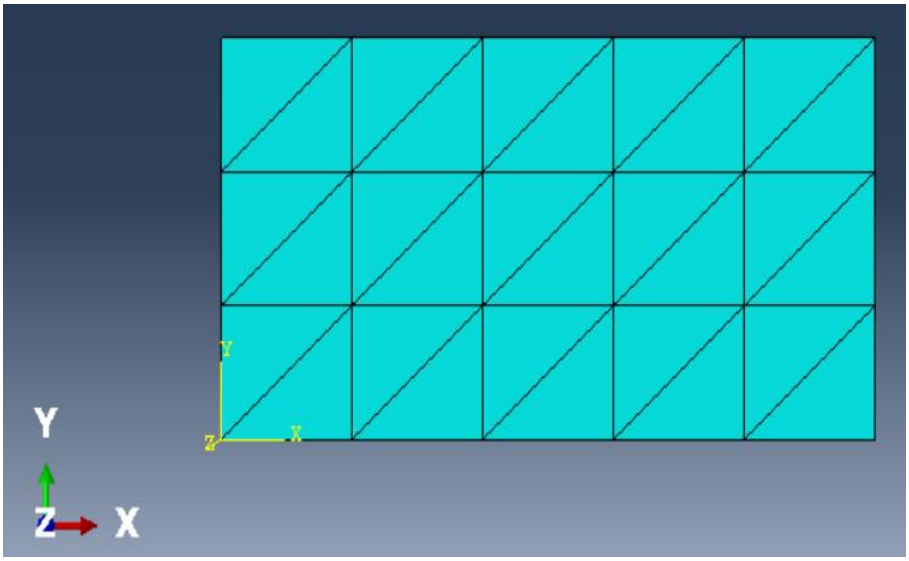


Fig. 3.28 Divide the mesh into 30 linear triangular elements

- **Visualization**

The visualization module is an essential stage for displaying simulation results in a visual manner, using color maps that show the distribution of displacement, stresses, strains and reactions at the supports, in order to understand the behavior of the structure under applied loads.

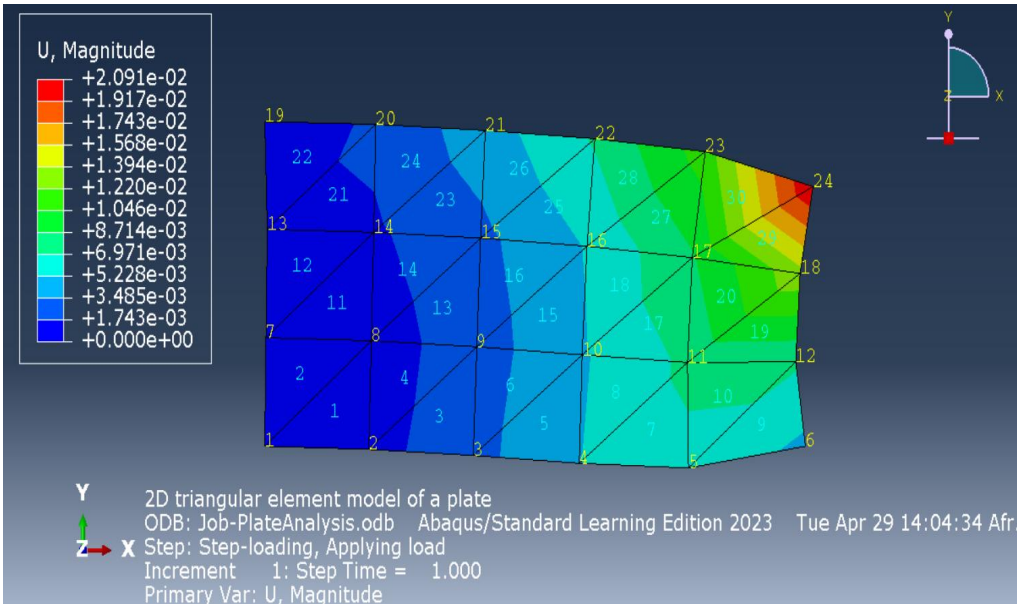


Fig. 3.29 Viewing the results

After completing the simulation process, the results of displacements, forces, stresses, and strains are extracted for use in the comparison process with the results of the developed program, with the aim of verifying the accuracy of the numerical model and determining its compatibility with approved simulation tools as in the Table 3.16 and Table 3.17.

Table 3.16 Displacement and force results for the second case provided by ABAQUS software

Node	RF1	RF2	U1	U2
1	636.14	-193.01	0.	0.
2	0.	0.	-548.66E-06	-942.54E-06
3	0.	0.	-915.89E-06	-2.7674E-03
4	0.	0.	-757.07E-06	-5.1231E-03
5	0.	0.	609.19E-06	-6.5381E-03
6	1000	9.1819E+03	4.8239E-03	0.
7	0.	0.	451.61E-06	96.914E-06
8	0.	0.	412.38E-06	-821.25E-06
9	0.	0.	386.93E-06	-2.6115E-03
10	0.	0.	295.64E-06	-5.0362E-03
11	0.	0.	248.13E-06	-7.3147E-03
12	0.	0.	1.2361E-03	-7.1284E-03
13	0.	0.	939.52E-06	169.20E-06
14	0.	0.	1.3020E-03	-738.16E-06
15	0.	0.	1.7859E-03	-2.4867E-03
16	0.	0.	2.1072E-03	-4.9091E-03
17	0.	0.	2.0087E-03	-8.3280E-03
18	0.	0.	2.8545E-03	-13.095E-03
19	-2.6361E+03	1.0111E+03	0.	0.
20	0.	0.	2.0045E-03	-807.52E-06
21	0.	0.	3.5132E-03	-2.6210E-03
22	0.	0.	4.9936E-03	-5.1242E-03
23	0.	0.	6.7933E-03	-8.9737E-03
24	1000	-10E+03	7.6265E-03	-19.472E-03

