FINITE ELEMENT METHOD AND POLYNOMIAL INTERPOLATION IN STRUCTURAL ANALYSIS

by

Patrick Vaugrante

Engineer, Ecole Nationale Superieure de Mecanique, France, 1980

M.Sc., Laval University, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Mathematics and Statistics

SIMON FRASER UNIVERSITY

APRIL 1985

APPROVAL

Name: Patrick Vaugrante

Degree: Master of Science

Title of Thesis: Finite Element Method and Polynomial Interpolation in Structural Analysis

Examining Committee:

Chairman: Professor C. Villegas

Dr. C.Y. Shen
Senior Supervisor

Dr. E. Pechlaner

Dr. M. Singh

Dr. G. Bader
External Examiner
Visiting Assistant Professor
Department of Mathematics and Statistics
Simon Fraser University

Date approved: April 10, 1985

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

Finite Element Method And Polynomial Interpolation in Structural Analyses

Author: _____
(signature)

VAUGRANTE
(name)

15/04/85
(date)

ABSTRACT

This thesis presents a numerical study of complex structural deformations via the example of Lee's frame. The structure of Lee's frame is discretized into 1-dimensional straight beam elements, with each element having two nodes and eight degrees of freedom. The deformation is assumed to take place in a plane, and the vertical and horizontal displacements of the nodes are approximated by cubic polynomials. The finite element method reduces this problem into solving a system of nonlinear algebraic equations; in the matrix form, it can be expressed as

$$[K]\{U\} = \{F\}$$

where $[K]$ is the global stiffness matrix of the system, $\{U\}$ the global vector of nodal variables, and $\{F\}$ the global load vector. To arrive at these equations, the integrals brought by the finite element method are evaluated by using the Gaussian quadrature method; and to solve these equations, the Newton-Raphson's method together with the polynomial interpolation method are employed. The polynomial interpolation method has not been extensively used in structural analysis, and its utilization here speeds up the accuracy of the Newton-Raphson's method. All these methods are implemented in an APL program, and the load-displacement diagram for the Lee's frame is obtained.

ACKNOWLEDGEMENTS

I especially wish to express my deep gratitude to my senior supervisor, Dr. C.Y. Shen, for his generous assistance.

I would also like to thank Dr. B. Russell, who gave me the opportunity to come to S.F.U., and Dr. G. Bader for his friendly advice.

Additional thanks go to the staff of the S.F.U. Department of Mathematics and Statistics and my colleagues, especially H.S. Kim, for their kindness.

Finally, I extend special thanks to Ms. Cindy Lister for her excellent work in typing this thesis.

TABLE OF CONTENTS

	Page
APPROVAL	(ii)
ABSTRACT	(iii)
ACKNOWLEDGEMENTS	(iv)
TABLE OF CONTENTS	(v)
LIST OF FIGURES	(ix)
LIST OF TABLES	(x)
SYMBOLS	(xi)
CHAPTER 1 - INTRODUCTION	1
1.1 Generalities	1
1.2 The Lee's frame	2
1.3 Numerical methods	2
1.4 Purpose, method and plan	4
CHAPTER 2 - MECHANICAL ANALYSIS	5
2.1 Generalities	5
2.2 Notation	5
2.3 Kinematic relations	6
2.3.1 Kinematic equations for a body	6
2.3.1.1 Expression of the deformation gradient tensor	6
2.3.1.2 Expression of the Cauchy strain tensor	9
2.3.1.3 Expression of the Green-Lagrange strain tensor	9
2.3.2 Kinematic relations for a point of a beam element	10
2.3.2.1 Calculation of a displacement vector of a point of a beam	11

Page

2.3.2.2 Deformation gradient tensor 16

2.3.2.3 Green-Lagrange strain tensor 16

2.3.2.3.1 First expression of the components of the tensor [ε] . . . 16

2.3.2.3.2 Second expression of the components of the tensor [ε] . . . 17

2.3.2.3.3 Third expression of the components of the tensor [ε] . . . 18

2.3.2.3.4 Fourth expression of the components of the tensor [ε] . . . 18

2.4 Differential equilibrium equations 19

2.4.1 Stress tensors 19

2.4.1.1 Cauchy stress tensor 20

2.4.1.2 1st and 2nd Piola-Kirchhoff stress tensor . . 22

2.4.1.3 Equilibrium equations 23

2.5 Expression of the virtual work 24

2.5.1 Expression of the virtual work for a body 24

2.5.1.1 Boundary conditions 24

2.5.1.2 Field of kinematically admissible virtual displacements 26

2.5.1.3 Expression of the virtual work in the deformed configuration 26

2.5.1.4 Expression of the virtual work with respect to the original configuration 28

2.5.1.4.1 Lagrangian formulation of the virtual work principle 29

2.5.2 Application of the virtual work principle to a beam . 31

2.5.2.1 Expression of the total internal work for a beam 32

Page

CHAPTER 3 - APPLICATION OF F.E.M. AND GAUSSIAN QUADRATURE TO BEAM STRUCTURES 35

3.1 Generalities 35

3.2 Finite Element Method 35

3.2.1 Introduction to the F.E.M. 36

3.2.2 Type of element 37

3.2.3 Interpolation functions 40

3.2.4 Discretized expressions 42

3.2.4.1 Calculation of the residual vector of an element V^e 45

3.2.4.2 Calculation of the tangent matrix of an element V^e 46

3.2.4.3 Calculation of the external work 48

3.2.4.4 Gathering of the discretized variables . . . 48

3.3 Gaussian quadrature 50

3.3.1 Legendre polynomials 51

3.3.2 Evaluation of the variables 51

CHAPTER 4 - NUMERICAL METHODS FOR SOLVING NONLINEAR SYSTEMS 53

4.1 Generalities 53

4.2 Solution of the linear system 54

4.2.1 Boundary conditions 54

4.2.2 Storage of the matrix $[K_T]$ 56

4.2.3 Block L-U factorization 56

4.3 Iterative process 58

4.3.1 Newton-Raphson algorithm 58

4.3.2 Imposition of a displacement 61

Page

4.4 Automatic incrementation 64

4.4.1 Choice of the increment 66

4.4.1.1 Polynomial interpolation 66

4.4.2 Choice of the stepsize 69

4.4.3 General improvements 71

4.4.3.1 Approximation of extrema 71

4.4.3.2 Prediction of the solution 72

CHAPTER 5 - RESULTS 74

5.1 Generalities 74

5.2 Flowcharts 74

5.2.1 Introduction 74

5.2.2 Flowchart of the main program 77

5.2.3 Flowchart of the "BLOCK RESOLUTION" 78

5.2.4 Flowchart of the "BLOCK DIVERGE" 79

5.2.5 Flowchart of the "BLOCK PREDICT" 80

5.2.6 Flowchart of the "BLOCK EXTREMA" 81

5.2.7 Flowchart of the "BLOCK INTERPOLATION" 82

5.2.8 Flowchart of the "BLOCK STEPSIZE" 83

5.3 Presentation of the results 84

5.3.1 Load-displacement curves 84

5.3.2 Comparison of procedures 89

5.3.2.1 Prediction in Newton-Raphson method 90

5.3.2.2 Comparison of the order of the polynomial interpolation 91

CHAPTER 6 - CONCLUSION 98

REFERENCES 100

LIST OF FIGURES

Figure		Page
1.1	Lee's frame	3
2.1	Deformation of a two-dimensional body	7
2.2	Thin rectangular beam before and after planar deformation	12
2.3	Deformation of the middle axis in the plane X-Z	12
2.4	Beam undergoing deformation	14
2.5	Rotation of the middle axis	14
2.6	Rotation of the normal section	14
2.7	Plane stress element	21
2.8	A two-dimensional body in equilibrium	21
3.1	Real element V^e.	38
3.2	Reference element V^r	38
4.1	Newton-Raphson's algorithm	59
4.2	Several deformations for one load	62
4.3	Displacements vs load	65
4.4	Choice of the increment	67
5.1	Different slopes of the Lee's frame	87
5.2	Load-displacement curves	88

LIST OF TABLES

Table		Page
3.1	Interpolation function	43
3.2	First derivative of the interpolation function	43
3.3	Second derivative of the interpolation function	44
5.1	Extrema of the curves load-displacement	85
5.2	Variations of F, u_p and w_p	86
5.3	Extrema for SMAX = 0.025	92
5.4	Extrema for SMAX = 0.05	93
5.5	Extrema for SMAX = 0.1	94
5.6	Extrema for SMAX = 1	95

SYMBOLS

$\forall$	For every
$\mathbb{N}$	set of positive integers
$\mathbb{R}$	set of real numbers
$\subset$	inclusion
$\in$	adherence
$\int$	integral sign
Σ	summation sign
$\| \ \|$	norm
$[\ ,\ ]$	closed interval
δ, Δ	variational operators
∂	partial derivative
$\simeq$	approximately equal
$\cup$	union
$[\ ]$	A matrix
$[\ ]^T$	the transpose of a matrix
$[\ ]^{-1}$	the inverse of a matrix
tr[ ]	the trace of a matrix
< >	row vector
{ }	column vector
F.E.M.	finite element method
Fig.	figure
Tab.	table
eq.	equation
det.	determinant

CHAPTER 1

INTRODUCTION

§1.1 GENERALITIES

The fast and unceasing evolution of digital computers in recent years has permitted a rapid increase in the use of numerical methods for solving partial differential equations governing the behaviour of solids and fluids. One of such numerical methods which has gained tremendous popularity for solving structural problems is the Finite Element Method (F.E.M.). In fact, its popularity has spilled over to other scientific fields such as heat transfer, fluid dynamics, etc.

For many structural problems, the deformation of the structure when it is submitted to external loads can be very complex, and even the numerical solution can be difficult to obtain by using the F.E.M. alone. And for such problems, one needs to deploy several numerical methods simultaneously. In this thesis we have carried out a numerical study of such complex problems. Although our numerical scheme is quite general and can be adopted to solve various complex structural problems, we elect to illustrate our method via the example of Lee's frame.

The physical and geometrical characteristics of the Lee's frame is described in the next section. In §1.3, we shall give a brief description of the main numerical methods used in this thesis, which will be followed, in §1.4, a summary of the mechanical formulation plus a plan of presentation.

§1.2 THE LEE'S FRAME

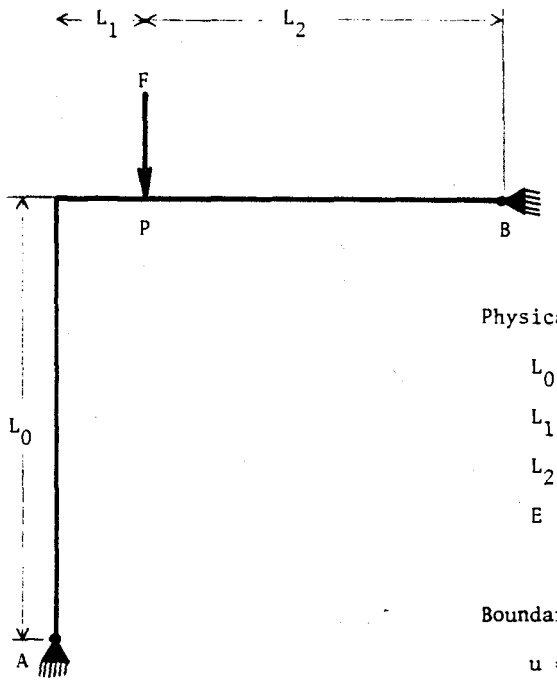
The Lee's frame is shown in the figure (1.1), and is submitted to a vertical load F at the point P. The two extremities of the frame are fixed and only allow rotations. The cross section of the Lee's frame is constant.

§1.3 NUMERICAL METHODS

The finite element method will be used to discretize the Lee's frame into beam elements. These elements will be modeled by a 1-dimensional reference element which has two nodes and eight degrees of freedom. The vertical and horizontal displacements at any point will be approximated by cubic polynomials. As a result, the mechanical behaviour of the Lee's frame can be characterized by a system of nonlinear algebraic equations:

$$[K]\{U\} = \{F\} \tag{1.1}$$

where $[K]$ is the global stiffness matrix of the Lee's frame, $\{U\}$ the global vector of nodal variables, and $\{F\}$ the global load vector. Before one arrives at the equation (1.1), the finite element method introduces a lot of integrals, and these integrals will be solved by using the Gaussian quadrature method. The nonlinear equation (1.1) will be solved by iterations, with the help of Newton-Raphson's method. Since it is possible to obtain several displacements for the same load, one needs to fix a displacement instead of the load. In the particular case of Lee's frame, the choice of this displacement must be made with care; the polynomial interpolation method will determine the right choice. The polynomial



Physical and geometrical characteristics

L_0 = 1.20m

L_1 = 0.24m

L_2 = 0.96m

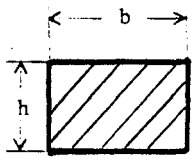
E = elasticity modulus = 720 $10^4 N/m^2$

Boundary conditions in A and B

u = w = 0

u = horizontal displacement

w = vertical displacement



SECTION OF LEE'S FRAME IN P

Thickness = h = 0.02m

Width = b = 0.03m

Fig. 1.1 - Lee's frame

interpolation method will also be used to speed up the convergence of the Newton-Raphson's method.

§1.4 PURPOSE, METHOD AND PLAN

The purpose of this thesis is to create an A.P.L. program which will completely determine the deformations of the Lee's frame. In order to apply the numerical methods, we must first obtain a mechanical formulation of the Lee's frame. In our analysis, we will follow all the particles of the frame in their motion, from the original to the final configuration of the frame. Thus we have adopted a total Lagrangian formulation. We shall calculate the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor, and shall utilize the principle of virtual displacement. Several other mechanical hypotheses will be used in this thesis including the Euler-Bernoulli hypothesis, plane stress hypothesis, and the existence of an elastic behaviour law.

The general plan of the thesis will be the following one:

- Mechanical Analysis
- Application of F.E.M. and Gaussian Quadrature to Beam Structures
- Numerical Methods for Solving Nonlinear Systems
- Flowcharts and Results
- Conclusion

CHAPTER 2

MECHANICAL ANALYSIS

§2.1 GENERALITIES

The purpose of this chapter is to present a theory concerning the bending of a beam. Many authors, e.g. Timoshenko [1] and Wang [2], studied the deformations of beams. One interesting mathematical publication, written by Sayir and Mitropoulos [3], provides several formulations depending on the approximations used.

In our case we will consider isotropic beams subjected to the action of planar forces. Therefore the calculations will be done in a two dimensional plane.

This chapter will be divided into two main parts. The first part establishes the governing differential equations of a two dimensional body in equilibrium and the second part involves the application of these equations to a beam element. We will first define some necessary notation.

§2.2 NOTATION

We will call configuration Γ^* the position of the set of points of a given body at the time t^*. Under a deformation, the trajectories of these points can be described in the vector form:

$$\begin{Bmatrix} x^*(P,t^*) \\ z^*(P,t^*) \end{Bmatrix} = \begin{Bmatrix} x^*(x,z,t^*) \\ z^*(x,z,t^*) \end{Bmatrix} \tag{2.1}$$

where x^* and z^* are the components of the point P^* at t^*. At $t = 0$ the position of P^* is given by P which has x and z for coordinates (Fig. 2.1). The coordinate system $(o;\vec{e}_1;\vec{e}_2)$ will be used for all configurations.

Throughout this thesis, the time t^* will be associated with the configuration after deformation. Therefore all the variables with the superscript "*" will be considered as belonging to the deformed configuration. The variables without the superscript "*" will be referred to the body before deformation. The equation (2.1) shows the relation between the initial configuration Γ and the deformed configuration Γ^*.

§2.3 KINEMATIC RELATIONS

We will establish in this paragraph the expression for the deformation gradient tensor [F], the Cauchy strain tensor [ε^*], and the Green-Lagrange strain tensor [ε]. The following developments can be found in Batoz [4].

2.3.1 Kinematic equations for a body

2.3.1.1 Expression of the deformation gradient tensor

Let x^* and z^* be the coordinates of the point P^* which corresponds to the point $P(x,z)$ in the initial configuration. The

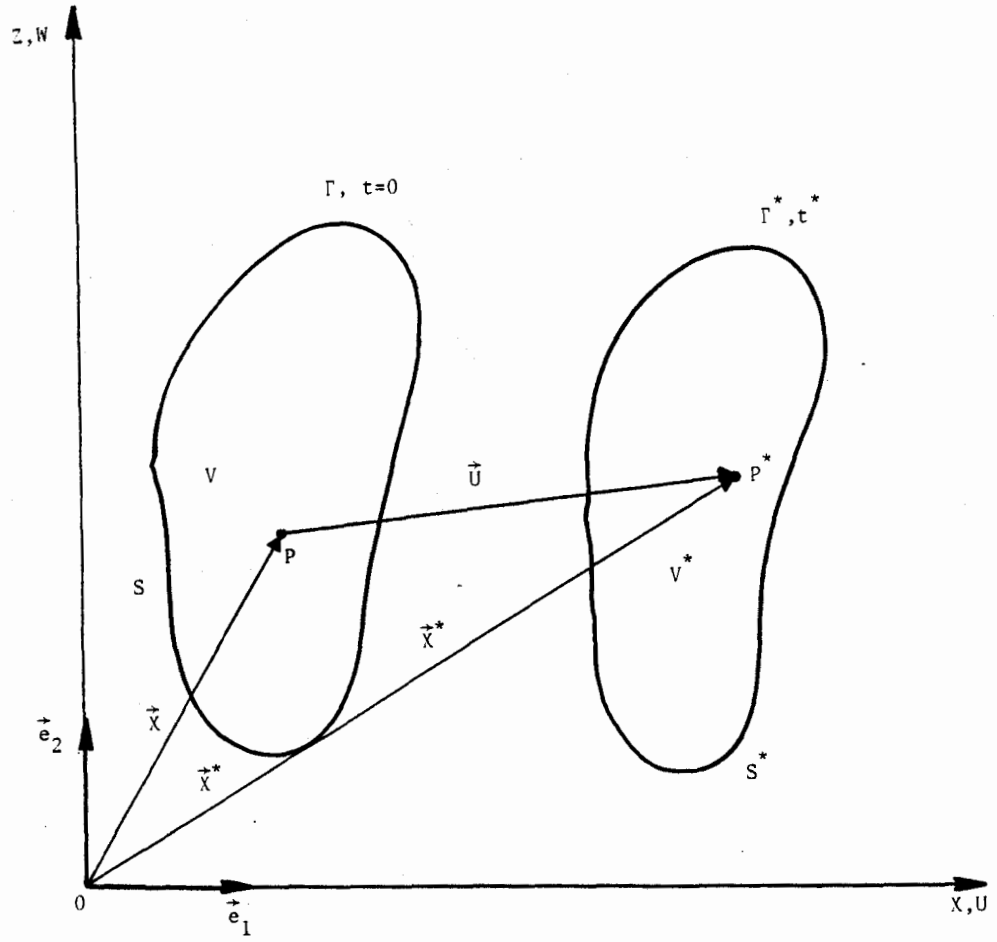


Fig. 2.1 - Deformation of a two dimensional body

components of the displacement vector $\overrightarrow{PP^*}$ are u and w. Thus,

$$\begin{Bmatrix} x^*(x,z) \\ z^*(x,z) \end{Bmatrix} = \begin{Bmatrix} x \\ z \end{Bmatrix} + \begin{Bmatrix} u(x,z) \\ w(x,z) \end{Bmatrix} .$$

Therefore,

$$\begin{Bmatrix} dx^* \\ dz^* \end{Bmatrix} = [F] \begin{Bmatrix} dx \\ dz \end{Bmatrix} \tag{2.2}$$

with

$$[F] = \begin{bmatrix} 1 + u_{,x} & u_{,z} \\ w_{,x} & 1 + w_{,z} \end{bmatrix} \tag{2.3}$$

and

$$u_{,x} = \frac{\partial u}{\partial x} , \quad \text{etc.}$$

The matrix [F] is called the deformation gradient tensor; its determinant is different of zero; and

$$dV^* = \det[F]dV$$

where dV^* is the volume element associated with the final configuration Γ^* and dV with the original configuration Γ.

2.3.1.2 Expression of the Cauchy strain tensor

The Cauchy strain tensor is defined by:

$$[\varepsilon^*] = \begin{bmatrix} \varepsilon^*_{xx} & \frac{1}{2}\varepsilon^*_{zx} \\ \frac{1}{2}\varepsilon^*_{xz} & \varepsilon^*_{zz} \end{bmatrix}$$

with $$\varepsilon^*_{xx} = \frac{\partial u^*}{\partial x^*}$$

$$\varepsilon^*_{zz} = \frac{\partial w^*}{\partial z^*}$$

and $$\varepsilon^*_{zx} = \varepsilon^*_{xz} = \frac{\partial u^*}{\partial z^*} + \frac{\partial w^*}{\partial x^*}$$

where u^* and w^* are the components of the displacement vector as functions of point P^* in the configuration Γ^*.

