



الدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

Département de Génie civil

Projet de Fin d'Etudes

Pour l'Obtention du Diplôme d'Ingénieur d'Etat en Génie Civil

Thème

ETUDE ET OPTIMISATION DES STRUCTURES EN  
PORTIQUES PLANS.  
(Réalisation d'un programme informatique *OPTIPORT*)

Etudié par :  
ABDELLAH EL HADJ Mohamed  
KARALI Rabie

Proposé et dirigé par :  
Dr. K. Silhadi  
Mr. B. Mezazigh

# Plan général



- 1.introduction générale
- 2.méthode des éléments finis
- 3.mise en œuvre de l'optimisation
- 4.présentation du programme OPTIPORT
- 5.application
- 6.conclusion

# شكر

نحمد الله العليّ القدير أولاً و آخراً على ما منّ به علينا من التوفيق لإتمام هذا العمل .  
ثم :

نشكر الوالدين الكريمين اللذين يرجع إليهما الفضل بعد الله عزّ و جلّ.

كما نتقدم بالشكر للأستاذ كمال سي الهادي و الأستاذ إبراهيم ميزازيغ على إقتراحهما لهذا الموضوع و تأطيرهما للعمل و بذلهما للتصائح و الإرشادات طوال مدة إنجاز المذكرة و نشكر أيضا أعضاء اللجنة المشرفة على تفحصها و تقييمها للعمل.

و لا يفوتنا شكر عمّال مكتبة المدرسة و عمّال مكتبة المركز الجامعي بالمدينة.

واخيرا نشكر الطالبين يونس أوس و أنيس حاج هنيّ و المهندس رفيق كراي و ابن العمّ محمد عبد الله الحاج وكل من ساعد على إنجاز هذا العمل.

## إهداء

إلى الوالدين الكريمين حفظهما الله  
و الإخوة وزوجاتهم و الأخوات و أزواجهن  
و جميع أفراد العائلة.  
و جميع الزملاء و الأصحاب...

محمد الأمين عبد الله الحاج

# إهداء

إلى الأم  
و الأب  
و الجدة  
بارك الله في أعمارهم  
و جميع الأقارب والأصدقاء...

كرالي ربيع

## الموضوع: تحليل و تخفيض المنشآت الإطارية في المستوي

**ملخص:** إن الهدف من هذه الدراسة هو إنجاز برنامج إعلام آلي يقوم بإعطاء أقل كتلة ممكنة للمنشآت الإطارية ذات عناصر لها مقاطع مستطيلة وتصرف خطي و ذلك بإعطاء الأبعاد المخفضة للعناصر المكونة لهذه المنشآت. يقوم هذا البرنامج بعمليتين أساسيتين هما: تحليل المنشأة باستعمال طريقة العناصر المنتهية لتحديد الإجهادات الأعظمية ثم تخفيض أقل ما يمكن عناصر هذه المنشأة باستعمال الطريقة التسلسلية ذات الدرجة الثانية بحيث تصبح الإجهادات أقرب ما يمكن من الحد المسموح به. وقد تم تجريب هذا البرنامج على منشآت مما أثبت نجاعته.

**كلمات مفتاحية:** عناصر منتهية, تسلسلية ذات درجة ثانية, تخفيض أقل ما يقل ما يمكن, إجهاد, مخطط, تقارب.

**Subject: Analysis and optimization of plane frame structures.**

**Abstract:** The objective of this study is the realization of a computer program which optimizes plane frame structures constituted of elements with rectangular cross-sections and with linear behavior. This program carries out the two following operations: analysis of structures using the finite element method in order to determine the maximum stresses, then to optimize the sections of the structure by using the sequential quadratic method to obtain the stresses closest to the ultimate one. This program has been tested on several structures and has showed its validity.

**Key words** finite elements, sequential quadratic, optimization, constraint, flow chart, convergence.

**Sujet : Etude et optimisation des structures en portiques plans.**

**Résumé:** L'objectif de cette étude est la réalisation d'un programme qui optimise les structures en portiques plan constituées d'éléments à sections droites rectangulaires et à comportement linéaire. Ce programme effectue les deux opérations suivantes: analyse de la structure à l'aide de la méthode des éléments finis afin de déterminer les contraintes maximales, ensuite optimiser les sections de la structure en utilisant la méthode séquentielle quadratique pour obtenir les contraintes les plus proches de la contrainte limite. Ce programme a été testé sur plusieurs structures et a montré sa validité.

**Mots clés:** éléments finis, séquentielle quadratique, optimisation, contrainte, organigramme, convergence.

Introduction générale:.....	1
-----------------------------	---

## Chapitre 1 Méthode des éléments finis

I.1. INTRODUCTION.....	3
I.2. APERÇU HISTORIQUE .....	3
I.3. ASPECTS GÉNÉRAUX DE LA MÉTHODE DES ÉLÉMENTS FINIS :.....	4
I.3.1. Description générale:.....	4
I.3.2. Démarche éléments finis:.....	5
I.4. FORMULATION DE LA MEF DANS LE CAS DES PORTIQUES:.....	6
I.4.1. Matrice de rigidité d'un élément portique.....	6
a) Élément barre.....	6
b) Élément poutre.....	7
c) Élément portique.....	7
d) Changement du repère.....	8
e) Matrice de rigidité dans le repère global.....	9
I.5. ASSEMBLAGE ET RÉSOLUTION:.....	10
I.5.1. Assemblage:.....	10
I.5.2. Conditions aux limites:.....	10
a) Méthode directe.....	11
I.5.3. Vecteur second membre $F$ .....	11
I.5.4. Détermination des efforts internes et des réactions.....	13
I.5.5. Calcul des contraintes dans les différents éléments.....	13
I.6. Conclusion.....	14

## Chapitre 2 Mise en œuvre de l'optimisation

II.1. introduction .....	15
II.1.1. Formulation mathématique du problème.....	15
II.2. Programmation linéaire.....	17
II.2.1. Définition .....	17
II.2.2. Forme standard.....	17
II.2.3. Espaces des solutions.....	18
II.2.4. Méthodes de résolution.....	19
II.2.4.1. Méthode du simplexe .....	19
□ Travail préparatoire .....	19
□ Critère d'entrée :.....	20
□ Critère de sortie :.....	20
□ Critère d'arrêt :.....	20
II.3. Programmation non linéaire.....	20
II.3.1. Définitions et théorèmes.....	21
II.3.2. Méthode d'optimisation non linéaire avec contraintes.....	22
II.3.3. Choix de la méthode d'optimisation.....	23
II.3.4. Pourquoi choisir la méthode SQP ?.....	23
II.3.5. Méthode séquentielle quadratique SQP.....	24
II.4. Conclusion.....	26

## Chapitre 3 : présentations du programme *OPTIPORT*

III.1.INTRODUCTION :	27
III.2.MISE EN ŒUVRE DU PROBLEME D'OPTIMISATION DES STRUCTURES EN PORTIQUES PLAN.....	28
III.3.DESCRPTION GÉNÉRALE DU PROGRAMME OPTIPORT.....	29
<i>III.3.1.Introduction des donnée:</i> .....	30
III.4.ORGANISATION GÉNÉRALE DE LA PARTIE ANALYSE DE LA STRUCTURE (PORT).....:	32
<i>III.4.1.Sous-programme EXSOL:</i> .....	35
<i>III.4.2.Sous-programme SOLE:</i> .....	37
<i>III.4.3.Sous-programme STIFFPOR</i> .....	39
<i>III.4.4.Sous-programme CON:</i> .....	41
III.5.DESCRPTION DU BLOC D'OPTIMISATION.....	44
<i>III.5.1.Sous-programme NLPQL:</i> .....	45
<i>III.5.2.Sous-programme FCN:</i> .....	46
<i>III.5.3.Sous-programme EXLINO:</i> .....	47
<i>III.5.4.Sous-programme DERIVAT</i> .....	47
<i>III.5.5.Sous-programme MODVFG</i> .....	50

## CHAPITRE 4: APPLICATIONS

IV.1.EXEMPLES D'APPLICATION:	51
<i>IV.1.1.Premier exemple</i> .....	51
<i>IV.1.2.Deuxième exemple</i> .....	54
<i>IV.1.3.Première étude</i> .....	56
<i>IV.1.4.Deuxième étude</i> .....	57
<i>IV.1.5.Troisième étude</i> .....	59
IV.2.CONCLUSION .....	61
Conclusion générale.....	62
Bibliographie.....	64

Annexe

المدسة الوطنفة المفعدة الففنفاف  
BIBLIOTHEQUE — المففففة  
Ecole Nationale Polytechnique

# INTRODUCTION GENERALE

## Introduction générale



### 1. Préambule:

La recherche opérationnelle, les sciences de la gestion et les méthodes d'optimisation mathématique sont des disciplines qui ont connu un grand développement au cours de ces dernières années. En utilisant des méthodes quantitatives, elles servent à la solution des problèmes de prise de décision, de planification et de contrôle.

Plusieurs de ces méthodes sont applicables à une grande gamme de problèmes de génie civil et la tendance actuelle est de bénéficier de leur usage pour orienter les quatre phases d'un projet (planification, conception, construction, et fonctionnement) vers le but de produire un projet complet et efficace.

Dans notre travail, nous nous sommes intéressés au dimensionnement automatique des structures en portique dans la construction métallique. Le problème est comment peut-on concevoir ces portiques avec un coût minimum et un degré de sécurité satisfaisant? Ce problème est plus compliqué que celui des structures en treillis car les éléments constituant les portiques (poteaux + poutres) sont soumis à des efforts axiaux, des moments fléchissants et des efforts tranchants simultanément, tandis que les barres dans les structures en treillis ne sont soumises qu'à des efforts axiaux.

### 2. Définition et buts de l'optimisation des structures

Déterminer les dimensions appropriées des composants structuraux est un problème de première importance pour l'ingénieur. Dans tous les domaines de la mécanique des structures, l'impact de la bonne conception d'une pièce est très important sur sa résistance, sa durée de vie et son utilisation en service. Ce défi est quotidien dans les secteurs de pointe tels que la recherche spatiale, l'aéronautique, l'automobile, les ouvrages en génie civil... . Le développement de l'art de l'ingénieur requiert des efforts considérables pour améliorer sans cesse les techniques de conception des structures. L'optimisation intervient de façon primordiale dans l'augmentation des performances et la réduction de masse, entraînant ainsi de substantielles économies de matériaux.

Le développement constant des techniques de conception assistée par ordinateur et des stratégies d'optimisation s'inscrit dans ce cadre. L'optimisation des structures soulève depuis plus de vingt ans beaucoup d'intérêt. Encore trop peu appliquée aux techniques classiques de bureau d'études, elle s'y intègre progressivement au fur et à mesure que s'accroît sa fiabilité. Parti des problèmes les plus simples, le champ d'application de l'optimisation structurale s'étend

aujourd'hui à de nouveaux défis toujours plus intéressants. L'optimisation des sections droites des éléments d'une structure en portiques constitue le sujet de ce projet de fin d'études. Dans tout ce travail on suppose qu'aucune modification du modèle géométrique (forme et topologie) n'est possible.

### **3. Elaboration du programme**

Une étape primordiale dans la conception et l'optimisation des structures est l'établissement d'un modèle numérique de base. Cette phase de modélisation, essentielle pour une compréhension future du comportement de la structure sous les différentes sollicitations, suppose le recours à un outil d'analyse numérique performant et maîtrisable, s'appuyant généralement sur la méthode des éléments finis.

Les fondements de la méthode des éléments finis (qui constitue à l'heure actuelle la technique la plus répandue de discrétisation) sont exposés en chapitre 1.

Notre présent travail, consiste à mettre en œuvre un outil informatique d'optimisation à travers le développement d'un programme (**OPTIPOINT**) de calcul et d'optimisation des structures en portiques plans.

La minimisation du poids de la structure est assurée par l'algorithme de la méthode séquentielle quadratique (**SQP**) qui sera explicité dans le chapitre 2.

En combinant ces deux méthodes: La méthode des éléments finis (**MEF**) pour l'analyse de la structure et la méthode **SQP** pour l'optimisation des sections, nous avons développé un programme de calcul et d'optimisation des structures en portiques (**OPTIPOINT**) écrit en langage FORTRAN 90. Il est basé sur le principe suivant :

En démarrant avec une valeur initiale des dimensions des sections des éléments de la structure, cette valeur sera perturbée par le processus d'optimisation de la méthode SQP, et à chaque itération une analyse de la nouvelle structure est effectuée par la méthode des éléments finis pour déterminer les contraintes maximales dans chaque élément du portique jusqu'à l'obtention du poids optimal. Une étude plus approfondie du programme sera détaillée en chapitre 3.

En fin, pour confirmer la performance de notre programme, des exemples pratiques seront étudiés en chapitre 4 et les résultats seront comparés avec ceux donnés par le logiciel SAP 2000.

**CHAPITRE 1**  
**METHODE DES ELEMENTS FINIS**

## 1.1. Introduction

La modélisation des phénomènes physiques est aujourd'hui très présente dans les métiers de l'ingénieur. Celui-ci est en effet amené à utiliser de plus en plus couramment des méthodes de calcul numérique face à la complexité sans cesse croissante des problèmes qu'il rencontre.

La méthode des éléments finis **MEF** est l'une des méthodes de simulation numérique les plus pratiquées de nos jours. Elle consiste à utiliser une approximation simple de la géométrie et des variables décrivant le phénomène physique telles que le déplacement, la vitesse, la pression,... afin de ramener le problème continu comportant une infinité d'inconnues vers un système algébrique à un nombre fini de degrés de liberté

Les problèmes traités par la **MEF** sont généralement liés au dimensionnement, à l'optimisation et à la fiabilité de l'objet à concevoir ou à construire. La finalité étant de simuler numériquement le comportement d'une structure soumise à diverses sollicitations mécaniques afin d'évaluer l'état de déformation, donc les valeurs des contraintes en supposant un comportement linéaire, et d'optimiser les paramètres structuraux (dans notre cas, les aires des sections des éléments constituant les portiques). Sans totalement se substituer aux essais expérimentaux, la simulation numérique constitue donc un outil complémentaire qui s'avère souvent beaucoup plus rapide et moins onéreux que ces derniers.

Avant d'entamer les fondements de base de cette méthode, nous commençons par donner un petit aperçu historique sur la **MEF**.

## 1.2. Aperçu historique [1], [2]

L'utilisation des méthodes de discrétisation et d'approximation numérique pour résoudre les problèmes d'ingénierie est connue depuis très longtemps et le concept de "l'élément fini" provenait de ces méthodes.

Si nous voulons identifier le premier "élément fini", probablement nous serons obligés à revenir en arrière jusqu'à l'approximation géométrique des pyramides par les égyptiens quelques 5000 ans avant. Si nous considérons, par exemple, l'approximation numérique de  $\pi$  comme un point de départ des éléments finis, nous trouvons des archives intéressantes dans la Chine, l'Égypte et la Grèce.

En effet l'approximation de  $\pi$  a commencé avec les chinois dès le premier siècle apr. j.c. Liu Hui (265-317 apr. j.c) a utilisé un polygone enfermé dans un cercle pour approximer sa circonférence et a trouvé  $\pi=3.1416$  en utilisant un polygone de 3072 arêtes (éléments finis),

Plus rigoureusement, si nous considérons les solutions approchées des problèmes d'élasticité comme le point de départ de la **MEF** nous pouvons nous référer à l'apport historique du développement de Timoshenko. Si nous considérons l'apparition de l'analyse des portiques comme le début de la **MEF**,

nous devons revenir aux travaux de Maxwell, Castigliano, et Mohr pendant la période 1850-1875.

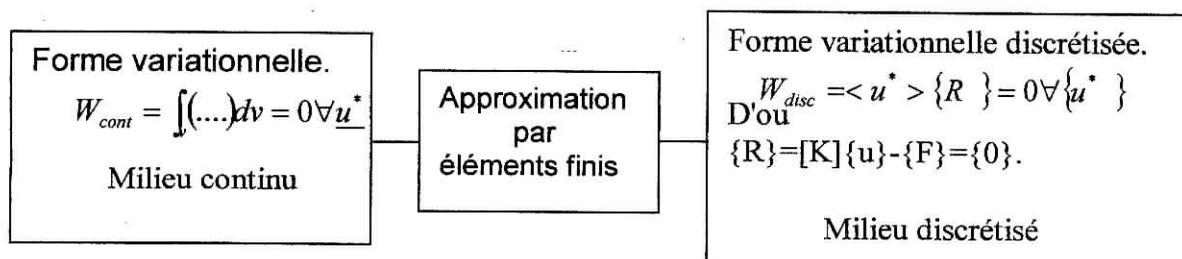
L'apparition des ordinateurs et les besoins de l'industrie ont provoqué un développement rapide de la mécanique des structures entre 1950 et 1960. Turner, Clough, Martin et Topp introduisent en 1956 le concept "moderne" de l'élément fini: Le comportement de chaque élément est caractérisé par *une matrice de rigidité élémentaire*. A partir de 1967, de nombreux livres sont publiés sur la méthode. Signalons en particulier les trois éditions très répandues du livre de Zienkiewicz. Des programmes généraux de type industriel sont apparus, et ne cessent de se développer. Ils sont capables de résoudre des problèmes variés de grandes tailles (de mille à quelques centaines de milliers de variables). Citons à titre d'exemple quelques noms de logiciels: ABAQUS, ANSYS, CASTEM 2000, NASTRAN, SAMCEF, etc. Les possibilités offertes par de tels programmes sont nombreuses :

- Analyse linéaire ou non d'un système physique continu,
- Analyse statique ou dynamique,
- Prise en compte de lois de comportement complexes,
- Problèmes d'optimisation; etc.

### 1.3. Aspects généraux de la méthode des éléments finis :

#### 1.3.1. Description générale:

La méthode des éléments finis est une technique particulière d'approximation des fonctions solutions par sous domaines. Les inconnues  $\{U\}$  sont les valeurs de ces fonctions en certains points ou noeuds de chaque sous domaine. La forme variationnelle définie sur le milieu continu est ainsi représentée par une forme variationnelle dite discrétisée qui fait intervenir les inconnues nodales  $\{U\}$ . Par exemple, pour un problème d'élasticité linéaire:



#### Problème d'élasticité linéaire

### 1.3.2. Démarche éléments finis: [3]

Les différentes étapes de la méthode des éléments finis sont les suivantes:

- Représentation du domaine de volume  $V$  par un ensemble de sous domaines de volume  $V_e$ :

$$V = \sum V^e; W = \sum W^e$$

- Représentation de la géométrie de chaque élément  $V_e$ :

$$\{X(\xi)\} = [N(\xi)]\{X_n^*\} \quad (1.1)$$

$\{X\}$  position d'un point;  $\{X_n\}$  coordonnées des nœuds définissant  $V_e$

$\xi$  coordonnées paramétriques  $\xi, \eta, \zeta$

$[N]$  fonction d'interpolation en variables paramétriques.

- Représentation (isoparamétrique) de la fonction solution  $\{u\}$  sur chaque élément :

$$\{u(\xi)\} = [N(\xi)]\{u_n\}; \{u^*(\xi)\} = [N(\xi)]\{u_n^*\}$$

$\{u\}$  fonction solution;  $\{u^*\}$  fonctions tests (ou virtuelles).

$\{u_n\}$  variables nodales caractérisant la fonction solution.

$\{u_n^*\}$  variables nodales virtuelles.

- Représentation de la forme variationnelle (discrétisation).
- Calcul élémentaire: sur chaque élément la quantité  $W$ , noté  $W_e$ , s'exprime en fonction de  $\{u_n\}$  et  $\{u_n^*\}$  (équation (1.2)):

$$W_e = \langle u_n^* \rangle ([k]\{u_n\} - \{f_n\}) \quad (1.2)$$

$[k]$  Matrice élémentaire (dite de rigidité);  $\{f_n\}$  vecteur élémentaire des sollicitations.

- Assemblage: construction de  $[K]$  et  $[F]$ :

$$W = \sum W^e = \sum_e \langle u_n^* \rangle ([k]\{u\} - \{f_n\}) \quad (1.3)$$

$$W = \langle U^* \rangle ([K]\{U\} - \{F\}) = 0 \quad \forall U^*{}^e$$

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

$[K]$  Matrice globale obtenue par assemblage des matrices élémentaires;

$\{F\}$  Vecteur global des sollicitations obtenu par assemblage des vecteurs sollicitations élémentaires.

- Résolution :

En tenant compte des conditions aux limites, trouver  $\{U\}$  tel que:

$$\{R\} = [K]\{U\} - \{F\} = 0 \quad (1.4)$$

Pour un problème linéaire:  $\{U\} = [K]^{-1}\{F\}$

- Evaluation des quantités relatives à chaque élément :

-Extraire  $\{u_n\}$  de  $\{U\}$

-Calculer les gradients de  $\{u\}$  en un point de l'élément (déformation)

-Calculer les quantités représentatives du problème (contraintes).

Dans le cadre de ce travail, les structures étudiées sont constituées de *portiques plans*, nous restreindrons l'étude sur ce type d'éléments.

#### 1.4. Formulation de la MEF dans le cas des portiques:

L'étude des structures en barres et poutres par la méthode des déplacements contient une bonne part d'une analyse par éléments finis et conduit au système linéaire suivant [1], [2], [4]:

$$[K] \{U\} = \{F\} \quad (1.5)$$

Où:

$[K]$  : matrice de rigidité,  $\{U\}$  : vecteur des inconnues cinématiques (déplacements) ou degrés de liberté,  $\{F\}$  : vecteur force.

Les éléments du portique sont soumis simultanément aux trois sollicitations suivantes: efforts normaux, efforts tranchants et moments fléchissants, ils sont donc considérés comme la superposition de deux types d'éléments:

- élément barre : sollicité par des efforts axiaux.
- élément poutre : sollicité par des efforts tranchants et des moments fléchissants.

Nous donnons dans ce qui suit la formulation de la matrice de rigidité élémentaire de chaque type de ces deux éléments puis celle résultant de leurs superposition.

##### 1.4.1. Matrice de rigidité d'un élément portique

###### a) Élément barre

Un élément du type « barre » de longueur  $L$ , d'un module d'élasticité  $E$  et d'une section  $A$  est représenté sur la figure (1.1). Ses deux extrémités numérotées 1 et 2 sont appelées « *points nodaux* ». Aux deux forces  $N_1$  et  $N_2$  appliquées selon la direction  $X$  correspondent deux déplacements  $u_1$  et  $u_2$ , respectivement. Ces deux déplacements sont souvent appelés "*degrés de liberté*". Il existe au total deux degrés de libertés pour cette barre.