Table 3.17 Stress and strain results for the second case provided by ABAQUS software

Element	E11	E22	E12	S11	S22	S12
1	-27.433E-06	7.2774E-06	10.535E-06	-5.7374	93.890E-03	884.96E-03
2	-1.9617E-06	5.8148E-06	-18.812E-06	-113.78E-03	1.1927	-1.5802
3	-18.361E-06	9.3569E-06	-13.076E-06	-3.5890	1.0677	-1.0984
4	-1.2724E-06	7.2774E-06	-31.850E-06	122.51E-03	1.5589	-2.6754
5	7.9408E-06	5.2098E-06	-54.620E-06	2.0705	1.6117	-4.5881
6	-4.5643E-06	9.3569E-06	-43.070E-06	-498.41E-03	1.8404	-3.6178
7	68.313E-06	-46.597E-06	-92.414E-06	12.693	-6.6121	-7.7628
8	-2.3755E-06	5.2098E-06	-50.760E-06	-240.37E-03	1.0340	-4.2638
9	210.73E-06	-427.70E-06	111.64E-06	23.253	-84.005	9.3775
10	49.397E-06	-46.597E-06	-12.349E-06	8.4556	-7.6714	-1.0374
11	-1.9617E-06	4.9852E-06	7.4711E-06	-160.24E-03	1.0068	627.57E-03
12	18.125E-06	4.3370E-06	-16.093E-06	4.3029	1.9865	-1.3518
13	-1.2724E-06	7.4844E-06	-5.5716E-06	134.10E-03	1.6053	-468.02E-03
14	24.195E-06	4.9852E-06	-34.050E-06	5.6988	2.4716	-2.8602
15	-4.5643E-06	7.6282E-06	-12.543E-06	-595.21E-03	1.4531	-1.0536
16	16.065E-06	7.4844E-06	-37.179E-06	4.0178	2.5762	-3.1230
17	-2.3755E-06	-60.796E-06	-8.2865E-06	-3.9367	-13.751	-696.07E-03
18	-4.9252E-06	7.6282E-06	-62.247E-06	-676.06E-03	1.4329	-5.2288
19	49.397E-06	-357.97E-06	106.42E-06	-8.9816	-77.420	8.9391
20	42.287E-06	-60.796E-06	-132.70E-06	6.0676	-11.250	-11.147
21	18.125E-06	-4.1613E-06	-3.2216E-06	3.8270	82.878E-03	-270.61E-03
22	100.22E-06	-10.152E-06	-96.747E-06	21.882	3.3385	-8.1268
23	24.195E-06	-8.0571E-06	16.209E-06	4.9685	-449.89E-03	1.3616
24	75.438E-06	-4.1613E-06	-48.529E-06	16.665	3.2924	-4.0764
25	16.065E-06	-12.907E-06	52.066E-06	2.8759	-1.9915	4.3735
26	74.021E-06	-8.0571E-06	-21.522E-06	16.129	2.3404	-1.8079
27	-4.9252E-06	-38.744E-06	116.13E-06	-3.2729	-8.9545	9.7550
28	89.983E-06	-12.907E-06	-19.290E-06	19.433	2.1478	-1.6203
29	42.287E-06	-382.67E-06	47.990E-06	-11.957	-83.350	4.0311
30	41.662E-06	-38.744E-06	-237.86E-06	7.1626	-6.3456	-19.981

By using expression (3.1), we find the relative error results as in the Tables 3.18 and 3.19

Table 3.18 Relative error of displacements and forces for second case between the results of the ABAQUS software and those calculated by the developed program

Nodes	*(e%) u*	*(e%) v*	*(e%) F/x-axis*	*(e%) F/y-axis*
Node 1	*0*	*0*	*0*	*0*
Node 2	*0*	*0*	*0*	*0*
Node 3	*0*	*0*	*0*	*0*
Node 4	*0*	*0*	*0*	*0*
Node 5	*0*	*0*	*0*	*0*
Node 6	*0*	*0*	*0*	*0*
Node 7	*0*	*0*	*0*	*0*
Node 8	*0*	*0*	*0*	*0*
Node 9	*0*	*0*	*0*	*0*
Node 10	*0*	*0*	*0*	*0*
Node 11	*0*	*0*	*0*	*0*
Node 12	*0*	*0*	*0*	*0*
Node 13	*0*	*0*	*0*	*0*
Node 14	*0*	*0*	*0*	*0*
Node 15	*0*	*0*	*0*	*0*
Node 16	*0*	*0*	*0*	*0*
Node 17	*0*	*0*	*0*	*0*
Node 18	*0*	*0*	*0*	*0*
Node 19	*0*	*0*	*0*	*0*
Node 20	*0*	*0*	*0*	*0*
Node 21	*0*	*0*	*0*	*0*
Node 22	*0*	*0*	*0*	*0*
Node 23	*0*	*0*	*0*	*0*
Node 24	*0*	*0*	*0*	*0*

Table 3.19 Relative error of stresses and strains for second case between the results of the ABAQUS software and those calculated by the developed program