We can define a vectorial representation $<\varepsilon^*>$ of the components of $[\varepsilon^*]$ as

$$<\varepsilon^*> = <\varepsilon^*_{xx} \quad \varepsilon^*_{zz} \quad \varepsilon^*_{zx}> \quad . \tag{2.4}$$

2.3.1.3 Expression of the Green-Lagrange strain tensor

The difference $ds^{*2} - ds^2$ of the square of the infinitesimal length between two points P^* and $P^{*\prime}$ in the deformed

configuration Γ^* and P and P' in the configuration Γ can be used as a measure for the deformation which occurred in the neighbourhood of P between the times t and t^*. Since,

$$ds^{*2} = \langle dx^* \rangle \{dx^*\} \quad ,$$

$$ds^2 = \langle dx \rangle \{dx\} \quad ,$$

the Green-Lagrange strain tensor $[\varepsilon]$ is defined by:

$$ds^{*2} - ds^2 = 2\langle dx \rangle [\varepsilon] \{dx\} \quad . \quad . \qquad (2.5)$$

Using the relation (2.2), we obtain:

$$2[\varepsilon] = [F]^T [F] - [I] \qquad (2.6)$$

where [I] represents the identity matrix.

2.3.2 Kinematic relations for a point of a beam element

A beam is a body with one of the dimensions, the length, which is predominant in comparison with the other two, the width and the thickness. Our study is being done in the X-Z plane. We will suppose the length along the axis OX and the thickness along the axis OZ. Therefore the width will be along a line perpendicular to the X-Z plane and will be neglected in the following calculations (it will just appear as a multiplicative factor for the area and for the momentum of inertia of a section of the beam).

For purpose of calculation, we choose a truss element with an orthogonal and tangential coordinate system. Therefore we will express the displacement of some point P of the beam as function of the displacement of its projection P_o on the middle axis. The first part of this paragraph is concerned with those calculations. In the second part we will discuss the deformation gradient tensor $[F]$ and the Green-Lagrange strain tensor $[\varepsilon]$ for the beam.

2.3.2.1 Calculation of a displacement vector of a point of a beam

Let us consider a beam of thickness h, width b and length L. The initial middle axis is the axis OX (Fig (2.2) and Fig (2.3)). Suppose the beam is deformed under the application of a force $\vec{F}$. The position of a point P is defined by its position vector (Fig (2.4)). Thus:

$$P = \overrightarrow{OP} = x\vec{e}_1 + z\vec{e}_2 = \begin{Bmatrix} x \\ z \end{Bmatrix}$$

where $x \in [0,L]$ and $z \in \left[-\frac{h}{2}, +\frac{h}{2}\right]$.

In the same way, $\overrightarrow{OP}_o = \begin{Bmatrix} x \\ o \end{Bmatrix}$.

If $\vec{U} = \begin{Bmatrix} U(x,z) \\ W(x,z) \end{Bmatrix}$ and $\vec{u} = \begin{Bmatrix} u(x) \\ w(x) \end{Bmatrix}$

are the respective displacement vectors of P and P_o, we get:

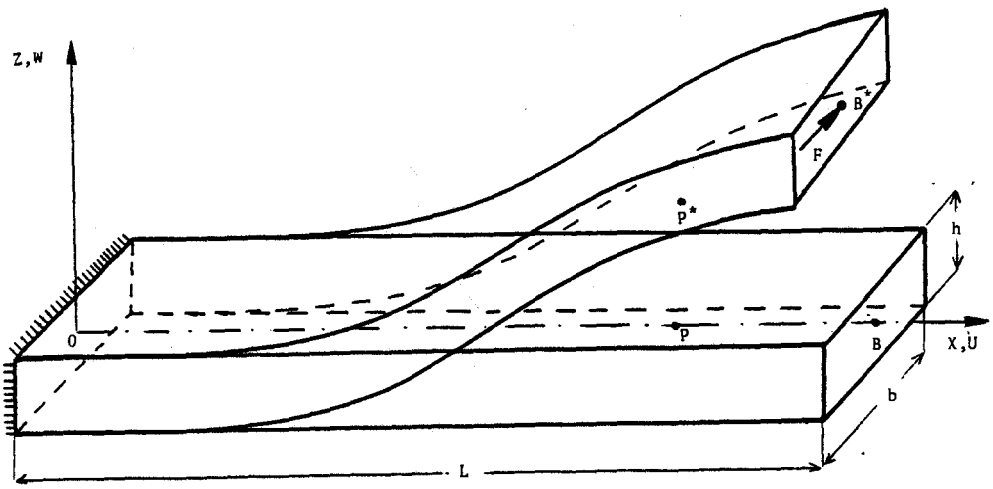


Fig. 2.2 - Thin rectangular beam before and after planar deformation ($h < b \ll L$)

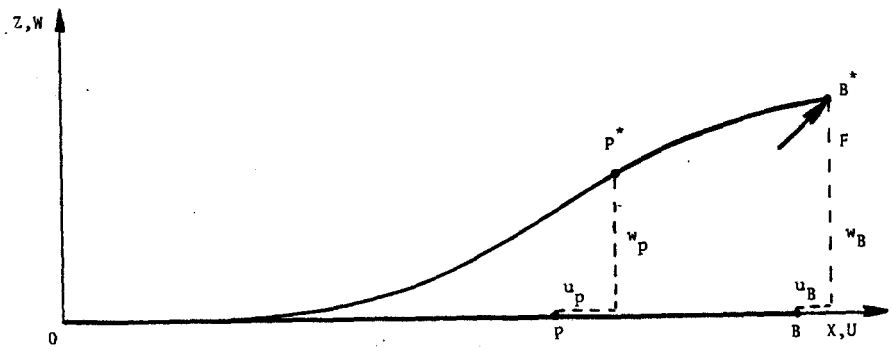


Fig. 2.3 Deformation of the middle axis in the plane X-Z

$$\vec{OP}^* = \vec{OP} + \vec{U} = \begin{Bmatrix} x^* \\ z^* \end{Bmatrix} = \begin{Bmatrix} x + U(x,z) \\ z + W(x,z) \end{Bmatrix}$$

and

$$\vec{OP}_o^* = \vec{OP}_o + \vec{u} = \begin{Bmatrix} x_o^* \\ z_o^* \end{Bmatrix} = \begin{Bmatrix} x + u(x) \\ w(x) \end{Bmatrix} .$$

Then the relation

$$\begin{Bmatrix} x_o^* \\ z_o^* \end{Bmatrix} = \begin{Bmatrix} x + u(x) \\ w(x) \end{Bmatrix}$$

describes the deformation of the middle axis. The unit tangent vector $\vec{t}^*$ at P_o^* is

$$\vec{t}^* = \begin{Bmatrix} 1 + u_{,x} \\ w_{,x} \end{Bmatrix} \times \frac{1}{m} = \begin{Bmatrix} \cos\theta \\ \sin\theta \end{Bmatrix}$$

with $m = \sqrt{(1 + u_{,x})^2 + w_{,x}^2}$ (Fig (2.4) and Fig (2.5)) .

The unit normal vector to $\vec{t}^*$ can be written as

$$\vec{n}^* = \frac{1}{m} \begin{Bmatrix} -w_{,x} \\ 1 + u_{,x} \end{Bmatrix} .$$

Thus the vectors $\vec{t}^*$ and $\vec{n}^*$ form a local orthogonal system at P_o^* .

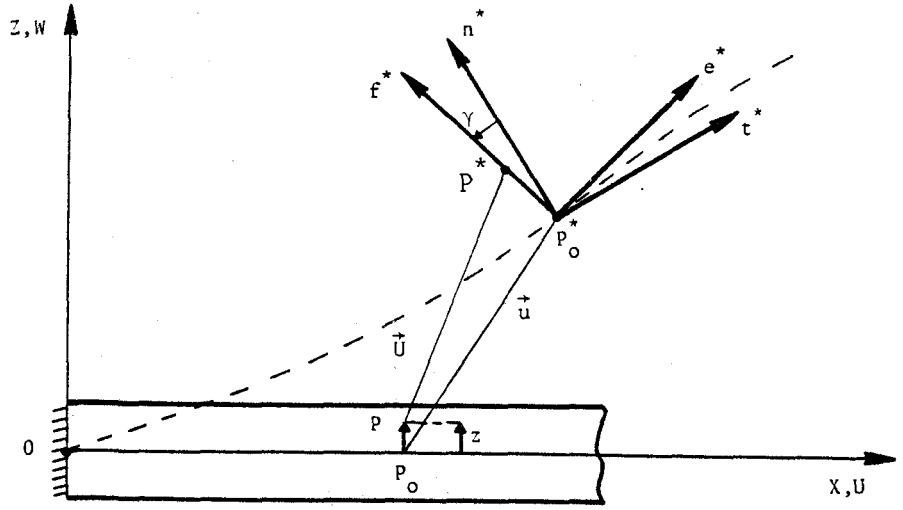


Fig. 2.4 - Beam undergoing deformation

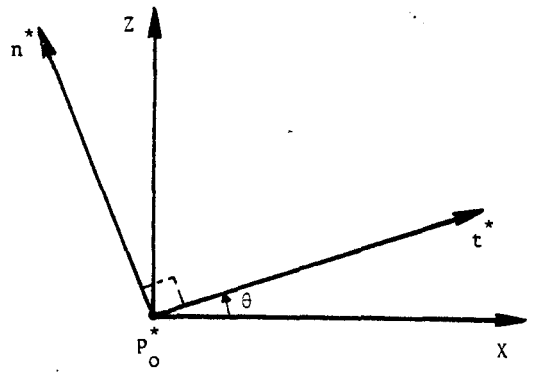


Fig. 2.5 - Rotation of the middle axis

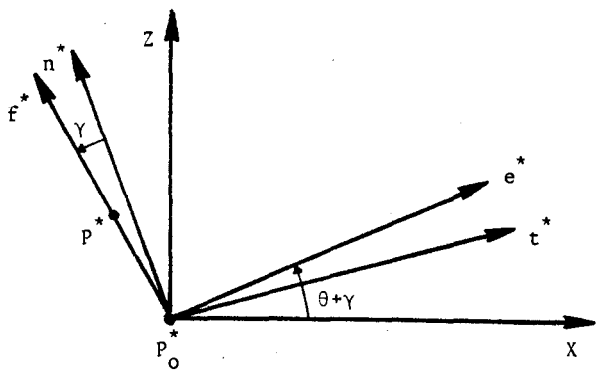


Fig. 2.6 - Rotation of the normal section

Let $\vec{f}^*$ be the unit vector along $\overrightarrow{P_o^* P^*}$ and let $\vec{e}^*$ be a unit vector orthogonal to $\vec{f}^*$ (Fig (2.4) and Fig (2.6)). Let us suppose that the normal section to the middle axis at P_o^* remains to be a plane after deformation. We will call αz the distance $\|\overrightarrow{P_o^* P^*}\|$. The constant α will be determined further after some supplementary hypotheses. Thus,

$$\overrightarrow{OP}^* = \overrightarrow{OP}_o^* + \alpha z \, \vec{f}^* \qquad (2.7)$$

Then $$\overrightarrow{P_o^* P^*} = \alpha z \, \vec{f}^* = \alpha z(-\vec{t}^* \sin \gamma + \vec{n}^* \cos \gamma)$$

or $$\overrightarrow{P_o^* P^*} = \alpha z(X_1 \vec{e}_1 + Z_1 \vec{e}_2)$$

where $$X_1 = - \frac{1}{m} (1 + u_{,x}) \sin \gamma - \frac{1}{m} w_{,x} \cos \gamma$$

$$Z_1 = - \frac{1}{m} w_{,x} \sin \gamma + \frac{1}{m} (1 + u_{,x}) \cos \gamma \quad .$$

The vectorial equation:

$$\overrightarrow{OP}^* = \overrightarrow{OP}_o^* + \overrightarrow{P_o^* P^*}$$

allows a relation between the displacement of P and the displacement of P_o :

$$U(x,z) = u(x) + \alpha z \, X_1(x)$$

$$W(x,z) = w(x) + \alpha z \, Z_1(x) - z \, .$$

2.3.2.2 Deformation gradient tensor

The relation (2.3) gives:

$$[F] = \begin{bmatrix} 1 + U,_x & U,z \\ W,_x & 1 + W,_z \end{bmatrix}$$

Since u , w , X_1 and Z_1 are independent of z , we have

$$[F] = \begin{bmatrix} 1 + u,_x + \alpha z \dfrac{\partial X_1}{\partial x} & \alpha X_1 \\ w,_x + \alpha z \dfrac{\partial Z_1}{\partial x} & \alpha Z_1 \end{bmatrix} . \qquad (2.8)$$

2.3.2.3 Green-Lagrange strain tensor

In the following discussion, we will introduce four different expressions for the components of the Green-Lagrange strain tensor, depending on various additional hypotheses.

2.3.2.3.1 First expression of the components of the tensor $[\varepsilon]$

The equation (2.6) allows us to obtain the components of the tensor $[\varepsilon]$ as:

$$\varepsilon_{xx} = \frac{1}{2}\left[\left(1 + u_{,x} + \alpha z \frac{\partial X_1}{\partial x}\right)^2 + \left(w_{,x} + \alpha z \frac{\partial Z_1}{\partial x}\right)^2 - 1\right]$$

$$\varepsilon_{zz} = \frac{1}{2}(\alpha^2 X_1^2 + \alpha^2 Z_1^2 - 1) \qquad (2.9)$$

$$\varepsilon_{xz} = \left(1 + u_{,x} + \alpha z \frac{\partial X_1}{\partial x}\right)\alpha X_1 + \left(w_{,x} + \alpha z \frac{\partial Z_1}{\partial x}\right)\alpha Z_1$$

2.3.2.3.2 Second expression of the components of the tensor [ε]

Here we advance the hypothesis of the conservation of orthogonality (Love-Kirchhoff hypothesis), i.e. $\gamma = 0$ (Fig (2.6)). After some calculations, we obtain the following expressions:

$$\varepsilon_{xx} = e_m + \alpha z\chi + \alpha^2 z^2 \psi$$

$$\varepsilon_{zz} = \frac{1}{2}(\alpha^2 - 1) \qquad (2.10)$$

$$\varepsilon_{xz} = 0$$

with $$e_m = u_{,x} + \frac{1}{2}(u_{,x})^2 + \frac{1}{2}(w_{,x})^2 = \frac{1}{2}(m^2 - 1) \quad ,$$

$$\chi = \frac{1}{m}(w_{,x}\, u_{,xx} - w_{,xx}\,(1 + u_{,x})) = -\, m \frac{d\theta}{dx} \quad ,$$

and $$\psi = \frac{1}{2m^4}\left((1 + u_{,x})w_{,xx} - u_{,xx}\, w_{,x}\right)^2 = \frac{1}{2}\left(\frac{d\theta}{dx}\right)^2 \quad .$$

2.3.2.3.3 Third expression of the components of the tensor [ε]

The hypothesis pertaining to the discussion in this section is the Bernoulli-Euler hypothesis which assumes:

- The sections of the beam remain orthogonal to the middle axis after the deformation (Love-Kirchhoff hypothesis)
- The sections of the beam remain the same after the deformation.

This hypothesis leads to the equality $\alpha = 1$; therefore

$$\varepsilon_{xx} = \bar{e}_m + z\chi + z^2\psi$$

$$\varepsilon_{zz} = 0 \qquad (2.11)$$

$$\varepsilon_{xz} = 0$$

2.3.2.3.4 Fourth expression of the components of the tensor [ε]

We will add two more hypothesis:

- hypothesis of thin beams $(z/L \ll 1)$
- hypothesis of small strains

This last hypothesis provides two facts:

- angle changes between fibers are small

- fiber extensions are small.

Since the value m represents the multiplicative factor of the elongation of the middle axis during the deformation, $m \simeq 1$ for a small strain. We get the equations:

$$\varepsilon_{xx} = e_m + z\underline{\chi}$$

$$\varepsilon_{zz} = 0 \qquad (2.12)$$

$$\varepsilon_{xz} = 0$$

where $\underline{\chi}$ is obtained by substituting the value 1 for m in the expression of χ. The expression of ε_{xx} is the same as the one obtained by Epstein and Murray and will be the one we will use.

From this expression, we could obtain the classical non-linear approximate expression of ε (Von Karman or Koiter Sanders type) by using the hypothesis of moderate rotations . Then we would have:

$$\varepsilon_{xx} = u_{,x} + \frac{1}{2} w_{,x}^2 - z\, w_{,xx}$$

§2.4 DIFFERENTIAL EQUILIBRIUM EQUATIONS

2.4.1 Stress tensors

In order to establish the equilibrium equations, we will

have first to define the Cauchy stress tensor $[\sigma^*]$. But the subsequent calculations need the knowledge of the 2nd Piola-Kirchhoff stress tensor $[\sigma]$. It will be defined in this paragraph after the determination of the Cauchy stress tensor. We will follow the development as given in Batoz [4].

2.4.1.1 Cauchy stress tensor

Let us consider an elementary rectangular element subjected to normal stresses σ_x^* and σ_z^* and shearing stresses τ_{xz}^* and τ_{zx}^* (Fig (2.7)). The plane Cauchy stress tensor, which allows us to define the stress state, can be written:

$$[\sigma^*] = \begin{bmatrix} \sigma_x^* & \tau_{zx}^* \\ \tau_{xz}^* & \sigma_z^* \end{bmatrix} .$$

It will be shown that:

$$\tau_{zx}^* = \tau_{xz}^* .$$

Thus we can represent the plane Cauchy stress tensor in the vector form as:

$$<\sigma^*> = <\sigma_x^* \quad \sigma_z^* \quad \tau_{zx}^*> .$$

The plane Cauchy stress tensor allows the determination of the stress vector $\vec{\sigma}_{(n)}^*$ at a point P^* acting on a line segment defined by the unit normal vector $\vec{n}^*$.

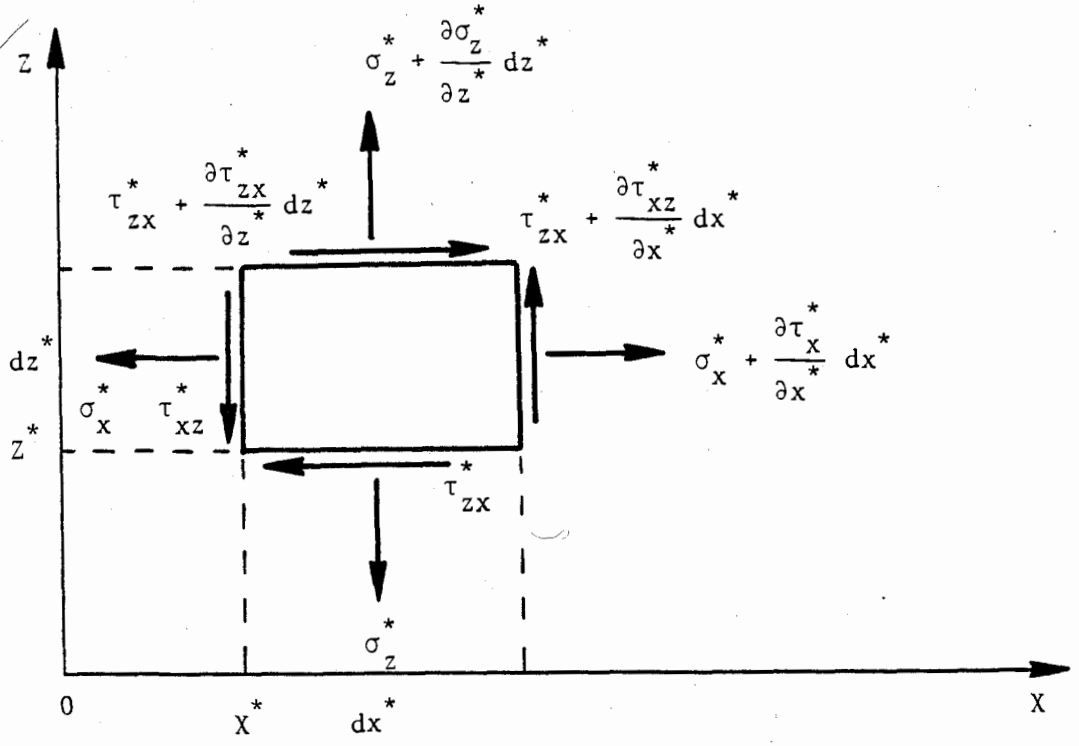


Fig. 2.7 - Plane stress element

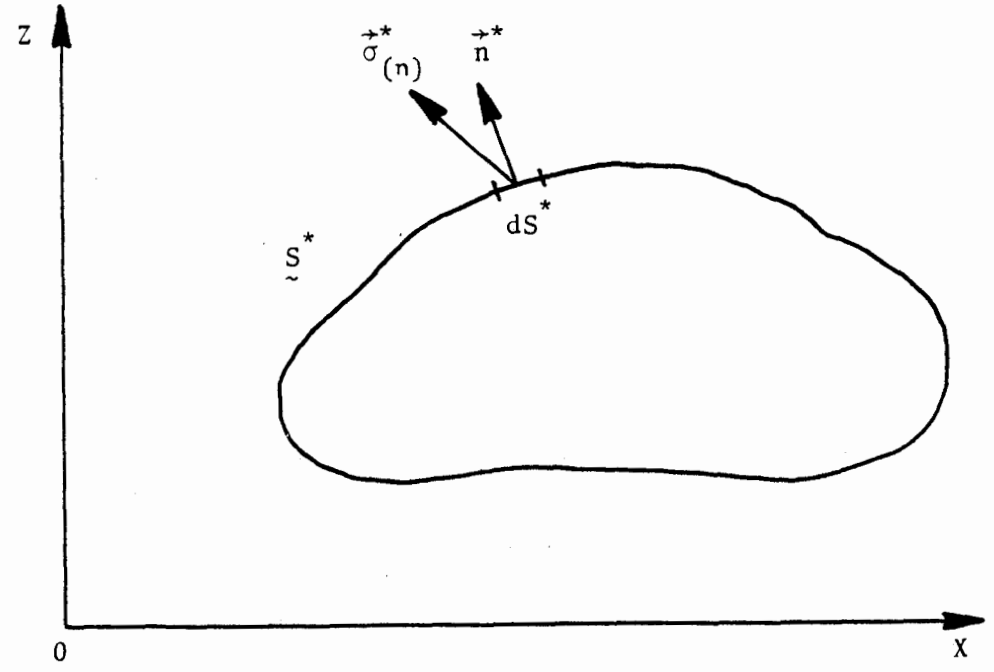


Fig. 2.8 - A two dimensional body in equilibrium

In the matrix form,

$$\left\{\sigma^*_{(n)}\right\} = [\sigma^*]^T\{n^*\} \quad . \tag{2.13}$$

A similar formula holds for the 3-dimensional case.