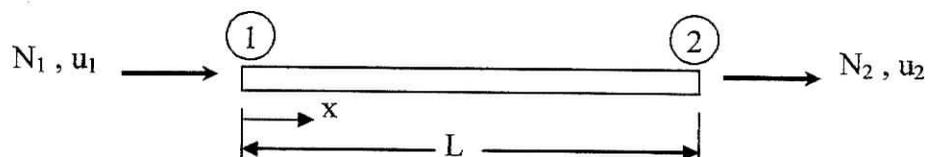


Figure 1.1: barre dans un repère local

Il est nécessaire de déterminer deux équations liants  $N_1$  et  $N_2$  à  $u_1$  et  $u_2$ . Cette détermination peut être effectuée par une approche énergétique ou bien par une approche équilibre contraintes déformation. La première est plus générale et plus puissante tandis que la deuxième est plus simple et physiquement plus claire, mais ne peut être appliquée que pour des éléments simples. Nous aboutissons au système suivant [1]

$$\begin{Bmatrix} N_1 \\ N_2 \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad (1.6)$$

Le terme  $EA/L$  est la rigidité axiale pour la barre. Cette dernière se comporte comme un ressort avec un coefficient de raideur  $k=EA/L$ .

b) *Élément poutre*

Considérons la poutre droite de la figure (1.2), cet élément a deux degrés de libertés en chaque point nodal, un déplacement transversal  $v$  et une rotation  $\theta$  (ou  $\partial u/\partial x$ ). A ces deux degrés de liberté  $v$  et  $\theta$  correspondent respectivement un effort tranchant  $V$  et un moment fléchissant  $M$  agissant en chaque point nodal.

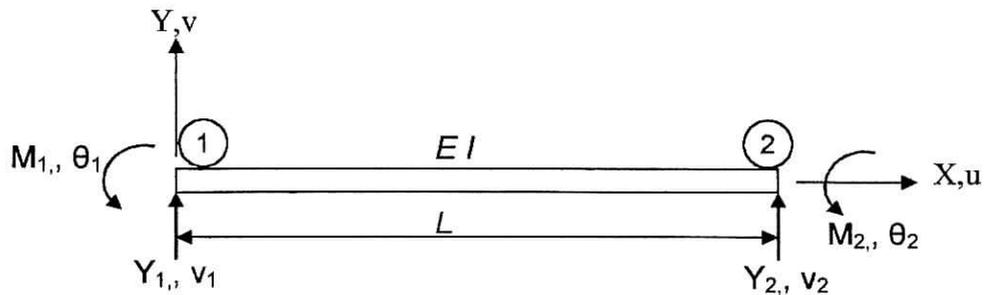


Figure1.2: élément "poutre" avec une section uniforme

Le système liant les efforts  $V$  et  $M$  au degrés de libertés  $v$  et  $\theta$  est le suivant:

$$\begin{Bmatrix} V_1 \\ M_1 \\ V_2 \\ M_2 \end{Bmatrix} = \frac{EI}{L} \begin{bmatrix} \frac{12}{L^2} & \frac{6}{L} & -\frac{12}{L^2} & \frac{6}{L} \\ \frac{6}{L} & 4 & -\frac{6}{L} & \\ -\frac{12}{L^2} & -\frac{6}{L} & \frac{12}{L^2} & -\frac{6}{L} \\ \frac{6}{L} & & \frac{6}{L} & 4 \end{bmatrix} \begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix} \quad (1.7)$$

Symboliquement

$$\{F\} = [k]\{u\} \quad (1.7a)$$

c) *Élément portique*

En combinant un élément barre (§IV.1.1) avec un élément poutre (§IV.1.2) nous obtenons l'élément portique suivant:

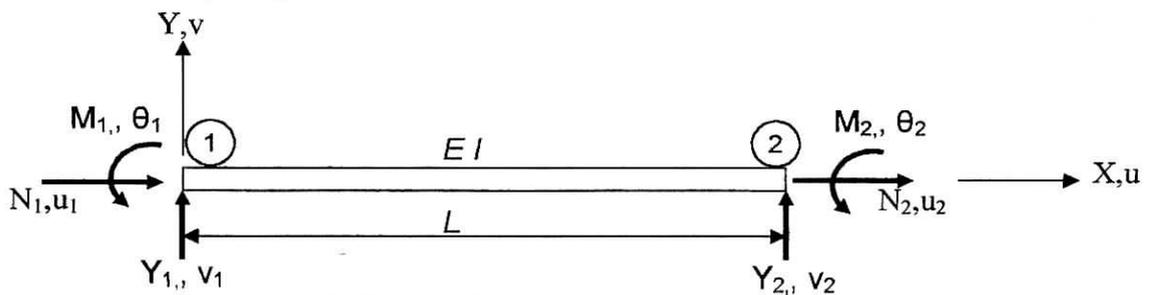


Figure1.3: élément "portique"

Les équations de rigidité sont:

$$\begin{Bmatrix} N_1 \\ N_2 \\ V_1 \\ M_1 \\ V_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} & 0 & 0 & 0 & 0 \\ -\frac{EA}{L} & \frac{EA}{L} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ 0 & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix} \quad (1.8)$$

Pour des raisons de commodité d'assemblage (que nous verrons vers la fin de cette partie consacrée à la **MEF**) il est préférable de numéroter tous les degrés de liberté en chaque point dans un certain ordre, dans ce but nous réarrangeons l'équation (1.8) Comme suit [1],[5]:

$$\begin{Bmatrix} N_1 \\ V_1 \\ M_1 \\ N_2 \\ V_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} \quad (1.9)$$

Symboliquement

$$\{\bar{\mathbf{F}}\} = [\bar{\mathbf{k}}] \{\bar{\mathbf{u}}\} \quad (1.9a)$$

Où le signe (  $\bar{\quad}$  ) indique que la formulation est effectuée par rapport au repère local, dans ce qui suit nous donnons la formulation dans le repère global en utilisant la matrice de rotation.

#### d) Changement du repère

La transformation géométrique du repère local vers le repère global est la suivante:

$$\begin{Bmatrix} \overline{N}_1 \\ \overline{V}_1 \\ \overline{M}_1 \\ \overline{N}_2 \\ \overline{V}_2 \\ \overline{M}_2 \end{Bmatrix} = \begin{bmatrix} \lambda & \mu & 0 & 0 & 0 & 0 \\ -\mu & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \lambda & \mu & 0 \\ 0 & 0 & 0 & -\mu & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} N_1 \\ V_1 \\ M_1 \\ N_2 \\ V_2 \\ M_2 \end{Bmatrix} \quad (1.10)$$

où  $\lambda = \cos \phi$   
 $\mu = \sin \phi$

On peut aussi écrire

$$\{\overline{F}\} = [T] \{F\} \quad (1.10a)$$

La matrice  $[T]$  est orthogonale c'est-à-dire  $[T]^{-1} = [T]^T$

L'équation (1.10a) peut être écrite en forme inverse

$$\{F\} = [T]^{-1} \{\overline{F}\} = [T]^T \{\overline{F}\} \quad (1.11)$$

Entre-temps, on peut aussi écrire

$$\{\overline{u}\} = [T] \{u\} \quad (1.12)$$

En remplaçant l'équation (1.10a) et l'équation (1.12) dans l'équation (1.9a) et en utilisant la propriété d'orthogonalité de  $[T]$ , nous obtenons:

$$\{F\} = [T]^T [\overline{k}] [T] \{u\} \quad (1.13)$$

En remplaçant chaque matrice par son expression nous obtenons la matrice de rigidité finale pour un élément portique.

e) *Matrice de rigidité dans le repère global*

Les équations de rigidité pour un élément portique plan orienté avec un angle  $\phi$  sont [1], [5] :

$$\begin{Bmatrix} N_1 \\ V_1 \\ M_1 \\ N_2 \\ V_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} r\lambda^2 + \frac{12}{L^2}\mu^2 & (r - \frac{12}{L^2})\lambda\mu & \frac{6}{L}\mu & 0 & 0 & 0 \\ (r - \frac{12}{L^2})\lambda\mu & r\mu^2 + \frac{12}{L^2}\lambda^2 & \frac{6}{L}\lambda & 0 & 0 & 0 \\ \frac{6}{L}\mu & \frac{6}{L}\lambda & 4 & 0 & 0 & 0 \\ \hline -r\lambda^2 - \frac{12}{L^2}\mu^2 & (-r + \frac{12}{L^2})\lambda\mu & \frac{6}{L}\lambda & r\lambda^2 + \frac{12}{L^2}\mu^2 & (-r + \frac{12}{L^2})\lambda\mu & \frac{6}{L}\mu \\ (-r + \frac{12}{L^2})\lambda\mu & -r\mu^2 - \frac{12}{L^2}\lambda^2 & -\frac{6}{L}\lambda & (r - \frac{12}{L^2})\lambda\mu & -r\mu^2 - \frac{12}{L^2}\lambda^2 & -\frac{6}{L}\lambda \\ \frac{6}{L}\mu & \frac{6}{L}\lambda & 2 & \frac{6}{L}\mu & -\frac{6}{L}\lambda & 4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} \quad (1.14)$$

Ce système élémentaire doit être assemblé avec les autres pour constituer le système global de la structure afin de pouvoir introduire les conditions aux limites (encastrement, appuis doubles,...) ainsi que les sollicitations extérieures.

Nous donnons dans ce qui suit la procédure d'assemblage des matrices pour pouvoir construire le système global.

### 1.5. Assemblage et résolution:

Cette section concerne l'assemblage des éléments (contribution de chaque élément pour l'équilibre global), la prise en compte des conditions aux limites ainsi que la résolution des systèmes d'équations linéaires dans le cas des sollicitations statiques.

#### 1.5.1. Assemblage:

Dans la méthode des éléments finis, l'expression du travail  $W$  sur le domaine est la somme des contributions élémentaires  $W^e$ . L'étape d'assemblage correspond à la construction de la matrice globale  $[K]$  et du vecteur sollicitation global  $\{F\}$  à partir des matrices et vecteurs élémentaires  $[k]$  et  $\{f_e\}$ :

$$W = \sum W^e = 0 \quad (1.15)$$

Avec 
$$W^e = \langle u_e^* \rangle^e ( [K]\{u_e\} - \{f\} ) \quad (1.15a)$$

$$W = \langle U^* \rangle ( [K]\{u_e\} - \{F\} ) = 0 \quad \forall \{U^*\} \quad (1.15b)$$

$\{U\}, \{U^*\}$  ensemble des variables nodales.

$\{u_e\}, \{q_e^*\}$  variables nodales appartenant à chaque élément.

Les variables  $\{u_e\}$  et  $\{q_e^*\}$  sont contenues dans les vecteurs  $\{U\}$  et  $\{U^*\}$ . L'équation (1.15) peut être traduite par l'addition de matrices et vecteurs. Si on exprime  $W^e$  sous la forme suivante :

$$W^e = \langle U^* \rangle ( [K^e]\{U\} - \{F^e\} ) \quad (1.16)$$

Où  $[K^e]$  (matrice élémentaire étendue) de dimension  $NDLT \times NDLT$  ( $NDLT$  est le nombre de degrés de liberté total) contient des lignes et des colonnes de zéros relatives aux variables de  $\{U^*\}$  et  $\{U\}$  n'appartenant pas à l'élément  $e$ . Les termes non nuls de position  $i, j$  dans  $[K^e]$  correspondent aux composantes de  $[K]$ . De même  $\{F^e\}$  de dimension  $NDLT$  contient des valeurs nulles relatifs aux composantes n'appartenant pas à l'élément  $e$ . Les équations (1.15a) et (1.15b) donnent:

$$\{F\} = \sum_e \{F^e\} \quad (1.17)$$

On écrit souvent symboliquement  $[K] = \sum_e [K]$  et  $\{F\} = \sum_e \{f_n\}$ .

#### 1.5.2. Conditions aux limites:

La matrice  $[K]$  et le vecteur  $\{F\}$  définis dans l'expression d'assemblage ne tiennent pas compte des conditions aux limites associées aux variables  $\{U\}$ . En général, on cherche le vecteur  $\{U\}$  tel que:

$$[K]\{U\} = \{F\} \quad (1.18)$$

Avec 
$$u_i = \bar{u}_i \quad (1.19)$$

Où  $u_i$  est la composante  $i$  du vecteur  $\{U\}$  dont la valeur est connue ( $= \bar{u}_i$ ).

La condition (1.19) peut être introduite directement dans (1.18) en modifiant [K] et {F}. Nous présentons ici une méthode conservant les dimensions de [K] et {F}.

a) *Méthode directe*

En introduisant (1.18) dans (1.19) on peut écrire:

$$\begin{bmatrix} K_{11} & \dots & K_{1i} & \dots & K_{1n} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \dots & K_{ii} & \dots & K_{in} \\ \vdots & & \vdots & & \vdots \\ K_{n1} & \dots & K_{ni} & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} F_1 \\ \vdots \\ F_i + R_i \\ \vdots \\ F_n \end{Bmatrix} \quad (1.20)$$

Où le terme  $F_i + R_i$  représente les forces externes garantissant l'équilibre au nœud lorsque  $u_i = \bar{u}_i$ ,  $R_i$  est la réaction (inconnue) et  $F_i$  est la composante des forces équivalentes (connue).

L'équation (1.20) fait intervenir n inconnues : n-1 variable {u} et la réaction  $R_i$ .

Ces inconnues peuvent être obtenues de la façon suivante:

- Evaluation des n-1 composantes de {U} par solution de:

$$\begin{bmatrix} K_{11} & \dots & 0 & \dots & K_{1n} \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ K_{n1} & \dots & 0 & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} F_1 - K_{1i}\bar{u}_i \\ \vdots \\ \bar{u}_i \\ \vdots \\ F_n - K_{ni}\bar{u}_i \end{Bmatrix}$$

- Evaluation de la réaction  $R_i$ :  $R_i = \sum_{j=1}^n K_{ij}u_j - F_i$

Finalement, nous aboutissons au système global suivant (après assemblage):

$$[K]\{u\} = \{F\} \quad (1.21)$$

[K]: matrice de rigidité du système global,

{u}: vecteur déplacement des nœuds de la structure,

{F}: vecteur force globale, que nous détaillerons dans le paragraphe suivant.

1.5.3. *Vecteur second membre F*

Dans ce paragraphe nous allons nous appliquer à déterminer le vecteur [F] qui représente l'ensemble des forces.

Il existe trois *contributions* au vecteur second membre [5]

$$F=Q+R+P$$

Dans cet équation:

Q : vecteur des charges nodales (de la structure).

R: vecteur des réactions d'appuis.

P: vecteur des charges équivalentes.

En ce qui concerne le terme P il est nécessaire pour chaque élément de calculer les efforts résultants des charges qui lui sont directement appliquées en le considérant isolé de la structure et encastré à ces deux extrémités [4]

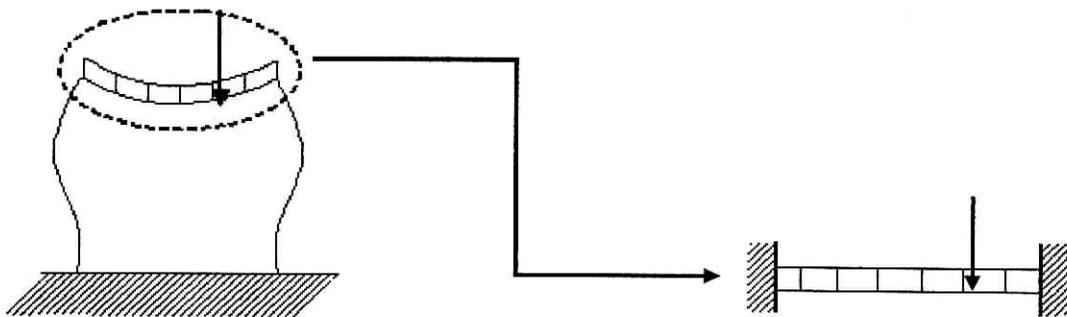
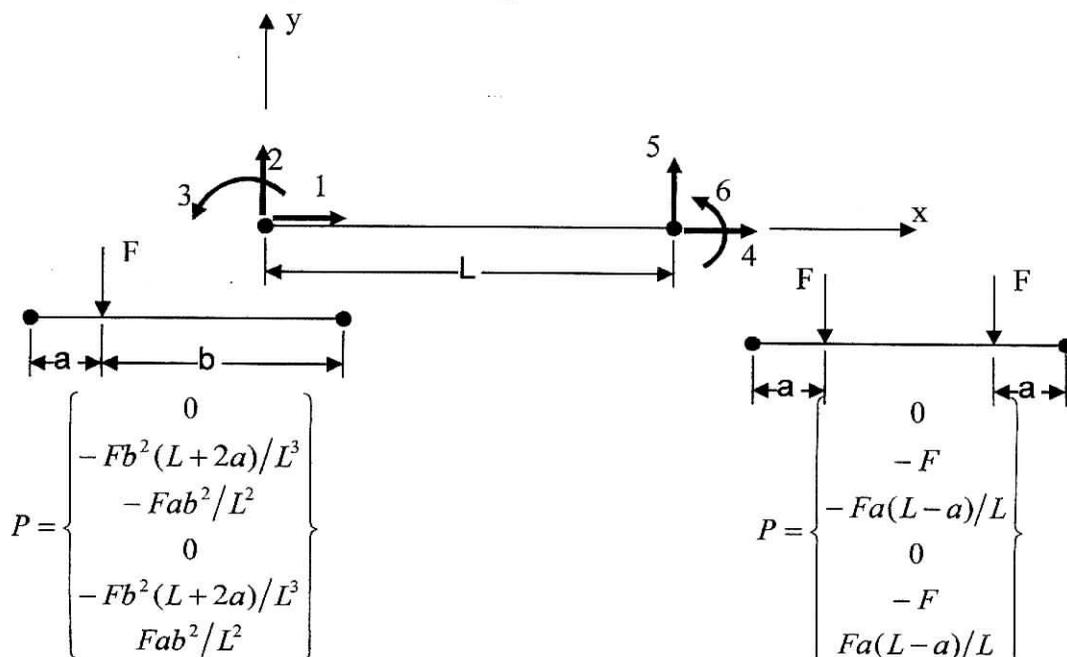
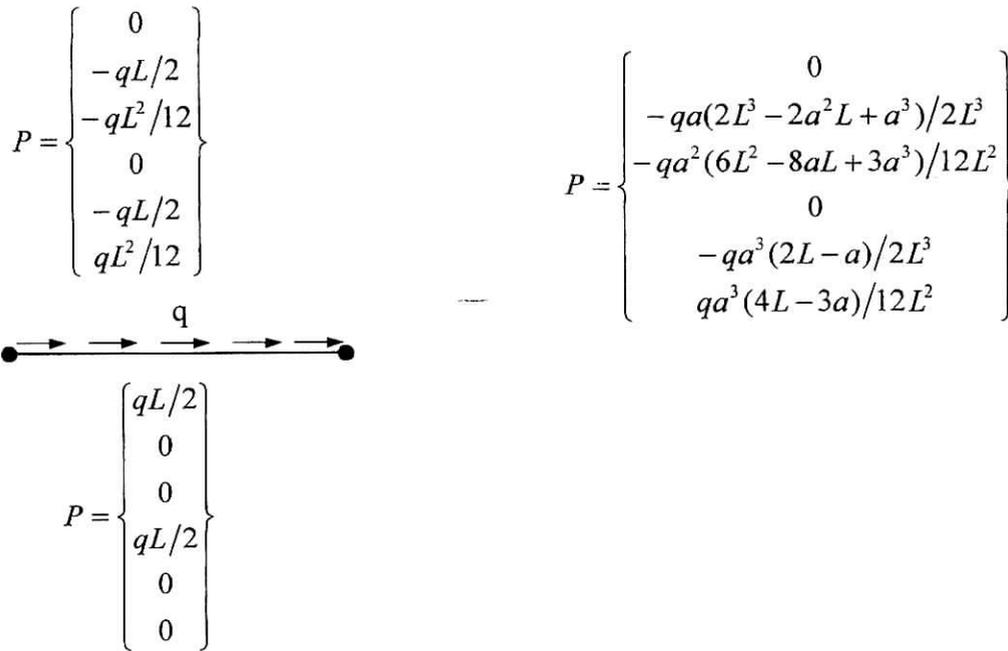


Figure1.4: calcul du terme P

Pour le calcul d'une barre biencastrée, soumise à des charges quelconques on peut décomposer les charges appliquées en une succession de charges simples : charges ponctuelles, charges réparties uniforme, ... dont les résultats ont été établis une fois pour toute. On effectue ensuite la somme des charges des différents efforts dues à ces charges simples, la figure 1.5 présente les vecteurs de charges équivalentes pour une poutre à six degrés de libertés :





**Figure 1.5 :** Vecteurs charges pour une poutre à 6 degrés de liberté axes locaux (x,y)

#### 1.5.4. Détermination des efforts internes et des réactions:

Après avoir déterminé les déplacements dans les noeuds, les efforts internes (N,M,T) dans les différents éléments et les réactions d'appuis seront obtenus en substituant les valeurs des déplacements et des rotations nodaux dans l'équation de rigidité élémentaires (1.14).

Le calcul des efforts dans les éléments est donc relativement simple une fois que l'on connaît les déplacements des nœuds. Les valeurs ainsi obtenues en résolvant le système  $\{F_e\} = [k_e]\{u_e\}$  représentent la superposition des efforts internes et des efforts équivalents dues aux charges réparties et aux forces concentrées en travées.

Donc les efforts internes réels seront obtenus en soustrayant le vecteur  $\{p_e\}$  pour chaque élément défini dans la figure 1.5 du vecteur  $\{F_e\}$  ainsi obtenu.

$$\{f_{int}\} = \{F_e\} - \{P\} \quad (1.22)$$

#### 1.5.5. Calcul des contraintes dans les différents éléments

Étant donné que les éléments du portique sont soumis à la flexion composée, les contraintes  $\sigma$  dans ces éléments seront obtenues à partir la formule:

$$\sigma_i = \frac{N_i}{A_i} + \frac{M.V}{I} \quad (1.23)$$

### **1.6. Conclusion**

La méthode des éléments finis basée sur le calcul des déplacements est actuellement la plus utilisée pour étudier les structures par ordinateur:

- Une fois les  $N$  inconnues choisies, la structure de référence est déterminée et elle est unique ;
- Les caractéristiques  $[k_e]$  et  $\{P_e\}$  sont codées une fois pour toute ;
- La matrice de rigidité  $[K]$  de la structure est construite mécaniquement à partir des matrices  $[k_e]$  des éléments et de la topologie de la structure; les seconds membres également; on trouve donc les déplacements des nœuds voire les réactions ;
- Les efforts internes ainsi que les contraintes se calculent élément par élément, à partir des déplacements exprimés en axes locaux, via des relations du type (1.22)

La formulation et la procédure de solution on été présentées systématiquement d'une manière à ce qu'elles puissent être facilement adaptées aux ordinateurs.

# CHAPITRE 2

## MISE EN ŒUVRE DE L'OPTIMISATION

## **II.1. introduction**

De très nombreux problèmes de prise de décision reposent sur la recherche de solutions optimales selon un critère donné. Nous présentons ici les principaux outils issus de la programmation mathématique (programmation linéaire et non linéaire, avec contraintes).

La programmation mathématique, et plus particulièrement l'optimisation, vise à résoudre des problèmes où l'on cherche à déterminer parmi un grand nombre de solutions candidates celle qui donne le meilleur rendement. Plus précisément, on cherche à trouver une solution satisfaisant un ensemble de contraintes qui minimise ou maximise une fonction donnée.

### **II.1.1. Formulation mathématique du problème**

La formulation mathématique d'un problème d'optimisation réel doit passer par les trois étapes suivantes:

La première étape consiste à déterminer les paramètres dont dépend le problème et à affecter une variable mathématique à chaque paramètre. Ces derniers sont "*les variables décisionnelles*".