Elements	(e%) strains			(e%) stresses		
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
Element 1	0	0	0	0	0	0
Element 2	0	0	0	0	0	0
Element 3	0	0	0	0	0	0
Element 4	0	0	0	0	0	0
Element 5	0	0	0	0	0	0
Element 6	0	0	0	0	0	0
Element 7	0	0	0	0	0	0
Element 8	0	0	0	0	0	0
Element 9	0	0	0	0	0	0
Element 10	0	0	0	0	0	0
Element 11	0	0	0	0	0	0
Element 12	0	0	0	0	0	0
Element 13	0	0	0	0	0	0
Element 14	0	0	0	0	0	0
Element 15	0	0	0	0	0	0
Element 16	0	0	0	0	0	0
Element 17	0	0	0	0	0	0
Element 18	0	0	0	0	0	0
Element 19	0	0	0	0	0	0
Element 20	0	0	0	0	0	0
Element 21	0	0	0	0	0	0
Element 21	0	0	0	0	0	0
Element 22	0	0	0	0	0	0
Element 23	0	0	0	0	0	0
Element 24	0	0	0	0	0	0
Element 25	0	0	0	0	0	0
Element 26	0	0	0	0	0	0
Element 27	0	0	0	0	0	0
Element 28	0	0	0	0	0	0
Element 29	0	0	0	0	0	0
Element 30	0	0	0	0	0	0

The results of the second case, similar to the first, showed complete agreement between the values of displacements, forces, and stresses and strains calculated using the developed program and the ABAQUS simulation results. The relative error between the results was zero for all results, indicating that the numerical model adopted in the program is accurate, making it a reliable tool for simulating and analyzing the behavior of plane structures using finite elements.

3.6 Validation result using RDM6 software

3.6.1 Overview of RDM6

RDM6 (Résistance Des Matériaux 6) is a French software developed to analyze the behavior of models under various loads. It is widely used in mechanical and civil engineering and is

adopted in educational sectors to train students in structural analysis and design. It's simple user interface makes it easy for students and researchers to study and understand the response of various elements such as beam, frames, and plates when subjected to complex loads. Its tools enable analysis of displacements, stresses, and strains, making it an effective tool for understanding and analyzing the stability of structures [43].

3.6.2 Main modules in RDM6

The RDM6 software contains 4 main modules designed to provide solutions for several types of analyses, allowing accurate and reliable results for analyzing the behavior of structures under various conditions. These modules include: frames, bending, rosette and finite elements. These modules and their functions can be explained below in Table 3.20 [43].

Table 3.20 Main modules in RDM6

Frames	*This module is designed for analyzing structural frameworks, including trusses and frames.*	
Bending	*Analyzes the behavior of beams under bending, a fundamental tool for studying structural elements subjected to flexural loads.*	
Rosette	*Focuses on analyzing strain gauge readings and calculating the resulting stresses.*	
Finite elements	*Analytical capabilities using the finite element method, in both 2D and 3D, to study the behavior of complex structures.*	

3.6.3 Comparing programming results with RDM6 results

In order to verify the accuracy of the program developed using MATLAB, the RDM6 software was used as a second validation tool. This was done after initial validation using ABAQUS, where the results were completely identical. This second comparison aims to assess the consistency of the results calculated in MATLAB and RDM6, particularly nodal displacements and reactions at the supports. Although RDM6 does not provide stress and strain computations, the consistency indicator of the validity of the adopted software model and the reliability of the analysis.

Consider a plate with a thickness of 10 mm, fixed on the left, subjected to a horizontal force of P=1000 N applied at two nodes on the right side of the plate, as shown in Fig. 3.33. The given data are: Young's modulus E=200000 MPa, Poisson's ratio $\nu=0.3$, and the stress type is plane stress. The objective of this study is to calculate the nodal displacements and reactions at the supports.

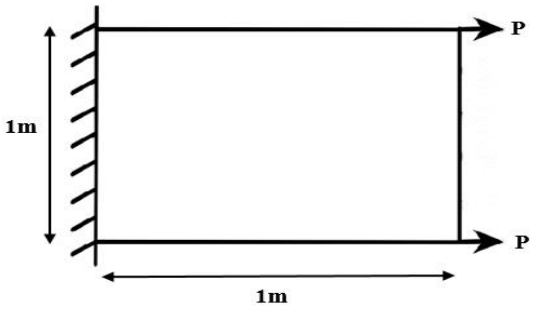


Fig. 3.30 Geometric model of a 2D flat plate

Using the program developed for element T3 and entering the data for this problem, the results were obtained as shown in Table 3.21.

Table 3.21 Results extracted from the developed calculation program for the plate

node	Ux	Uy	Fx	Fy
1	0.0000e+00	0.0000e+00	-3.3929e+02	-1.1905e+02
2	0.0000e+00	0.0000e+00	-3.1931e+02	6.0268e+01
3	1.9831e-03	-7.5917e-04	1.0000e+03	0.0000e+00
4	1.7996e-03	4.0849e-04	1.0000e+03	0.0000e+00
5	0.0000e+00	0.0000e+00	-4.4548e+02	4.5231e+01
6	0.0000e+00	0.0000e+00	-4.1048e+02	3.1086e+01
7	0.0000e+00	0.0000e+00	-4.8544e+02	-1.7537e+01
8	3.0177e-04	-4.9842e-05	0.0000e+00	0.0000e+00
9	6.4343e-04	-3.9891e-05	0.0000e+00	0.0000e+00
10	1.1349e-03	-1.1577e-04	0.0000e+00	0.0000e+00
11	7.7617e-04	-3.0743e-04	0.0000e+00	0.0000e+00
12	5.3022e-04	1.9087e-05	0.0000e+00	0.0000e+00
13	7.5788e-04	3.1650e-04	0.0000e+00	0.0000e+00
14	1.1050e-03	1.2669e-04	0.0000e+00	0.0000e+00
15	6.3338e-04	7.1931e-05	0.0000e+00	0.0000e+00
16	2.8632e-04	6.4112e-05	0.0000e+00	0.0000e+00
17	2.2003e-04	-5.7760e-06	0.0000e+00	0.0000e+00
18	4.3250e-04	-1.0196e-06	0.0000e+00	0.0000e+00
19	6.1282e-04	-3.8104e-05	0.0000e+00	0.0000e+00
20	1.8926e-04	-8.3729e-06	0.0000e+00	0.0000e+00
21	3.5124e-04	-1.6048e-05	0.0000e+00	0.0000e+00
22	4.4984e-04	-7.2219e-06	0.0000e+00	0.0000e+00
23	2.2376e-04	-7.1421e-07	0.0000e+00	0.0000e+00
24	4.4882e-04	-1.9432e-05	0.0000e+00	0.0000e+00
25	6.5151e-04	-5.7672e-07	0.0000e+00	0.0000e+00

To compare the extracted results with those provided by the RDM6 software for this plate, we perform various simulation steps:

1. **Initial configuration**

- Open RDM6 - Éléments finis.
- Specify the unit system in the program according to the units used (mm, N, MPa) and define the origin of the plate.