2.4.1.2 1st and 2nd Piola-Kirchhoff stress tensor

Let dS be the infinitesimal element of surface which becomes dS^* after deformation and $\vec{n}$ be the unit normal vector to dS and $\vec{n}^*$ the unit normal vector to dS^*. We will define the stretch λ^* of the element of surface dS by:

$$dS^* = \lambda^* dS \quad .$$

Let $\vec{\sigma}^*_{(n)}$ be the stress vector at P^* associated to dS^*. The stress vector $\vec{\underset{\sim}{\sigma}}^*_{(n)}$ corresponds to $\vec{\sigma}^*_{(n)}$ on the non-deformed surface and is defined by:

$$\left\{\vec{\underset{\sim}{\sigma}}^*_{(n)}\right\} = \lambda^* \left\{\sigma^*_{(n)}\right\}$$

The stress vector $\vec{\sigma}_{(n)}$ is linked to the vector $\vec{\underset{\sim}{\sigma}}^*_{(n)}$ as the vector dx is with the vector $\vec{dx}^*$, i.e.

$$\left\{\sigma_{(n)}\right\} = [F]^{-1}\left\{\underset{\sim}{\sigma}^*_{(n)}\right\} \quad .$$

The components of the 1st Piola-Kirchhoff stress tensor $[\underset{\sim}{\sigma}^*]$ and of the 2nd Piola-Kirchhoff stress tensor $[\sigma]$ connect the components of the stress vectors $\vec{\underset{\sim}{\sigma}}^*_{(n)}$ to the vector $\vec{\sigma}^*_{(n)}$ and $\vec{\sigma}_{(n)}$ to the vector

$\vec{n}$ by

$$\left\{\underset{\sim}{\sigma}^{*}_{(n)}\right\} = [\underset{\sim}{\sigma}^{*}]^{T}\{n\}$$

$$\left\{\sigma_{(n)}\right\} = [\sigma]^{T}\{n\}$$

Using the Hansen formula $\{n^{*}\}dS^{*} = \det[F]\left([F]^{-1}\right)^{T}\{n\}dS$, we can connect the Cauchy stress tensor $[\sigma^{*}]$ to the 2nd Piola Kirchhoff stress tensor $[\sigma]$ as

$$[\sigma^{*}] = \frac{1}{\det[F]}[F][\sigma][F]^{T} \qquad (2.14)$$

2.4.1.3 Equilibrium equations

Let $\underset{\sim}{S}^{*}$ be an arbitrary closed surface subjected to the stress vector $\vec{\sigma}^{*}_{(n)}$ (Fig (2.8)). The principle of equilibrium yields:

$$\int_{\underset{\sim}{S}^{*}} \vec{\sigma}^{*}_{(n)}\, dS^{*} + \int_{\underset{\sim}{V}^{*}} \vec{F}^{*}_{V}\, dV^{*} = 0$$

where

$\underset{\sim}{V}^{*}$ is the corresponding volume of $\underset{\sim}{S}^{*}$, and

$\vec{F}^{*}_{V}$ is the body force per unit volume with components $F^{*}_{V_x}$ and $F^{*}_{V_z}$.

In the differential form and for 2-dimensional case, the above equations can be written as;

$$\frac{\partial \sigma_x^*}{\partial x^*} + \frac{\partial \tau_{zx}^*}{\partial z^*} + F_{V_x}^* = 0$$

$$\frac{\partial \tau_{xz}^*}{\partial x^*} + \frac{\partial \sigma_z^*}{\partial z^*} + F_{V_z}^* = 0 \quad . \tag{2.15}$$

The vanishing of moments can be stated as:

$$\int_{\underset{\sim}{S}^*} \vec{OP}^* \wedge \vec{\sigma}_{(n)}^* \, dS^* + \int_{\underset{\sim}{V}^*} \vec{OP}^* \wedge \vec{F}_V^* \, dV^* = 0 \ ,$$

from which one can deduce, in particular, that

$$\tau_{xz}^* = \tau_{zx}^* \tag{2.16}$$

The equilibrium equations for beams will be the same equations ((2.15) and (2.16)).

§2.5 EXPRESSION OF THE VIRTUAL WORK

2.5.1 Expression of the virtual work for a body

2.5.1.1 Boundary conditions

The boundary conditions can be separated into 2 sets:

- the mechanical conditions acting on a part S_σ^* of the surface: These conditions are related to the forces and they describe the principle action-reaction on the portion of surface S_σ^* .

- the geometrical conditions acting on S_u^* (the complementary part of S_σ^*): These conditions are related to the displacements and the rotations of points of S_u^* and they impose some restrictions for these variables.

Mechanical conditions

Let F_s^* be the external force per unit area applied to S_σ^* . Suppose its components are F_{sx}^* and F_{sz}^* . Then the mechanical conditions on every point $P^* \in S_\sigma^*$ can be written:

$$F_{sx}^* = \sigma_x^* p^* + \tau_{zx}^* n^*$$

$$F_{sz}^* = \tau_{xz}^* p^* + \sigma_z^* n^* \qquad (2.17)$$

where p^* and n^* are the components of unit normal vector $\vec{n}^*$ at P^* for the considered surface element.

Geometrical conditions

The geometrical conditions impose some displacements or rotations for each point $P^* \in S_u^*$. Let $\vec{u}^*$ be the displacement vector of P^* and let $\vec{\theta}^*$ be a vector whose components are the values of the rotations at P^* . Then at this point the geometrical conditions can be written:

$$\vec{u}^* = \bar{\vec{u}}^*$$

$$\vec{\theta}^* = \bar{\vec{\theta}}^*$$

where $\bar{\vec{u}}^*$ is a vector containing the values of the imposed displacements and $\bar{\vec{\theta}}^*$ is a vector containing the values of the imposed rotations.

2.5.1.2 Field of kinematically admissible virtual displacements

A field of arbitrary virtual displacements $\vec{\delta u}^*$ is said to be kinematically admissible if

$$\vec{\delta u}^* = \vec{0} \quad \text{on} \quad S_u^*$$

2.5.1.3 Expression of the virtual work in the deformed configuration

The principle of virtual work states that the equilibrium of a body requires that, for any kinematically admissible small virtual displacements $\vec{\delta u}^*$, the total internal virtual work δU is equal to the total external virtual work δW.

Let δu^* and δw^* be the components of $\vec{\delta u}^*$. Then the total external virtual work is

$$\delta W = \int_{V^*} \vec{F}_v^* \cdot \vec{\delta u}^* \, dV^* + \int_{S_\sigma^*} \vec{F}_s^* \cdot \vec{\delta u}^* \, dS^*$$

Then, after application of the equations (2.15), the above equation becomes

$$\delta W = - \int_{V^*} \left(\left(\frac{\partial \sigma_x^*}{\partial x^*} + \frac{\partial \tau_{zx}^*}{\partial z^*} \right) \delta u^* + \left(\frac{\partial \sigma_z^*}{\partial z^*} + \frac{\partial \tau_{xz}^*}{\partial x^*} \right) \delta w^* \right) dV^*$$

$$+ \int_{S_\sigma^*} \vec{F}_s^* \cdot \vec{\delta u}^* \, dS^* \quad . \tag{2.18}$$

Since $\vec{\delta u}^* = \vec{0}$ on S_u^*, using the equation (2.17), we will obtain:

$$\int_{S_\sigma^*} \vec{F}_s^* \cdot \vec{\delta u}^* \, dS^* = \int_{S_\sigma^* \cup S_u^*} \vec{F}_s^* \cdot \vec{\delta u}^* \, dS^* = \int_{S^*} \Big((\sigma_x^* p^* + \tau_{zx}^* n^*) \, \delta u^*$$

$$+ (\sigma_z^* n^* + \tau_{xz}^* p^*) \delta w^* \Big) \, dS^* \, .$$

Using the equations (2.4), we can define the transposed vector $<\delta\varepsilon^*>$ which represents the vector of virtual strains by:

$$<\delta\varepsilon^*> = <\delta\varepsilon_{xx}^* \quad \delta\varepsilon_{zz}^* \quad \delta\varepsilon_{zx}^*>$$

with

$$\delta\varepsilon_{xx}^* = \delta \frac{\partial u^*}{\partial x^*} = \frac{\partial \delta u^*}{\partial x^*}$$

$$\delta\varepsilon_{zz}^* = \delta \frac{\partial w^*}{\partial z^*} = \frac{\partial \delta w^*}{\partial z^*}$$

$$\delta\varepsilon_{zx}^* = \delta \left(\frac{\partial u^*}{\partial z^*} + \frac{\partial w^*}{\partial x^*} \right) = \frac{\partial \delta u^*}{\partial z^*} + \frac{\partial \delta w^*}{\partial x^*} \quad .$$

Thus, applying integration by parts to the surface integral of the equation (2.18), we get

$$\delta W = \int_{V^*} (\sigma_x^* \, \delta\varepsilon_{xx}^* + \sigma_z^* \, \delta\varepsilon_{zz}^* + \tau_{zx}^* \, \delta\varepsilon_{xz}^*) \, dV^*$$

or $$\delta W = \int_{V^*} <\sigma^*>\{\delta\varepsilon^*\} \, dV^*$$

The integral on the right will be called the total internal virtual work and will be denoted by δU , i.e.

$$\delta U = \int_{V^*} <\delta^*>\{\delta\varepsilon^*\} \, dV^*$$

Thus the virtual work principle can be stated either as $\delta U = \delta W$ or

$$\int_{V^*} <\sigma^*>\{\delta\varepsilon^*\} \, dV^* = \int_{V^*} <F_V^*>\{\delta u^*\} \, dV^* + \int_{S_\sigma^*} <F_s^*>\{\delta u^*\} \, dS^* \qquad (2.19)$$

2.5.1.4 Expression of the virtual work with respect to the original configuration

The virtual work principle presented previously is expressed with respect to the deformed configuration Γ^* . Because the quantities V^* and S^* about the configuration Γ^* are unknown, it is important to restate the principle of the virtual work with respect

to the original configuration Γ . We will express the virtual work principle in terms of Green-Lagrange strains and 2nd Piola-Kirchhoff stresses.

2.5.1.4.1 Lagrangian formulation of the virtual work principle

The expression of the total internal virtual work is:

$$\delta U = \int_{V^*} <\sigma^*>\{\delta\varepsilon^*\}\, dV^* \quad .$$

Using the relation $<\sigma^*>\{\delta\varepsilon^*\} = tr([\sigma^*][\delta\varepsilon^*])$, we get

$$\delta U = \int_{V^*} tr([\sigma^*][\delta\varepsilon^*])\, dV^*$$

If we set $[\delta\underset{\sim}{\varepsilon}] = [F]^T[\delta\varepsilon^*][F]$ and if we apply the equation (2.14), we get

$$\delta U = \int_{V} tr([\sigma][\delta\underset{\sim}{\varepsilon}])\, dV$$

The virtual work principle is valid for all fields of kinematically admissible virtual displacements. In particular it is valid for $\delta\vec{u}^* = \delta\vec{u}$. In this case,

$$[\delta\varepsilon] = [\delta\underset{\sim}{\varepsilon}] = [F]^T[\delta\varepsilon^*][F]$$

Using the equations (2.3) and (2.4), we have

$$[\delta\varepsilon^*] = \frac{1}{2}\left([B]^T + [B]\right) \quad \text{and}$$

$$[\delta F] = [B][F]$$

where
$$[B] = \begin{bmatrix} \delta \dfrac{\partial u}{\partial x^*} & \delta \dfrac{\partial u}{\partial z^*} \\ \delta \dfrac{\partial w}{\partial x^*} & \delta \dfrac{\partial w}{\partial z^*} \end{bmatrix} .$$

After substituting the last 2 equations into the one before, we obtain:

$$2[\delta\varepsilon] = [\delta F]^T[F] + [F]^T[\delta F] \quad ,$$

or after integration,

$$2[\varepsilon] = [F]^T[F] + [C_o]$$

where $[C_o]$ is a constant matrix.

Thus, $\forall$ dx :

$$\langle dx\rangle[F]^T[F]\{dx\} + \langle dx\rangle[C_o]\{dx\} = 2\langle dx\rangle[\varepsilon]\{dx\} \quad .$$

We can choose $[C_o]$ in order to obtain the most interesting formulation for $[\varepsilon]$. In fact, if we choose $[C_o] = -[I]$, then tensor $[\varepsilon]$ is the Green-Lagrange strain tensor. Thus:

$$\delta U = \int_V \mathrm{tr}([\sigma][\delta\varepsilon])\, dV \qquad (2.20)$$

with $[\varepsilon] = \frac{1}{2}([F]^T[F] - [I])$ -- the Green-Lagrange strain tensor.

The equation (2.20) can also be written:

$$\delta U = \int_V <\sigma>\{\delta\varepsilon\}\, dV$$

where $<\sigma>$ is the vectorial representation of the components of the 2nd Piola-Kirchhoff stress tensor,

$<\delta\varepsilon>$ is the vectorial representation of the components of the Green-Lagrange strain variation tensor.

The expression of the virtual work principle, with respect to Lagrangian variables, can be written, for all $\vec{\delta u}$ such that $\vec{\delta u} = \vec{0}$ on S_u^*,

$$\int_V <\sigma>\{\delta\varepsilon\}dV = \int_V <\bar{F}_V>\{\delta u\}dV + \int_{S_\sigma} <\bar{F}_s>\{\delta u\}\, dS$$

where S_σ is the portion of S which becomes S_σ^* after deformation,

$<\bar{F}_V> = <F_V^*>\det[F]$ and is the body forces in Γ, and

$<\bar{F}_s> = \lambda^* <F_s^*>$ and is the surface forces on S_σ.

2.5.2 Application of the virtual work principle to a beam

The expression of the virtual work due to external forces varies as a function of the forces applied and needs to be calculated for each case. For a beam, nevertheless, we can calculate the internal virtual work.

2.5.2.1 Expression of the total internal virtual work for a beam

The total internal virtual work is (eq. (2.20)):

$$\delta U = \int_V \mathrm{tr}([\sigma][\delta\varepsilon])\, dV \quad ,$$

therefore

$$\delta U = \int_V (\sigma_{xx}\, \delta\varepsilon_{xx} + \tau_{xz}\, \delta\varepsilon_{xz} + \sigma_{zz}\, \delta\varepsilon_{xx})\, dV \ .$$

The equation (2.12) gives the Green-Lagrange strain tensor

$$\varepsilon_{xx} = e_m + z\underline{\chi}$$

$$\varepsilon_{zz} = 0$$

$$\varepsilon_{xz} = 0 \quad ,$$

where

$$e_m = u,_x + \frac{1}{2} u,_x^2 + \frac{1}{2} w,_x^2 \qquad (2.21)$$

and

$$\underline{\chi} = w,_x\, u,_{xx} - (1 + u,_x) w,_{xx} \quad .$$

We will have then:

$$\delta\varepsilon_{xx} = \delta e_m + z\delta\underline{\chi}$$

$$\delta\varepsilon_{zz} = 0$$

$$\delta\varepsilon_{xz} = 0 \qquad (2.22)$$

where $\delta e_m = (1 + u,_x)\delta u,_x + w,_x \delta w,_x$,

and $\delta \underline{\chi} = u,_{xx} \delta w,_x + w,_x \delta u,_{xx} - (1 + u,_x) \delta w,_{xx}$

$- w,_{xx} \delta u,_x$.

Our study restricts itself to elastic deformations. That means the strained body will recover its original shape and size when the stresses are removed.

In this hypothesis, stress is a function of strain only and Hooke's law gives this relation. This law can be written:

$$\sigma_{xx} = E \varepsilon_{xx} ,$$

where E represents the Young's modulus of elasticity.

Therefore δU can be expressed:

$$\delta U = E \int_V (\delta e_m e_m + z(\delta \underline{\chi} e_m + \delta e_m \underline{\chi}) + z^2 \delta \underline{\chi} \underline{\chi}) dV$$

If we complete the orthogonal system OXZ by the axis OY, we will obtain:

$$dV = dx\, dy\, dz .$$

After integration along the length L of the beam, δU can be written:

$$\delta U = E \int_L \left(\delta e_m e_m \iint dy\, dz + (\delta \underline{\chi} e_m + \delta e_m \underline{\chi}) \iint z\, dy\, dz \right.$$

$$\left. + \underline{\chi}\, \delta \underline{\chi} \iint z^2\, dy\, dz \right) dx$$

The beam is supposed to be symmetric in the plane OXY with respect to its middle axis. Therefore all integration of an odd factor of z will be zero.

Thus:

$$\delta U = \int_L (\delta e_m \, EA \, e_m + \delta \underline{\chi} \, EI \, \underline{\chi}) \, dx$$

where $A = \iint dy \, dz$ represents the area of the cross section of the beam,

and $I = \iint z^2 \, dy \, dz$ represents the moment of inertia of the cross section.

The normal stress N and the bending moment M are

$$N = EA \, e_m \quad ,$$

$$M = EI \, \underline{\chi} \quad .$$

For the particular case of a rectangular beam with a width b along the axis OY and a thickness h along the axis OZ :

$$A = b \, h \quad ,$$

$$I = \frac{b \, h^3}{12} \quad .$$

CHAPTER 3

APPLICATION OF FINITE ELEMENT METHOD AND GAUSSIAN QUADRATURE TO BEAM STRUCTURES

§3.1 GENERALITIES

The Finite Element Method is a very popular technique for solving partial-differential equations that occur in engineering applications. In the recent years a lot of publications have been written. Some of the leading books are: Zienkiewicz [5], Gallagher [6], Bathe [7], Dhatt and Touzot [8], Strang and Fix [9] and Oden and Reddy [10]. The last two books consider principally the mathematical aspect of the Finite Element Method. The books by Zienkiewicz, Bathe, and Dhatt and Touzot are very detailed and with some numerical programs included. The book of Gallagher is more related to the structural analysis and is suited as an introductory book to F.E.M.

The Gaussian quadrature is a technique used in numerical integration and is employed to solve the numerous integrals brought by the F.E.M. analysis. This technique is discussed in almost all the books concerning numerical analysis, e.g. Burden, Faires and Reynolds [11], Davis and Rabinowitz [12], and Engels [13]. However, we shall give a brief presentation of this technique at the end of this chapter.

§3.2 FINITE ELEMENT METHOD

This section begins with a presentation of the Finite Element Method

and ends with some matrix representations of the variables used.

3.2.1 Introduction to the F.E.M.

We will base our study on the "displacements method". The basic principle of the Finite Element Method consists of subdividing the domain of study V into a set of sub-domains V^e, called finite elements. Two distinct elements can have in common only points located on their common boundary. These elements are represented by a finite number of nodes. The F.E.M. defines some interpolation functions N which allow us to approximate an unknown function u over V^e as a function of a certain set of values, called the nodal variables, at the nodes of this element. These interpolation functions must assure the continuity of the variable on the domain V. Thus, for each element V^e of a beam,

$$u = \langle N \rangle \{u_n^e\} \tag{3.1}$$

where u is the displacement of points in V^e, N the interpolation function defined over V^e, and $\{u_n^e\}$ represents the vector of the nodal variables assigned to the nodes of the element V^e.

Assuming $V = \sum_{e \in K} V^e$ where $K \in \mathbb{N}$, we can express the virtual work principle as:

$$\delta U = \delta W$$

or

$$\sum_{e \in K} \delta U^e = \sum_{e \in K} \delta W^e$$

after transformation, this relation becomes:

$$\sum_{e\in K} <\delta u_n^e>[k^e]\ u_n^e = \sum_{e\in K} <\delta u_n^e>\{f_n^e\} \qquad (3.3)$$

where $[k^e]$ is the elementary stiffness matrix of the element V^e and $\{f_n^e\}$ the elementary load vector.