La deuxième étape est de rechercher quelles sont les restrictions imposées sur les variables et construire un modèle mathématique des relations entre ces différentes variables. Les restrictions sont dues à une variété de sources telles que:

- Le comportement mécanique;
- Les propriétés des matériaux utilisés;
- Les limites imposées par les règlements du domaine;
- Les limites imposées par la disponibilité du matériau;
- Les limites imposées par les conditions d'esthétique;
- Les limites imposées par les méthodes de constructions envisagées...etc.

Chaque restriction (ou bien *contrainte*) doit être examinée en détail et exprimée mathématiquement sous forme d'une ou plusieurs fonctions des variables décisionnelles. Souvent, cette opération n'est pas difficile puisque les lois mécaniques gouvernant le comportement mécanique ainsi que les propriétés des matériaux sont maîtrisées. De plus, les règlements sont disponibles et peuvent être consultés pour poser les contraintes nécessaires.

La troisième étape consiste à choisir les critères qui garantissent l'excellence du projet et d'écrire *la fonction objectif* pour exprimer ces critères sous forme d'une fonction des variables décisionnelles, ces critères peuvent être d'origine technologique ou économique ou bien la combinaison des deux par exemple: "*le poids minimum*", qui constitue le but de ce P.F.E.

Notre objectif sera de minimiser le poids des structures. La "fonction objectif" est alors "le poids de la structure". Les variables de cette fonction objectif sont les aires des sections droites des éléments du portique, ce sont "les variables décisionnelles" du problème. Les contraintes à respecter sont la résistance de la construction, c'est-à-dire que les contraintes dans ces éléments doivent être inférieures aux contraintes limites, ainsi que la forme et les dimensions de ces éléments doivent être conformes aux règlements appliqués. Nous obtenons la formulation mathématique suivante:

*Minimiser  $f(x)$ =poids total de la structure*

*Sous les contraintes:*

*Résistance des constituants.*

*Formes des éléments dans les limites admises.*

*Dimensions des éléments varient dans un intervalle donné*

Sous une autre forme:

$$(P) \begin{cases} \text{minimiser } f(x) = \gamma \sum_1^m l_i \cdot h_i \cdot b_i \\ \text{sous les contraintes:} \\ |\sigma_{\text{lim}}| - |\sigma_{i \text{ max}}| \geq 0 \\ b_i h^{-1}_i - R_{\text{min}} \geq 0 \\ R_{\text{max}} - b_i h^{-1} \geq 0 \\ b_{\text{min}} \leq b_i, h_i \leq h_{\text{max}} \end{cases} \quad i = 1, \dots, m$$

où

$f(x)$  est la fonction objectif: poids de la structure

$x$ : vecteur, à  $2m$  composantes, des dimensions des éléments

$\gamma$  : Poids spécifique du matériau constituant les éléments de la structure

$m$ : nombre d'éléments

$b_i$ : largeur de l'élément  $i$

$h_i$ : hauteur de l'élément  $i$

$l_i$  : longueur de l'élément  $i$

$\sigma_i$  : contrainte maximale dans l'élément  $i$ , qui est égale à :  $2400 \text{ kg/cm}^2$

$\sigma_{\text{lim}}$  : contrainte limite de traction/compression de l'acier utilisé

$R_{\text{min}}$ : borne inférieure du rapport  $b/h$

$R_{\text{max}}$ : borne supérieure du rapport  $b/h$

$b_{\text{min}}$ : borne inférieure des dimensions

$h_{\text{max}}$  : borne supérieure des dimensions.

Cette formulation du problème nous conduit à choisir une méthode d'optimisation non linéaire avec contraintes. La méthode *SQP* est le candidat le plus puissant (son choix sera justifié dans ce qui suit), mais avant d'entamer le

fondement de cette méthode il est intéressant d'aborder dans ce chapitre des définitions de base ainsi qu'une méthode de programmation linéaire (méthode du simplex) sur la quelle dépend grandement la méthode **SQP**.

## II.2. Programmation linéaire

### II.2.1. Définition

Un programme linéaire est un programme qui consiste à optimiser (maximiser ou minimiser) une fonction linéaire à plusieurs variables soumise à un ensemble de contraintes linéaires.

D'une manière générale, un programme linéaire s'écrit sous la forme suivante

$$\left\{ \begin{array}{l} \text{Min } F = a_0 + a_1x_1 + \dots + a_nx_n \\ \text{Sous les contraintes} \\ b_{11}x_1 + \dots + b_{1n}x_n \quad R_1 \quad c_1 \\ b_{21}x_1 + \dots + b_{2n}x_n \quad R_2 \quad c_2 \\ \dots \\ b_{m1}x_1 + \dots + b_{mn}x_n \quad R_m \quad c_m \end{array} \right. \quad (2.1)$$

Les relations  $R_1, R_2, \dots, R_m$  sont des opérateurs d'égalités ou d'inégalités. Les coefficients  $a_0, a_1, \dots, a_n$  dans la fonction objectif sont des nombre réels. De même que tous les coefficients dans le système des contraintes sont des réels. Ce programme général peut aussi être exprimé sous la forme matricielle suivante :

$$\begin{array}{l} \text{Min } F = [a] \{x\} \\ \text{sous les contraintes} \\ [B] \{x\} \quad R \quad \{c\} \end{array} \quad (2.2)$$

où

[B] est la matrice  $m \times n$  des coefficients des variables dans le système des contraintes.

{a} est le vecteur des coefficients dans la fonction objectif

{x} est le vecteur représentant les variables de décision

{c} est le vecteur second membre du système des contraintes .

### II.2.2. Forme standard

Les méthodes de résolution des programmes linéaires utilisent souvent la forme standard. Nous allons donner sa définition ainsi que les transformations utiles pour ramener un programme linéaire donné sous forme générale à une forme standard.

On appelle forme standard d'un programme linéaire, la forme suivante :

$$\left\{ \begin{array}{l} \text{Min } F = \sum_{j=1}^n a_j x_j \\ \text{Sous les contraintes} \\ \sum_{j=1}^n b_{ij} x_j = c_i \quad , \quad i = 1, \dots, m \\ x_j \geq 0 \end{array} \right. \quad (2.3a)$$

Cette forme impose aussi que les seconds membres des relations soient des nombres réels positifs ou nuls, c'est-à-dire  $c_i \geq 0$ ,  $i=1,2,\dots,m$ . La traduction matricielle de la forme standard d'un programme linéaire est :

$$\begin{cases} \text{Min} F = \{a\}^T \{x\} \\ \text{sous les contraintes} \\ [B]\{x\} = \{c\} \quad (i) \\ \{x\} \geq 0 \quad (ii) \end{cases} \quad (2.3b)$$

Lorsque le programme n'est pas donné sous la forme standard, il est utile d'utiliser les transformations suivantes [7] :

- 1) La fonction objectif consiste à maximiser une fonction linéaire  $F$ . Ceci est équivalent à minimiser  $(-F)$  et on a la relation suivante  $\text{Max } F = -\text{Min } (-F)$ .
- 2) Lorsque une contrainte (par exemple la  $i^{\text{ème}}$ ) n'est pas une égalité deux cas peuvent se produire :

a)  $b_{i1}x_1 + \dots + b_{in}x_n \leq c_i$ , on introduit une variable  $e_i$  d'écart tel que :

$$b_{i1}x_1 + \dots + b_{in}x_n + e_i = c_i$$

b)  $b_{i1}x_1 + \dots + b_{in}x_n \geq c_i$ , on introduit une variable  $e_i$  d'écart tel que

$$b_{i1}x_1 + \dots + b_{in}x_n - e_i = c_i$$

3) Il arrive parfois qu'un ou plusieurs seconds membres des coefficients ne soient pas positifs. Pour mieux fixer les idées, supposons que la  $i^{\text{ème}}$  contrainte possède un  $b_i$  négatif en d'autre terme :

$$b_{i1}x_1 + \dots + b_{in}x_n = c_i \quad \text{avec } b_i < 0$$

il suffit, dans ce cas, de multiplier les deux termes de cette contrainte par  $(-1)$  et on obtient

$$-b_{i1}x_1 - \dots - b_{in}x_n = -c_i \quad \text{avec } -b_i \geq 0$$

### II.2.3. Espaces des solutions

Une solution d'un programme linéaire est un vecteur  $(x_1, \dots, x_n)$  de  $\mathbb{R}^n$  satisfaisant aux contraintes  $(i)$  du système (2.3b).

Un vecteur de valeurs  $(x_1, \dots, x_n)$  est dit *solution réalisable* s'il vérifie les contraintes  $(i)$  et  $(ii)$  du système (2.3b).

Une solution est dite de *base* si elle possède  $m$  variables et  $n-m$  variables hors base. Les variables hors base doivent être nécessairement nulles. Une solution de base réalisable vérifie en plus toutes les contraintes. Cette dernière est obtenue en annulant les variables de décision. Elles sont donc des variables hors base et les variables d'écart sont de base. Evidemment cette solution de base réalisable n'est pas unique et on peut trouver d'autres.

La solution optimale est nécessairement de base.

### II.2.4. Méthodes de résolution

Dans la littérature existante on distingue deux classes de méthodes de résolution d'un programme linéaire (fonction objectif et contraintes linéaire) à savoir la méthode graphique et la méthode du *simplex* avec ses différentes variantes. Nous ne nous intéressons qu'à celle du *simplex* car la méthode graphique n'est applicable qu'aux programmes linéaires possédant au plus deux variables.

#### II.2.4.1. Méthode du *simplex*

Cette méthode a été mise au point par DANTZIG en 1948. Elle permet de résoudre les problèmes de la programmation linéaire. Son principe est de passer d'une solution de base réalisable à une autre solution de base réalisable de meilleure valeur de la fonction objectif. Dans le cas où il n'est pas possible d'améliorer cette valeur alors la solution optimale est atteinte, lorsqu'elle existe. Néanmoins, cette méthode nécessite une solution de base réalisable de départ. Cette condition n'est pas un handicap puisqu'on peut la trouver en utilisant la méthode des deux phases [7].

Considérons le problème

$$\left\{ \begin{array}{l} \text{Minimiser } f = \sum_{i=1}^N a_i x_i \\ \text{Sous les contraintes} \\ g_j = \sum_{i=1}^N b_{ji} x_i \geq c_j \quad j = 1, \dots, M_1 \\ h_j = \sum_{i=1}^N b_{ji} x_i \leq c_j \quad j = M_1 + 1, \dots, M \\ x_i \geq 0 \quad i = 1, \dots, N \end{array} \right. \quad (2.4)$$

#### - Travail préparatoire

Tout d'abord, on doit réécrire le problème (2.4) sous sa forme standard et ceci en ajoutant ou en soustrayant des variables hors base ( $x_{N+i}$ ) non négatives. Il devient donc sous la forme :

$$\left\{ \begin{array}{l} g_j = \sum_{i=1}^N b_{ji} x_i - x_{N+j} = c_j \quad j = 1, \dots, M_1 \\ h_j = \sum_{i=1}^N b_{ji} x_i + x_{N+j} = c_j \quad j = M_1 + 1, \dots, M \\ x_i \geq 0 \quad i = 1, \dots, N + M \end{array} \right. \quad (2.5)$$

Si une contrainte d'inégalité ( $\leq$  ou  $\geq$ ) est satisfaite en égalité alors la contrainte est dite *active* ou bien *serrée*.

La deuxième étape étant de choisir une variable pour chaque contrainte et de la représentée en fonction des autres, pour simplifier la formule nous choisissons les variables d'écart, ce qui nous donne :

$$\begin{aligned}
 g_j = x_{N+1} &= -c_j + \sum_{i=1}^N b_{ji} x_i & j = 1, \dots, M_1 \\
 h_j = x_{N+1} &= c_j - \sum_{i=1}^N b_{ji} x_i & j = M_1 + 1, \dots, M \\
 x_i &\geq 0 & i = 1, \dots, N + M
 \end{aligned} \tag{2.6}$$

Sous forme matricielle, on peut écrire :

$$\begin{aligned}
 [x_s] &= [c] + [b][x_D] \\
 \text{Min } f &= [a][x_D]
 \end{aligned} \tag{2.7}$$

où  $[x_s]$ ,  $[x_D]$  sont les vecteurs contenant les variables de base et hors base respectivement et  $[a]$ ,  $[b]$ ,  $[c]$  contiennent les coefficients du problème.

Les prochaines itérations consistent à faire entrer de variables hors base dans la base et sortir autres variables de la base jusqu'à l'itération finale où le minimum de la fonction est atteint selon les critères suivants :

– *Critère d'entrée* :

La variable candidate à entrer en base ( $x_r$ ) est celle qui possède le coefficient le plus élevé et positif dans le vecteur F.

– *Critère de sortie* :

La variable sortante ( $x_s$ ) est celle qui vérifie la condition  $\text{Min} \{c_i/b_{ir}, a_{ir} > 0\} = c_s/b_{sr}$   $b_{ir}$  est appelé *pivot*

– *Critère d'arrêt* :

L'optimum est atteint lorsque les coefficients des variables hors base dans le vecteur F sont tous négatifs ou nuls.

### II.3. Programmation non linéaire

La programmation non linéaire est une extension directe de la programmation linéaire, lorsque nous remplaçons le modèle linéaire de la fonction par un modèle non linéaire.

Les algorithmes numériques et programmes informatiques sont largement applicables sous forme de logiciels. Cependant, pour comprendre : comment les méthodes de l'optimisation travaillent, comment les programmes correspondants sont organisés, comment les résultats seront interprétés, et finalement, quelles sont les limitations de ces programmes; il est nécessaire de comprendre au moins la terminologie de base.

Nous présentons donc, une brève introduction sur les différentes méthodes d'optimisation en particulier celle que nous avons utilisée (**SQP**).

Les classes les plus populaires de la programmation non linéaire sont : pénalité, Lagrangien augmenté, séquentielle quadratique.

Avant de donner le principe de la méthode d'optimisations utilisée, nous donnons dans ce qui suit quelques définitions nécessaires.

### II.3.1. Définitions et théorèmes

1) On appelle  $V(x_0)$  voisinage de  $x_0$ , l'ensemble des vecteurs  $X$  vérifiant [7]:

$$\exists \varepsilon \text{ petit} / (x-x_0)^T (x-x_0) \leq \varepsilon \quad (2.8)$$

La fonction  $f$  admet un minimum au point  $x_0$  s'il existe un  $V(x_0)$  tel que  $f(x_0) \leq f(x)$  pour tout  $x$  dans  $V(x_0)$ . Dans le cas où  $\varepsilon$  est infini le minimum est global.

2) on appelle gradient de  $f$  le vecteur :

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (2.9)$$

La matrice Hessienne de  $f$  est la matrice  $H_f$  définie comme suit:

$$H_f = (h_{ij})_{ij} \quad \text{où} \quad h_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \quad (2.10)$$

3) la fonction  $f$  est dite convexe sur un domaine convexe  $D$ , si elle vérifie la condition suivante [13]:

$$\forall x, y \in D \text{ et } \lambda \in [0,1] f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \quad (2.11)$$

4) Optimum local et optimum global :

Dans plusieurs cas la fonction peut présenter plusieurs points vérifiant les conditions d'optimalité, ces optimums sont appelés optimums *locaux* et le point présentant la valeur la plus basse de la fonction est appelé *optimum global* [8].

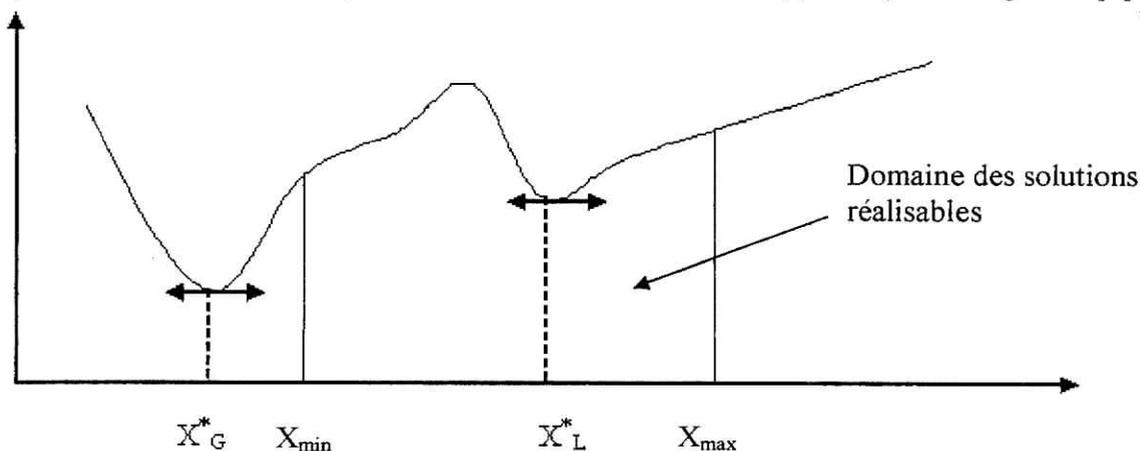


Figure 2.1: optimum local et optimum global

5) fonction de Lagrange:

L'un des outils fondamentaux pour tirer les conditions d'optimalité est la fonction de Lagrange:

$$L(x, u) = f(x) - \sum_{j=1}^m u_j g_j(x) \quad (2.12)$$

définie pour tout  $x \in \mathbb{R}^n$  et  $u = (u_1, u_2, \dots, u_m)^T$ .

les variables  $u_j$  sont appelées: *multiplicateurs de Lagrange*.

L'intérêt d'utiliser ces fonctions de Lagrange est de lier la fonction objectif  $f(x)$  aux contraintes  $g_i(x)$  et de permettre d'écrire un algorithme systématique pour la résolution du problème [12].

6) condition d'optimalité de Kuhn et Tucker:

On suppose que les fonctions  $f$  et  $g_i$  ( $i=1,2,\dots, m$ ) sont continument différentiables, et que l'hypothèse de qualification des contraintes est vérifiée en  $x^0 \in X$  Avec:

$$X = \{x \in \mathbb{R}^n / g_i(x) \leq 0, \quad \forall i = 1, 2, \dots, m\}$$

Alors une condition nécessaire pour que  $x^0$  soit un optimum local du problème est qu'il exist des nombres  $\lambda_i \geq 0$ , ( $i=1, \dots, m$ ) appelé multiplicateurs de Kuhn et Tucker, tel que [14], [9]

$$\begin{cases} \nabla f(x^0) + \sum_i \lambda_i \nabla g_i(x^0) = 0 \\ \lambda_i g_i(x^0) = 0 \end{cases}$$

7) Soit  $\bar{x} \in S$  et  $\bar{\lambda} \geq 0$ , on dit que  $(\bar{x}, \bar{\lambda})$  est un *point-coll* de  $L(x, \lambda)$  si [14], [9]:

$$\begin{aligned} \text{a) } & L(\bar{x}, \bar{\lambda}) \leq L(x, \bar{\lambda}) \quad \forall x \in S \\ \text{b) } & L(\bar{x}, \bar{\lambda}) \leq L(\bar{x}, \lambda) \quad \forall \lambda \geq 0 \end{aligned}$$

ou  $S$  l'intervalle dans le quel peut varier  $x$

8) Si  $(\bar{x}, \bar{\lambda})$  est un *point-coll* de  $L(x, \lambda)$  alors  $\bar{x}$  est un optimum global de  $P$ . [9]

### II.3.2. Méthode d'optimisation non linéaire avec contraintes

La plupart des méthodes existantes en programmation non linéaire sous contraintes peuvent se rattacher à deux grandes familles [14]:

- méthodes primales (directes)
- méthodes duales.

Les méthodes primales se caractérisent par le fait qu'elles opèrent directement sur le problème, elles engendrent une séquence de solutions (c'est-à-dire de points satisfaisant les contraintes) en assurant une décroissance monotone de la fonction à minimiser. Elles présentent un avantage important: si le processus itératif est interrompu elles procurent une solution approchée satisfaisant les contraintes. Par contre, elles sont de mise en œuvre délicate et la convergence globale est souvent difficile à obtenir. De plus la vitesse de convergence vers la solution dépend grandement, pour les méthodes différentiables, de l'estimation des dérivées premières des fonctions du problème étudié, sans exploiter les information

du second ordre ce qui limite, pour certains problèmes, l'efficacité de ces méthodes. Plus généralement il est préférable d'utiliser une des méthodes duales que nous allons décrire dans ce qui suit.

Les méthodes duales sont plus robustes et la convergence globale est souvent plus facile à obtenir; en contre partie elles présentent l'inconvénient de ne fournir une solution primale réalisable qu'en fin de convergence, le principe commun de ces méthodes consiste à ramener le problème initial à la résolution d'une séquence de problèmes d'optimisation sans contraintes en exploitant les informations données par la fonction de Lagrange.

### **II.3.3. Choix de la méthode d'optimisation**

Déclarer que tous les problèmes d'optimisation des structures sont non linéaires n'est pas une exagération, en effet, dans notre cas (optimisation de la fonction poids d'une structure en portiques plans) la non linéarité de la fonction objectif est évidente puisqu'elle est en fonction du produit  $b_i \cdot h_i$  qui sont deux variables indépendantes.

Les contraintes d'optimisation, quant à elles, sont aussi non linéaires, leur non linéarité provient du fait que le calcul des contraintes dans les éléments nécessite la détermination des déplacements dont leurs valeurs dépendent de l'inverse des matrices de rigidité. Ces dernières sont en fonction du moment d'inertie ( $b \cdot h^3$ ) et de la section ( $b \cdot h$ ).

Donc il est nécessaire de choisir une méthode d'optimisation non linéaire avec contraintes pour résoudre le problème d'optimisation des structures en portiques plans.

### **II.3.4. Pour quoi choisir la méthode SQP ?**

Notre problème étant non linéaire avec contraintes. Donc le choix de la méthode d'optimisation doit s'effectuer entre l'une des méthodes traitant cet ensemble de problèmes, en effet **SQP** est la méthode la plus adéquate pour les raisons suivantes [8], [11], [10] :

- la convergence de la méthode est assurée quelque soit le point de départ (dans le domaine faisable ou non), ceci est dû au caractère dual de la méthode.
- la mise en œuvre en algorithme informatique est facilement réalisable en FORTRAN (l'algorithme NLPQL conçu par Schittkowski)
- elle constitue une synthèse des autres méthodes donc elle profite de leurs avantages (calcul de la direction, détermination du Hessien,...)

*Remarque:*

Le programme **NLPQL** utilisant la méthode **SQP** constitue une part des systèmes d'optimisation commercialisés tel que

- ANSYS/POPT (CAD-FEM, Gra.ng) pour l'optimisation des structures.
- STRUREL (RCP, Munich) pour la fiabilité des analyses,

- Microwave O.c.e Suit (Applied Wave Research, El Segundo) pour la conception électronique,
- POINTER (Synaps, Atlanta) pour la conception automatique en mécanique.

Notons que ces logiciels d'optimisation sont très chères et ne sont disponibles que chez les grandes entreprises.

### II.3.5. Méthode séquentielle quadratique SQP

Considérons le problème d'optimisation non linéaire avec contraintes général suivant :

$$(P) \begin{cases} \text{Minimiser } f(x) \\ \text{Sous les contraintes} \\ g_i(x) \leq 0 \quad i = 1, \dots, m \end{cases}$$

Les conditions d'application de cette méthode sont :

- le problème n'est pas trop grand (nombre de variables et de contraintes limité);
- les fonctions et leurs gradients peuvent être évaluées avec une bonne précision.