2. **Geometric model definition**

The geometric model was prepared by entering the coordinates of two points (x, y), and then defining a rectangle using these two points, which formed the basic geometric shape of the model (Fig. 3.31).

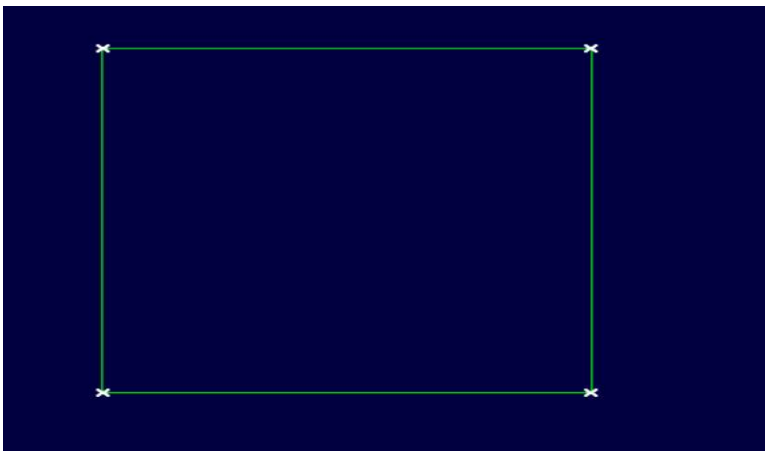


Fig. 3.31 Geometric model of a plate

3. Element mesh generation

The block mesh technique was used to divide the geometric model into blocks of triangular or quadrilateral elements, to facilitate the analysis process (Fig. 3.32).

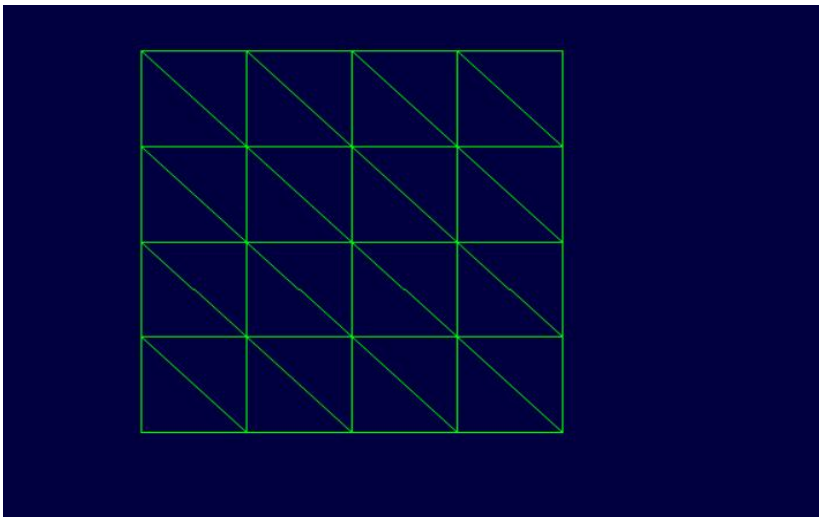


Fig. 3.32 Element mesh using block mesh technique

4. Determine the types of physical study

Choosing the appropriate physical condition for the problem. One can define “plane stress” to analyze thin plates, “plane strain” for long objects, or the “rotation problem” for objects with axial symmetry, etc. (Fig. 3.33).

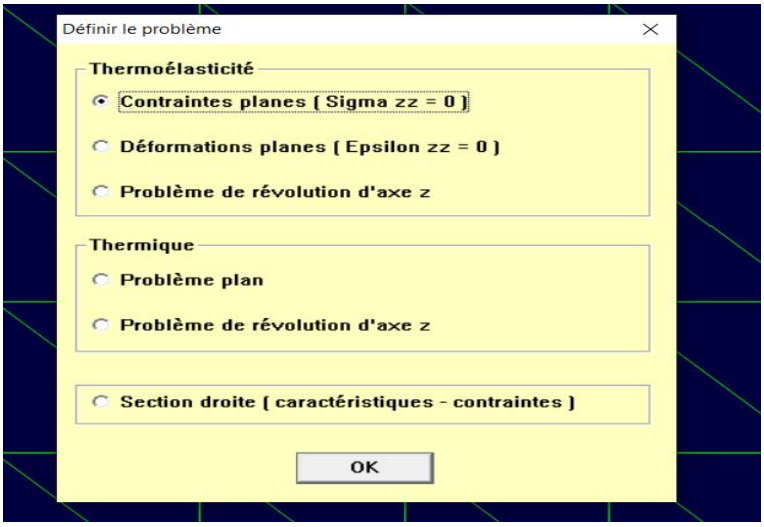


Fig. 3.33 Definition of study type

5. Material properties definition

Enter the mechanical properties of the material used in the model, such as Young's modulus, thickness and Poisson's ratio, which are necessary to analyze the mechanical response of the structure under the action of loads (Fig. 3.34).