We can define a gathering technique which allows us to write the relation (3.3) in the form:

$$<\delta U_n>[K]\{U_n\} = <\delta U_n>\{F_n\} \qquad (3.4)$$

where $[K]$ is the global stiffness matrix, $\{U_n\}$ is the global vector of nodal variables, and $\{F_n\}$ is the global load vector.

The preceding equation is valid for all $<\delta U_n>$, therefore,

$$[K]\{U_n\} = \{F_n\} \qquad (3.5)$$

The above discussion about equation 3.5 will become more transparent as the reader continues his reading of the next few sections. The procedure used to solve this equation will be described in chapter 4.

3.2.2 Type of element

The theoretical formulation requires us to deal with functions of class C^1 so that the continuity of the displacements and of their derivatives at the nodes are guaranteed. Thus we opted for a Hermite's element defined in the book of Dhatt and Touzot [8] (Fig (3.1)) and for which the vector of nodal variables can be written as

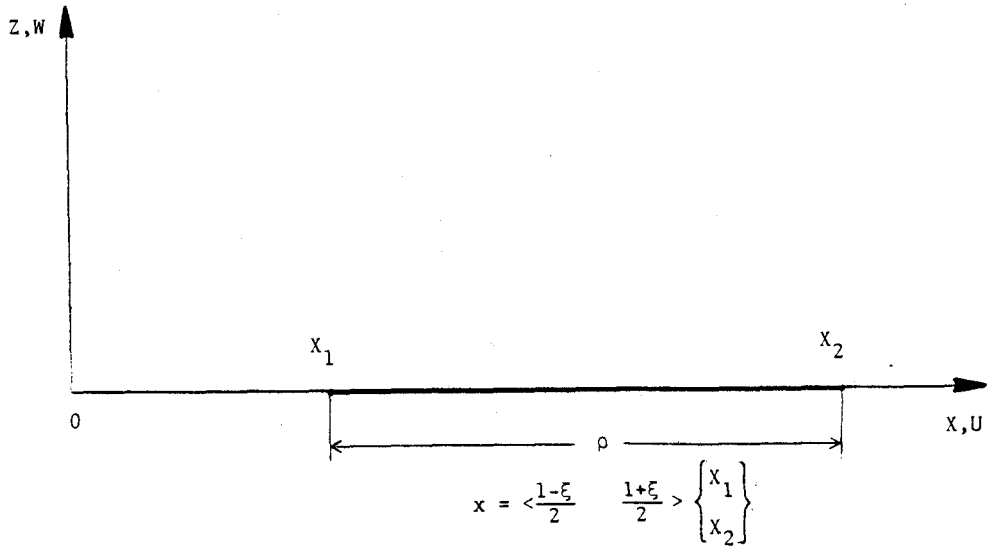


Fig. 3.1 - Real element V^e

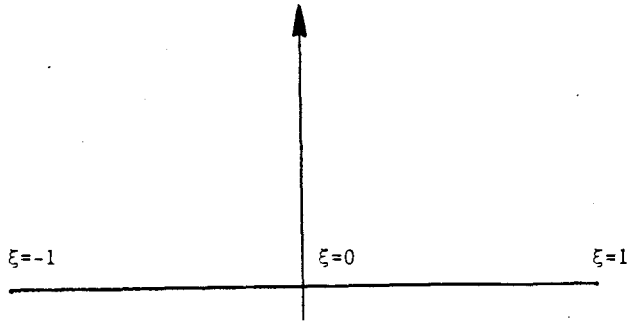


Fig. 3.2 - Reference element V^r

$$<u_n^e> = <u_1 \quad u_{1,x} \quad w_1 \quad w_{1,x} \quad u_2 \quad u_{2,x} \quad w_2 \quad w_{2,x}>$$

where the index 1 corresponds to the first node of the element and the index 2 corresponds to the second node of the element (here we deal with a finite element of a beam. See Fig. 3.1), and u , w represents the horizontal and vertical displacements respectively.

But the elements V^e can have different lengths. In order to simplify the calculations we will relate all the elements V^e with a unique element V^r , called reference element (Fig (3.2)). This can be done by a bijective geometrical transformation which defines the coordinate x of each point of the real element V^e as function of the coordinate ξ of the corresponding point in the reference element.

The displacements u and w of a point of V^e will be written as

$$u(\xi) = <N_u(\xi)>\{u_n^e\}$$

$$w(\xi) = <N_w(\xi)>\{u_n^e\}$$

where $<N_u> = <N_1 \quad N_2 \quad 0 \quad 0 \quad N_3 \quad N_4 \quad 0 \quad 0>$ and

$<N_w> = <0 \quad 0 \quad N_1 \quad N_2 \quad 0 \quad 0 \quad N_3 \quad N_4>$. The functions N_1, N_2, N_3 and N_4 will be defined later.

Let $u,_x$ be the first derivative of U with respect to x . Then

$$u,_x = \frac{du}{d\xi} \cdot \frac{d\xi}{dx} = \frac{2}{x_2-x_1} \cdot \frac{du}{d\xi} = \frac{2}{\rho} \frac{d<N_u(\xi)>}{d\xi} \{u_n^e\}$$

where $\rho = x_2-x_1$. In the same way,

$$u_{,xx} = \left(\frac{2}{\rho}\right)^2 \frac{d^2\langle N_u(\xi)\rangle}{d\xi} \{u_n^e\}$$

The derivatives of the interpolation functions will be defined in the next section.

3.2.3 Interpolation functions

Suppose we want to approximate the horizontal displacement u at a point with coordinate ξ .

$u(\xi)$ can be expressed in the reference element as a linear combination of independent functions $\{p_i(\xi) : i \in I\}$. The choice of these functions is an important part of the F.E.M. These functions $p_i(\xi)$ will be chosen from a polynomial basis with their number equal to the number of nodal variables of the element. The nodal variables corresponding to the horizontal displacement are u_1, $u_{1,x}$, u_2 and $u_{2,x}$. Therefore, the polynomial basis $p_i(\xi)$ is defined by:

$$p_i(\xi) = \xi^i \qquad i = 0,1,2,3 \quad .$$

We will then obtain

$$u(\xi) = \langle 1 \quad \xi \quad \xi^2 \quad \xi^3 \rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \tag{3.6}$$

$$u,_{\xi}(\xi) = \langle 0 \quad 1 \quad 2\xi \quad 3\xi^2 \rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \tag{3.7}$$

where a_1, a_2, a_3 and a_4 are the generalized coordinates. The approximations (3.6) and (3.7) must give the actual values at the nodes; therefore, $u(-1) = u_1$, $u,_{\xi}(-1) = u_{1,\xi}$, $u(1) = u_2$ and $u,_{\xi}(2) = u_{2,\xi}$. The equalities $u_{1,\xi} = \frac{\rho}{2} u_{1,x}$ and $u_{2,\xi} = \frac{\rho}{2} u_{2,x}$ lead to the relation

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = [p_n]^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \rho/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \rho/2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_{1,x} \\ u_2 \\ u_{2,x} \end{Bmatrix}$$

with $$[p_n] = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} .$$

Finally we obtain

$$u(\xi) = \langle N(\xi) \rangle \begin{Bmatrix} u_1 \\ u_{1,x} \\ u_2 \\ u_{2,x} \end{Bmatrix}$$

with $\langle N(\xi)\rangle = \langle 1 \;\; \xi \;\; \xi^2 \;\; \xi^3 \rangle$

$$\left[\begin{array}{cccc} \frac{2}{4} & \frac{1}{4} & \frac{2}{4} & -\frac{1}{4} \\ -\frac{3}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{array}\right] \left|\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & \frac{\rho}{2} & 0 & 0 \\ 1 & 0 & 1 & 0 \\ & 0 & 0 & \frac{\rho}{2} \end{array}\right]$$

Let us set $\langle N(\xi)\rangle = \langle N_1 \;\; N_2 \;\; N_3 \;\; N_4 \rangle$. Then

$$N_1 = \frac{1}{4}(1-\xi)^2(2+\xi)$$

$$N_2 = \frac{\rho}{8}(1-\xi^2)(1-\xi)$$

$$N_3 = \frac{1}{4}(1+\xi)^2(2-\xi)$$

$$N_4 = -\frac{\rho}{8}(1-\xi^2)(1+\xi) \quad .$$

We would obtain the same result for the function of interpolation of the vertical displacement w . For convenience, we supply three tables which give the interpolation functions for u, $u_{,x}$ $u_{,xx}$, w, $w_{,x}$ and $w_{,xx}$. These tables are the tables (3.1), (3.2) and (3.3).

3.2.4 Discretized expressions

After a short presentation of the Newton-Raphson method, developed more fully in the next chapter, we will define the residual vector and the tangent matrix. These explanations can be found in the book of Dhatt and Touzot [8] and the discretized expressions can be deducted from the materials of Batoz [14] .

TAB. 3.1 - Interpolation function

$u = \langle N_1 \quad N_2 \quad 0 \quad 0 \quad N_3 \quad N_4 \quad 0 \quad 0 \rangle \{u_n^e\}$	
$w = \langle 0 \quad 0 \quad N_1 \quad N_2 \quad 0 \quad 0 \quad N_3 \quad N_4 \rangle \{u_n^e\}$	
N_1	$\frac{1}{4}(1-\xi)^2(2+\xi)$
N_2	$\frac{\rho}{8}(1-\xi^2)(1-\xi)$
N_3	$\frac{1}{4}(1+\xi)^2(2-\xi)$
N_4	$-\frac{\rho}{8}(1-\xi^2)(1+\xi)$

TAB. 3.2 - First derivative of the interpolation function

$u,_x = \langle N_{1,x} \quad N_{2,x} \quad 0 \quad 0 \quad N_{3,x} \quad N_{4,x} \quad 0 \quad 0 \rangle \{u_n^e\}$	
$w,_x = \langle 0 \quad 0 \quad N_{1,x} \quad N_{2,x} \quad 0 \quad 0 \quad N_{3,x} \quad N_{4,x} \rangle \{u_n^e\}$	
$N_{1,x}$	$-\frac{3}{2\rho}(1-\xi^2)$
$N_{2,x}$	$\frac{1}{4}(-1+\xi)(1+3\xi)$
$N_{3,x}$	$\frac{3}{2\rho}(1-\xi^2)$
$N_{4,x}$	$-\frac{1}{4}(1+\xi)(1-3\xi)$

TAB. 3.3 - Second derivative of the interpolation function

$u,_{xx} = \langle N_{1,xx} \quad N_{2,xx} \quad 0 \quad 0 \quad N_{3,xx} \quad N_{4,xx} \quad 0 \quad 0 \rangle \{u_n^e\}$ $w,_{xx} = \langle 0 \quad 0 \quad N_{1,xx} \quad N_{2,xx} \quad 0 \quad 0 \quad N_{3,xx} \quad N_{4,xx} \rangle \{u_n^e\}$	
$N_{1,xx}$	$\frac{6}{\rho^2}\xi$
$N_{2,xx}$	$\frac{1}{\rho}(3\xi-1)$
$N_{3,xx}$	$-\frac{6}{\rho^2}\xi$
$N_{4,xx}$	$\frac{1}{\rho}(3\xi+1)$

The Newton-Raphson method allows us to solve the nonlinear equation (3.5) ($[K]\{U_n\} = \{F_n\}$) by iteration. Let $\delta\Pi = \delta W - \delta U$. One looks for an approximation U_{n+1} of the solution by expanding $\delta\Pi$ in Taylor's serie in the neighbourhood of U_n :

$$\delta\Pi(U_{n+1}) = \delta\Pi(U_n) + \delta^2\Pi(U_n) + ... = 0$$

We will then define $\{U_{n+1}\}$ by $\{U_{n+1}\} = \{U_n\} + \{\Delta U_n\}$.

In order to apply this method we must calculate the residual vector $\{R\}$ and the tangent matrix $[K_T]$. The residual vector is defined by:

$$\delta\Pi(U_n) = <\delta U_n>(\{F_n\} - [K]\{U_n\}) = <\delta U_n>\{R\} \qquad (3.8)$$

The tangent matrix $[K_T]$ satisfies the equation

$$[K_T]\{\Delta U_n\} = \{R\} \quad . \qquad (3.9)$$

To calculate $\{R\}$ and $[K_T]$, one needs first to calculate the residual vector and tangent matrix for each element V^e .

3.2.4.1 Calculation of the residual vector of an element V^e

The application of the equation (3.1) for δu gives

$$\delta u = <N>\{\delta u_n^e\} \ ,$$

where $<\delta u_n^e> = <\delta u_1 \quad \delta u_{1,x} \quad \delta w_1 \quad \delta w_{1,x} \quad \delta u_2 \quad \delta u_{2,x} \quad \delta w_2 \quad \delta w_{2,x}>$.

The discretization of the equations (2.22) allows us to write

$$\delta e_m = \langle VM \rangle \{\delta u_n^e\} \quad ,$$

$$\delta \underline{\chi} = \langle VK \rangle \{\delta u_n^e\} \quad ,$$

where $\langle VM \rangle = (1 + u,_x)\langle Nu,_x \rangle + w,_x \langle Nw,_x \rangle$,

and $\langle VK \rangle = u,_{xx}\langle Nw,_x \rangle + w,_x \langle Nu,_{xx} \rangle - (1 + u,_x)\langle Nw,_{xx} \rangle - w,_{xx}\langle Nu,_x \rangle$

We can express the expression of the internal virtual work of an element by:

$$\delta U^e = \langle \delta u_n^e \rangle \{r^e\}$$

where $$\{r^e\} = \int_{-1}^{+1} (EAe_m \{VM\} + EI\underline{\chi}\{VK\}) \frac{\rho}{2} d\xi$$

($\{r^e\}$ is called the elementary residual vector).

3.2.4.2 Calculation of the tangent matrix of an element V^e

The variation of the internal virtual work is needed for the calculation of the tangent matrix and can be written as

$$\Delta\delta U = \int_L (\delta e_m EA\Delta e_m + \delta\underline{\chi} EI \Delta\underline{\chi} + EAe_m \Delta\delta e_m + EI\underline{\chi}\Delta\delta\underline{\chi})\, dx \quad .$$

Because u and w are essential variables, we will set the following conditions:

$$\Delta\delta u = \Delta\delta w = \Delta\delta u,_x = \Delta\delta w,_x = \Delta\delta u,_{xx} = \Delta\delta w,_{xx} = 0$$

Then the expression of $\Delta\delta e_m$ and $\Delta\delta\underline{\chi}$ can be written as

$$\Delta\delta e_m = \delta u,_x \Delta u,_x + \delta w,_x \Delta w,_x \quad ,$$

$$\Delta\delta\underline{\chi} = \delta w,_x \Delta u,_{xx} + \Delta w,_x \delta u,_{xx} - \Delta w,_{xx} \delta u,_x - \delta u,_{xx} \Delta u,_x \quad .$$

Therefore, on the element V^e ,

$$\Delta\delta e_m = <\delta u_n^e>[VM2]\{\Delta u_n^e\}$$

$$\Delta\delta\underline{\chi} = <\delta u_n^e>[VK2]\{\Delta u_n^e\}$$

where $[VM2] = \{Nu,_x\}<Nu,_x> + \{Nw,_x\}<Nw,_x>$

$$[VK2] = \{Nw,_x\}<Nu,_{xx}> + \{Nu,_{xx}\}<Nw,_x> - \{Nw,_{xx}\}<Nu,_x>$$

$$- \{Nu,_x\}<Nw,_{xx}> \quad .$$

Then

$$\Delta\delta U^e = <\delta u_n^e>[k_t^e]\{\Delta u_n^e\}$$

with

$$[k_t^e] = \int_{-1}^{+1} (\{VM\}EA<VM> + \{VK\}EI<VK> + EAe_m[VM2]$$

$$+ EI\underline{\chi}[VK2]) \frac{\rho}{2} d\xi \quad .$$

The matrix $[k_t^e]$ is called the elementary tangent matrix.

3.2.4.3 Calculation of the external work

The load F is vertical and is applied on the point P of the frame. Then the external work is given by

$$\delta W = F\delta w_p$$

or

$$\delta W = <\delta u_n^e>\{f^e\}$$

with P located at the second node of the element V^e and $<f^e> = <0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad F \quad 0>$.

3.2.4.4 Gathering of the discretized variables

All the variables are calculated with respect of an element V^e and need to be gathered in order to obtain the global and final equation. This will be done by adding these variables together. Our discretization of the Lee's frame uses 10 elements and 11 nodes. On each element we can write the displacement vector $\{U_e\}$ as

$$\{U_e\} = \begin{Bmatrix} \{Ue,_1\} \\ \{Ue,_2\} \end{Bmatrix} ,$$

where

$$\{Ue,_1\} = \begin{Bmatrix} u_1 \\ u_{1,x} \\ w_1 \\ w_{1,x} \end{Bmatrix}_e$$

$$\{Ue,_2\} = \begin{Bmatrix} u_2 \\ u_{2,x} \\ w_2 \\ w_{2,x} \end{Bmatrix}_e \quad .$$

By adding on the ten elements, we will obtain the final vector $\{U\}$,

$$\{U\} = \begin{Bmatrix} \{u_{1,1}\} \\ \{u_{1,2}\} + \{u_{2,1}\} \\ \vdots \\ \{u_{9,2}\} + \{u_{10,1}\} \\ \{u_{10,2}\} \end{Bmatrix} .$$

In the same way we can write the tangent matrix

$$[k_t^e] = \begin{bmatrix} k_{e,1} & k_{e,2} \\ k_{e,3} & k_{e,4} \end{bmatrix}$$

Then the final matrix will be

$$[K_T] = \begin{bmatrix} k_{1,1} & k_{1,2} & 0 & & & & & & \\ k_{1,3} & (k_{1,4}+k_{2,1}) & k_{2,2} & & & & & & \\ 0 & k_{2,3} & (k_{2,4}+k_{3,1}) & k_{3,2} & & & & & \\ & & & & & & & & 0 \\ & & & & & & & (k_{9,4}+k_{10,1}) & k_{10,2} \\ & & & & & & 0 & k_{10,3} & k_{10,4} \end{bmatrix}$$

But all the terms of the matrices $[k_t^e]$ involve integrals, therefore, we must first calculate these integrals. This will be done by the gaussian quadrature method.

§3.3 GAUSSIAN QUADRATURE

This procedure solves the integral $I = \int_a^b f(x)\,dx$ by choosing n values $x_1, x_2 \ldots x_n$ and n constants $c_1, c_2 \ldots c_n$ with the objective of minimizing the error term

$$E = \int_a^b f(x)\,dx - \sum_{i=1}^{n} c_i\, f(x_i) \qquad (3.10)$$

The purpose of this section is to find the right x_i's and c_i's. We will need some knowledge about Legendre polynomials.

3.3.1 Legendre polynomials

The following explanation can be found in Godsil [15] .

A set of functions $\{P_o, P_1, \ldots P_n\}$ is said to be orthogonal on $[a,b]$ with respect of the positive weight function $w(x)$ if

$\int_a^b P_k(x)\, P_j(x)\, w(x)\, dx$ is equal to zero when $j \neq k$ and positive when $j=k$.

An important property of orthogonal polynomials is the following one:

if the functions $P_o, P_1 \ldots P_n$ are polynomials and if P_k is a polynomial of order k for $k = 0, 1, \ldots, n$ then P_k has k distinct roots in the interval $[a,b]$.

The set of Legendre polynomials is a set of orthogonal polynomials on $[-1,1]$ with respect to the weight function $w(x) = 1$ which satisfies the recursive relation:

$$P_{k+1}(x) = \frac{2k+1}{k+1}\, x\, P_k(x) - \frac{k}{k+1}\, P_{k-1}(x)$$

where $k \geq 1$, $P_o(x) = 1$, $P_1(x) = x$.

3.3.2 Evaluation of the variables

The values of the points $x_1, \ldots, x_n$ minimizing the error term (eq 3.10) are found to be the n distinct roots of the Legendre polynomial $P_n(x)$ (Davis and Rabinowitz [12]).

The coefficients c_i are given by

$$c_i = \frac{2(1-x_i^2)}{[n\ P_{n-1}(x_i)]^2} \quad .$$

The gaussian quadrature allows the exact integration of a polynomial of order (2n-1). The choice of $n = 3$ will give us a good enough approximation.

Therefore the values of x_i will be the roots of the polynomial

$$P_3(x) = \frac{5}{2} x^3 - \frac{3}{2} x \quad .$$

These values are $x_1 = -\sqrt{\frac{3}{5}}$, $x_2 = 0$, $x_3 = +\sqrt{\frac{3}{5}}$. The corresponding coefficients c_i will be $c_1 = c_3 = \frac{5}{9}$, $c_2 = \frac{8}{9}$.

The finite element method and the gaussian quadrature allow us to characterize the behaviour of Lee's frame by the equation:

$$[K]\{U_n\} = \{F_n\} \quad .$$

The following chapter will introduce the methods used to solve this equation.