L'idée de base de la méthode est de formuler et résoudre à chaque itération un sous problème de *programmation quadratique* (fonction objectif quadratique et contraintes linéaires) qui obtenu par la linéarisation des contraintes et l'approximation quadratique de la fonction de Lagrange:  $L(x, \lambda) = f(x) - \sum_{j=1}^m \lambda_j g_j(x)$  où  $x \in \mathbb{R}^n$  est le vecteur des variables du problème et  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T \in \mathbb{R}^m$  le vecteur multiplicateur.

Notre problème est alors amené à la résolution du "problème de programmation quadratique" suivant :

$$d \in \mathbb{R}^n \begin{cases} \min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B_k d + \nabla f(x_k)^T d \\ \text{sous les contraintes} \\ \nabla g_j(x_k)^T d + g_j(x_k) = 0 \quad j = 1, \dots, m_e \\ \nabla g_j(x_k)^T d + g_j(x_k) \geq 0 \quad j = m_e, \dots, m \end{cases}$$

Où  $d$  est le vecteur direction variable de ce sous problème de minimisation et soit  $d_k$  la solution optimale de ce problème, alors la nouvelle itération  $X_{k+1}$  est obtenue

$$\text{par: } \begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}$$

où  $\alpha_k \in ]0,1]$  est le pas à l'itération donnée.

On voit bien, comme dans le reste des méthodes d'optimisation faisant appel au différentielles des fonctions et contraintes étudiées, que la convergence de la méthode **SQP** dépend de la direction de recherche  $d_k$  et du pas  $\alpha_k$  à l'itération  $k$ . La direction de recherche  $d_k$  est déterminée, comme nous l'avons vu par la résolution d'un problème d'optimisation quadratique ainsi la direction de recherche se trouve elle-même être optimale parmi toutes les directions possibles.

Il ne reste donc que le paramètre pas " $\alpha_k$ " à définir de manière efficiente. Le pas doit satisfaire la condition de décroissance d'une fonction d'évaluation dite "mérite" [10],[14]. Le pas  $\alpha_k$  est déterminé de manière unique en satisfaisant la condition générale d'Armijo [10], cette condition est fondamentale car elle impose au pas:

- De ne pas être trop grand, sinon l'algorithme risque d'avoir un comportement oscillatoire.
- De ne pas être trop petit sinon l'algorithme risque de converger prématurément.

Algorithme de la méthode :

L'algorithme de la méthode SQP est le suivant :

- a) à l'itération 0 on est en  $x_0$  (point de départ), avec  $v_0=0$ ,  $B_0=I$  et  $u_0>0$
- b) à l'itération k on est en  $x_k$ .
- c) calculer  $f$ ,  $\nabla f$ ,  $g_j$ ,  $B_k$  à l'itération courante k.
- d) formuler le problème quadratique à l'itération courante.

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B_k d + \nabla f(x_k)^T d$$

sous les contraintes

$$\nabla g_j(x_k)^T d + g_j(x_k) = 0 \quad j = 1, \dots, m_e$$

$$\nabla g_j(x_k)^T d + g_j(x_k) \geq 0 \quad j = m_e, \dots, m$$

Résoudre le problème par la méthode de Simplex (modifiée).

- e) Formuler la fonction d'évaluation dite "mérite" :

$$\phi_r(\alpha) = \psi_r \left[ \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} \right]$$

en fonction de  $d_k$ ,  $x_k$ ,  $u_k$ ,  $v_k$ .

- f) Fixer  $\beta$  et  $\sigma$ , constantes à l'itération k, poser  $i=0$ .

Varier  $i < 0$  jusqu'à satisfaire une première fois la condition la condition d'Armijo:

$$\phi_r(\sigma \beta^i) \leq \phi_r(0) + \sigma \beta^i \mu \phi'(0)$$

Poser  $\alpha_k = \sigma \beta^i$

- g) Si le couple  $(x_k, v_k)$  est un point-coll de la fonction de Lagrange, alors le vecteur  $x_k$  est un optimum global du problème.

Sinon faire:  $\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}$  et retour en (b).

Les détails de cette méthode (choix du pas, choix de la direction détermination du Hessien sont dans [10] et [12].

## **II.4. Conclusion**

L'optimisation des structures est indispensable pour concevoir des systèmes de plus en plus performants. L'objet de l'optimisation est de minimiser une fonction objectif tout en respectant un certain nombre de contraintes physiques et mécaniques (limitations du domaine de solution). La méthode séquentielle quadratique **SQP** –permettant de traiter les problèmes généraux non linéaires avec contraintes- est robuste et sa convergence est assurée quelque soient les conditions du problème sans perdre vue que les contraintes ne doivent pas être contradictoires. Le seul inconvénient de cette méthode est qu'elle nécessite le calcul de l'Hessien à chaque itération, ce qui est consommateur en temps de calcul.

11

CHAPITRE 3  
PRESENTATION DU PROGRAMME  
*"OPTIPOINT"*

### III.1. Introduction :

Le programme *OPTIPORT* que nous avons réalisé consiste à calculer les contraintes maximales (aux niveaux des sections les plus sollicitées) dans les éléments constituant les portiques plans, ensuite à déterminer la section optimale de chaque élément en se basant sur les valeurs des contraintes déjà calculées.

Nous avons intégré dans ce programme le code de calcul MEF3 conçu par G.DHATT et G.TOUZOT basé sur la méthode des éléments finis sur lequel nous avons opéré les modifications nécessaires pour le rendre spécifique au calcul des portiques plans.

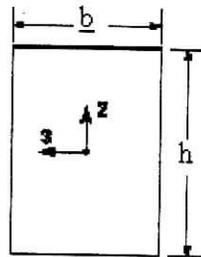
Ce chapitre sera consacré à la présentation du programme *OPTIPORT* ainsi que la manière dont ces différents sous programmes s'articulent.

### III.2. Mise en œuvre du problème d'optimisation des structures en portiques plans:

Le programme *OPTIPORT* que nous avons réalisé fournit le poids optimal (c'est-à-dire les sections optimales) de la structure sous un chargement imposé.

L'analyse de la structure est effectuée par la partie *PORT* basée sur la méthode des éléments finis afin de déterminer la contrainte maximale dans chaque élément du portique (nous détaillerons ce point dans les paragraphes qui suivent)

Les sections droites ont une forme rectangulaire dont les variables seront :  
La hauteur  $h$   
La largeur  $b$



Le matériau utilisé est l'acier avec hypothèse d'une relation contrainte déformation linéaire, et d'une résistance caractéristique à la traction égale à celle à la compression. La détermination des contraintes maximales se fera après avoir déterminé les efforts maximaux en utilisant les équations décrites dans le chapitre 1.

La partie optimisation *OPTI* est basée sur la méthode *SQP* (séquentiel quadratic programming). Le choix de cette méthode est justifié dans le chapitre 2.

Le problème étant formulé de la manière suivante:

$$\begin{array}{l}
 \text{minimiser } f(x) = \gamma \sum_1^m l_i \cdot h_i \cdot b_i \\
 \text{sous les contraintes:} \\
 p) \left\{ \begin{array}{l}
 |\sigma_{\text{lim}}| - |\sigma_{i \text{ max}}| \geq 0 \\
 b_i h_i^{-1} - R_{\text{min}} \geq 0 \\
 R_{\text{max}} - b_i h_i^{-1} \geq 0 \\
 b_{\text{min}} \leq b_i, h_i \leq h_{\text{max}}
 \end{array} \right. \quad i = 1, \dots, m
 \end{array}$$

où

$f(x)$  est la fonction objectif : poids de la structure

$x$ : vecteur, à  $2 \cdot m$  composantes, des dimensions des éléments

$\gamma$  : Poids spécifique du matériau constituant les éléments de la structure

$m$ : nombre d'éléments

$b_i$ : largeur de l'élément  $i$

$h_i$ : hauteur de l'élément  $i$

$l_i$  : longueur de l'élément  $i$

$\sigma_i$  : contrainte maximale dans l'élément  $i$ , qui est égale à : 2400 kg/cm<sup>2</sup>

$\sigma_{lim}$  : contrainte limite de traction/compression de l'acier utilisé

$R_{min}$ : borne inférieure du rapport  $b/h$

$R_{max}$ : borne supérieure du rapport  $b/h$

$b_{min}$ : borne inférieure des dimensions

$h_{max}$  : borne supérieure des dimensions.

En notant que: les valeurs de  $\sigma_{lim}, R_{min}, R_{max}, b_{min}, h_{max}$  sont fixées par les règlements et elles sont introduites par l'utilisateur du programme, donc on peut faire une étude comparative et ceci en variant les valeurs de ces limites.

Nous donnons dans ce qui suit l'organisation générale du programme *OPTIPORT* puis nous détaillons chaque partie séparément.

### ***III.3. Description générale du programme OPTIPORT***

Le programme *OPTIPORT* est composé de deux parties:

- La partie *PORT* qui se charge de l'analyse de la structure avec les sections proposées.
- La partie *OPTI* qui calcule la fonction objectif et l'optimise sous les contraintes données.

Ces deux parties s'enchaînent de la manière suivante:

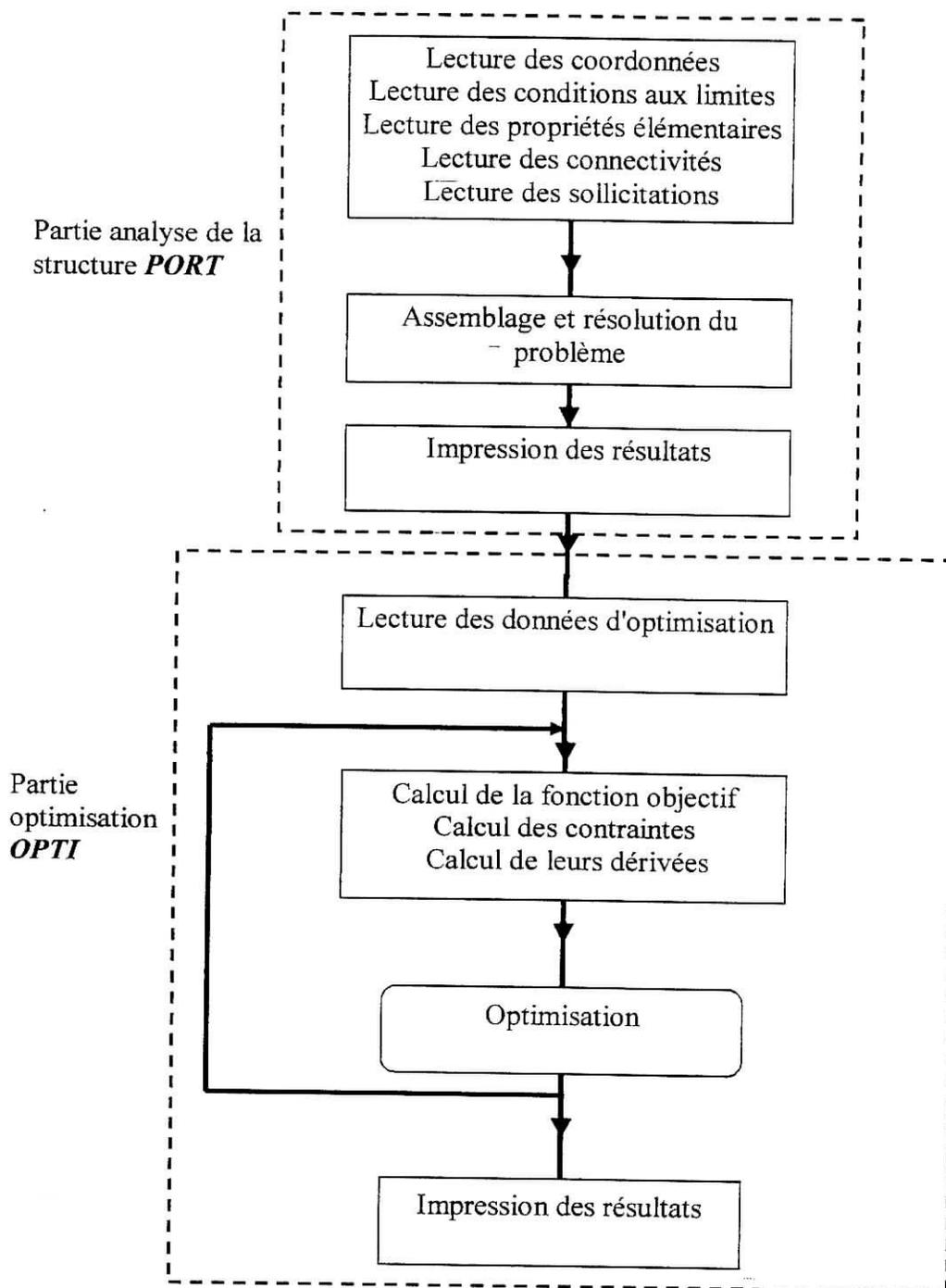


Figure 3.1: organigramme général du programme *OPTIPORT*

### III.3.1. Introduction des données:

Le programme *OPTIPORT* nécessite un nombre assez grand de variables à introduire suivant des formats spécifiques avec un enchaînement donné.

Pour éviter toute erreur due à la confusion des données et pour faciliter à l'utilisateur l'opération d'introduction des données, nous avons réalisé un sous programme d'introduction des données que nous avons appelé (*PRDON*).

Ce sous programme se charge de l'écriture des données suivant les formats et l'enchaînement nécessaires (sans obliger l'utilisateur à le faire). Ceci nous permet d'avoir un fichier de données cohérent avec les différents sous programmes chargés de l'écriture des données.

Les différentes paramètres demandés par le sous programme (*PRDON*) sont:

*Nom* de fichier des données.

*NNT*: Nombre de noeuds total.

*NDLN*: Nombre de degrés de liberté par noeud.

*NDIM*: Dimension du problème.

$X_i, Y_i$ : Les coordonnées du noeud  $i$ .

*COLI*: Les différents types de conditions aux limites (nombre d'encastrement, nombre d'appuis double, nombre d'appuis simples suivant X, nombre d'appuis simple suivant Y)

*NAP*: Les noeuds correspondants à chaque type d'appui.

*NPRE*: Nombre de propriété par élément.

*NGPE*: Nombre de groupes de propriétés élémentaires (égal au nombre d'éléments).

$V_i$ : Valeurs des différentes propriétés (module de Young et les dimensions des sections).

*NELT*: Nombre d'éléments total.

*KNE*: liste des numéros des noeuds de chaque élément.

*VSN*: Valeur des sollicitations nodales correspondant à chaque degré de liberté.

*KVSN*: Liste des numéros des noeuds ainsi sollicités.

*VSC*: valeurs des sollicitations concentrées en travées.

*KVSC*: Liste des numéros des noeuds ainsi sollicités.

*VCR*: Valeurs des charges réparties appliquées.

*KVCR*: Liste des éléments ainsi sollicités.

*N*: Nombre de variables d'optimisation.

*NC*: Nombre de contraintes d'optimisation.

*MAXITN*: Nombre max d'itérations.

*XLB*: La limite inférieure de B et H.

*XUB*: La limite supérieure de B et H.

*CLIM*: Limite élastique des contraintes.

*RILIM*: Limite inférieure du rapport B/H.

*RSLIM*: Limite supérieure du rapport B/H.

*Remarques*:

- Nous avons veillé à ce que ce sous programme (*PRDON*) soit aussi général que possible pour offrir la possibilité de l'utiliser pour d'autres cas (3 dimensions, autres types de profils,...).
- L'appel de ce sous programme est facultatif, l'utilisateur peut utiliser des fichiers de données déjà préparés.

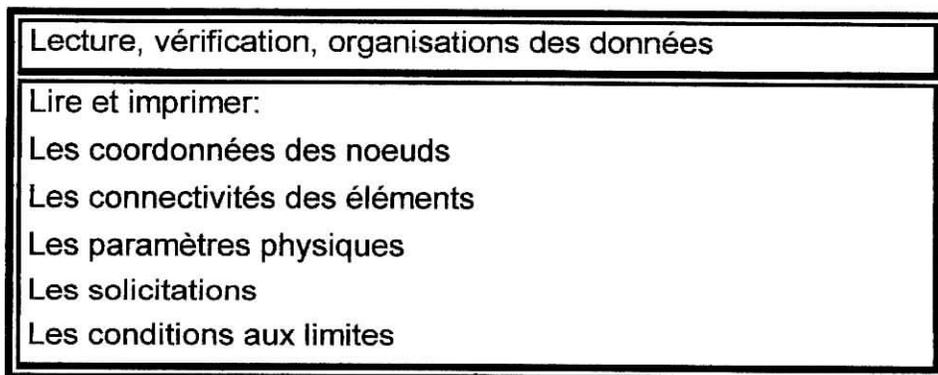
Après avoir introduit les données, le programme commence par analyser la structure à l'aide de la partie (*PORT*) décrite ci-dessous.

#### **III.4. Organisation générale de la partie analyse de la structure (port):**

Tout programme basé sur la méthode des éléments finis inclut quelques blocs fonctionnels caractéristiques:

- Lecture, vérification et organisation des données décrivant le maillage (noeuds et éléments), les paramètres physiques (poids spécifique, module d'élasticité, etc.), les sollicitations et conditions aux limites;
- Construction des matrices et vecteurs élémentaires, puis assemblage de ceux-ci pour former la matrice globale et le vecteur global des sollicitations;
- Résolution du système d'équations après prise en compte des conditions aux limites;
- Impression des résultats après calcul éventuel de variables additionnelles (contraintes, réactions, etc.).

Le tableau ci dessous montre l'enchaînement de ces différents blocs:



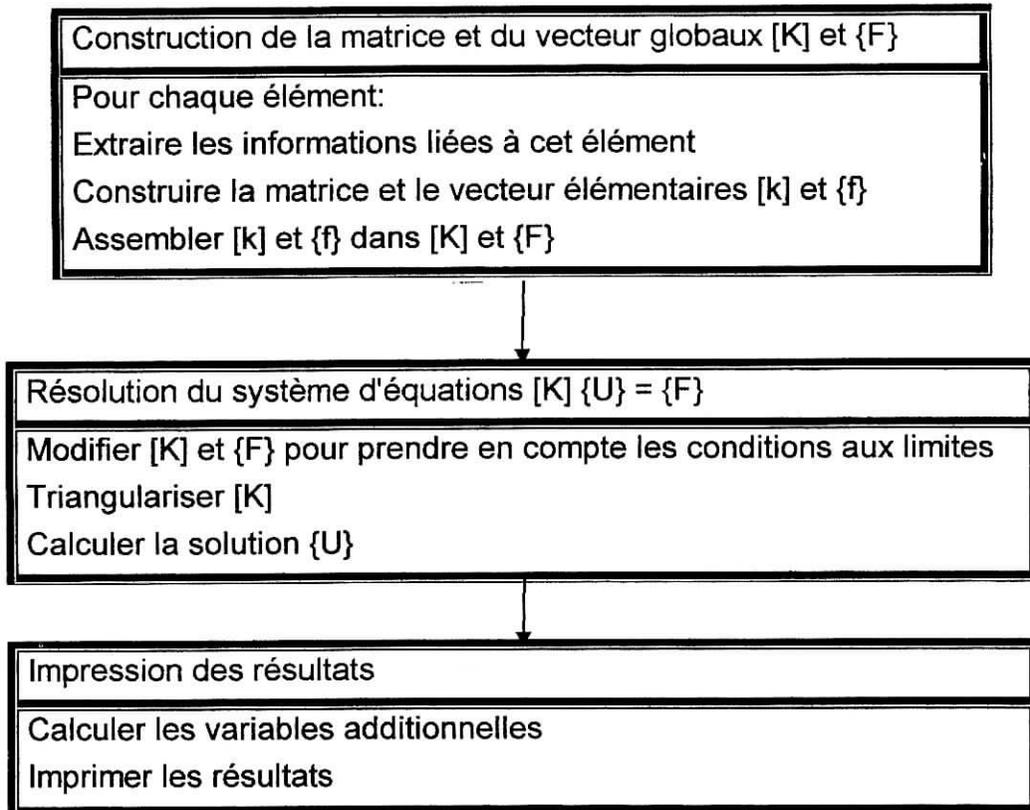


Tableau 3.1: blocs fonctionnels caractéristiques d'un programme de la MEF

L'organigramme suivant représente les différents sous programmes constituant cette partie :

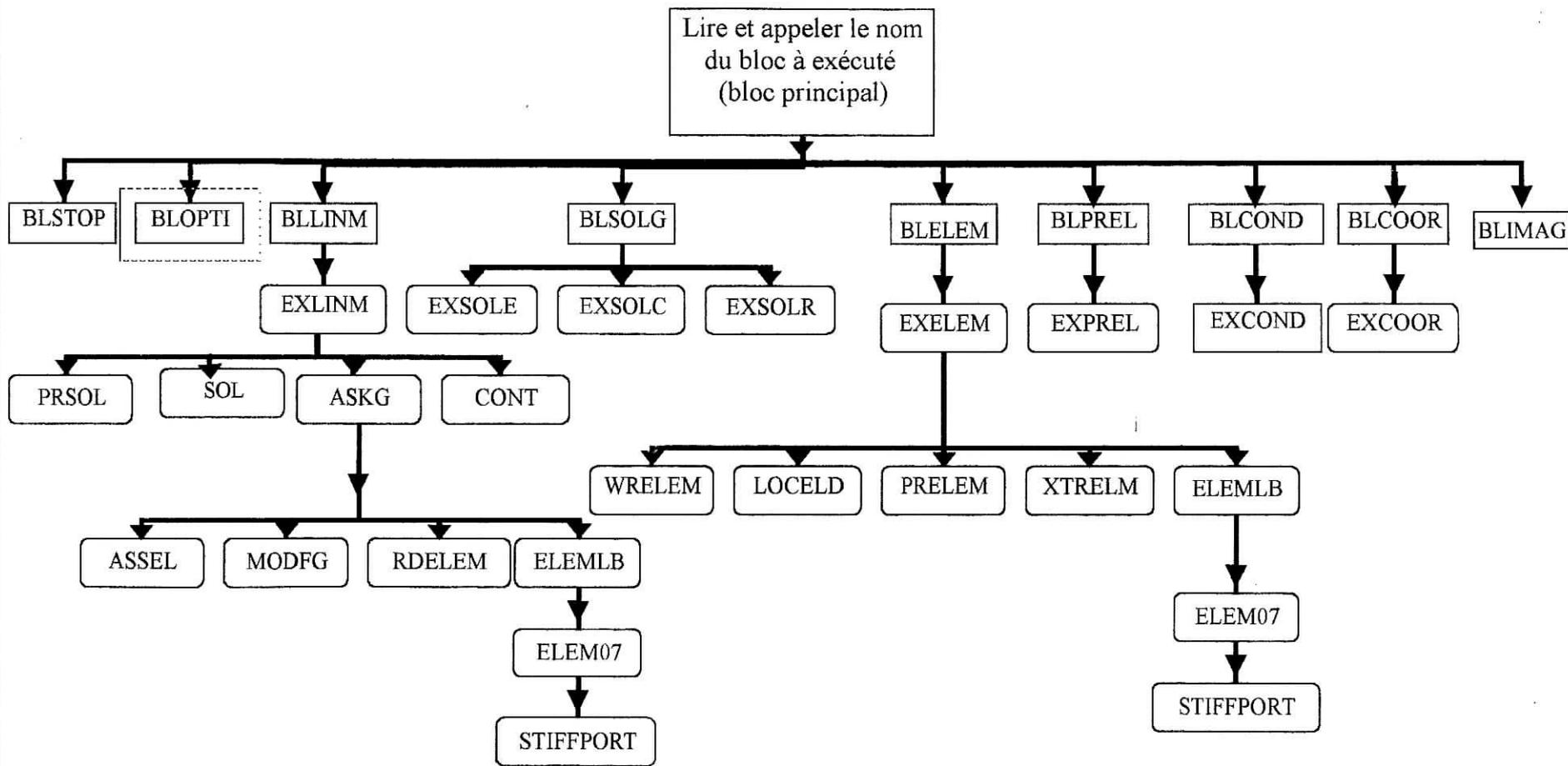


Figure 3.2: organigramme général du programme OPTIPORT

Nous nous restreindrons dans ce qui suit sur le détail des sous-programmes *EXSOLR*, *EXSOLE*, *STIFFPORT*, *CONT*, spécifiques aux calcul des portiques plans, les autres sous-programmes sont détaillés dans [2].