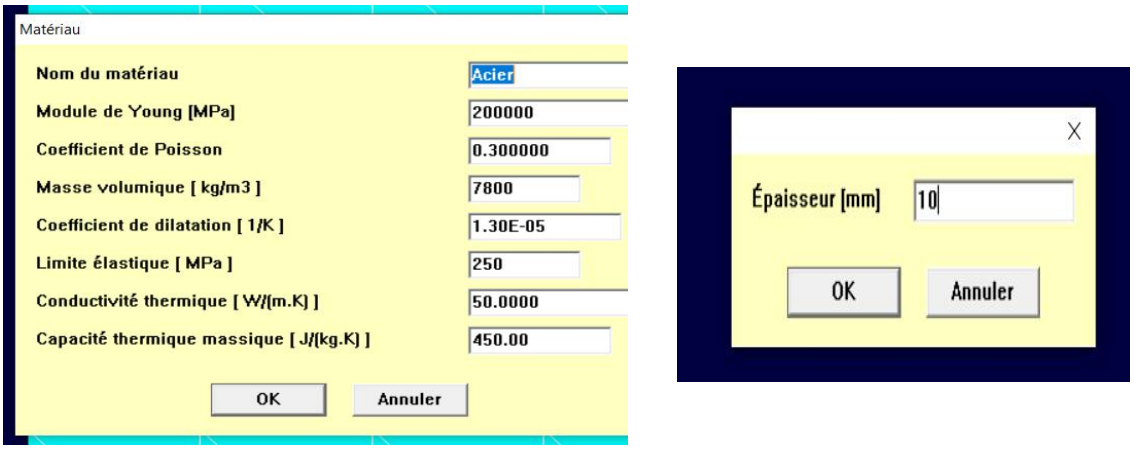


Fig. 3.34 Properties of the material used

6. Applying load and boundary conditions

Applying forces to the model, as well as constraints that limit the movement of certain nodes, such as fixation or movement in a specific direction. These constraints play a role in determining the system's response under the influence loads, allowing for the calculation of displacements and reactions (Fig. 3.35).

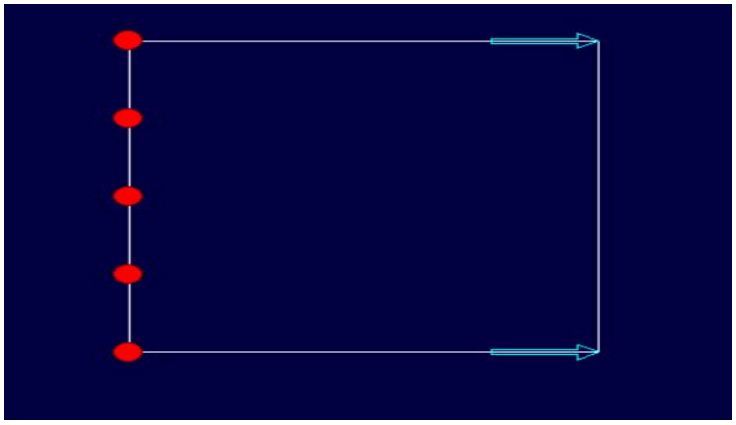


Fig. 3.35 Applying load and boundary conditions

7. Analyzing the results

Numerical analysis was performed using RDM6 software. The deformed plate under the influence of loads was visualized (Fig. 3.36), showing the displacement in different regions, helping to identify the areas experiencing the greatest displacement.

The results of calculations related to the reactions at the supports and the nodal displacements were also extracted (Fig. 3.37). These results illustrate the distribution of forces at the supports and the displacement values, enabling analysis of the structure's response to the applied loads.

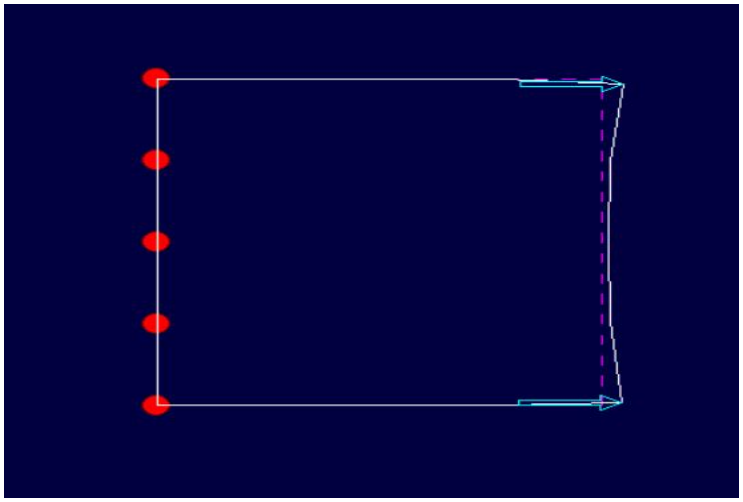


Fig. 3.36 Viewing the results

| Nodal displacements [mm] |

Node	Ux	Uy	Node	Ux	Uy	Node	Ux	Uy
1	0.0000E+00	0.0000E+00	2	0.0000E+00	0.0000E+00	3	1.9831E-03	-7.5917E-04
4	1.7996E-03	4.0849E-04	5	0.0000E+00	0.0000E+00	6	0.0000E+00	0.0000E+00
7	0.0000E+00	0.0000E+00	8	3.0177E-04	-4.9842E-05	9	6.4343E-04	-3.9891E-05
10	1.1348E-03	-1.1577E-04	11	7.7617E-04	-3.0743E-04	12	5.3022E-04	1.9087E-05
13	7.5788E-04	3.1650E-04	14	1.1049E-03	1.2669E-04	15	6.3338E-04	7.1931E-05
16	2.8632E-04	6.4112E-05	17	2.2003E-04	-5.7760E-06	18	4.3250E-04	-1.0196E-06
19	6.1282E-04	-3.8104E-05	20	1.8926E-04	-8.3729E-06	21	3.5124E-04	-1.6048E-05
22	4.4984E-04	-7.2219E-06	23	2.2376E-04	-7.1421E-07	24	4.4882E-04	-1.9432E-05
25	6.5151E-04	-5.7672E-07						