CHAPTER 4

NUMERICAL METHODS FOR SOLVING NONLINEAR SYSTEMS

§4.1 GENERALITIES

This chapter presents the numerical methods we will use in order to solve the nonlinear equation:

$$[K]\{U_n\} = \{F_n\} \tag{4.1}$$

We will proceed by iteration; and at each iterative step, we will solve the linear equation

$$[K_T(U_{i-1})]\{\Delta U_i\} = \{R(U_{i-1})\} \quad . \tag{4.2}$$

This chapter will be divided into three parts:

- the first part concerns the method for solving the relation (4.2)
- the second part presents the usual iterative process
- the third part, which is the most important one, will introduce a special technique needed for finding the deformations of the Lee's frame.

§4.2 SOLUTION OF THE LINEAR SYSTEM

4.2.1 Boundary conditions

The equation (4.2) doesn't take into account the boundary conditions, we will introduce them by using the method of the diagonal dominant term.

Let $\{U_n\}$ be the approximation of the solution vector $\{U\}$. Then $\{U_n\}$ is obtained by iteration, using the following process:.

$$\{U_1\} = \{U_o\} + \{\Delta U_1\}$$

$$\{U_2\} = \{U_1\} + \{\Delta U_2\}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\{U_n\} = \{U_{n-1}\} + \{\Delta U_n\}$$

Let m be the dimension of the vector space containing $\{U\}$. Then the m components of $\{U_o\}$ are usually chosen equal to zero and

$$\{U_o\} = \begin{Bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix}$$

Let $\overline{U}_{s_1}$, $\overline{U}_{s_2}$, ... and $\overline{U}_{s_p}$ be the boundary conditions for the vector $\{U\}$. We will define $\{U_o\}$ by

$$\{U_o\} = \begin{Bmatrix} U^1 \\ U^2 \\ \cdot \\ \cdot \\ \cdot \\ U^m \end{Bmatrix} ,$$

with
$$\begin{cases} U^i = 0 & i \neq s_1,\ldots,s_p \\ U^i = \overline{U}_i & \text{in the other cases.} \end{cases}$$

In order to keep the same boundary conditions for the final approximation, we have to modify the matrix $[K_T]$ and the vector $\{R\}$ in the equation (4.2).

We will bring the following modifications:

$$\begin{cases} K_{T_{s_j s_j}} = K_{T_{s_j s_j}} + \alpha \\ R_{s_j} = 0 \end{cases}$$

where $j = 1,2 \ldots p$ and $\alpha = 10^8$.

These two modifications will insure that

$$\Delta U_{s_1} = \Delta U_{s_2} = \ldots = \Delta U_{s_p} = 0$$

in the vectors

$$\{\Delta U_i\} = \begin{Bmatrix} \Delta U^1 \\ \cdot \\ \cdot \\ \cdot \\ \Delta U^m \end{Bmatrix} , \quad i = 1,2,\ldots,n \quad .$$

4.2.2 Storage of the matrix $[K_T]$

The matrix $[K_T]$ is a symmetric, positive definite block tridiagonal matrix, and can be written:

$$[K_T] = \begin{bmatrix} KD(1) & KS(1) & & & & \\ KS(1)^T & KD(2) & KS(2) & & & \\ & KS(2)^T & KD(3) & \ddots & & \\ & & \ddots & \ddots & KS(10) & \\ & & & KS(10)^T & KD(11) & \end{bmatrix}$$

The order of $[K_T]$ is 44 and all the internal matrices are matrices of order 4. The storage scheme will consist of 2 vectors $\{KD\}$ and $\{KS\}$ such that

$$\{KD\} = \begin{Bmatrix} KD(1) \\ KD(2) \\ \vdots \\ KD(11) \end{Bmatrix} \quad \text{and} \quad \{KS\} = \begin{Bmatrix} KS(1) \\ KS(2) \\ \vdots \\ KS(10) \end{Bmatrix} .$$

4.2.3 Block L-U factorization

All the details concerning this method can be found in the book by Golub and Van Loan [16] .

We use this method in order to take advantage of the block structure of the symmetric positive definite matrix $[K_T]$. By considering a block factorization for the matrix $[K_T]$, we obtain

$$[K_T] = \begin{bmatrix} I & & & \\ ML(2)\,I & & & \\ & ML(3)\,I & & \\ & & \ddots & \\ & & & ML(11)\,I \end{bmatrix} \begin{bmatrix} MU(1) & KS(1) & & \\ & MU(2) & KS(2) & \\ & & \ddots & \\ & & & KS(10) \\ & & & MU(11) \end{bmatrix}$$

where the matrices $ML(i)$ and $MU(i)$ are matrices of order 4 and obtained by the following algorithm:

(1) $MU(1) = KD(1)$

(2) For $i = 2, \ldots 11$

(3) Solve $ML(i) \cdot MU(i-1) = KS(i)^T$ for $ML(i)$

(4) $MU(i) = KD(i) - ML(i) \cdot KS(i-1)$

This procedure is well defined as $MU(1), \ldots MU(10)$ are non-singular due to the construction of $[K_T]$. The vector $\{\Delta U\}$ can be obtained by block forward elimination and backward substitution using the following algorithm:

```
(5)  For  i = 1, ... 11
(6)  y(i) = R(i) - ML(i)y(i-1)    (ML(1)y(0) = 0)
(7)  For  i = 11, ... 1
(8)  Solve  MU(i)ΔU(i) = y(i) - KS(i)ΔU(i+1)
     for  ΔU(i)          (KS(11)ΔU(12) = 0)
```

To carry out the instructions (3) and (8) we use the matrix inverse function included in the A.P.L. instructions.

§4.3 ITERATIVE PROCESS

This section begins by a presentation of the Newton-Raphson's method and ends with the description of a modification of this method --this modification is the imposition of an increment.

4.3.1 Newton-Raphson algorithm (Fig. (4.1))

The following expansion is given in the book of Dhatt and Touzot [8] .

The equation brought by the F.E.M. is

$$[K(U)]\{U\} = \{F\} \tag{4.3}$$

This equation can be written:

$$\{R(U)\} = \{F\} - [K(U)]\{U\} = 0 \quad .$$

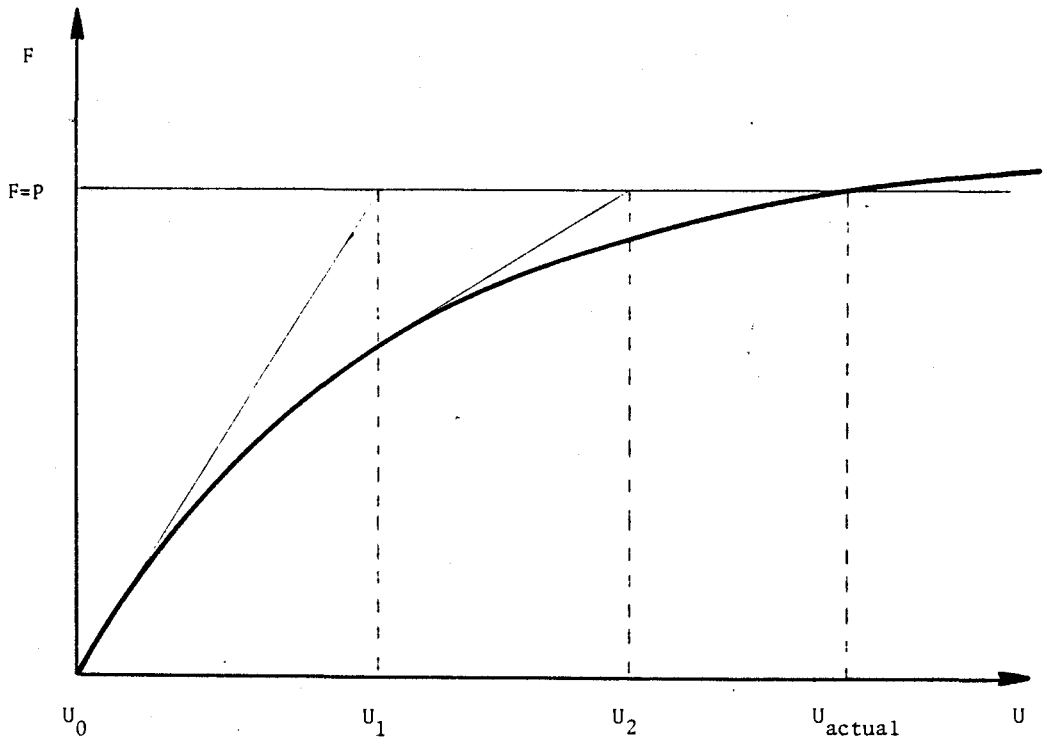


Fig. 4.1 - Newton-Raphson's algorithm

The solution of the equation (4.3) requires us to find a vector $\{U\}$ such that $\{R(U)\}$ is as close to zero as possible. The actual solution gives $\{R(U)\} = 0$.

The search for the solution will be done in an iterative manner: if $\{U_{i-1}\}$ is the solution after $(i-1)$ loops, we have:

$$[K(U_{i-2})]\{U_{i-1}\} = \{F\} \quad .$$

But the residual vector

$$R(U_{i-1}) = \{F\} - [K(U_{i-1})]\{U_{i-1}\}$$

is different of zero. Therefore the new solution must satisfy

$$\{U_i\} = \{U_{i-1}\} + \{\Delta U_i\}$$

with $$\{R(U_i)\} = \{R(U_{i-1} + \Delta U_i)\} = 0 \quad .$$

Expanding this last equation into a Taylor's series of first order, we get

$$\{R(U_i)\} = \{R(U_{i-1})\} + \left[\left(\frac{\partial R}{\partial U}\right)_{U_{i-1}}\right]\{\Delta U_i\} = 0 \quad .$$

Therefore, we must have

$$-\left[\left(\frac{\partial R}{\partial U}\right)_{U_{i-1}}\right]\{\Delta U_i\} = \{R(U_{i-1})\} \quad .$$

The expression of the tangent matrix $[K_T(U_{i-1})]$ is then

$$[K_T(U_{i-1})] = - \left[\left(\frac{\partial R}{\partial U} \right)_{U_{i-1}} \right] .$$

The Newton-Raphson's method solves at each step

$$[K_T(U_{i-1})]\{\Delta U_i\} = \{R(U_{i-1})\} .$$

The termination of this iterative processes will depend on the value of $\{\Delta U_i\}$.

4.3.2 Imposition of a displacement

The following explanations are given in the publication of Batoz and Dhatt [17] .

So far we have supposed that the load was given and that the displacements $\{U_n\}$ were the unknowns. For some problems (Fig. 4.2) it's possible to obtain several displacements for a given load F . In this example the Newton-Raphson's method can't give the solution in the interval [B,C] . In order to obtain a solution in this interval, we must fix one component U of the displacement vector $\{U_n\}$ and consider the load F ($F = \lambda F_o$, F_o fixed) as unknown. As a matter of fact, in [B,C] there is only one load F corresponding to a given value of U .

For each iterative step the Newton-Raphson's method imposes the solution of the equation

$$[K_T]\{\Delta U\} = \{R\} .$$

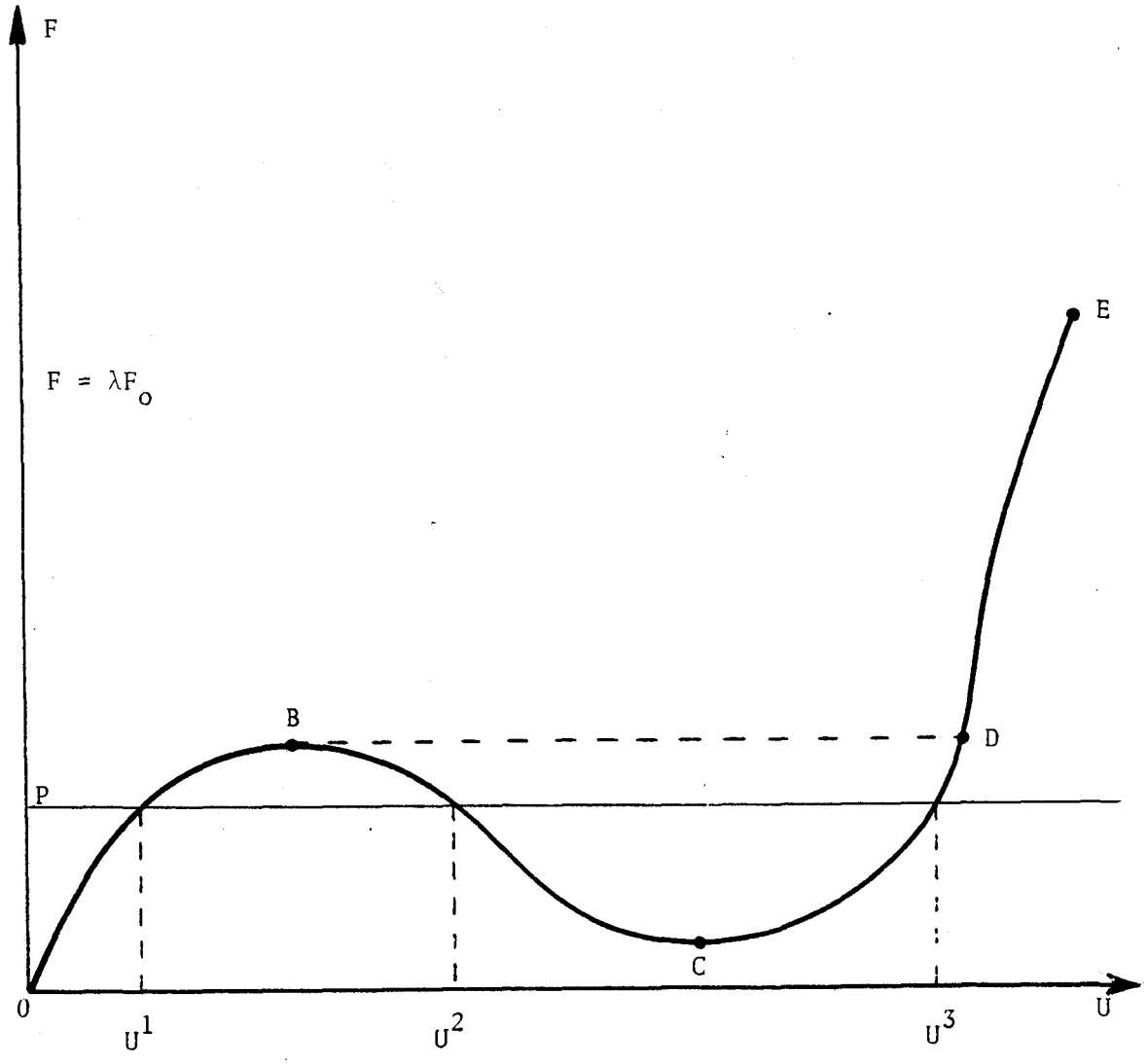


Fig. 4.2 - Several displacements for one load

The residual vector $\{R\}$ can be written in the form

$$\{R\} = \lambda\{F_o\} - [K]\{U\} \tag{4.4}$$

where $\{F_o\}$ is the initial load vector, and λ the load parameter with the initial value 1 .

Let U_L be the value of the imposed component of the solution vector $\{U\}$. The technique used is to substitute the considered component of the initial vector $\{U_o\}$ by U_L and then to fix $\Delta U_L = 0$ for each loop. Then we must solve the two equations:

$$[K_T]\{\Delta U_2\} = \{F_o\} \tag{4.5}$$

$$[K_T]\{\Delta U_1\} = \{R\} \tag{4.6}$$

where $\{R\}$ satisfies the equation (4.4).

Let $(\Delta U_L)_1$ and $(\Delta U_L)_2$ be the variations of the component relative to the imposed displacement, obtained respectively by the equations (4.6) and (4.5). Then the total variation $\{\Delta U\}$ of the vector solution is defined by

$$\{\Delta U\} = \{\Delta U_1\} + \Delta\lambda\{\Delta U_2\} \ ,$$

where $\Delta\lambda = -\dfrac{(\Delta U_L)_1}{(\Delta U_L)_2}$.

Therefore for the loop considered here, $\Delta U_L = 0$ and the new load vector is $(\lambda + \Delta\lambda)\{F_o\}$.

§4.4 AUTOMATIC INCREMENTATION

The figure (4.3) shows the load-displacements curve of the Lee's frame. In order to find the curve in the intervals [B,C] and [A,D] we need to impose a displacement (u or w) . We will call this imposed displacement "increment". In the interval [E,G] , we can't use w as increment; and in [F,H] , u can't be the increment either. These 2 impossibilities are due to the fact that in these intervals we reach an extrema for one displacement, and therefore, we can't always increase the value of this displacement. But u is increasing in [E,G] and w in [F,H] . So, we can obtain the complete curve by using the 2 increments u and w , if we choose the increment carefully at each stage of the deformation (u must be the increment in [E,G] and w the increment in [F,H]) .

This thesis presents an automatic method for this choice and proposes values for stepsizes (values of the increase of the increment). The general technique is based on polynomial interpolation. We also tried to take advantage of this method to improve the precision of our program. Our plan, for this section, will be the following one:

- choice of the increment
- choice of the stepsize
- general improvements.

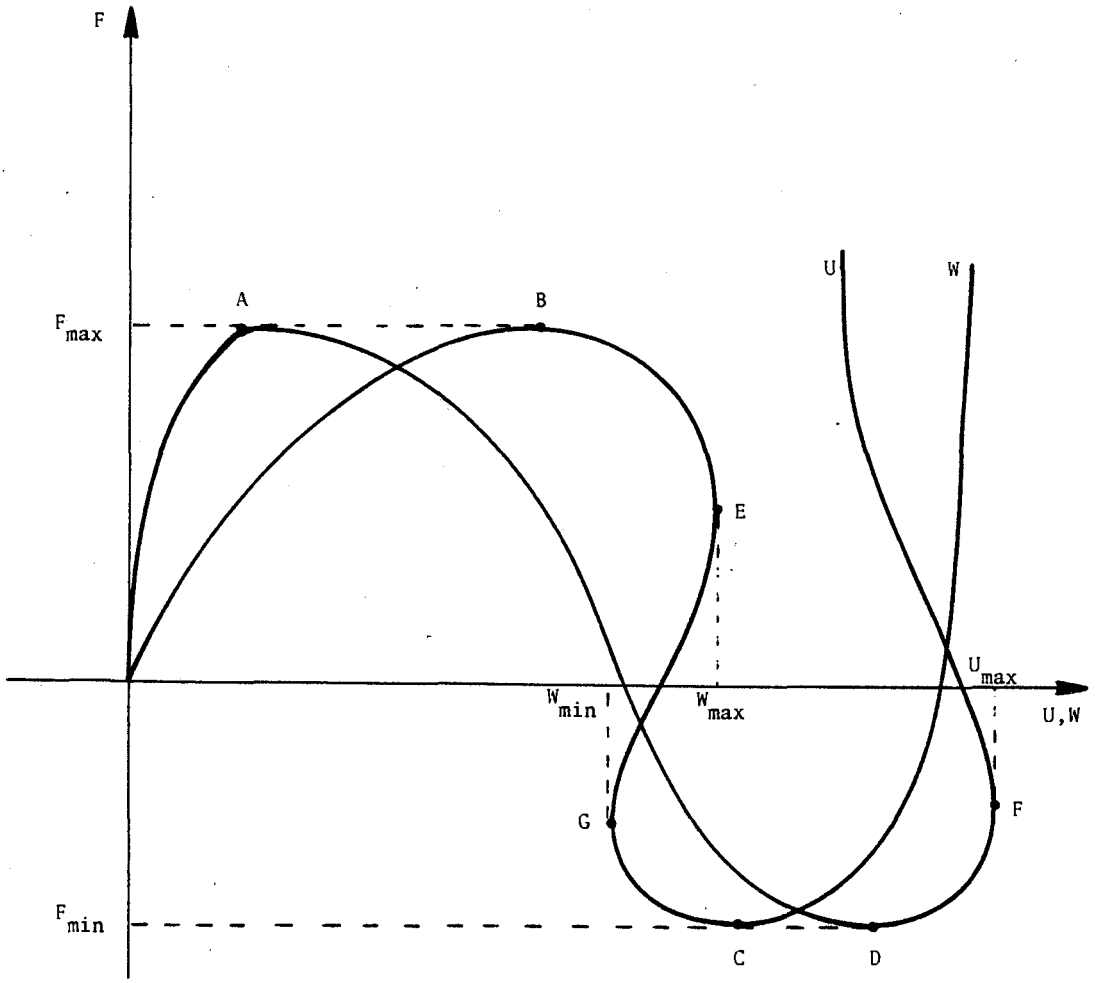


Fig. 4.3 - Displacements vs load

4.4.1 Choice of the increment

The purpose of this study is to find an increasing increment. We shall choose the displacement which increases the most as an increment. Consequently, the slope of the tangent will allow us to predict the right choice.