#### **III.4.1. Sous-programme EXSOLR:**

Ce sous-programme est appelé par le bloc *SOLG* chargé de la lecture et de l'assemblage des différents vecteurs de sollicitations (réparties, nodales, concentrées en travées, poids propre) dans un vecteur de sollicitation global (*VFG*) qui sera actualisé à chaque itération (par le sous-programme *MODVFG*) du fait de la variation du poids propre au cours du processus d'optimisation.

Il effectue deux opérations:

- La première consiste à calculer les sollicitations équivalentes élémentaires dues au poids propre.
- La seconde est de calculer ces mêmes sollicitations dues aux autres types de charges réparties.

Il crée les fichiers suivants:

**Solep**: où il enregistre les **SOL**licitations **E**quivalentes dues au **P**oids propre dans chaque nœud de la structure.

**Soler**: où il enregistre les **SOL**licitations **E**quivalentes dues aux charges **R**éparties autres que le poids propre. Si un élément ne supporte que son poids propre la valeur zéro est affectée à cet élément dans ce fichier.

A partir de ces deux fichiers le programme lit ces sollicitations lors du calcul des efforts internes élémentaires dans le sous-programme *CONT*.

Les expressions des efforts équivalents aux charges réparties sont donnés dans le chapitre 1 § 1.5.3

L'organigramme de ce sous-programme est le suivant:

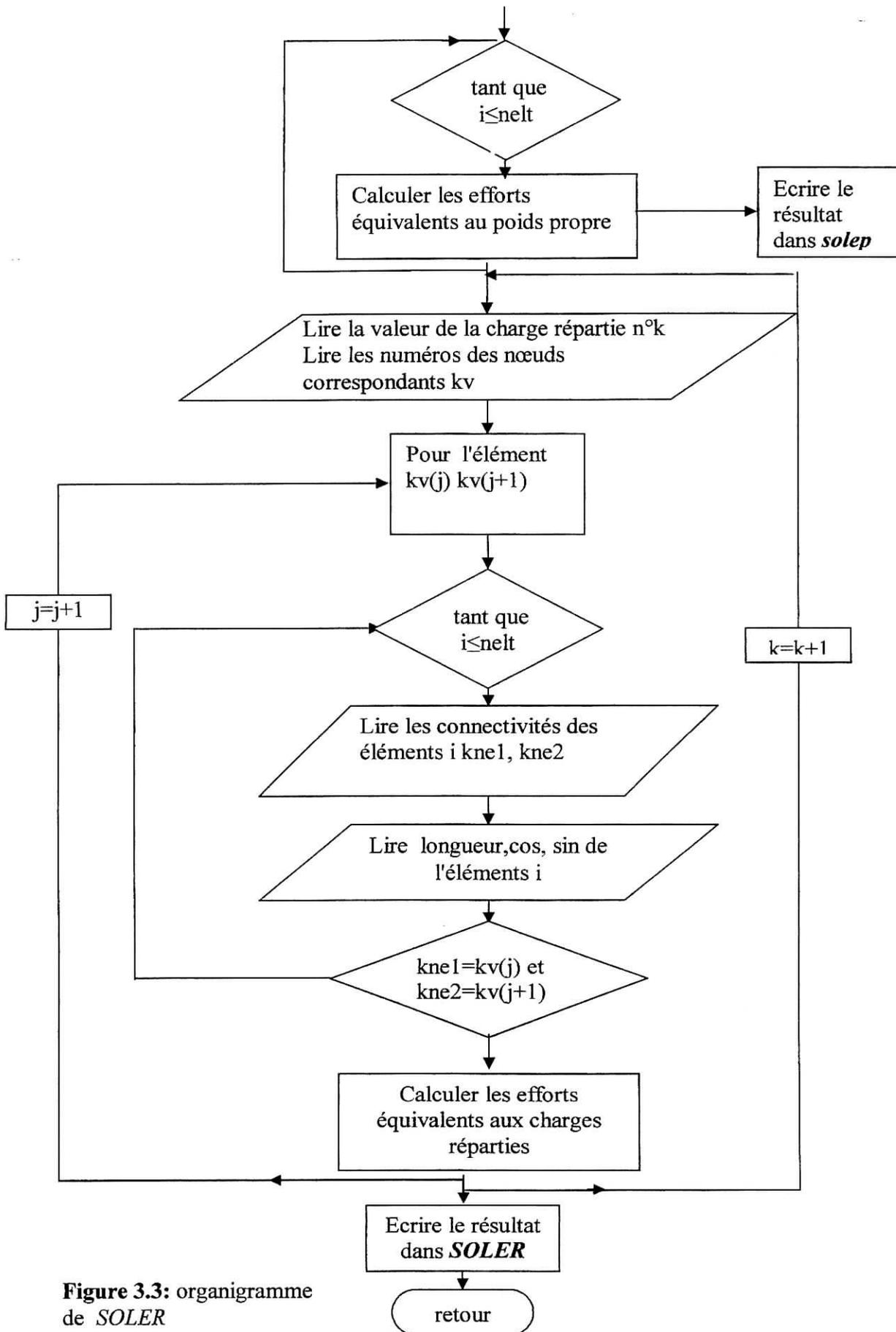


Figure 3.3: organigramme de *SOLER*

### III.4.2. Sous-programme *SOLE* :

Ce sous-programme est appelé par le bloc *BLSOLG*.

Il est chargé des opérations suivantes:

- Détermination des sollicitations équivalentes aux niveaux des nœuds dues aux charges concentrées en travées.
- Création du fichier *SOLC* où il enregistre les forces équivalentes pour les utiliser dans le calcul des efforts internes.

L'organigramme général de ce sous-programme est représenté sur la figure:(3.4).

*Remarques:*

a) Les vecteurs obtenus par les trois sous-programmes *SOLC* (sollicitations concentrées), *SOLE* (sollicitations concentrées en travées), *SOLR* (sollicitations réparties) sont assemblés dans le vecteur force global (*VFG*) en utilisant les tables :

-*KDLNC*: crée par le bloc *COOR*, de dimension  $(NNT+1)$  et qui contient les degrés de liberté par nœud cumulative tel que:

$KDLNC(L+1)$ =le nombre total de degrés de liberté des noeuds 1,2,..., L dans le cas des portiques plans, il est égal à  $3*L$ .

$KDLNC(NNT+1)$ =le nombre total de degrés de liberté= $3*NNT$  (dans le cas des portiques plans).

-*KNEQ*: crée par le bloc *ELEM*, elle est de dimensions *NDLT* (nombre de degrés de liberté total)

si on pose  $J=KNEQ(L)$

$J > 0 \rightarrow$  le degré de liberté L est inconnu et correspond à l'équation j dans le système d'équation.

$J < 0 \rightarrow$  le degré de liberté L est imposé (conditions au limite).

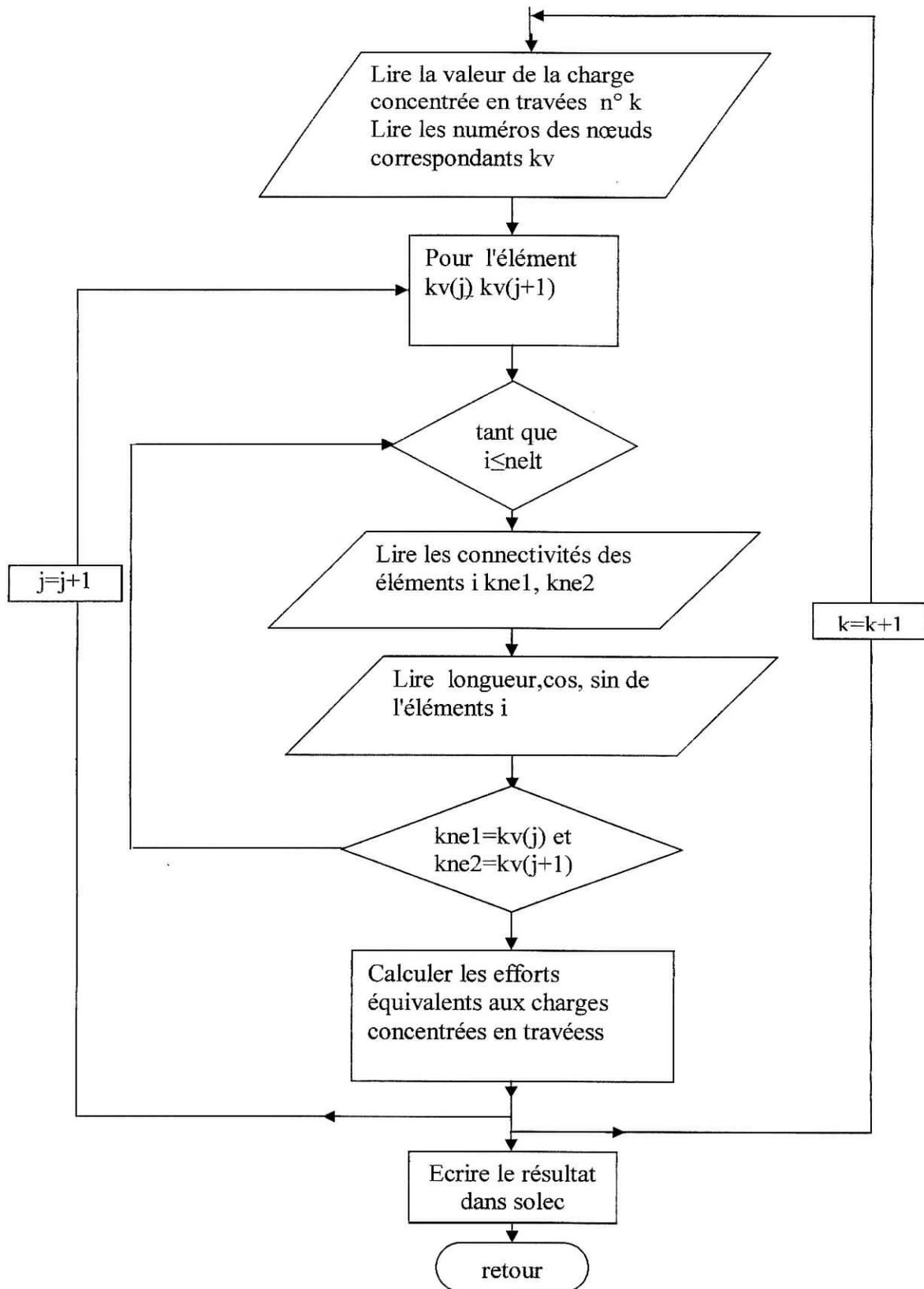


Figure 3.4: organigramme de *SOLE*

b) le vecteur sollicitations globales (*VFG*) est assemblé selon l'algorithme suivant:

Début

À la sollicitation n°:  $i$

Lire la valeur de la charge  $V(i)$

Lire les nœuds concernés  $KV(i)$

$i_1 = KV(i)$

Si ( $i \leq 0$ ) aller à la fin (fin des noeuds)

$id_1 = KDLNC(i_1) + 1$  numéro du premier degré de liberté du nœud  $i_1(U_{i_1})$

$id_2 = KDLNC(i_1 + 1)$  numéro du troisième degré de liberté du nœud  $i_1(\theta_{i_1})$

Pour  $id$  allant de  $id_1$  à  $id_2$  faire

$j = j + 1$

$ieq = KNEQ(id)$  ( position finale du noeud  $id$  dans le system d'équation)

    Si ( $ieq \leq 0$ ) reboucler (noeud appuis)

$VFG(ieq) = VFG(ieq) + V(j)$  (ajouter les sollicitations appliquées sur ce nœud dans sa place dans le système global)

Fin pour

FIN

Cette même procédure est répétée pour les tables contenant les sollicitations équivalentes (à chaque nœud) aux sollicitations réparties et concentrées en travées.

### **III.4.3. Sous-programme STIFFPOR:**

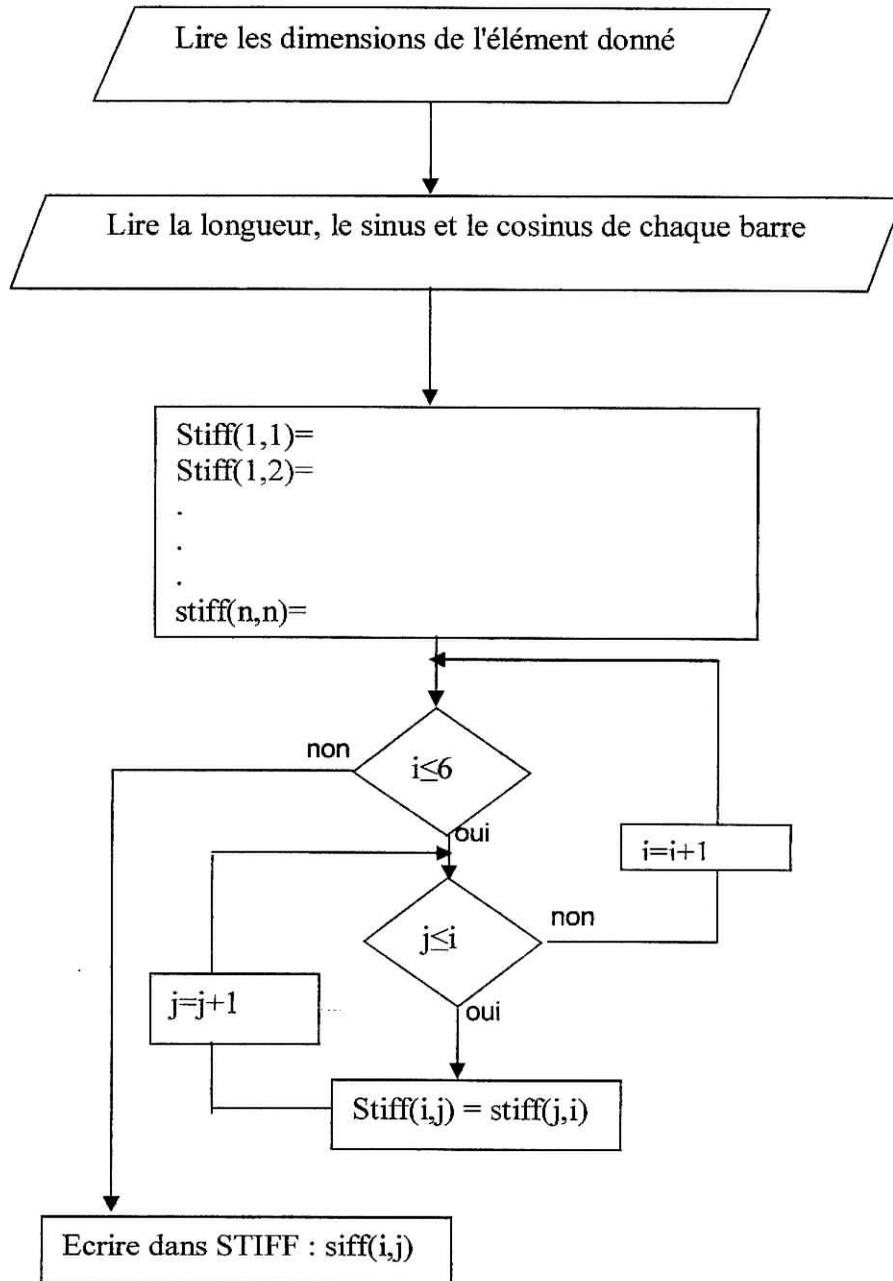
Ce sous-programme a pour rôle la construction de la matrice de rigidité élémentaire pour un élément type portique, cette dernière est donnée par l'équation (1.14) chapitre 1.

Ce sous-programme est appelé par :

- Le bloc **ELEM** pour écrire la matrice de rigidité de chaque élément.
- Le sous-programme **ASKG** pour assembler les différentes matrices élémentaires et construire la matrice de rigidité globale.
- Le sous-programme **ASKGO** pour le même but que **ASKG** mais dans le processus d'optimisation.

Ce sous-programme a pour objet la création du fichier **STIFF** où il enregistre les matrices élémentaires.

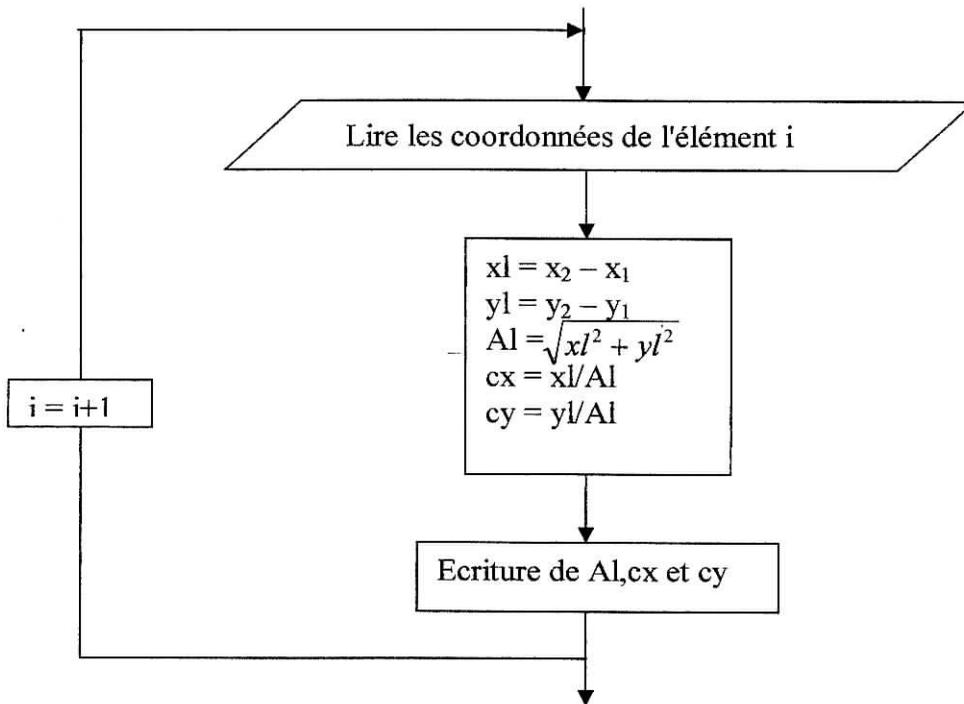
L'organigramme général de ce sous-programme est le suivant:



**Figure 3.5:** Organigramme du sous-programme *STIFFPOR*

*Remarque :*

Le calcul des longueurs, sinus et cosinus des différents éléments s'effectue dans le bloc *ELEM* en suivant l'organigramme ci dessous :



**Figure 3.6:** Organigramme de calcul des longueurs, sinus et cosinus des différents éléments

#### III.4.4. Sous-programme *CONT*:

Ce sous-programme a pour objet :

- Le calcul des efforts internes (N, T, M) au niveau de chaque nœud
- Le calcul des contraintes maximales selon la procédure suivante :

Pour l'élément *iel*

- Déterminer les efforts internes dans les nœuds à partir de la formule  $\{F\} = [K]\{u\} - \{f\}$  tel que

$\{F\} = \langle N_1, T_1, M_1, N_2, T_2, M_2 \rangle^t$  : efforts internes dans l'élément *iel*

$[K]$  : matrice de rigidité élémentaire construite par le sous-programme *STIFFPORT* et sauvegardée dans le fichier *STIFF*.

$\{u\} = \langle u_1, v_1, \theta_1, u_2, v_2, \theta_2 \rangle^t$  : déplacements nodaux de l'élément *iel* déterminé par le sous-programme *EXLINM*.

$\{f\} = \langle N_{eq1}, T_{eq1}, M_{eq1}, N_{eq2}, T_{eq2}, M_{eq2} \rangle^t$  : efforts équivalents dans les nœuds calculés par les sous-programmes *EXSOLR* et *EXSOLE* et sauvegardés dans les fichiers *SOLC* et *SOLP*.

Pour les éléments supportant des charges réparties et des charges concentrées en travées ; déterminer la valeur du moment maximal en travées (voir figure 3.6):

Pour  $x \leq a$

$$M(x) = T_1 \cdot x - M_1 - q \cdot (x^2) / 2$$

Le moment maximal est à la position :  $x = T_1 / q$

Pour  $a \leq x \leq L$

$$M(x) = T_1 \cdot x - M_1 - q \cdot (x^2) / 2 - p \cdot (x - a)$$

Le moment maximal est à la position :  $x = (T_1 - p) / q$ .

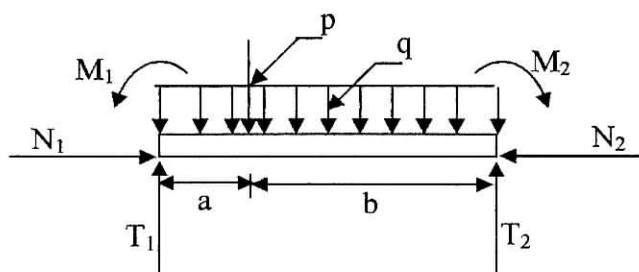
– Calculer les trois contraintes suivantes :

La contrainte dans le premier nœud de l'élément.

La contrainte dans le deuxième nœud de l'élément.

La contrainte max en travée.

La contrainte dans l'élément sera le maximum de ces trois contraintes.