| Support reaction [N] |

Node	Rx	Ry	Node	Rx	Ry	Node	Rx	Ry
1	-339.29	-119.05	2	-319.31	60.27	3	1000.00	0.00
4	1000.00	0.00	5	-445.48	45.23	6	-410.48	31.09
7	-485.44	-17.54	8	0.00	0.00	9	0.00	0.00
10	0.00	0.00	11	0.00	0.00	12	0.00	0.00
13	0.00	0.00	14	0.00	0.00	15	0.00	0.00
16	0.00	0.00	17	0.00	0.00	18	0.00	0.00
19	0.00	0.00	20	0.00	0.00	21	0.00	0.00
22	0.00	0.00	23	0.00	0.00	24	0.00	0.00
25	0.00	0.00						

Fig. 3.37 Exporting the report results

The results obtained from the analysis using the RDM6 software are presented in Table 3.22, were these results show the values of displacements and reactions at various points of the studied model.

Table 3.22 Results of structural analysis of the model using RDM6 software

Nodes	Displacements (mm)		Forces (N)	
	u	v	F/x-axis	F/y-axis
Node 1	0	0	-339.29	-119.05
Node 2	0	0	-319.31	60.27
Node 3	1.9831×10^{-3}	-7.5917×10^{-4}	1000	0
Node 4	1.7996×10^{-3}	4.0849×10^{-3}	1000	0
Node 5	0	0	-445.48	45.23
Node 6	0	0	-410.48	31.09
Node 7	0	0	-485.44	-17.54
Node 8	3.0177×10^{-4}	-4.9842×10^{-5}	0	0
Node 9	6.4343×10^{-4}	-3.9891×10^{-5}	0	0
Node 10	1.1348×10^{-3}	-1.1577×10^{-4}	0	0
Node 11	7.7617×10^{-4}	-3.0743×10^{-4}	0	0
Node 12	5.3022×10^{-4}	1.9087×10^{-5}	0	0
Node 13	7.5788×10^{-4}	3.1650×10^{-4}	0	0
Node 14	1.1049×10^{-3}	1.2669×10^{-4}	0	0
Node 15	6.3338×10^{-4}	7.1931×10^{-5}	0	0
Node 16	2.8632×10^{-4}	6.4112×10^{-5}	0	0
Node 17	2.2003×10^{-4}	-5.7760×10^{-6}	0	0
Node 18	4.3250×10^{-4}	-1.0196×10^{-6}	0	0
Node 19	6.1282×10^{-5}	-3.8104×10^{-5}	0	0
Node 20	1.8926×10^{-4}	-8.3729×10^{-6}	0	0
Node 21	3.5124×10^{-4}	-1.6048×10^{-5}	0	0
Node 22	4.4984×10^{-4}	-7.2219×10^{-6}	0	0
Node 23	2.2376×10^{-4}	-7.1421×10^{-7}	0	0
Node 24	4.4882×10^{-4}	-1.9432×10^{-5}	0	0
Node 25	6.5151×10^{-4}	-5.7672×10^{-7}	0	0

In order to compare between these results, the relative difference was calculated using formula (3.1), and presented in Table 3.23.

Table 3.23 Relative error between the results of the proposed program and those extracted from the RDM6 software for the studied structure

Nodes	*(e%) u*	*(e%) v*	*(e%) F/x-axis*	*(e%) F/y-axis*
Node 1	*0*	*0*	*0*	*0*
Node 2	*0*	*0*	*0*	*0.003*
Node 3	*0*	*0*	*0*	*0*
Node 4	*0*	*0*	*0*	*0*
Node 5	*0*	*0*	*0*	*0.002*
Node 6	*0*	*0*	*0*	*0.01*
Node 7	*0*	*0*	*0*	*0.01*
Node 8	*0*	*0*	*0*	*0*
Node 9	*0*	*0*	*0*	*0*
Node 10	*0.008*	*0*	*0*	*0*
Node 11	*0*	*0*	*0*	*0*
Node 12	*0*	*0*	*0*	*0*
Node 13	*0*	*0*	*0*	*0*
Node 14	*0.009*	*0*	*0*	*0*
Node 15	*0*	*0*	*0*	*0*
Node 16	*0*	*0*	*0*	*0*
Node 17	*0*	*0*	*0*	*0*
Node 18	*0*	*0*	*0*	*0*
Node 19	*0*	*0*	*0*	*0*
Node 20	*0*	*0*	*0*	*0*
Node 21	*0*	*0*	*0*	*0*
Node 22	*0*	*0*	*0*	*0*
Node 23	*0*	*0*	*0*	*0*
Node 24	*0*	*0*	*0*	*0*

Table 3.23 represents the relative error between the results of the proposed program and those of the RDM6 software. We observe that the displacements error at node 10 is 0.008%, while at node 11 it was 0.009%. Regarding the forces, the error at node 2 is 0.003%, at node 5 it was 0.002%, and at node 6 and 7 it is 0.01% each. By adding all these partial errors together gives a total relative error of 0.042%. This value is very small compared to 1%. Therefore, it can be said that the results obtained by the proposed program are in strong agreement with those of RDM6, which strengthens the reliability of the algorithm used and the validity of the mathematical modeling.