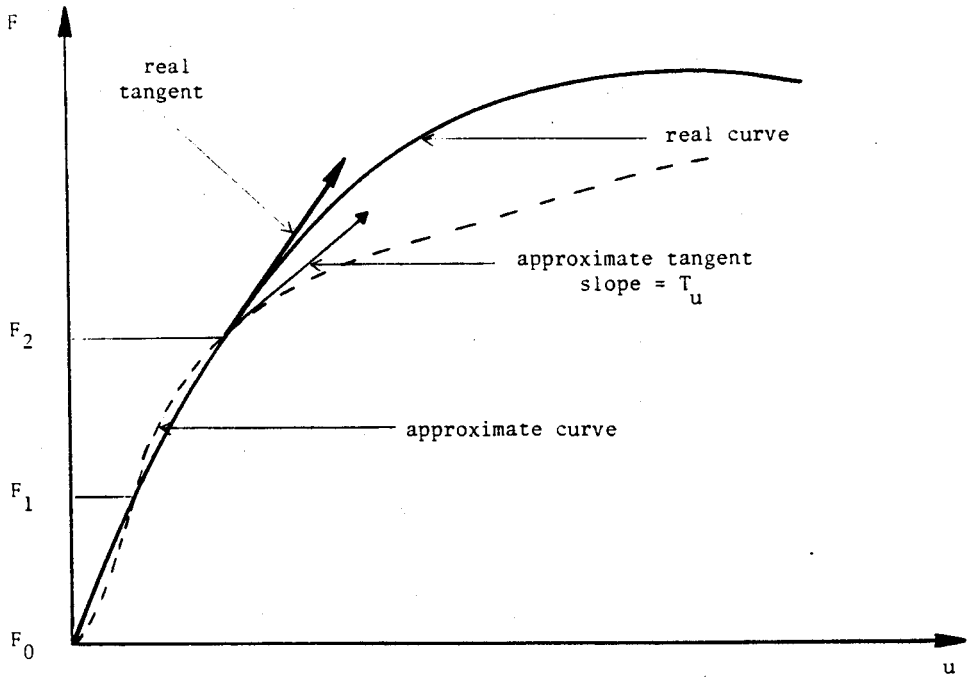
We will use polynomial interpolation of order 3 or 4 to determine the slope of the tangent for u and w (Fig. (4.4) polynomial interpolation of order 3, points chosen have load F_o, F_1 and F_2). The value that we obtain for the slope won't be the exact value, but the precision is not very important and we just need the order of magnitude. The smallest absolute value of the slope will give the displacement which increases the most.

4.4.1.1 Polynomial interpolation

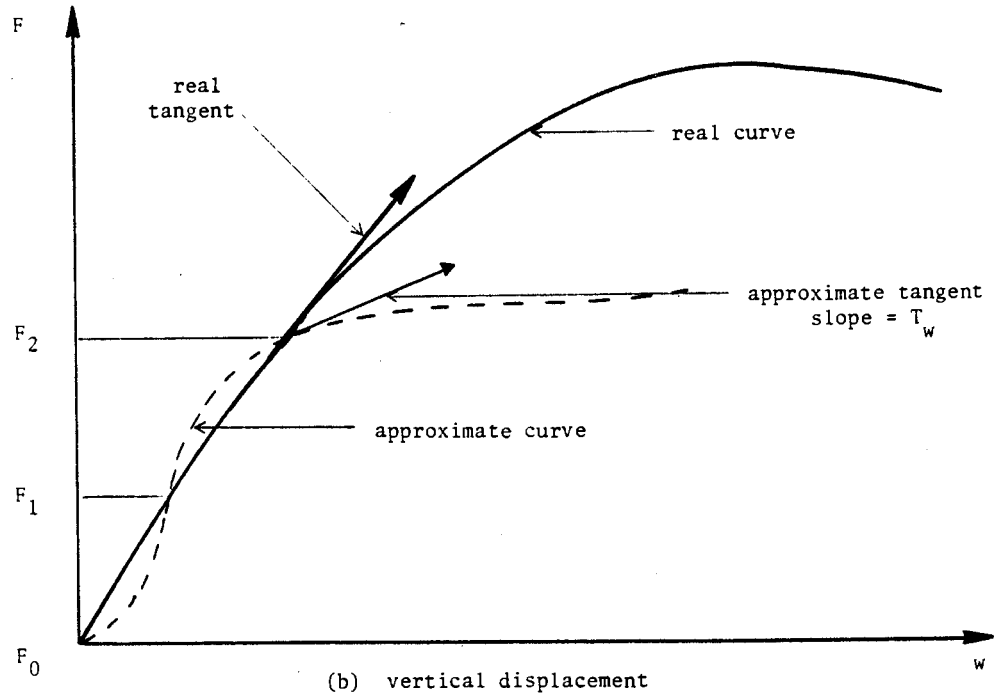
A summary on this method can be found in the book of Burden, Faires, and Reynolds [11] .

Interpolation can be used to approximate an original function in the calculation of integrals or derivatives or in other operations. The most useful interpolating class is then the class of polynomials.

The curves of load-displacements (Fig. (4.3)) can be considered as functions over intervals. These intervals mustn't contain any extrema for the displacement considered. On these intervals, let us define the function f and g such that



(a) horizontal displacement



(b) vertical displacement

if $|T_w| < |T_u|$ next increment = w

$|T_w| \geq |T_u|$ next increment = u

Fig. 4.4 - Choice of the increment

$$F = f(u)$$

$$F = g(w) \quad .$$

In order to approximate f by a polynomial P_s of order $(s+1)$ choose $(s+1)$ points in the interval of definition of f and we will use a divided-difference table. Let us suppose that we know the values of the function f at the different displacements $u_o, u_1, \ldots, u_s$. Then the polynomial P_s of degree at most s that agrees with the function f at those points is

$$P_s(u) = f(u_o) + f[u_o,u_1](u-u_o) + f[u_o,u_1,u_2](u-u_o)(u-u_1)$$

$$+ \ldots\ldots + f[u_o,u_1, \ldots, u_s](u-u_o)(u-u_1) \ldots (u-u_{s-1})$$

where $f[u_i,u_j] = \dfrac{f(u_j) - f(u_i)}{u_j - u_i}$, $\quad i \neq j$,

and $f[u_i, \ldots, u_{i+k}] = \dfrac{f[u_{i+1}, \ldots, u_{i+k}] - f[u_i, \ldots, u_{i+k-1}]}{u_{i+k} - u_i}$.

Once the polynomial P_s is found it's easy to find the value of the derivative, which is the slope of the tangent, at the last point. The algorithm used is the following one:

$F_o, F_1, \ldots, F_s$

$u_o, u_1, \ldots, u_s$ are known

$w_o, w_1, \ldots, w_s$

(1) find $P_s(u)$ which interpolates $f(u)$
and $Q_s(w)$ which interpolates $g(w)$

(2) calculate $P'_s(u_s)$ and $Q'_s(w_s)$

(3) if $|P'_s(u_s)| \leq |Q'_s(w_s)|$ then u is the increment and
$u_{s+1} = u_s$ + stepsize .

Otherwise, w is the increment and $w_{s+1} = w_s$ + stepsize.

The stepsize is generally small, therefore we don't need a high value of s for a good interpolation of the functions f and g . Our program will compute 2 polynomial interpolations, one with $s = 3$ and the other with $s = 4$.

4.4.2 Choice of the stepsize

Once we found the increment, we have to define its variation, called stepsize. The value of the stepsize is important, and it determines the convergence and also the speed of convergence of the Newton-Raphson's method. Because the imposition of a relaxation factor has the same effect, our program will use the following two techniques.

The technique of imposition of a relaxation factor fixes a definite value ΔS for the stepsize and then modifies slightly the iterations of the equation (4.2) in order to attain the fastest convergence. The modification is the introduction of a coefficient w_i such that each loop solves:

$$[K_T(U_{i-1})]\{\Delta U_i\} = \{R(U_{i-1})\}$$

and $$\{U_i\} = \{U_{i-1}\} + w_i\{\Delta U_i\} \quad .$$

If $w_i < 1$ we have an under relaxation method which improves the possibility of convergence

and if $w_i > 1$ we have an over relaxation method which improves the speed of convergence.

The optimum value for w_i depends on the problem studied; it is determined by numerical experimentation. The experimental results we obtained are not very good and we decided to use a variation of stepsize.

The technique we used is based principally on the number of iteration we needed to obtain the preceding deformation.

Let us define:

NITi as the previous number of iteration,

NITW as the number of iteration wished,

ΔSi as the previous stepsize.

Then the general formulation for the stepsize ΔS_{i+1} is

$$\Delta S_{i+1} = \Delta S_i \left(\frac{NITi}{NITW}\right)^{\alpha}$$

where α is a coefficient which the experimental tries and the publication of Deuflhard [18] shows that its optimum value is 1. We linked the value of ΔS_{i+1} with the value of the slope but this modification doesn't bring a big change in the total number of iterations.

The stepsize chosen previously with the increment determined by polynomial interpolation gives the complete load-displacements curves of the Lee's frame. The next section will be only concerned with the amelioration of the method.

4.4.3 General improvements

The 2 ameliorations, discussed in this section, will concern the precision of the curves and the speed of convergence of the Newton-Raphson's method.

4.4.3.1 Approximation of extrema

In engineering practice, the determination of the buckling point is quite important. It occurs at the first extremum of the load. The polynomial interpolation, which allowed us to find P_s and Q_s, will approximate this extrema for a load corresponding to the points where the values of the derivatives of P and Q are zero. In the same way we will estimate all the extrema for u, w and F. The extrema for u or w occur at points where the derivative of P or Q are infinite. In order to palliate this inconvenience we will then consider the polynomials R and S which interpolate $u = h(F)$ and $w = k(F)$,

and the extrema will be for the points where the derivatives of R or S is zero.

4.4.3.2 Prediction of the solution

The deformations of the Lee's frame are represented by 44 variables which are all the displacements and rotations at each node. Therefore we have to find a prediction for all these variables.

Let us suppose that u_p is the horizontal displacement at the point P (Fig. (1.1)) and is the next increment. The value of the stepsize found is Δu_p . For the node j (j=0, ..., 11), we will construct 4 polynomials PU_1, PU_2, PU_3 and PU_4 approximating the functions U_1, U_2, U_3 and U_4 defined by:

$$
\begin{aligned}
u_j &= U_1(u_p) \\
u_{j,x} &= U_2(u_p) \\
w_j &= U_3(u_p) \\
w_{j,x} &= U_4(u_p) \quad .
\end{aligned}
$$

The prediction of the new position for the node j will be:

$$
\begin{aligned}
u_j &= PU_1(u_p + \Delta u_p) \\
u_{j,x} &= PU_2(u_p + \Delta u_p) \\
w_j &= PU_3(u_p + \Delta u_p) \\
w_{j,x} &= PU_4(u_p + \Delta u_p)
\end{aligned}
$$

The last operation will consist of the prediction of the new value for the load F .

All of these preceding strategies are implemented in our program, and the algorithms and results will be shown in the next chapter.

CHAPTER 5

RESULTS

§5.1 GENERALITIES

In this chapter we shall present the numerical program which is broken into many succinct algorithms. In this way, our presentation will allow one to understand the different stages of this program. The application of this program to the Lee's frame will be done in the last part of this chapter.

§5.2 FLOWCHARTS

5.2.1 Introduction

To comprehend the following flowcharts, one needs first to define some variables.

IPAS : represents the number of the solution considered

NITER: determines the maximum number of iterations

SMIN : represents the value of the minimum stepsize

SMAX : represents the value of the maximum stepsize

STEP : represents the value of the stepsize at the iteration considered

DIV : indicates if we obtained a convergence or not for the previous solution; DIV = 0 means we obtained convergence; DIV = 1 means we had divergence for the solution considered but we obtained convergence for the preceding solution; DIV = 2 means we had divergence for the second time in a row; DIV = 3 means we had divergence for the third time in a row with the minimum stepsize and this value of DIV stops the program.

{VFF}: is the vector of the n last values obtained for the force F when using polynomial interpolation of order n.

{VUU}: is the vector of the n last values obtained for the horizontal displacement at the point P.

{VWW}: is the vector of the n last values obtained for the vertical displacement at the point P.

{U} : is the vector solution and has 44 components.

u_I : indicates the value of the horizontal displacement at the node I.

NIW : represents the number of iterations we would like to have at each step.

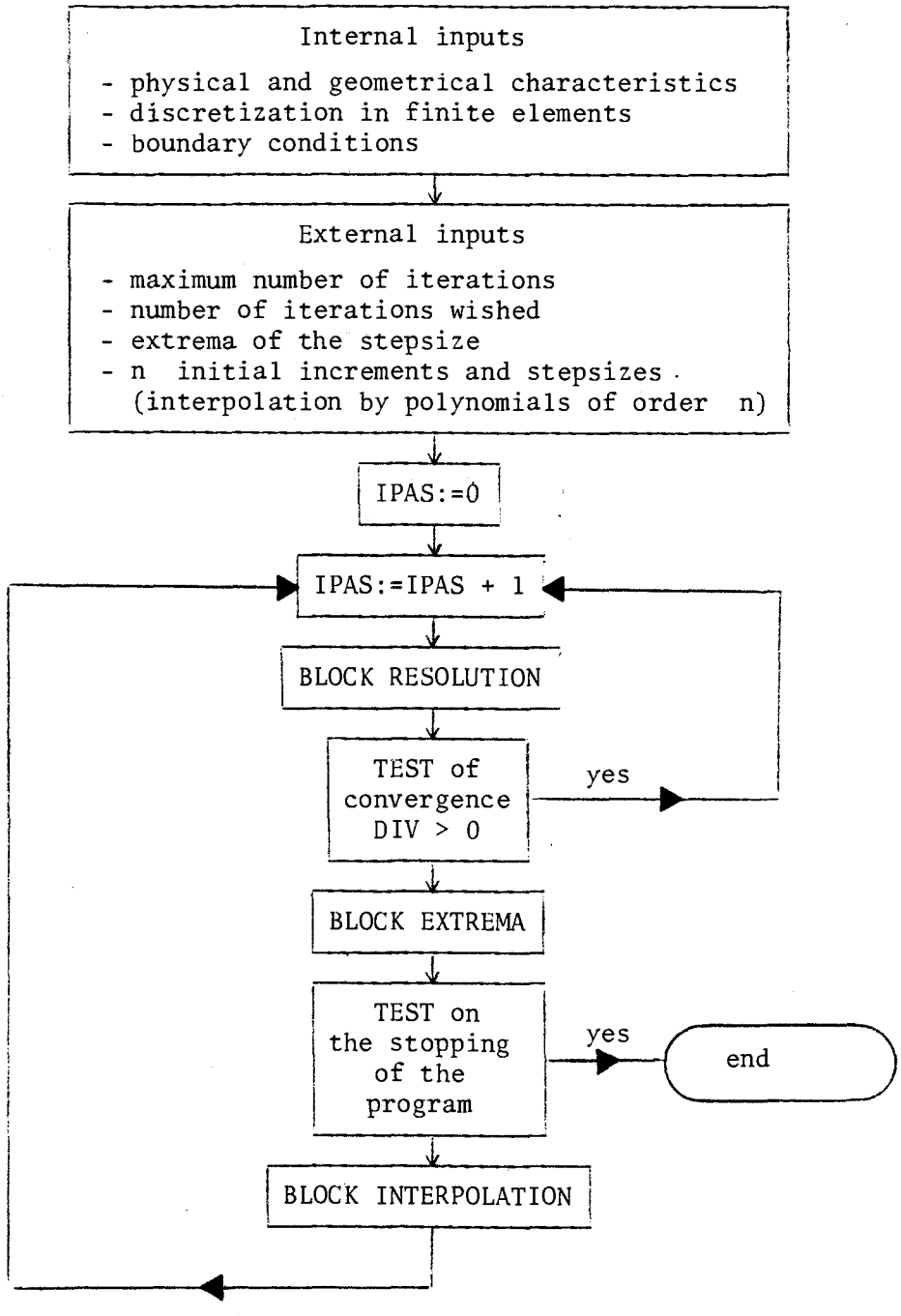
β : if the absolute value of the tangent of one of the load-displacement curve is greater than β, we divide the stepsize by 2.

α : if the absolute value of the tangent of one of the load-displacement curve is smaller than α , we divide the stepsize by 2.

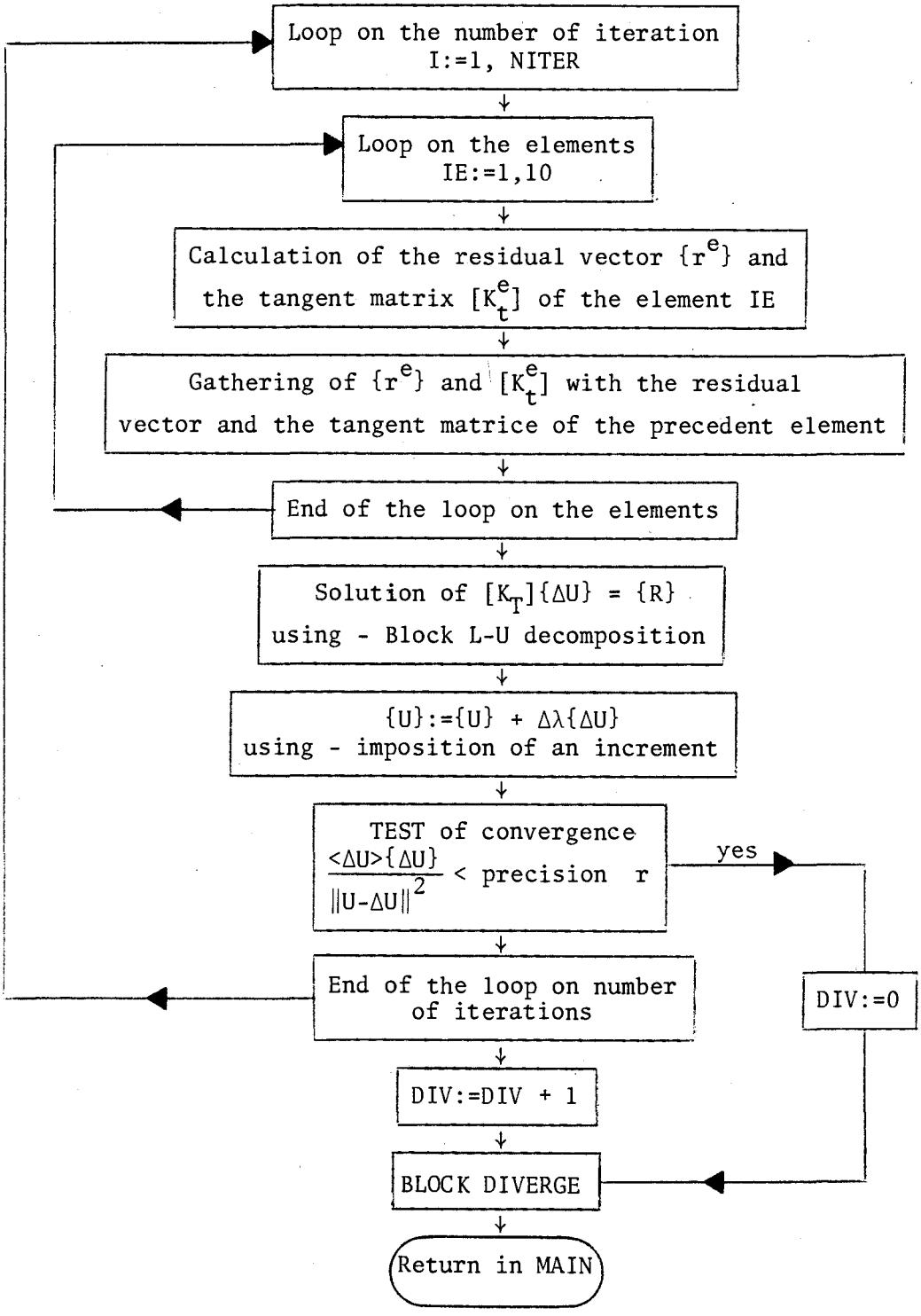
We will present 7 flowcharts:

- Algorithm of the main program.
- Algorithm of the block "resolution": this algorithm is concerned with the solution of the equation $[K]\{U\} = \{F\}$.
- Algorithm of the block "diverge": this algorithm studies the change of stepsize in a case of divergence.
- Algorithm of the block "predict": this algorithm is used for the prediction of the solution.
- Algorithm of the block "extrema": this algorithm calculates the values of the extrema.
- Algorithm of the block "interpolation": this algorithm defines a polynomial interpolation using the 3 last points obtained.
- Algorithm of the block "stepsize": this algorithm defines the value of the stepsize.