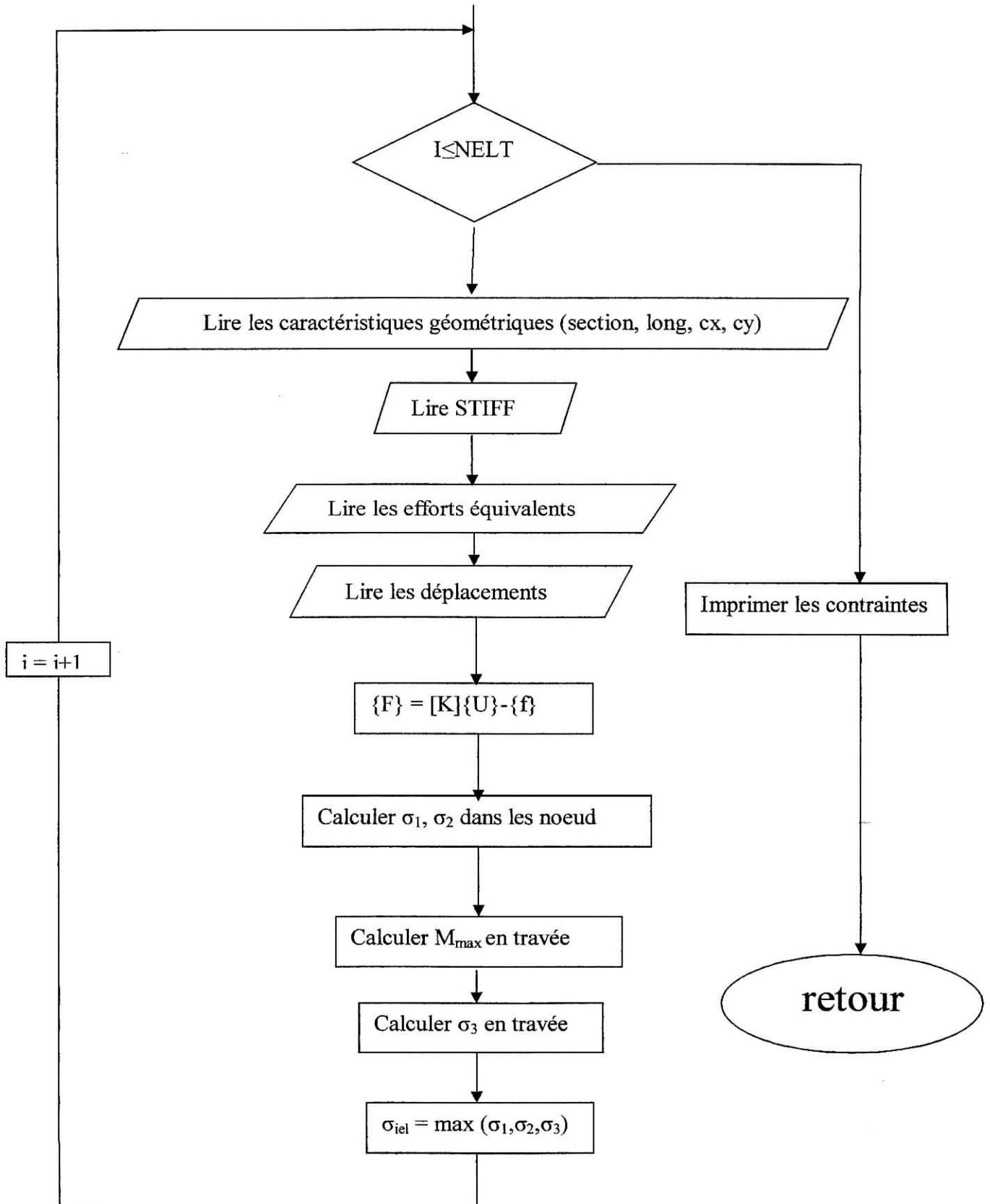
**Figure 3.6:** poutre bi encastée soumise à une charge répartie et une force concentrée en travées

*Remarque :*

Le sous-programme **CONT** effectue les changements nécessaires pour tenir compte des cas des poteaux (barres verticales).

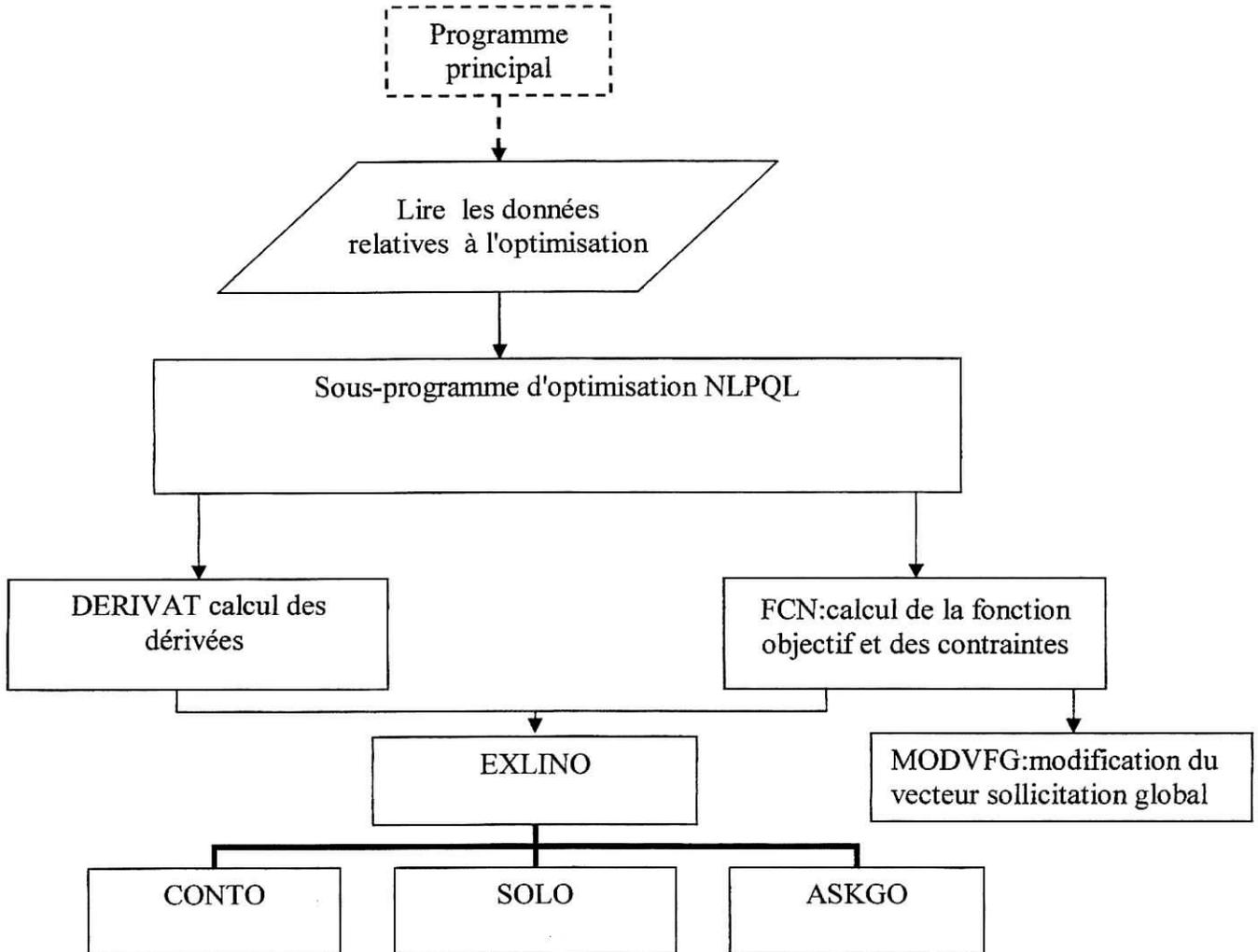
Ce sous-programme est appelé par le bloc **BLINM** chargé de l'assemblage et de la résolution d'un système linéaire.

L'organigramme du sous-programme **CONT** est le suivant :

Figure 3.7: Organigramme du sous-programme *CONT*

### III.5. Description du bloc d'optimisation

Nous allons dans ce qui suit décrire le programme d'optimisation ainsi que les sous programmes qui s'y rattachent. Commençons tout d'abord par donner l'organigramme général du bloc *OPTI*



**Figure 3.8** organigramme général du bloc d'optimisation

Les données relatives à l'optimisation sont :

- .MAXITN: nombre maximum d'optimisation
- .N : nombre de variables
- .NC : nombre de contraintes d'optimisation
- .CLIM: contrainte admissible
- .RILIM: limite inférieure du rapport b/h
- .RSLIM: limite supérieure du rapport b /h
- .XLB: limite inférieure des variables d'optimisation (b et h)

*XUB*: limite supérieure des variables d'optimisation (b et h)

*Remarque* :

Nous donnons maintenant les détails de certains sous programme constituant ce bloc:

### III.5.1. *Sous-programme NLPQL* :

C'est le sous programme chargé de l'optimisation des problèmes non linéaires avec contraintes en utilisant la méthode séquentielle quadratique.

Le sous-programme NLPQL est basé sur le code de FORTRAN développé par Schittkowski (1986). Il utilise la méthode séquentielle quadratique pour résoudre les problèmes non linéaires généraux déclarés comme suit:

$$(P) \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{sous les contraintes} \\ g_j(x) = 0 & j = 1, \dots, m_e \\ g_j(x) \geq 0 & j = m_e, \dots, m \\ x_l \leq x \leq x_u \end{cases}$$

Son appel se fait de la manière suivante:

CALL NLPQL (FCN, GRAD, M, ME, N, XGUESS, IBTYPE, XLB, XUB, IPRINT, MAXITN, X, FVALUE )

Où

- *FCN* : sous programme fourni par l'utilisateur pour évaluer les fonctions à un point donné.
- *GRAD* : sous programme fourni par l'utilisateur pour évaluer les gradients à un point donné.
- *M* : nombre total de contraintes. (Input)
- *ME* : Nombre de contraintes d'égalité. (Input)
- *N* : Nombre de variables. (Input)
- *XGUESS* : Vecteur de dimension N qui contient une estimation initiale de la solution calculée (input)
- *IBTYPE* : Scalaire indiquant les types des limites sur les variables (input).  
 0 l'utilisateur fournira tous les limites .  
 1 toutes les variables sont non négatives .  
 2 toutes les variables sont non positives .  
 3 l'utilisateur ne fournit que les limites sur la première variable, toutes les autres variables auront les mêmes limites.
- *XLB* : Vecteur de dimension N qui contient les limites inférieures sur les variables .

S'il n'y a aucune limite inférieure sur les variable, alors la valeur de **XLB** correspondante est prise égale à  $-1.0E6$

- **XUB** : Vecteur de dimension N qui contient les limites supérieures sur les variables.

S'il n'y a aucune limite supérieure sur les variables, alors la valeur **XUB** correspondante est prise égale à  $1.0E6$

- **IPRINT** : Paramètre qui indique le niveau du détail des résultats désiré. (Input) .

0: aucun output à imprimer

1: seule la dernière analyse de la convergence est donnée

2: une ligne de résultats intermédiaires est imprimée pour chaque itération

3: information détaillée est imprimée pour chaque itération

- **MAXITN** : Nombre maximum d'itérations permis. (Input)

- **X** : Vecteur de dimension N qui contient la solution calculée. (Output)

- **FVALUE** : Scalaire contenant la valeur de la fonction objectif à la solution calculée (Output).

### **III.5.2. Sous-programme FCN :**

Ce sous-programme calcule la valeur de la fonction objectif à l'itération courante ainsi que les valeurs des contraintes d'optimisation  $G_i$ .

La fonction objectif est évaluée après la réception des dimensions de tous les éléments à l'itération courante stockés dans un seul vecteur appelé **VSEC1** de dimension  $2*N$ . Ce vecteur **VSEC1** est envoyé par **NLPQL** et sauvegardé dans le fichier **SECT**.

La fonction objectif est calculée selon la formule suivante :

$$f = \gamma \sum_{i=1}^{NELT} h_i b_i l_i$$

Le nombre total des contraintes est égal à  $3*nelt$  (3 contraintes pour chaque élément):

- $nelt$  contraintes  $\sigma_{lim} - \sigma_i \geq 0$
- $nelt$  contraintes  $(b/h) - (b/h)_{min} \geq 0$
- $nelt$  contraintes  $(b/h)_{max} - (b/h) \leq 0$

Les  $nelt$  premières contraintes ne sont pas calculées directement car elles dépendent des contraintes dans les barres à l'itération K. Or la détermination de ces dernières nécessite l'exécution d'une étude linéaire de toute la structure avec les sections calculées pour cette itération. Pour cet effet le sous-programme **FCN** doit appeler le sous-programme **EXLINO** à chaque itération.

Les  $2 \cdot nelt$  autres contraintes sont calculées directement selon l'algorithme suivant :

Pour  $i$  allant de  $nelt+1$  à  $2 \cdot nelt$  faire

$$G(i) = RSLIM - VSEC1(2 \cdot (i - nelt)) \cdot VSEC1[(2(i - nelt) - 1)]^{-1}$$

Fin pour

Pour  $i$  allant de  $2 \cdot nelt+1$  à  $3 \cdot nelt$

$$G(i) = VSEC1(2(i - 2 \cdot nelt)) [VSEC1(2(i - 2 \cdot nelt) - 1)]^{-1} - RILIM$$

Fin pour

### III.5.3. Sous-programme *EXLINO* :

Ce sous-programme est semblable au sous-programme *EXLINM* de la partie *PORT*. Ils ne se diffèrent que dans la lecture des sections des éléments: le premier lit les sections initiales, alors que le deuxième lit les sections perturbées (utilisées dans la dérivation numérique) et les sections des différents éléments à l'itération courante (pour le calcul des contraintes d'optimisation).

### III.5.4. Sous-programme *DERIVAT* :

Ce sous-programme est chargé d'effectuer les opérations suivantes:

- calcul du gradient de la fonction objectif ; son expression étant donnée, l'expression du gradient est déterminée analytiquement.
- calcul du jacobien des contraintes d'optimisation.

La dimension du jacobien est de  $(3 \cdot nelt \times 2 \cdot nelt)$ .

Le nombre de lignes du jacobien représente le nombre de contraintes.

Le nombre de colonnes du jacobien représente le nombre de variables d'optimisation.

Les dérivées des  $nelt$  premières contraintes (contraintes liées à la résistance limite des éléments) sont calculées en utilisant les méthodes de dérivation numérique puisque leurs expression analytique est très compliquée alors que les dérivés des  $2 \cdot nelt$  dernières contraintes (contraintes liées à la forme géométrique) sont calculées analytiquement.

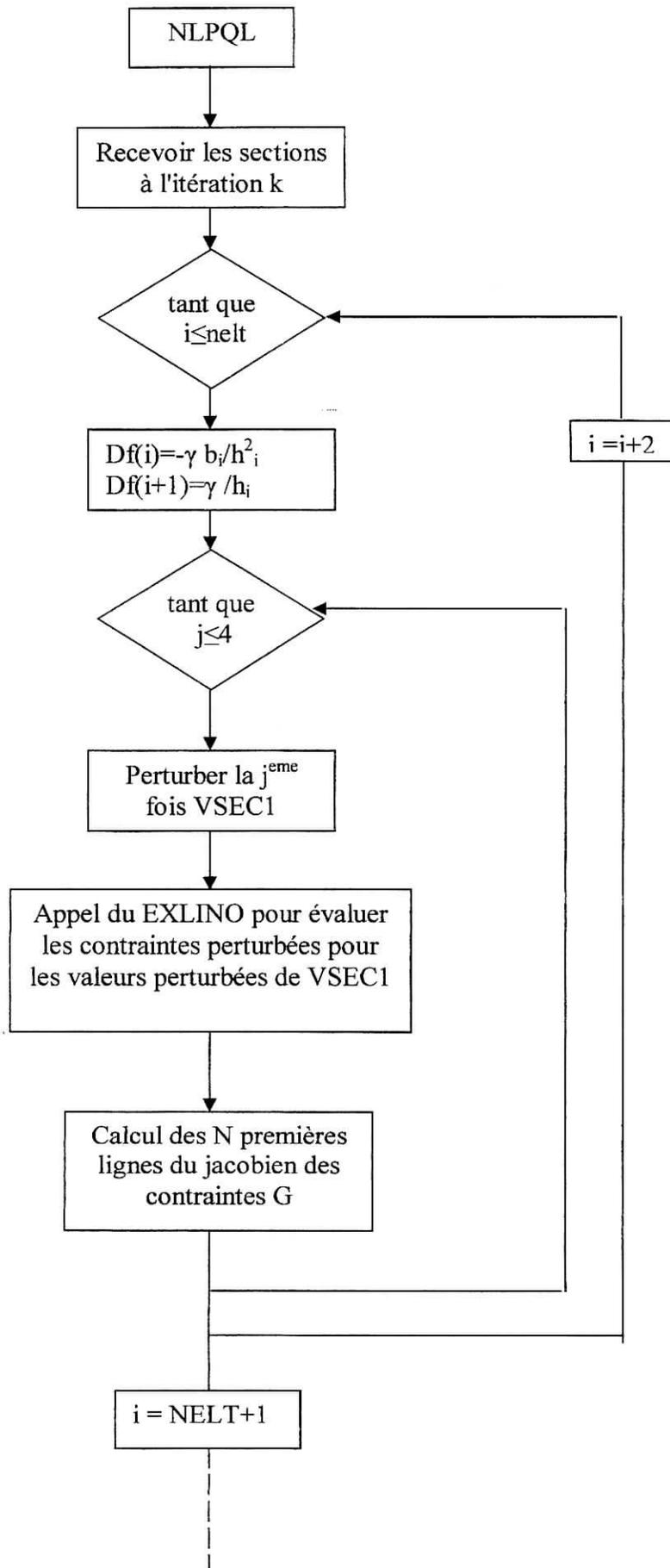
La formule de dérivation numérique utilisée est [14]:

$$f'(x_i) = \frac{16f(x_i + \Delta) - 16f(x_i - \Delta) + 2f(x_i - 2\Delta) - 2f(x_i + 2\Delta)}{24\Delta} + \frac{1}{30} \Delta^4 f^{(5)}(x_i)$$

où  $\Delta$  est le pas de dérivation.

Le terme  $\frac{1}{30} \Delta^4 f^{(5)}(x_i)$  représente l'erreur de la dérivation.

L'organigramme général du sous-programme *DERIVAT* est le suivant :



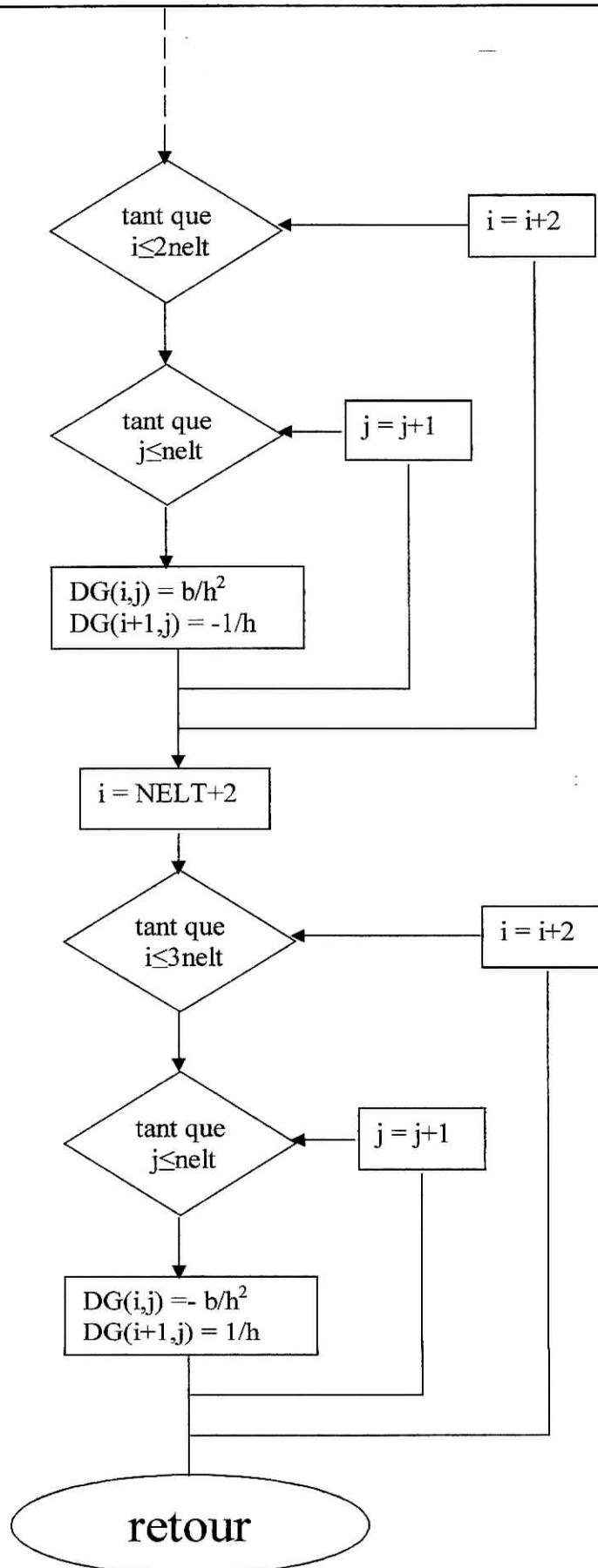


Figure 3.9 : Organigramme du sous-programme *DERIVAT*

### III.5.5. Sous-programme *MODVFG*:

Puisque le poids propre est pris en compte dans le vecteur de sollicitations globales *VFG* nous rencontrons les problèmes suivants :

- si les sections initiales sont choisies petites les sollicitations appliquées aux différents éléments seront sous estimées. Ceci nous donne une structure sous dimensionnée puisque après optimisation on aura un poids plus grand que celui de la première itération alors qu'on garde toujours les sollicitations dues au poids propre initial.
- Pour les mêmes raisons si les sections initiales sont choisies grandes le processus d'optimisation nous donnera une structure surdimensionnée.

Pour éviter ces problèmes nous avons réalisé le sous-programme *MODVFG* qui a pour objet l'actualisation du vecteur forces globales à chaque itération en remplaçant les valeurs des sollicitations équivalentes dues aux poids propre des éléments de l'itération précédente par ceux de l'itération courante.

L'algorithme de ce sous-programme est le suivant :

```

A l'itération k
pour l'élément i
- recevoir les nouvelles dimensions de cet élément à l'itération k
- lire les anciennes dimensions de l'itération k-1 de cet élément à partir
du fichier SECT
- lire la longueur, sin et cos de cet élément à partir le fichier LONG
- lire les connectivités de cet élément à partir le fichier KNE
- lire v1(n) les efforts équivalents dues au poids propre calculés avec les
anciennes dimensions à partir le fichier SOLEP
  - calculer les efforts v2(n) dues au poids propre avec les nouvelles
sections.
  - Se positionner dans les composantes (j, j+1, j+2) correspondantes à
l'élément i dans le vecteur des sollicitations globales VFG.
  - Effectuer les opérations suivantes:
VFG(j)=vfg(j)-v1(1)+v2(1)
VFG(j+1)=vfg(j+1)-v1(2)+v2(2)
VFG(j+2)=vfg(j+2)-v1(3)+v2(3)
Fin pour
Ecrire les nouvelles dimensions dans le fichier SECT
Ecrire les nouveaux efforts équivalents dans le fichier SOLEP
Retour

```

# CHAPITRE 4

## APPLICATIONS

## IV.1. Exemples d'application:

### IV.1.1. Premier exemple:

Cet exemple d'application est un portique constitué de deux travées de 4m de longueur chacune et d'un seul étage de 3m de hauteur.

Les charges appliquées sur ce portique sont :

-Poids propre, pris en considération dans tous les éléments, avec  $\gamma=7.8$  t/m<sup>3</sup>

-une charge répartie sur les deux poutres  $g=2$  t/m

-deux forces  $P_1, P_2$ , concentrées sur les deux poutres avec  $P_1=P_2=3$  t.