3.7 Conclusion

In this chapter, a numerical program for the analyzing planar structures using the finite element method was developed in MATLAB. The program's validity is initially verified by comparing its results with manual calculations for a simple case. Subsequently, ABAQUS and RDM6 software were used to validate the results for more complex cases, testing the program's accuracy in representing the mechanical responses of structures. Finally, the error difference

between the results is calculated. It is found that the proposed program performs well and is based on an efficient algorithm, which enhances its reliability in finite element applications.

General Conclusion

In conclusion, this dissertation focused on the topic of analyzing the behavior of plane structures, which is an important component of applications in mechanical and civil engineering. Despite the simplicity of the geometric model, addressing this type of problem requires precise numerical modeling and a good understanding of mechanical properties. To simplify this process, a program was developed using MATLAB that allows for the analysis of first-order triangular element (T3). This program has proven its ability to calculate basic variables and contribute to accelerating analysis processes associated with the finite element method.

To ensure the reliability and accuracy of the developed program, a validation process was conducted using three different approaches. First, the results were compared with a manual solution for a simple case, which showed complete agreement. Then, tested using ABAQUS software on two more complex numerical examples, and the results were found to be perfectly consistent with those form MATLAB, with an error rate of 0%. Finally, a third example was validated using RDM6, with a total error rate of approximately 0.042%, confirming the accuracy proposed numerical approach. These results enhance the effectiveness of the developed tool and its potential for application in various fields such as engineering formation and educational simulation. It also helps students and engineers focus on studying the behavior of the system instead of being distracted by complex calculations that delay understanding the problem.

This work is not the end, but rather represents a first step toward expanding the program in the future, whether by using more precise elements such as the second-order triangular element (T6) or by studying the nonlinear behavior of structures, thus contributing to the advancement of numerical analysis tools in the field of applied mechanics.

References

[1] A. Seghir, Cours : Méthode des Éléments Finis, Université Abderrahmane Mira - Bejaia, Faculté de Technologie, Département de Génie Civil, 2014.

[2] A. Hrennikof, "Solution of Problems of Elasticity by the Framework Method," J. Appl. Mech., vol. 8, no. 4, pp. A169-A175, 1941. [Online]. Available: https://asmedigitalcollection.asme.org/appliedmechanics/article/8/4/A169/1101846/ .

[3] D. McHenry, "Limit Analysis of Rotationally Symmetric Plates," J. Appl. Mech., vol. 10, no. 1, pp. A31–A37, 1943.

[4] N. M. Newmark, "An Approximate Method of Computation for Structural Dynamics," Proc. ASCE, vol. 73, no. 6, pp. 905–922, 1947. [Online]. Available: https://engineering.purdue.edu/~ce573/Documents/Newmark_A%20Method%20of%20Computation%20for%20Structural%20Dynamics.pdf .

[5] J. H. Argyris and S. Kelsey, "Energy Theorems and Structural Analysis," Aircraft Eng. Aero. Tech., vol. 27, no. 10, pp. 347–356, 1955.

[6] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp, "Stiffness and Deflection Analysis of Complex Structures," J. Aeronaut. Sci., vol. 23, no. 9, pp. 805–823, 1956. [Online]. Available: https://docslib.org/doc/11249201/ .

[7] Lord Rayleigh, The Theory of Sound, vol. 1. London: Macmillan, 1877. [Online]. Available: https://archive.org/details/theorysound06raylgoog [accessed: May. 19, 2025].

[8] W. Ritz, "Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik," J. Reine Angew. Math., vol. 135, pp. 1–61, 1909. [Online]. Available: https://www.degruyterbrill.com/document/doi/10.1515/crll.1909.135.1/html .

[9] C. F. Gauss, Methodus nova integralium valores per approximationem inveniendi. Manuscript, 1795. [Online]. Available: https://www.math.unipd.it/~alvise/PHD_2024/LEZIONI/GAUSSIAN/PDF/Methodus_nova_integralium_valores_per.pdf [accessed: May. 19, 2025].

[10] B. G. Galerkin, "Series Solution of Some Problems of Elastic Equilibrium of Rods and Plates," Vestnik Inzhenerov i Tekhnikov, vol. 19, pp. 897–908, 1915.

[11] C. B. Biezeno and H. Koch, Technische Mechanik, Springer, 1923.

[12] L. F. Richardson, "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations," Phil. Trans. R. Soc. Lond. A, vol. 210, pp. 307–357, 1911. [Online]. Available: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1911.0009 .

[13] H. Liebmann, Die numerische Lösung der partiellen Differentialgleichungen, Teubner, 1918.

[14] R. V. Southwell, Relaxation Methods in Engineering Science, Oxford Univ. Press, 1946.

[15] R. Courant, “Variational Methods for the Solution of Problems of Equilibrium and Vibrations,” Bull. Amer. Math. Soc., vol. 49, pp. 1–23, 1943. [Online]. Available: https://www.ams.org/journals/bull/1943-49-01/S0002-9904-1943-07818-4/ .

[16] W. Prager and J. L. Synge, “Approximations in Elasticity Based on the Concept of Function Space,” Q. Appl. Math., vol. 5, no. 3, pp. 241–269, 1947. [Online]. Available: https://www.jstor.org/stable/43633616 .

[17] O. C. Zienkiewicz, The Finite Element Method in Structural and Continuum Mechanics, New York: McGraw-Hill, 1967. [Online]. Available: https://archive.org/details/finiteelementmet0000zien_c3j7 [accessed: May. 19, 2025].

[18] R. S. Varga, Matrix Iterative Analysis, Englewood Cliffs: Prentice-Hall, 1962. [Online]. Available: https://archive.org/details/matrixiterativea0000varg [accessed: May. 19, 2025].

[19] O. C. Zienkiewicz and R. L. Taylor, The Finite Element Method: Volume 1 - The Basis, 5th ed., Butterworth-Heinemann, 2000.