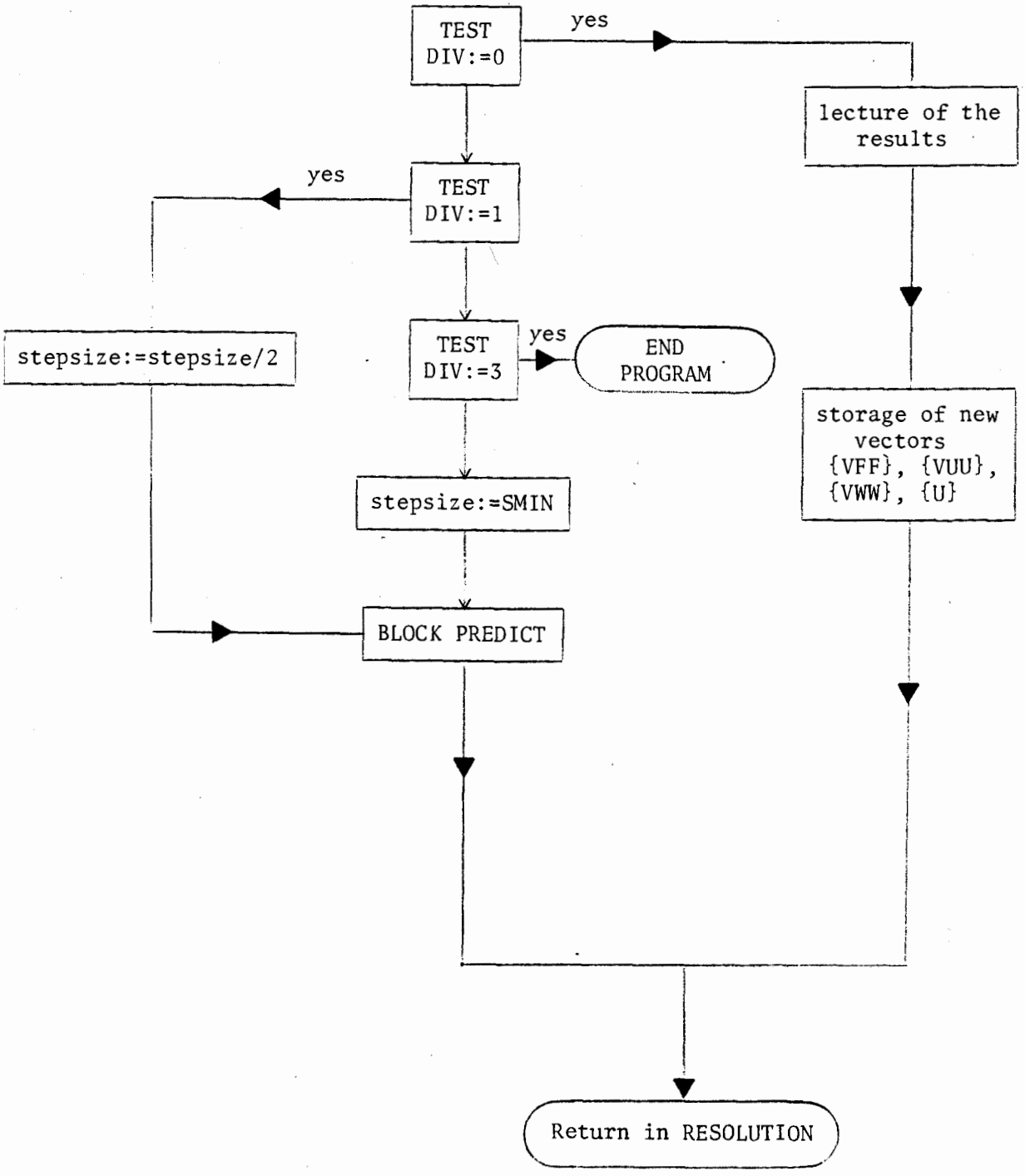
5.2.2 Flowchart of the main program



5.2.3 Flowchart of the "BLOCK RESOLUTION"

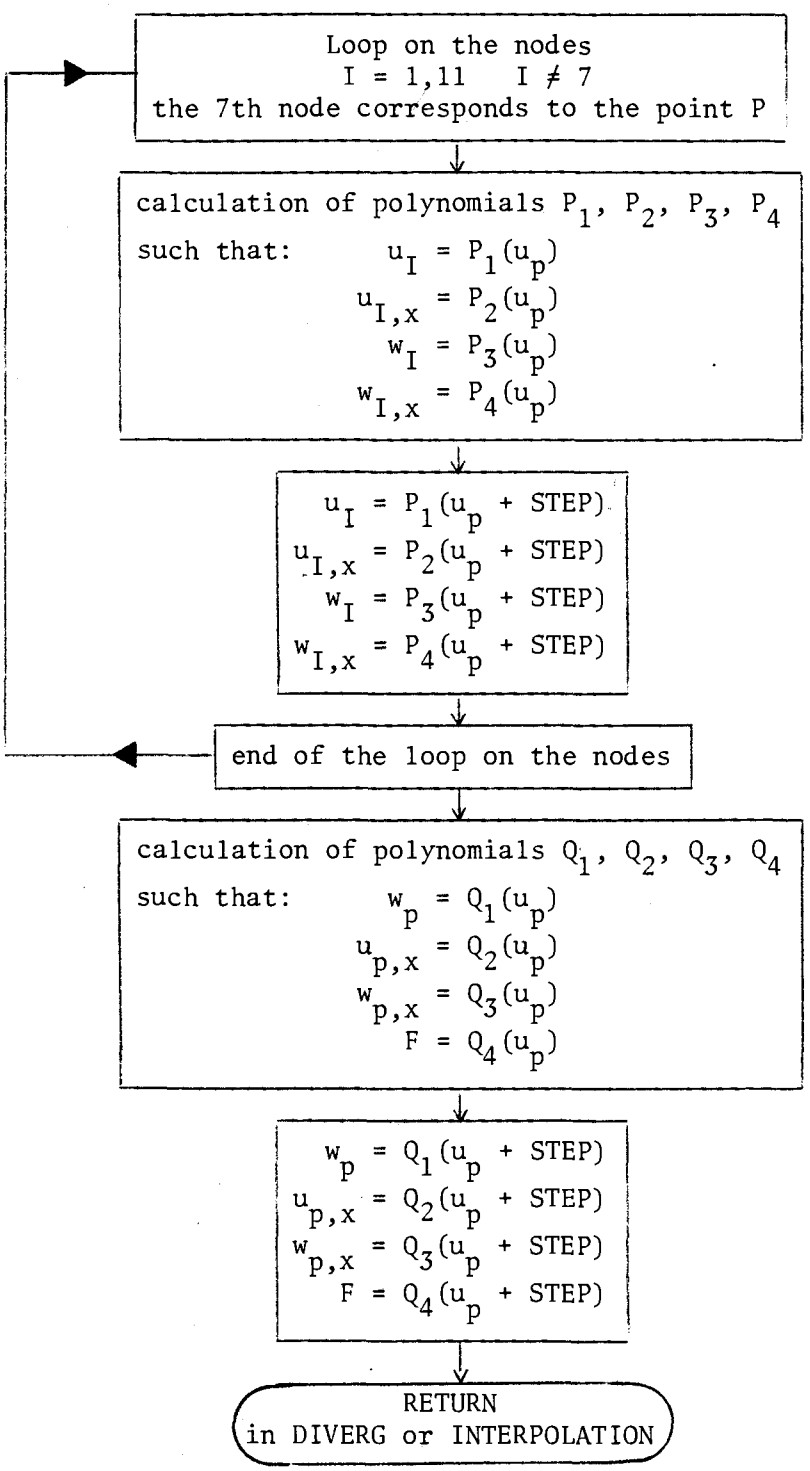


5.2.4 Flowchart of the "BLOCK DIVERGE"

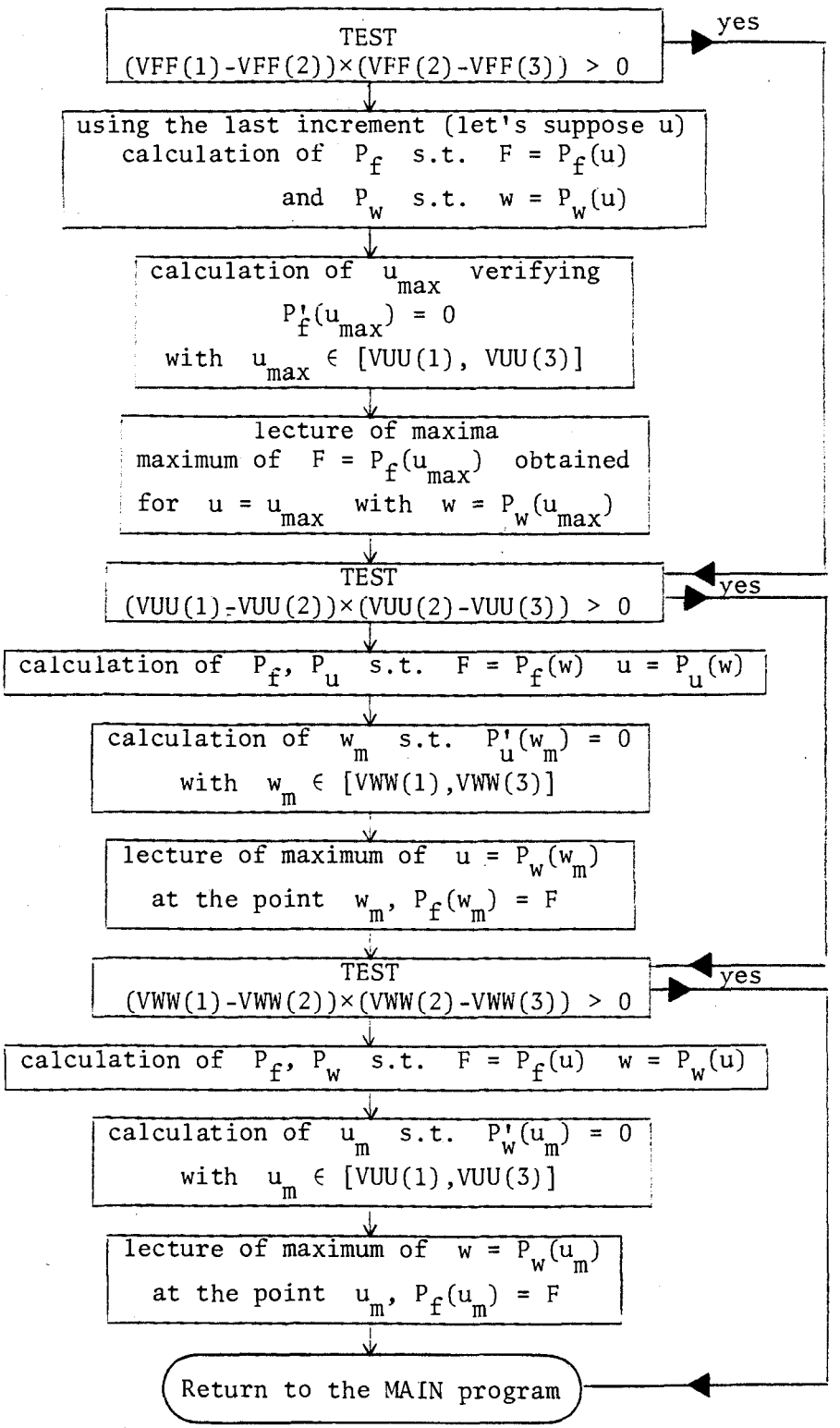


5.2.5 Flowchart of the "BLOCK PREDICT"

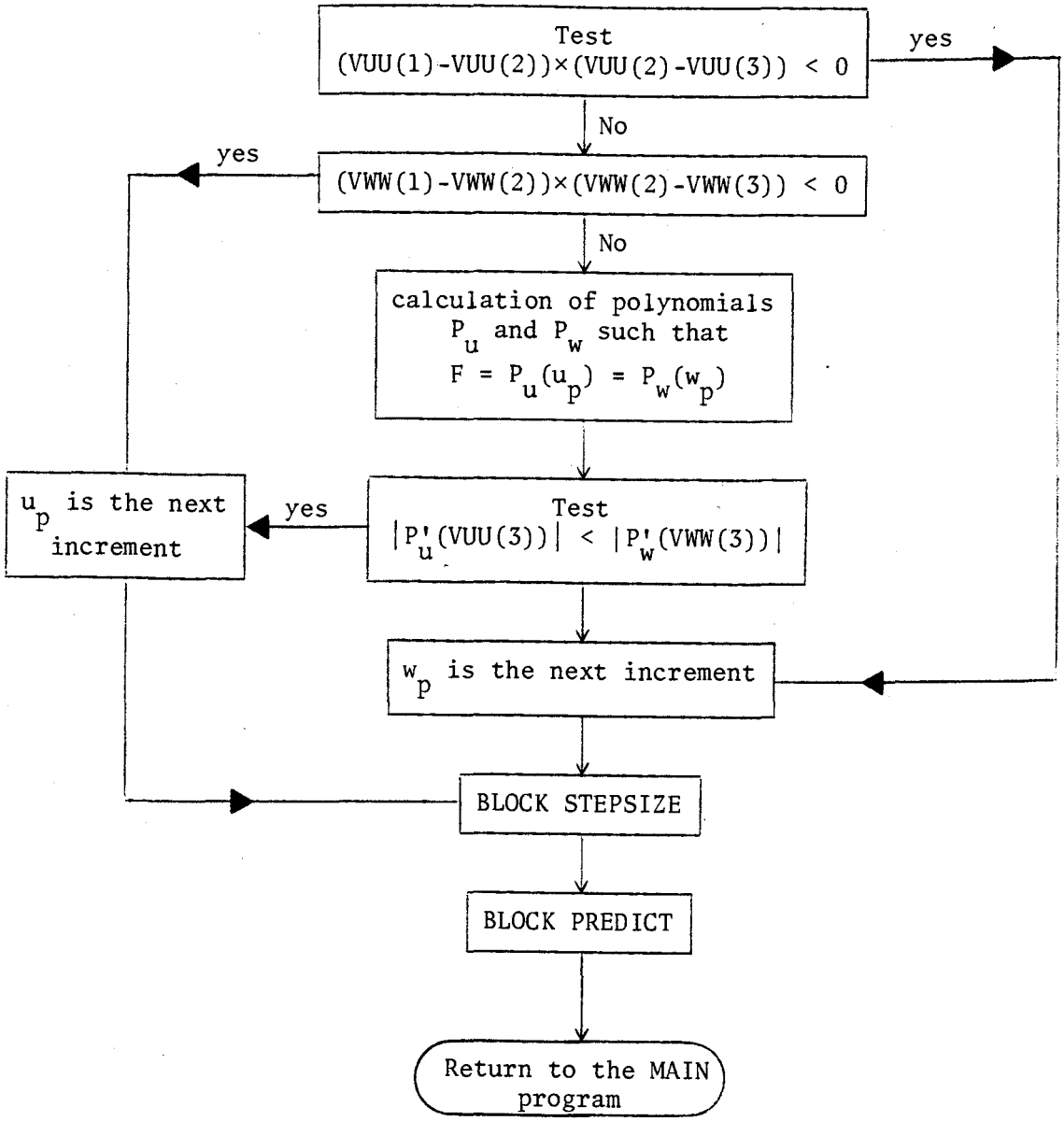
Let's suppose u_p is the increment.



5.2.6 Flowchart of the "BLOCK EXTREMA"

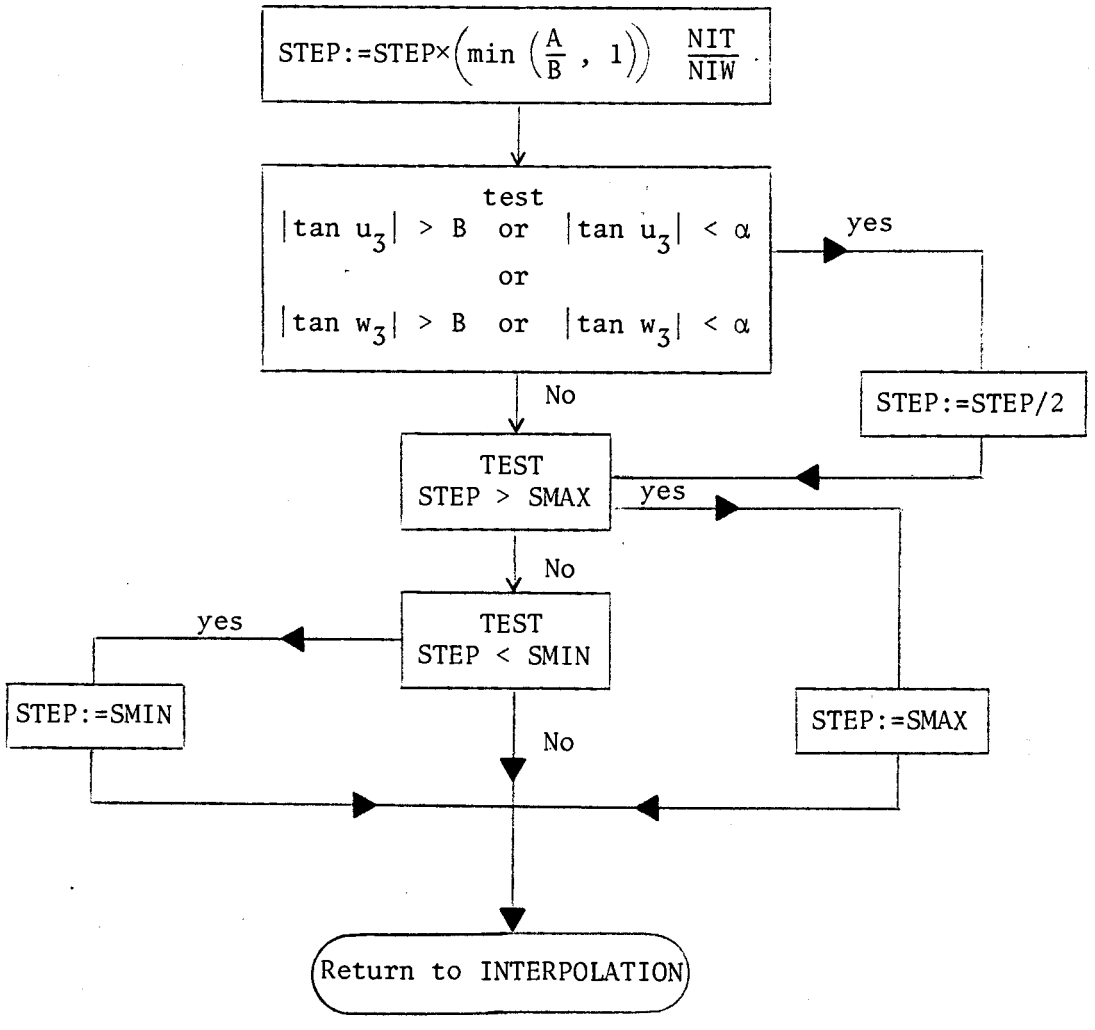


5.2.7 Flowchart of the "BLOCK INTERPOLATION"



5.2.8 Flowchart of the "BLOCK STEPSIZE"

Let's suppose the value of the new increment at the last point is A ($A = u_3$ or $A = w_3$) (u_3 = VUU(3) and w_3 = VWW(3)), the value of the preceding increment at the last point is B ($B = u_3$ or $B = w_3$), and NIT is the value of the number of iteration at the previous step.



§5.3 PRESENTATION OF THE RESULTS

The main purpose of this thesis is to create a program which allows us to determine the load-displacement curves of the Lee's frame. This objective has been realized by using a polynomial interpolation of order 3 or 4. The program, with the algorithms given in the previous section, allows us to find the load-displacement curves without any failure. Thus, the next step concerns the optimization of this program. In order to try to reduce the cost of a run we use polynomial interpolation to predict the solution. The precision of the curves can be increased by using polynomial interpolation to find the extrema of the curves. In order to appreciate the precision of the extrema found by polynomial interpolation we must first find the load-displacement curves with exact values for these extrema. This discussion will be done in the next section. All the curves and the tables presented below correspond to the displacements at the point P of the frame (fig. (1.1)).