Le module de YOUNG  $E = 2000$  t/cm<sup>2</sup>

La contrainte limite d'élasticité  $\sigma_{lim}=2.4$ t/cm<sup>2</sup>

Les sections initiales sont prises égales à :40\*30 cm<sup>2</sup>

La limite inférieure de b et h  $XLB=4$  cm

La limite supérieure de b et h  $XUB=25$  cm

La limite inférieure du rapport b/h  $Rilim=0.2$

La limite supérieure du rapport b/h  $Rslim=1$

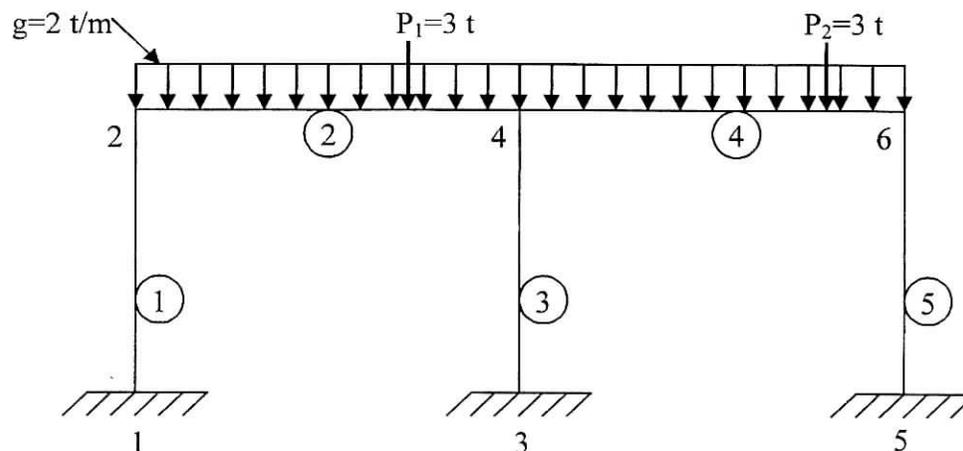


Figure 4.1: schéma du premier portique étudié

L'exécution du programme à donner les résultats suivants :

- Affichage donné lors de l'exécution du programme:

\*\*\*\*\*

OPTIMISATION DES STRUCTURES  
OPTIPORT

\*\*\*\*\*

donner le nom du fichier des donnees

pfedat

donner le nom du fichier des resultats

pferes

?????????voulez vous optimiser votre structure:o/n???????????

o

ANALYSE EFFECTUEE LE 26- 6- 2003

DEBUT D EXECUTION A 14h: 32min: 19sec

.. .execution en cours veuillez patientez...

\*\*\*\*\*

OPTIMISATION EFFECTUEE AVEC SUCCES

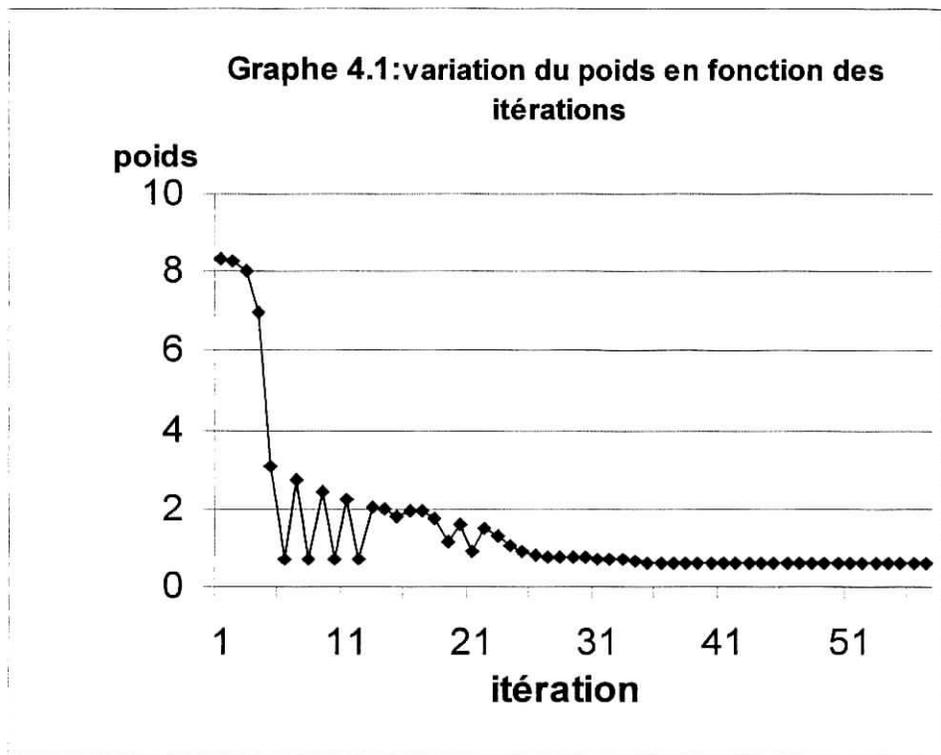
poids optimal = 0.59

\*\*\*\*\*

fin d execution a 14h: 32min: 41sec

*Variation du poids de la structure au cours de l'optimisation:*

La courbe (4.1) représente la variation du poids total du portique au cours de l'optimisation:

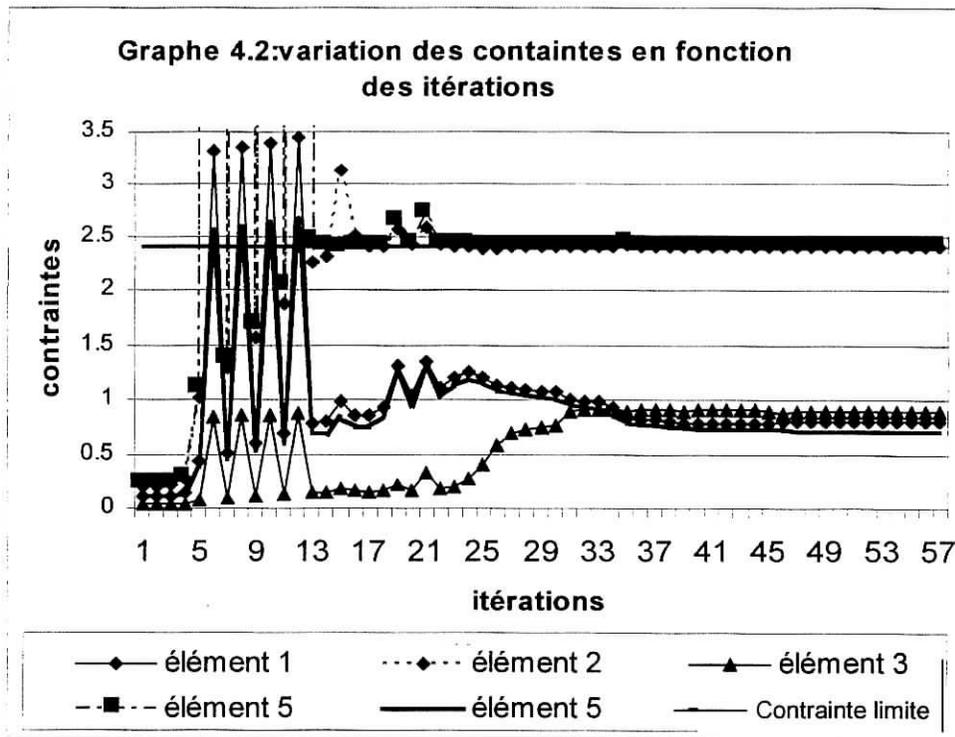


Interprétation des résultats :

La courbe -comme le montre la figure- à une allure générale de descente (le poids a diminué de 8.3t pour atteindre 0.6 t à l'itération finale), néanmoins elle présente un comportement oscillatoire entre les itérations 6 et 22. Ceci montre le caractère dual de la méthode **SQP**. En effet l'algorithme sort du domaine de solutions réalisables plusieurs fois (voir la courbe de variation des contraintes). Les contraintes dans certains éléments dépassent la contrainte limite ce qui oblige l'algorithme à augmenter les sections des éléments où les contraintes sont violées pour satisfaire les contraintes d'optimisation  $\sigma_i \leq \sigma_{lim}$ .

*Variation des contraintes au cours de l'optimisation:*

Le graphe (4.2) représente la variation des contraintes dans les éléments constituant le portique en fonction des itérations :



Nous remarquons que les contraintes dans les éléments 2 et 4 sont saturées (leurs valeurs ont atteint la contrainte limite 2400 kg/cm<sup>2</sup>) alors que celles dans les autres éléments restent loin de la contrainte limite malgré la convergence vers le poids optimal. En fait ce n'est pas une erreur puisque le poids obtenu est réellement le poids optimal et si l'on diminue les sections des éléments 1, 3, ou 5 pour augmenter les valeurs de leurs contraintes nous constatons que les contraintes saturées en 3 et 4 dépassent la limite donc ces éléments sortent du domaine réalisable.

élément	Hauteur h (cm)	Largeur b (cm)	Contrainte t/cm <sup>2</sup>
1	4.36	4.00	0.7947
2	18.86	4.00	2.4000
3	4.00	4.00	0.8920
4	18.86	4.00	2.4000
5	4.35	4.00	0.7178

**Tableau 4.1:**valeurs des sections et des contraintes de la dernière itération

Si nous diminuons les dimensions des éléments 1,3 et 5 (éléments dont les contraintes ne sont pas saturées) nous obtenons le tableau suivant:

élément	Hauteur h (cm)	Largeur b (cm)	Contrainte t/cm <sup>2</sup>
1	4.16	4.00	0.7845
2	18.86	4.00	<u>2.4023*</u>
3	4.00	4.00	0.8919
4	18.86	4.00	<u>2.4028*</u>
5	4.15	4.00	0.7035

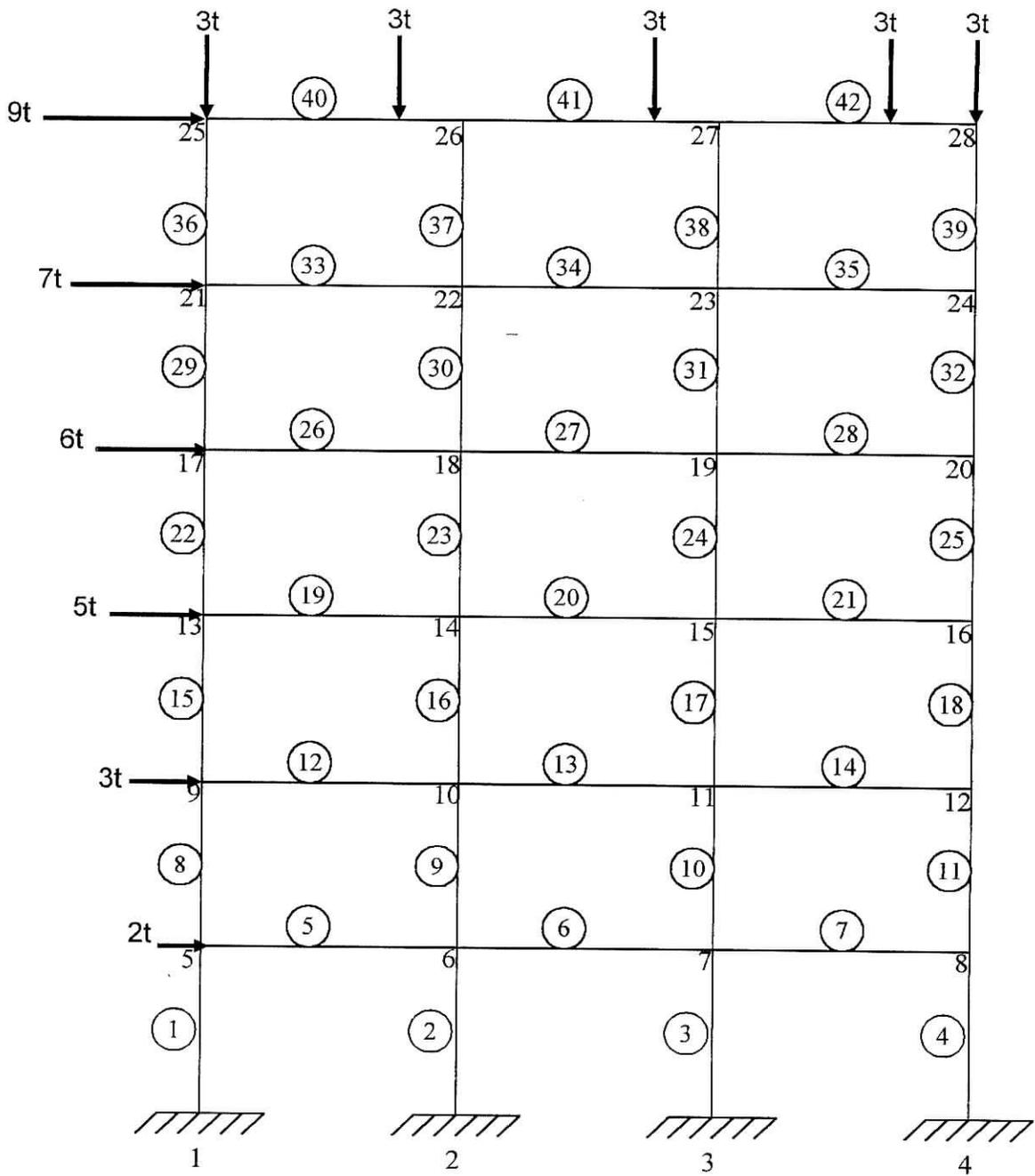
**Tableau 4.2:**valeurs des contraintes après modification des sections

#### IV.1.2. Deuxième exemple:

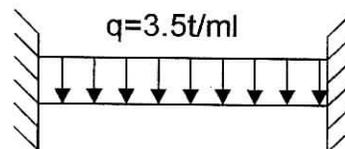
Cet exemple d'application est un portique dont les caractéristiques sont les suivantes :( voir figure. 4.2)

- Nombre de niveaux: 6
- Nombre de travées: 4 *rectiligne*
- Hauteur des étages: 3m
- Longueur des travées: 4m.
- Sollicitations appliquées:
  - poids propre, pris en considération dans tous les éléments, avec  $\gamma = 7.8 \text{ t/m}^3$
  - une charges répartie sur toutes les poutres,  $g=3.5 \text{ t / ml}$
  - des forces concentrées horizontales représentent la répartition de la force sismique sur les différents étages.
  - deux forces concentrées sur les deux nœuds extrêmes du dernier étage (représente les surcharges de l'acrotère).
  - quatre forces concentrées en travées de 3m sur toutes les travées du dernier étage.

- Le module de YOUNG ,  $E=2000 \text{ t/ cm}^2$
- La contrainte limite d'élasticité,  $\sigma_e =2.4 \text{ t/ cm}^2$ .



Charge répartie sur toutes les poutres de la structure



**Figure 4.2:** schéma de la structure étudiée

### IV.1.3. Première étude:

Les sections initiales sont sur-estimées et prises égales à :  $40 \times 30 \text{ cm}^2$

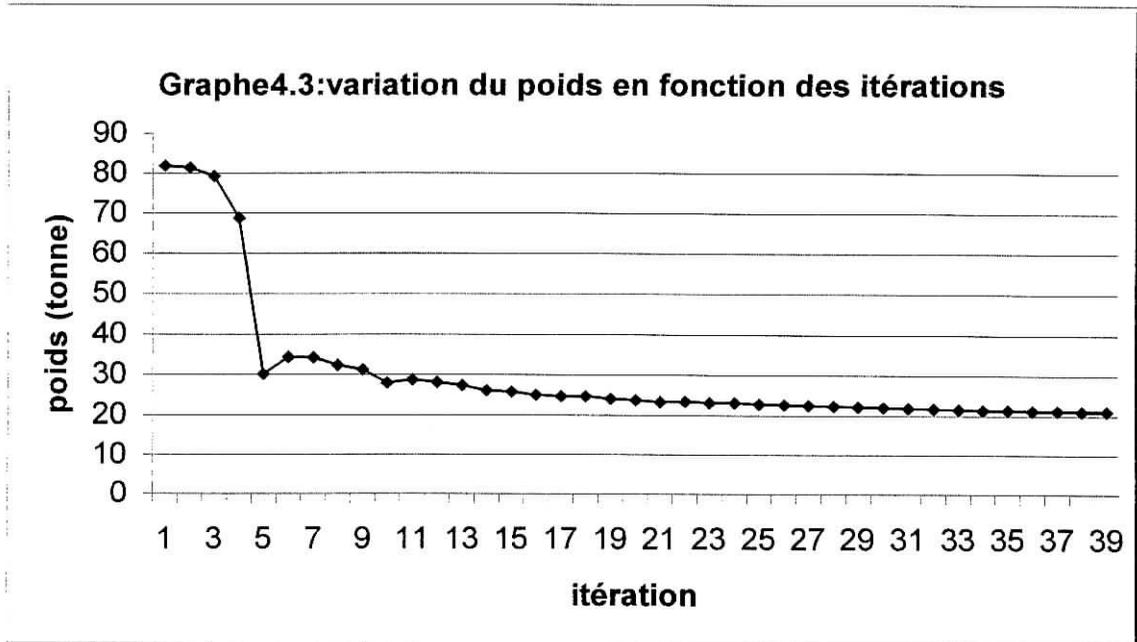
La limite inférieure des valeurs de  $b$  et  $h$   $X_{LB} = 5$

La limite supérieure des valeurs de  $b$  et  $h$   $X_{UB} = 20$

La limite inférieure du rapport  $b/h$   $R_{LIM} = 0.25$

La limite supérieure du rapport  $b/h$   $R_{SLIM} = 1$

La courbe 4.3 montre la variation du poids de la structure au cours de l'optimisation :



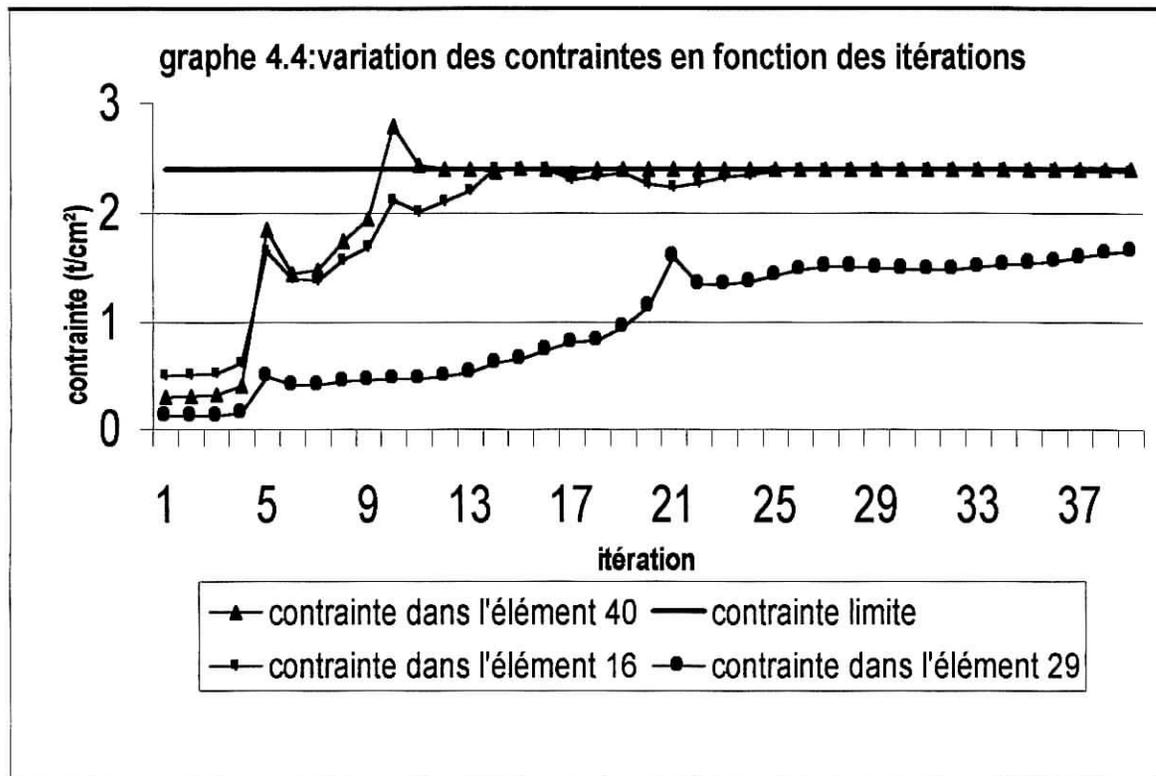
#### Interprétation des résultats:

- La courbe -comme le montre la figure 4.3- a une allure générale de descente.
- Nous remarquons que le poids total de la structure a diminué de plus de 63% dès les cinq premières itérations (donc 85% de la différence entre le poids initial et le poids optimal est atteinte), en effet, cette chute est justifiée par le fait que le poids initial est pris très loin du poids optimal (surdimensionnement excessif).
- L'augmentation du poids entre les itérations 5,6 (de 30.2 t à 34.2 t) , 9,10(de 27.9t à 28.6t) ainsi qu'entre 20,21(de 23.2t à 23.4t) est dûe au fait que les contraintes dans certains éléments de la structure dépasse la contrainte limite d'élasticité.(voir graphe 4.4) ce qui a obligé l'algorithme d'augmenter les dimensions de ces éléments pour rattraper l'écart.

- Après la 33<sup>ème</sup> itération la pente de la courbe tend vers zéro, la courbe admet donc une asymptote ( $y = \text{poids optimal}$ )

La figure (4.4) représente la variation de la contrainte dans trois éléments de la structure :

- Un élément trop chargé (poutre 16)
- Un élément faiblement chargé (poteau 29)
- Un élément moyennement chargé (poutre 40)



Interprétation des résultats:

- La courbe n'a pas une allure constante, en effet la contrainte dépasse plusieurs fois la contrainte limite c'est-à-dire que les solutions sortent du domaine réalisable, ceci justifie le caractère dual de la méthode SQP ( le processus de convergence n'est pas restreint à rester dans le domaine des solutions réalisables )
- La contrainte dans l'élément 29 n'est pas trop serrée comme les deux autres puisque cet élément n'est pas trop chargé et si on diminue les dimensions de ces éléments la contrainte dans les autres éléments chargés dépasse la contrainte limite (voir premier exemple).