[20] R. D. Cook, D. S. Malkus, and M. E. Plesha, Concepts and Applications of Finite Element analysis, University of Wisconsin-Madison, John Wiley & Sons, 1988.

[21] Poly-Cours : Méthodes des Éléments Finis, Université d’Ain Témouchent, Faculté des Sciences et Technologie, Département de Génie Mécanique.

[22] F. Bourada, Cours : Méthode des Éléments Finis. Centre universitaire Ahmed Ben Yahia El-Wancharissi, Tissemsilt, Institut des Sciences et de la Technologie, Département des Sciences et de la Technologie.

[23] Y. Liu, Lecture Notes: An Introduction to the Finite Element Method, University of Cincinnati, 2015.

[24] J. N. Reddy, An Introduction to the Finite Element Method, McGraw-Hill Education, 2019.

[25] T. R. Chandrupatla and A. D. Belegundu, Introduction to Finite Elements in Engineering, 4th ed., Pearson Education, 2011.

[26] O. C. Zienkiewicz, R. L. Taylor and J. Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, 6th ed., Oxford, Butterworth-Heinemann, 2005.

[27] A. Khennane, Introduction to Finite Element Analysis Using MATLAB and ABAQUS, LLC by Taylor and Francis Group, 2013.

[28] R. Alubady, Programming Fundamentals, University of Babylon, College of Information Technology, Information Network, Department-First Class, Fourth Lecture.

[29] A. Chen and P. Tsai, Basic Principals of Programming Languages, 4th ed., Chapter1.

[30] I. Sommervielle, Software Engineering, 10th ed., Harlow, England: Pearson, 2015.

[31] M. Yoann, Les langages de programmation, Mémoire académique, 2022 - 2023

[32] M. Ben-Ari, Understanding Programming Languages, Weizmann Institute of Science, published by John Wiley & Sons, Chichester, 1996. Online edition, 2006.

[33] K. C. Louden and K. A. Lambert, Programming Languages: Principles and Practice, Louden. 3rd ed.

[34] R. W. Sebesta, Concepts of Programming Languages, 11th ed., Global Edition contributions by S. Mukherjee and A. K. Bhattacharjee, University of Colorado at Colorado Springs, Pearson, 2016.

[35] H. Fangohr. A Comparison of C, MATLAB and Python as Teaching Languages in Engineering. University of Southampton, UK.

[36] D. Hanselman and B. Littlefield, Mastering MATLAB 7, Prentice Hall, 2004.

[37] S. J. Chapman, MATLAB Programming for Engineers, 6th ed., Cengage Learning, 2019.

[38] D. Houcque, Introduction to MATLAB for Engineering Students. Northwestern University, version 1.2, August. 2005.

[39] S. De, ABAQUS Handout, MANE 4240/CIVL 4240: Introduction to Finite Element, Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute.

[40] K. Meftah, Cours : Introduction générale au logiciel ABAQUS. Université de Biskra, 2013.

[41] P. Malik and S. K. Mishra, “ABAQUS – An introduction to Finite Element Software”, International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 10, pp. 333 - 347, Oct. 2021.

[42] M. Ait Taleb Aïssa, Modélisation du comportement de structures tubulaires traitées et non traitées, Mémoire de Master, Université Mouloud Mammeri de Tizi-Ouzou, Faculté de Génie Mécanique et de Génie de Procédés, 2016.

[43] Y. Debard, RDM Version 6 – Résistance des Matériaux : Calcul des structures par la méthode des éléments finis, IUT Le Mans, France, Jan. 9, 2009. [Online]. Available: https://www.enseignementsup-recherche.gouv.fr/ressources-pedagogiques/notice/view/oai%253Awww.unit.eu%253Aunit-ori-wf-1-3135 [accessed: May. 3, 2025].

Annexes

The following lines represent the user interaction sequence as it appears in the MATLAB command window during the execution of the finite element program. Each message is accompanied by a clear instruction on what the user is required to enter or define.

When the program starts, the following message is displayed:

1) **Young's Modulus:** Enter its value in MPa.
2) **Poisson's ratio:** Enter the value of the Poisson's ratio.
3) **thickness:** Enter the thickness of the structure in mm.
4) **Enter 1 or 2:** Type 1 for plane stress and 2 for plane strain.
5) **Number of nodes:** Enter the total number of nodes in the finite element mesh.
6) **Coordinates for each node [x y]:** Enter the coordinates of each node in the format [x, y], were x and y represent the position of the node in the Cartesian plane.
7) **Number of elements:** Enter the total number of triangular elements.
8) **Node connectivity for each element [n1 n2 n3]:** Define the connectivity of each element by specifying the numbers of the three nodes that form each triangular element in the following format [n1 n2 n3] were n1: is the number of the first node, n2: the number of the second node, and n3: the number of the third node.
9) **How many boundary condition are known?** Enter the total number of known displacements (e.g., fixing both x and y at a node counts as 2).

   BC: [DOF value], where:

   Value: Represent the displacement imposed at a specified DOF.

   DOF: Represent the degree of freedom for each coordinate is calculated as follows:

- DOF= 2 * number of node – 1, for displacement in x-direction at node.
- DOF= 2 * number of node, for displacement in y-direction at node.

10) **Are there any distribution load? (Yes=1, No=2):** Enter 1 if a distribution load is applied, or 2 if not.
11) **Are there any concentrated forces? (Yes/No):** Enter "Yes" if a concentrated force is applied, or "No" if not.
12) **number of Nodes were applied the load:** Enter the total number of nodes to which an external force was applied.
13) **Node number:** Enter the number of node were applied the force.
14) **force in x-direction:** Enter its value in the x-direction.
15) **force in y-direction:** Enter its value in the x-direction.