5.3.1 Load-displacement curves

In order to determine the extrema precisely we supplied a program which allows the user to choose the next increment and stepsize. The values we obtained for the extrema are shown in the table (5.1). The precision on these extrema, nevertheless, depends on the value of the precision chosen for the Newton-Raphson convergence criterion. The values obtained for the curves are shown in the table (5.2), and the schematization of the curves is done on the figure (5.2). The figure (5.1) shows some deflected shapes of the Lee's

EXTREMUM	LOAD F (expressed in Newton)	Horizontal displacement u_p (in meters)	Vertical displacement w_p (in meters)
Local maximum of the load (Buckling point)	1.87160	0.2719	0.4875
Local maximum of the vertical displacement w_p	1.194	0.6231	0.61023
Local minimum of the vertical displacement w_p	-0.462	0.8128	0.50827
Local minimum of the load F	-0.97420	0.9028	0.5820
Local minimum of the horizontal displacement u_p	-0.706	0.94503	0.7140

TAB. 5.1 - Extrema of the curves load-displacement

Load F (in Newton)	Horizontal displacement u_p (in meters)	Vertical displacement w_p (in meters)
0.8	0.0101	0.073
0.96	0.0171	0.1
1.35	0.0528	0.2
1.54	0.0867	0.27
1.75	0.1522	0.37
1.85	0.2208	0.445
1.85	0.3201	0.5198
1.68	0.4451	0.5777
1.51	0.5201	0.5984
1.37	0.5701	0.6068
1.20	0.6201	0.6102
1.00	0.6701	0.6070
0.74	0.7201	0.5936
0.57	0.7451	0.5809
0.34	0.7701	0.5612
-0.04	0.7951	0.5276
-0.39	0.8094	0.5091
-0.56	0.8189	0.5093
-0.79	0.8439	0.5232
-0.95	0.8811	0.5556
-0.94	0.9252	0.6211
-0.74	0.9449	0.7044
-0.61	0.9429	0.7444
-0.39	0.9304	0.7944
-0.07	0.9066	0.8444
0.17	0.8908	0.8694
0.58	0.8734	0.8944
1.48	0.8595	0.9194
2.53	0.8590	0.9319

TAB. 5.2 - Variations of F, u_p and w_p

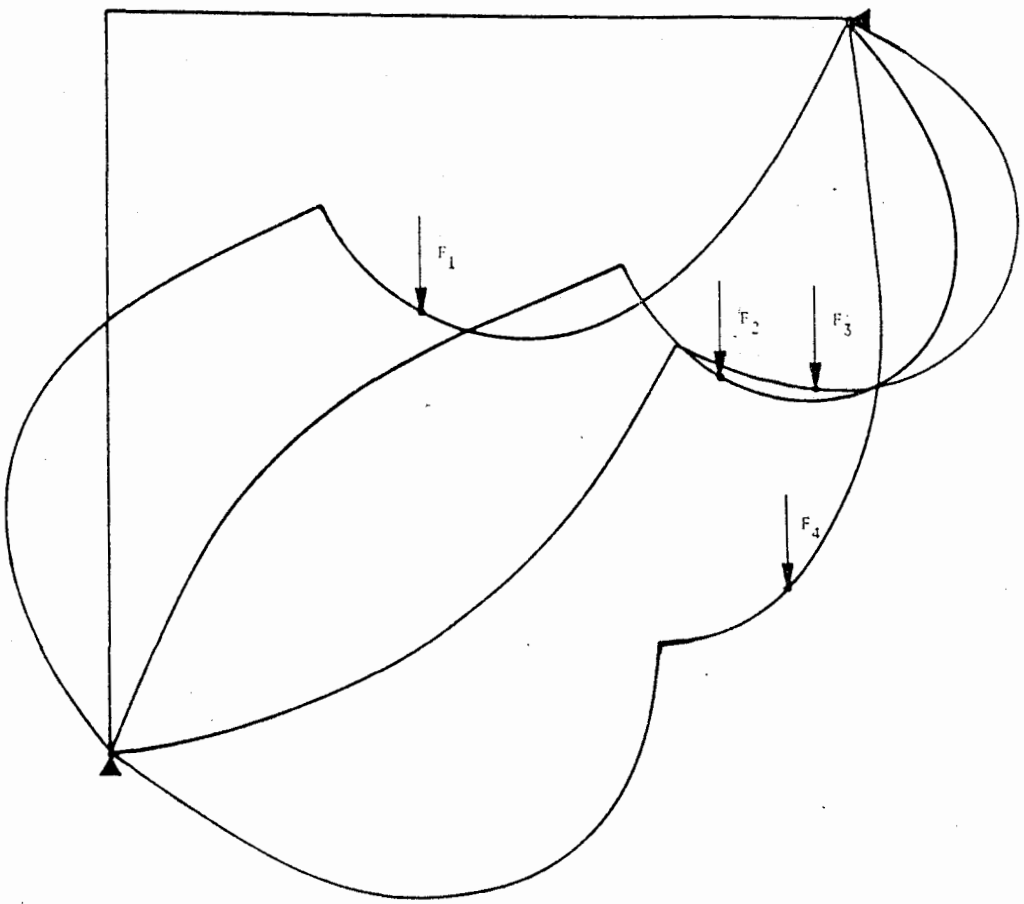


Fig. 5.1 - Different shapes of the Lee's frame

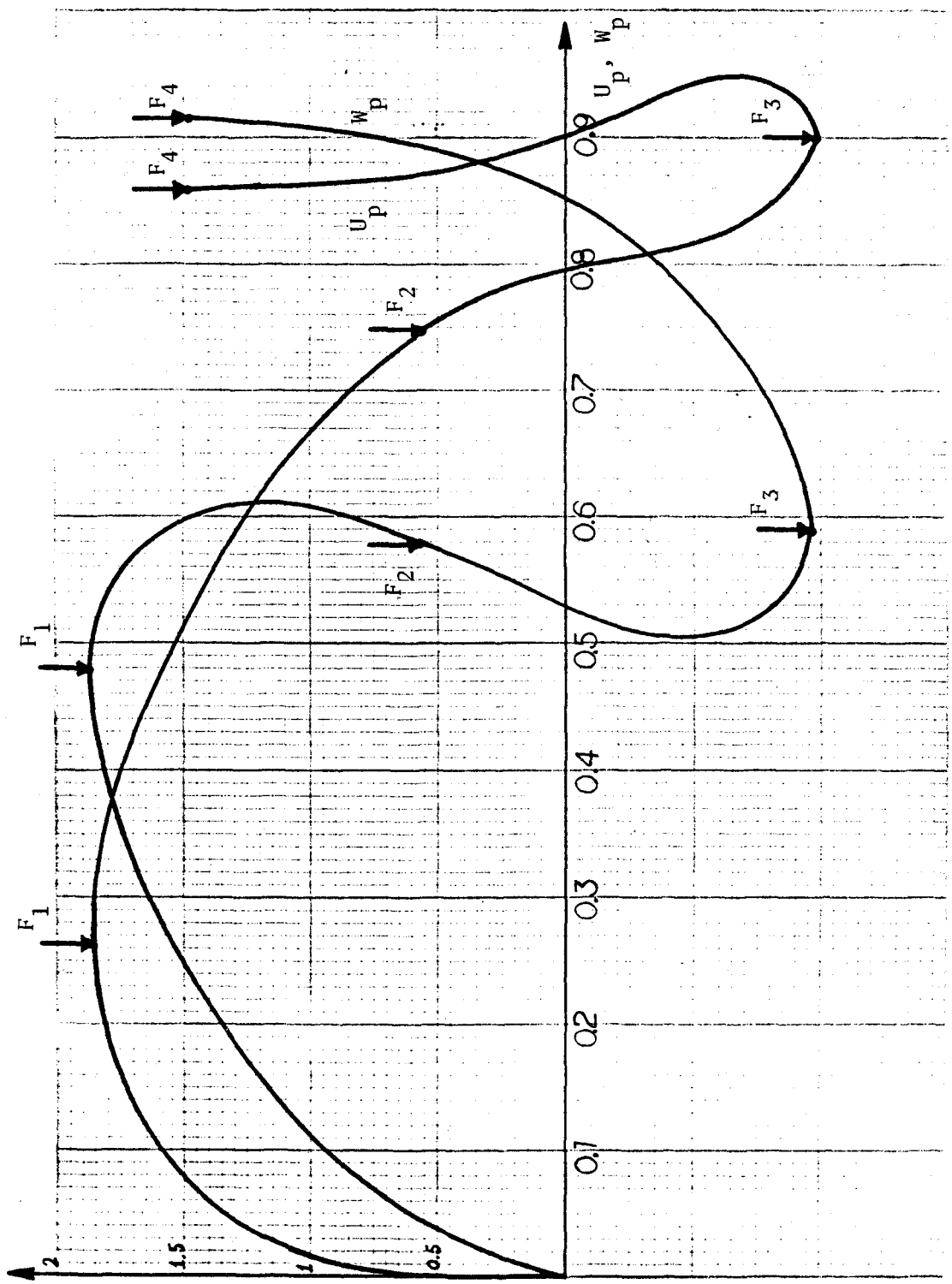


Fig. 5.2 - Load-displacement curves

frame at some points pointed in the figure (5.1).

5.3.2 Comparison of procedures

In order to be able to understand the following results we will briefly define some values.

The "preceding number of iterations" (NIT) represents how many times we solved the equation $[K]\{U\} = \{F\}$ before obtaining the last point from the point preceding it. Then the value of the stepsize for the following point will be an inverse factor of NIT.

The "total number of iterations" represents how many times we solved the equation $[K]\{U\} = \{F\}$ before obtaining all the points needed to schematize the curves load-displacement.

Sometimes it appears that the value of the stepsize (between two points) is too big then we obtain a divergence (no convergence of the Newton-Raphson's method before the maximum number of iterations fixed by the user). The program will then come back to the previous point and try again with a smaller stepsize. The "total number of divergences" represents how many times we get a divergence before the obtainment of the curves.

The first comparison will be done roughly but will indicate the utility of the prediction.

5.3.2.1 Prediction in Newton-Raphson method

A first test has been done for a dozen of steps, using a very small stepsize (STEP = 0.01). For each step the prediction method allowed us to obtain the convergence in 2 or 3 iterations and the simple Newton Raphson's needed 7 or 8 iterations before the convergence.

This test seems already to have proved the merit of the prediction method. But it would be wrong to conclude now of the superiority of this method because the values of the stepsize have a big influence on the convergence and we are not sure that the values we used don't advantage a lot the prediction method over the other one. To be absolutely sure of our conclusion we must use some other values for the stepsize. Therefore we tried 2 more tests.

The first one used a stepsize depending only on the preceding number of iterations; the second one used a stepsize also depending on the slope of the curves. With the method without prediction we obtained the curves in respectively 450 and 550 iterations (total number of iterations). Using the method with prediction we needed only 150 iterations in both cases.

The previous examples clearly prove the superiority of the method with prediction which is also more reliable than the simple Newton-Raphson method. To obtain the curves we got, with this method, only 1 or 2 divergences (total number of divergences). When we didn't use the

prediction we got globally 15 divergences before the obtainment of the curves.

The next section will be concerned by a comparison between 2 different predictions, a prediction done in approximating the deformations by a polynomial of order 3 and a prediction done in approximating the deformations by a polynomial of order 4.

5.3.2.2 Comparison of the order of the polynomial interpolation

A general observation given by the results concerns the stepsize strategy (or the method we will use to determine the value of the stepsize between 2 points). The simple Newton-Raphson's method needs a well defined stepsize in order to obtain good results. This need implies an important study for the length of the stepsize which influences a lot on the total number of iterations. The interval of admissible stepsizes is very small and a value of the stepsize outside this interval would immediately lead to a divergence. In the opposite direction the convergence is almost always obtained for any stepsize when using the Newton-Raphson's method with prediction. In this case, due to the rapidity of convergence, the value of the stepsize determined by our algorithm is very often equal to the maximum stepsize SMAX . Therefore the values chosen for NIW, α, and β don't bring big modifications. We will fix NIW=5, α=10 and β=0.5. The only factor concerning the precision and the cost of a run will then be the value of the maximum stepsize. The tables (5.3), (5.4), (5.5) and (5.6) show

MAXIMUM STEPSIZE = 0.025					
Order of Polynomial interpolation		4		3	
Total number of iterations		201		213	
Buckling point					
F	$\Delta F/F$	1.87160	0	1.87161	0
u_p	$\Delta u_p/u_p$	0.2708	0.004	0.2712	0.003
w_p	$\Delta w_p/w_p$	0.4867	0.002	0.4860	0.003
Local maximum of w_p					
F	$\Delta F/F$	1.193	0.001	1.191	0.003
u_p	$\Delta u_p/u_p$	0.6233	0	0.6227	0.001
w_p	$\Delta w_p/w_p$	0.61023	0	0.61024	0
Local minimum of w_p					
F	$\Delta F/F$	-0.474	0.026	-0.467	0.011
u_p	$\Delta u_p/u_p$	0.8141	0.002	0.8139	0.001
w_p	$\Delta w_p/w_p$	0.50808	0	0.50849	0
Local minimum of F					
F	$\Delta F/F$	-0.97352	0.001	0.97436	0
u_p	$\Delta u_p/u_p$	0.9022	0.001	0.9024	0
w_p	$\Delta w_p/w_p$	0.5786	0.006	0.5829	0.002
Local maximum of u_p					
F	$\Delta F/F$	-0.711	0.007	-0.708	0.003
u_p	$\Delta u_p/u_p$	0.94503	0	0.94503	0
w_p	$\Delta w_p/w_p$	0.7131	0.001	0.7131	0.001

TAB. 5.3 - Extrema for SMAX = 0.025

MAXIMUM STEPSIZE = 0.05					
Order of polynomial interpolation		4		3	
Total number of iterations		150		154	
Buckling point					
F	$\Delta F/F$	1.87160	0	1.87154	0
u_p	$\Delta u_p/u_p$	0.2708	0.004	0.2720	0
w_p	$\Delta w_p/w_p$	0.4865	0.002	0.4842	0.007
Local maximum of w_p					
F	$\Delta F/F$	1.195	0.001	1.189	0.004
u_p	$\Delta u_p/u_p$	0.6228	0	0.6220	0.002
w_p	$\Delta w_p/w_p$	0.61020	0	0.61034	0
Local minimum of w_p					
F	$\Delta F/F$	-0.468	0.013	-0.476	0.030
u_p	$\Delta u_p/u_p$	0.8156	0.003	0.8136	0.001
w_p	$\Delta w_p/w_p$	0.50894	0.001	0.49666	0.023
Local minimum of F					
F	$\Delta F/F$	-0.97496	0.001	-0.97426	0
u_p	$\Delta u_p/u_p$	0.9017	0.001	0.9010	0.002
w_p	$\Delta w_p/w_p$	0.5795	0.004	0.5862	0.007
Local maximum of u_p					
F	$\Delta F/F$	-0.711	0.007	-0.701	0.007
u_p	$\Delta u_p/u_p$	0.94504	0	0.94462	0.004
w_p	$\Delta w_p/w_p$	0.7131	0.001	0.7134	0.001

TAB. 5.4 - Extrema for SMAX = 0.05

MAXIMUM STEPSIZE = 0.1					
Order of polynomial interpolation		4		3	
Total number of iterations		144		143	
Buckling point					
F	$\Delta F/F$	1.87135	0	1.87097	0
u_p	$\Delta u_p/u_p$	0.2704	0.006	0.2763	0.016
w_p	$\Delta w_p/w_p$	0.4854	0.004	0.4750	0.026
Local maximum of w_p					
F	$\Delta F/F$	1.196	0.002	1.192	0.002
u_p	$\Delta u_p/u_p$	0.6223	0.001	0.6229	0
w_p	$\Delta w_p/w_p$	0.61030	0	0.61019	0
Local minimum of w_p					
F	$\Delta F/F$	-0.465	0.006	-0.291	0.370
u_p	$\Delta u_p/u_p$	0.8233	0.013	0.8113	0.002
w_p	$\Delta w_p/w_p$	0.51287	0.009	0.50319	0.010
Local minimum of F					
F	$\Delta F/F$	-0.97231	0.002	-0.97404	0
u_p	$\Delta u_p/u_p$	0.9019	0.001	0.9024	0
w_p	$\Delta w_p/w_p$	0.5790	0.005	0.5834	0.002
Local maximum of u_p					
F	$\Delta F/F$	-0.716	0.014	-0.688	0.025
u_p	$\Delta u_p/u_p$	0.94502	0	0.94483	0.002
w_p	$\Delta w_p/w_p$	0.7130	0.001	0.7136	0.001

TAB. 5.5 - Extrema for SMAX = 0.1

MAXIMUM STEPSIZE = 1					
Order of polynomial interpolation		4		3	
Total number of iterations		160		178	
Buckling point					
F	$\Delta F/F$	1.87079	0	1.87192	0
u_p	$\Delta u_p/u_p$	0.2653	0.024	0.2738	0.007
w_p	$\Delta w_p/w_p$	0.4706	0.035	0.4798	0.016
Local maximum of w_p					
F	$\Delta F/F$	1.300	0.089	1.140	0.045
u_p	$\Delta u_p/u_p$	0.6218	0.002	0.6066	0.026
w_p	$\Delta w_p/w_p$	0.61022	0	0.61110	0.001
Local minimum of w_p					
F	$\Delta F/F$	-0.460	0.004	-0.511	0.106
u_p	$\Delta u_p/u_p$	0.8171	0.005	0.7945	0.023
w_p	$\Delta w_p/w_p$	0.51289	0.009	0.50909	0.002
Local minimum of F					
F	$\Delta F/F$	-0.97478	0.001	-0.97423	0
u_p	$\Delta u_p/u_p$	0.9017	0.001	0.9026	0
w_p	$\Delta w_p/w_p$	0.5791	0.005	0.5821	0
Local maximum of u_p					
F	$\Delta F/F$	-0.756	0.071	-0.705	0.001
u_p	$\Delta u_p/u_p$	0.94498	0.001	0.94460	0.004
w_p	$\Delta w_p/w_p$	0.7125	0.002	0.7128	0.002

TAB. 5.6 - Extrema for SMAX = 1

results obtained from 4 different values for the maximum stepsize. The tabulated values are the extrema of the curves, their relative errors and the total number of iterations. The 2 first values will allow us to evaluate the precision on the extrema of the curves. The last value of the table allows us to approximate the cost of a run.

The comparison of the total number of iterations must be done roughly because the end points are not always the same. This fact can bring as far as a difference of 8 iterations (on the total number of iterations). It appears then that the total number of iterations of a run doesn't depend very much on the order of the polynomial interpolation used. The main difference appears when the value of the maximum stepsize is 1 and, in this case, the polynomial interpolation of order 3 needs only 10 per cent more iterations. Furthermore this value of the maximum stepsize doesn't allow a good drawing of the load-displacement curves because we didn't obtain enough points. Therefore we will restrict our comparison to smaller values of the maximum stepsize. These values don't bring a big difference to the values obtained for the extrema of the curves. In fact the accuracy of our results for the extrema seems to depend in major part on the distribution of the points found and their proximity to the extrema.

Therefore the more elaborate polynomial interpolation of order 4 doesn't seem to prevail on the polynomial interpolation of order 3 and we can suspect that we would have the same conclusion for a higher order polynomial interpolation.

The value of the maximum stepsize appears to be more influential although all the stepsizes gave a very good approximation of the curves. We obtained the best ratio (number of points obtained/number of iterations needed) for a value of 0.05 for the maximum stepsize.

CHAPTER 6

CONCLUSION

The first and primordial objective of this thesis--which was to find a way to obtain the load-displacement curves of the Lee's frame--has been solved by using a program with an automatic incrementation based on polynomial interpolation. The polynomial interpolation technique has also been used in the following connections:

i. to approximate the extrema of the curves, and

ii. to predict an initial value for the solution in order to improve the rapidity of convergence of the Newton-Raphson's method.

In the first instance, the estimation of the extrema is interesting only when a small number of points on the load-displacement curve are known. Since the task of finding the extrema can be accomplished after the curve has been found, we do not recommend the use of polynomial interpolation method for this purpose alone.

As for the second point, the prediction brought a tremendous improvement in the speed of convergence of Newton-Raphson's method, and furthermore, it eliminated most of the risks of divergence. It is our opinion that the polynomial interpolation technique should be used as often as possible in making an initial prediction of the solution. This is not quite the case for the automatic incrementation scheme as it is not useful for the situations where the load-displacement curves have no local extrema.

The final conclusion is that: this thesis has reached its objective. Its main contribution resides in the "prediction" method, which is very effective and henceforth should be used for solving nonlinear static problems.

REFERENCES

[1] Timoshenko, S., and Goodier, Theory of Elasticity, 3rd edition, McGraw-Hill, 1970.

[2] Wang, C., Applied Elasticity, McGraw-Hill, 1953.

[3] Sayir, M. and Mitropoulos, C., On Elementary Theories of Linear Elastic Beams, Plates and Shells, Journal of Applied Mathematics and Physics (Z.A.M.P.), Vol. 31, 1979.

[4] Batoz, J.L., Cours de DEA sur les grandes deformations en formulation lagrangienne totale, Compiegne University, France.

[5] Zienkiewicz, O.C., The Finite Element Method in Engineering Science, McGraw-Hill, 3rd edition, 1977.

[6] Gallagher, R.H., Finite Element Analysis Fundamentals, Prentice-Hall, 1975.

[7] Bathe, K.J., Finite Element Procedures in Engineering Analysis, Prentice-Hall, 1982.

[8] Dhatt, G. and Touzot, G., Une presentation de la methode des elements finis, Maloine, 1981.

[9] Strang, G. and Fix, G.J., An Analysis of the Finite Element Method, Prentice-Hall, 1973.

[10] Oden, J.T. and Reddy, J.N., An Introduction to the Mathematical Theory of Finite Elements, John Wiley & Sons, 1976.

[11] Burden, R.L., Faires, J.D. and Reynolds, A.C., Numerical Analysis, 2nd edition, Prindle, Weber & Schmidt, 1981.

[12] Davis, P.J. and Rabinowitz, P., Methods of Numerical Integration, Academic Press, 1975.

[13] Engels, H., Numerical Quadrature and Cubature, Academic Press, 1980.

[14] Batoz, J.L., Grands deplacements et grandes rotations de poutres minces elastiques, Report, Compiegne University, France, 1981.

[15] Godsil, C., Linear Analysis, Lecture Notes, Simon Fraser University, 1983.

[16] Golub, G.H. and Van Loan, C.F., Matrix Computations, The Johns Hopkins University Press, 2nd printing, 1984.

[17] Batoz, J.L. and Dhatt, G., Une evaluation des methodes du type Newton-Raphson imposant l'accroissement d'un deplacement, Report, Laval and Compiegne Universities, 1978.

[18] Deuflhard, P., A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques, Numerische Mathematik, 1979.