#### IV.1.4. Deuxième étude

Les sections initiales sont sous estimées et prises égales à :  $4 \times 3 \text{ cm}^2$

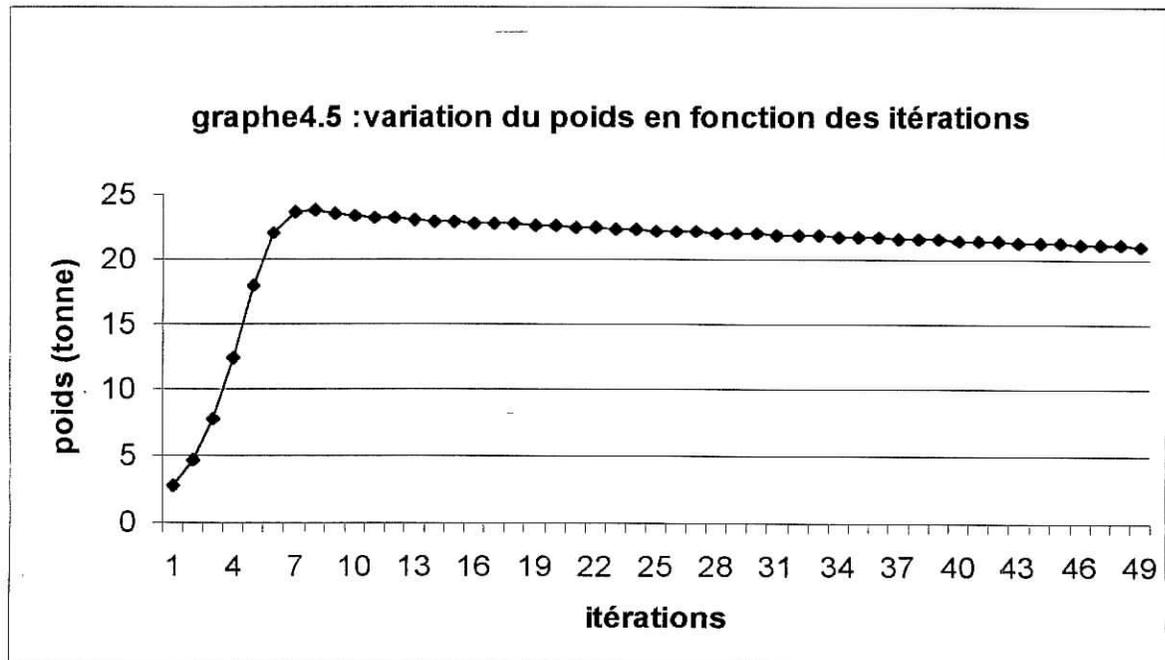
La limite inférieure des valeurs de  $b$  et  $h$   $XLB = 5 \text{ cm}$

La limite supérieure des valeurs de  $b$  et  $h$  XUB= 20 cm

La limite inférieure du rapport  $b/h$  RILIM= 0.25

La limite supérieure du rapport  $b/h$  RSLIM= 1

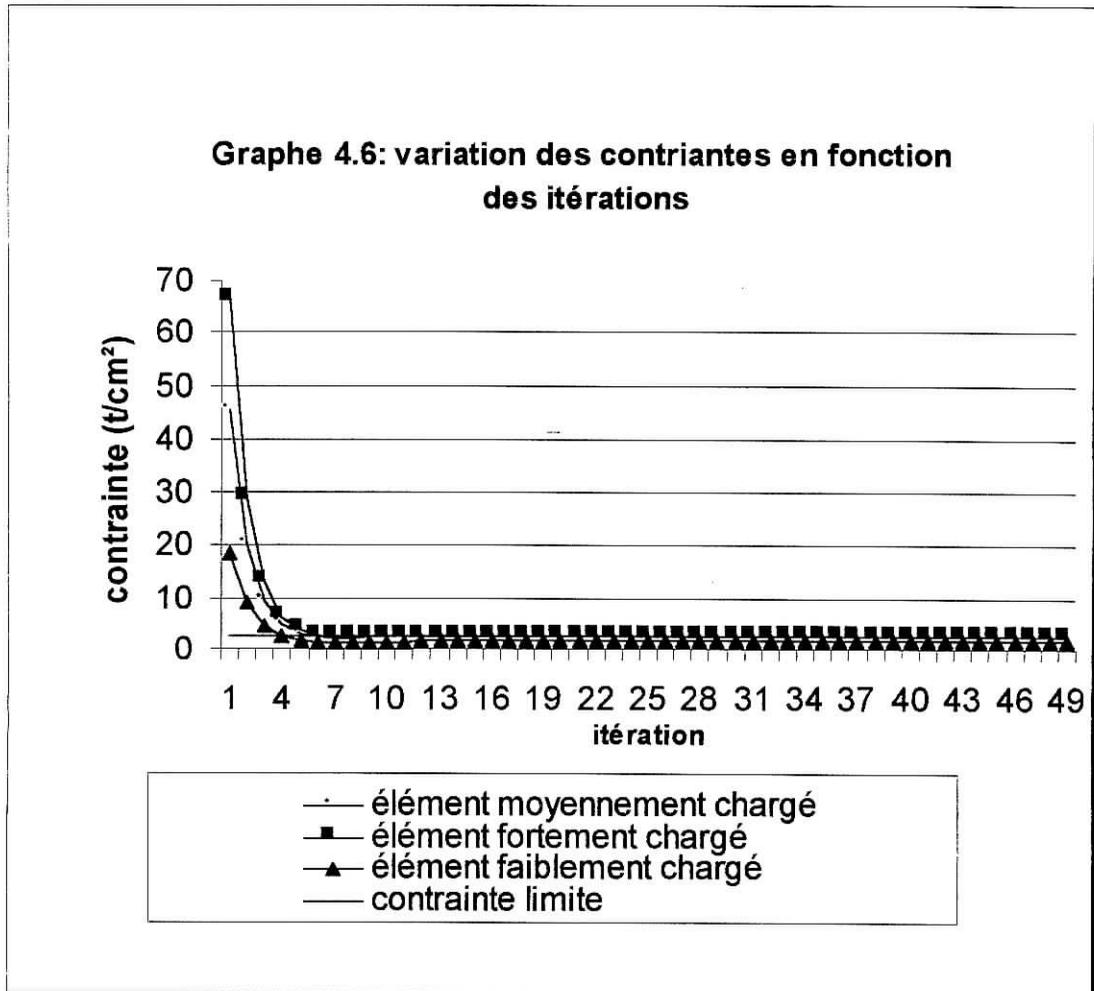
LA courbe 4.5 montre la variation du poids de la structure au cours de l'optimisation :



#### *Interprétation des résultats:*

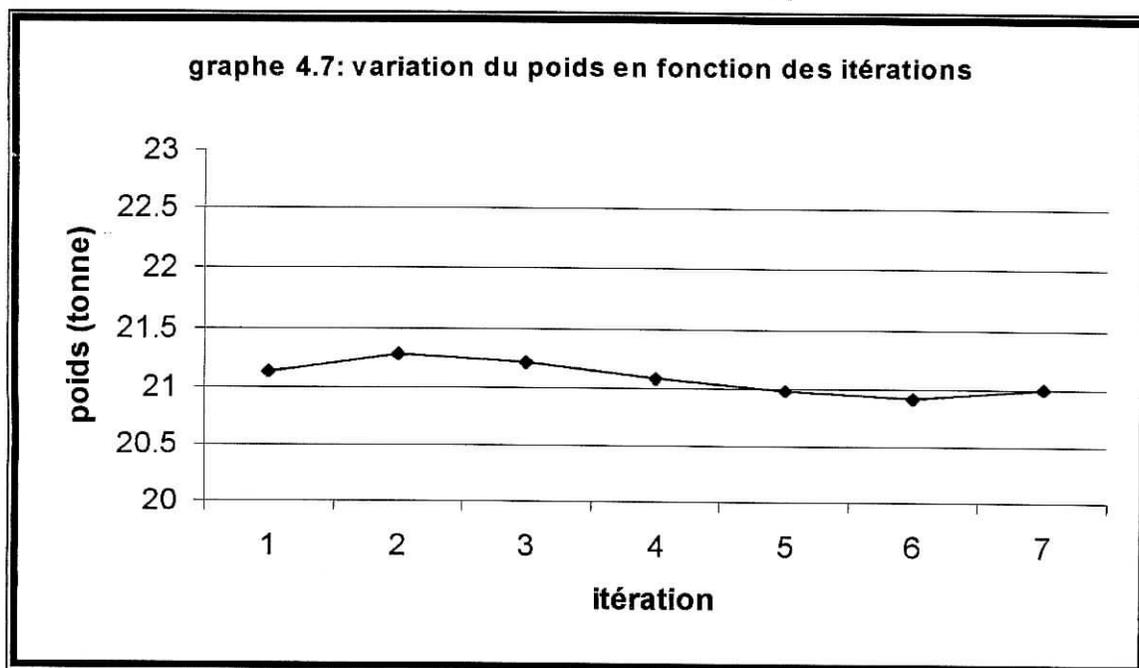
- L'ascendance de cette courbe est prévue puisque la structure est sous dimensionnée.
- La convergence vers le poids optimal est très rapide, en effet le poids de la structure a augmenté de 20 tonnes après 6 itérations (1 tonne seulement loin du poids optimal).
- Nous remarquons que le poids commence à diminuer après la huitième itération en convergeant lentement vers la solution optimale. Ceci est justifié par le fait que l'algorithme commence par vérifier les contraintes dès les premières itérations avant de les saturer c'est à dire trouver les sections pour lesquelles les contraintes dans les éléments soient le plus proche possible de la contrainte limite.

Le graphe 4.6 représente la variation des contraintes dans les mêmes éléments de la première étude:



#### IV.1.5. Troisième étude :

Les dimensions initiales ne sont pas identiques et sont proches de la solution optimale. La courbe 4.7 montre la variation du poids de la structure dans ce cas.



Nous remarquons que le nombre d'itération a été réduit d'une manière considérable. Ceci est prévu puisque les sections initiales ont été choisies au voisinage de la solution optimale.

Les détails des résultats (efforts internes, déplacements, contraintes, sections optimales) obtenues par ces trois applications sont données dans l'annexe.

## **IV.2. Conclusion :**

Le premier exemple est intéressant car il nous montre que l'optimalité du poids n'implique pas la saturation des contraintes dans tous les éléments.

Les résultats obtenus pour cette structure sont identiques dans les trois cas, en effet les poids optimaux sont tous égale à 21.0 t . Néanmoins trois remarques s'imposent:

- Le nombre d'itérations varie d'un cas à un autre car ce paramètre dépend du vecteur de départ de l'algorithme. Le nombre d'itération des cas 1 et 2 est élevé en raison de l'éloignement du vecteur de départ de la solution optimale. Le cas 3 montre qu'un vecteur de départ proche du point d'optimisation réduit considérablement le nombre d'itération donc le temps d'exécution
- Les contraintes d'optimisation doivent être posées d'une manière qu'elles ne soient pas contradictoires. Dans le cas ou ceci n'est pas vérifié le processus s'arrête prématurément mais malgré cet handicap on peut aboutir à de bons résultats lorsque on serait proche de l'asymptote de la courbe.
- La convergence de l'algorithme ne dépend pas du point initial, en effet , que le point initial soit à l'intérieur du domaine des solutions réalisables ou non le programme converge .
- L'observation des valeurs de b et h des éléments des structures optimales d'un cas à un autre montre que ces dernières changent , ceci est dû au fait que le problème étudié ne présente pas de vecteur de solution unique

# CONCLUSION GENERALE

## *Conclusion générale*

Après avoir abordé la méthode des éléments finis pour les portiques plans ainsi que la méthode d'optimisation séquentielle quadratique et mis en œuvre la procédure d'optimisation des structures en portiques plans à travers le développement d'un programme informatique, les résultats fiables des applications traitées ont montré la validité du programme *OPTIPORT* et la puissance de l'algorithme *SQP*. Ce travail traite les sections rectangulaires en acier, néanmoins il pourra être développé par la suite pour traiter d'autres types de structure.

En fin ce programme présente un intérêt économique certain en minimisant le poids des structures en portiques plans et peut constituer un outil d'aide et de comparaison pour la recherche dans le domaine d'optimisation des structures en utilisant d'autres approches.

# BIBLIOGRAPHIE

---

**BIBLIOGRAPHIE**

- [1] YANG T.Y (1986), *Finite Element Structural Analysis*  
Prentice-Hall, Inc., Englewood Cliffs, New Jersey ,543p
- [2] DHATT G. TOUZOT G.(1984),*Une présentation de la méthode des éléments finis*  
Maloine S.A. Paris, 543p.
- [3] BATOZ J.L , DHATT G.(1990), *Modélisation des structures par éléments finis*  
(tome 1 et 2) , HERMES, Paris, 455 p.
- [4] AUBOIN (1983), *Calcul des structures et informatique*  
Editions EYROLLES, Paris 113p
- [6] FREY F. JIROUSEK J. (2001), *Traité de Génie Civil ,Vol 6 ,Analyse des structures et milieux continus, Méthode des éléments finis*  
Presses Polytechniques et Universitaires Romandes, Lausanne, 298p
- [7] BELHACEN B. (1998) *Recherches Opérationnelles*  
(tome 1 et 2) Institut National d'Informatique
- [8] TEMPLEMAN A.B (1982) *Civil Engineering Systems*  
The Macmillan Press LTD. London, 370p
- [9] MINOUX M. (1983) *Programmation mathématique*  
(tome 1) Dunod, Paris, 294p
- [10] SCHITTKOWSKI K. (2001), *NLPQLP: A New Fortran Implementation of a Sequential Quadratic Programming Algorithm - User's Guide - Report*, Department of Mathematics University of Bayreuth.  
Web site <http://www.klaus-schittkowski.de>
- [11] SCHITTKOWSKI K. (2002) *Non linear programming software.*  
Web site <http://www.klaus-schittkowski.de>
- [12] SCHITTKOWSKI K., ZILLOBER CH. (2000), *Non linear programming Report*, Departement of mathematics, University of Bayreuth.  
Web site <http://www.klaus-schittkowski.de>
-

- [13] CHARLES AUDET (2001), *Eléments de programmation non linéaire*  
*Département de mathématique, Ecole*  
*Polytechnique de Montréal.*
- [14] Aoues younes, Hadj henni annis (2002) *Optimisation des structures*  
*spatiales en treillis*, mémoire de fin d'études  
*Ecole National Polytechnique ENP.*
- [15] Bernardo Horowitz, Silvana M.B. Afonso ,(2000) *Quadratic programming*  
*solver for structural optimization using SQP algorithm.*  
Advances in Engineering Software  
Web site [http:// www.elsevier.com/locate/advengsoft](http://www.elsevier.com/locate/advengsoft)
- [16] J. Isenberg, V. Pereyra., D. Lawver (2002) *Optimal design of steel frame*  
*structures. Applied Numerical Mathematics*  
Web site [http:// www.elsevier.com/locate/advengsoft](http://www.elsevier.com/locate/advengsoft)

# ANNEXE

**Premier exemple**

Variation des sections au cours de l'optimisation

itér	élém	h	b	itér	élém	h	b	itér	élém	h	b
1	1	25	25	20	1	12.24	11.42	39	1	4.58	4
	2	25	25		2	11.08	8.91		2	18.84	4
	3	25	25		3	11.95	11.32		3	4.06	4
	4	25	25		4	11.31	8.94		4	18.86	4
	5	25	25		5	12.19	11.41		5	4.37	4
2	1	24.94	24.94	21	1	8.91	6.32	40	1	4.56	4
	2	24.92	24.92		2	13.66	6.58		2	18.86	4
	3	24.94	24.94		3	7.94	5.99		3	4.03	4
	4	24.92	24.92		4	13.69	6.28		4	18.86	4
	5	24.94	24.94		5	8.75	6.27		5	4.36	4
3	1	24.65	24.65	22	1	11.7	10.6	41	1	4.55	4
	2	24.53	24.53		2	11.5	8.53		2	18.86	4
	3	24.65	24.65		3	11.3	10.46		3	4.02	4
	4	24.53	24.53		4	11.7	8.51		4	18.86	4
	5	24.65	24.65		5	11.63	10.58		5	4.35	4
4	1	23.21	23.21	23	1	10.78	9.28	42	1	4.55	4
	2	22.62	22.62		2	12.23	8		2	18.86	4
	3	23.21	23.21		3	10.18	9.08		3	4.01	4
	4	22.62	22.62		4	12.41	7.89		4	18.86	4
	5	23.21	23.21		5	10.68	9.25		5	4.34	4
5	1	16.51	16.51	24	1	9.38	7.32	43	1	4.54	4
	2	13.7	13.7		2	13.35	7.27		2	18.86	4
	3	16.51	16.51		3	8.47	7.03		3	4.01	4
	4	13.7	13.7		4	13.55	7.04		4	18.86	4
	5	16.51	16.51		5	9.24	7.28		5	4.34	4
6	1	9.17	9.17	25	1	8.28	5.88	44	1	4.53	4
	2	4	4		2	14.22	6.77		2	18.86	4
	3	9.17	9.17		3	7.12	5.52		3	4.01	4
	4	4	4		4	14.51	6.48		4	18.86	4
	5	9.17	9.17		5	8.11	5.83		5	4.34	4
7	1	15.69	15.69	26	1	7.31	4.75	45	1	4.49	4
	2	12.62	12.62		2	14.85	6.36		2	18.86	4
	3	15.69	15.69		3	5.93	4.34		3	4.02	4
	4	12.62	12.62		4	15.26	6.01		4	18.86	4
	5	15.69	15.69		5	7.12	4.69		5	4.36	4
8	1	9.13	9.13	27	1	6.94	4.33	46	1	4.4	4
	2	4	4		2	15.06	6.2		2	18.86	4
	3	9.13	9.13		3	5.47	4		3	4.05	4
	4	4	4		4	15.52	5.84		4	18.87	4
	5	9.13	9.13		5	6.74	4.28		5	4.38	4
9	1	15.03	15.03	28	1	6.65	4.04	47	1	4.38	4
	2	11.75	11.75		2	15.22	6.09		2	18.86	4
	3	15.03	15.03		3	5.11	4		3	4.03	4
	4	11.75	11.75		4	15.73	5.71		4	18.87	4
	5	15.03	15.03		5	6.44	4		5	4.37	4
10	1	9.09	9.09	29	1	6.58	4	48	1	4.37	4
	2	4	4		2	15.26	6.07		2	18.86	4
	3	9.09	9.09		3	5.02	4		3	4.01	4
	4	4	4		4	15.77	5.68		4	18.86	4
	5	9.09	9.09		5	6.37	4		5	4.36	4
11	1	14.44	14.44	30	1	6.46	4	49	1	4.37	4
	2	10.98	10.98		2	15.31	6.03		2	18.86	4
	3	14.44	14.44		3	4.88	4		3	4.01	4

	4	10.98	10.98		4	15.85	5.62		4	18.86	4
	5	14.44	14.44		5	6.26	4		5	4.36	4
12	1	9.05	9.05	31	1	5.89	4	50	1	4.36	4
	2	4	4		2	15.6	5.82		2	18.86	4
	3	9.05	9.05		3	4.15	4		3	4	4
	4	4	4		4	16.25	5.36		4	18.86	4
	5	9.05	9.05		5	5.68	4		5	4.35	4
13	1	13.9	13.9	32	1	5.75	4	51	1	4.36	4
	2	10.28	10.28		2	15.69	5.76		2	18.86	4
	3	13.9	13.9		3	4	4		3	4	4
	4	10.28	10.28		4	16.37	5.29		4	18.86	4
	5	13.9	13.9		5	5.54	4		5	4.35	4
14	1	13.84	13.84	33	1	5.68	4	52	1	4.36	4
	2	10.2	10.21		2	15.82	5.67		2	18.86	4
	3	13.83	13.84		3	4	4		3	4	4
	4	10.4	10.31		4	16.51	5.2		4	18.86	4
	5	13.84	13.84		5	5.47	4		5	4.35	4
15	1	13.05	13.03	34	1	5.37	4	53	1	4.36	4
	2	9.1	9.18		2	16.46	5.22		2	18.86	4
	3	12.99	13.01		3	4	4		3	4	4
	4	10.55	9.88		4	17.21	4.77		4	18.86	4
	5	13.05	13.03		5	5.15	4		5	4.35	4
16	1	13.55	13.48	35	1	4.87	4	54	1	4.36	4
	2	9.95	9.83		2	17.63	4.51		2	18.86	4
	3	13.52	13.47		3	4.07	4		3	4	4
	4	10.5	10.1		4	18.48	4.1		4	18.86	4
	5	13.55	13.48		5	4.65	4		5	4.35	4
17	1	13.54	13.4	36	1	4.76	4	55	1	4.36	4
	2	10.19	9.88		2	17.99	4.39		2	18.86	4
	3	13.49	13.39		3	4.07	4		3	4	4
	4	10.55	10.05		4	18.84	4		4	18.86	4
	5	13.54	13.4		5	4.54	4		5	4.35	4
18	1	12.93	12.47	37	1	4.73	4	56	1	4.36	4
	2	10.59	9.42		2	18.05	4.36		2	18.86	4
	3	12.76	12.42		3	4.04	4		3	4	4
	4	10.88	9.51		4	18.85	4		4	18.86	4
	5	12.9	12.47		5	4.51	4		5	4.35	4
19	1	10.16	8.24	38	1	4.68	4	57	1	4.36	4
	2	12.55	7.36		2	18.27	4.26		2	18.86	4
	3	9.47	8.01		3	4.03	4		3	4	4
	4	12.62	7.18		4	18.85	4		4	18.86	4
	5	10.04	8.2		5	4.46	4		5	4.35	4

Tableau 4.1: variation des dimensions en fonction des itérations

**Deuxième exemple**

**Tous les résultats qui suivent sont de la deuxième étude (sections initiales (4\*3 cm<sup>2</sup>))**

**Fichier de données :**

IMAG

COOR

28	3	2	1.	1.
1		0.	0.	
2		400.	0.	
3		800.	0.	
4		1200.	0.	
5		0.	300.	

6	400.	300.
7	800.	300.
8	1200.	300.
9	0.	600.
10	400.	600.
11	800.	600.
12	1200.	600.
13	0.	900.
14	400.	900.
15	800.	900.
16	1200.	900.
17	0.	1200.
18	400.	1200.
19	800.	1200.
20	1200.	1200.
21	0.	1500.
22	400.	1500.
23	800.	1500.
24	1200.	1500.
25	0.	1800.
26	400.	1800.
27	800.	1800.
28	1200.	1800.

-1

COND

111

1	2	3	4
0			

PREL

42	3		
1	2000.	4.	3.
2	2000.	4.	3.
3	2000.	4.	3.
4	2000.	4.	3.
5	2000.	4.	3.
6	2000.	4.	3.
7	2000.	4.	3.
8	2000.	4.	3.
9	2000.	4.	3.
10	2000.	4.	3.
11	2000.	4.	3.
12	2000.	4.	3.
13	2000.	4.	3.
14	2000.	4.	3.
15	2000.	4.	3.
16	2000.	4.	3.
17	2000.	4.	3.
18	2000.	4.	3.
19	2000.	4.	3.
20	2000.	4.	3.
21	2000.	4.	3.
22	2000.	4.	3.
23	2000.	4.	3.
24	2000.	4.	3.
25	2000.	4.	3.
26	2000.	4.	3.
27	2000.	4.	3.
28	2000.	4.	3.
29	2000.	4.	3.
30	2000.	4.	3.
31	2000.	4.	3.

32	2000.	4.	3.
33	2000.	4.	3.
34	2000.	4.	3.
35	2000.	4.	3.
36	2000.	4.	3.
37	2000.	4.	3.
38	2000.	4.	3.
39	2000.	4.	3.
40	2000.	4.	3.
41	2000.	4.	3.
42	2000.	4.	3.

-1

ELEM

42	2	7	1	0	0		
1	1	0	0	1	0	1	5
2	1	0	0	2	0	2	6
3	1	0	0	3	0	3	7
4	1	0	0	4	0	4	8
5	1	0	0	5	0	5	6
6	1	0	0	6	0	6	7
7	1	0	0	7	0	7	8
8	1	0	0	8	0	5	9
9	1	0	0	9	0	6	10
10	1	0	0	10	0	7	11
11	1	0	0	11	0	8	12
12	1	0	0	12	0	9	10
13	1	0	0	13	0	10	11
14	1	0	0	14	0	11	12
15	1	0	0	15	0	9	13
16	1	0	0	16	0	10	14
17	1	0	0	17	0	11	15
18	1	0	0	18	0	12	16
19	1	0	0	19	0	13	14
20	1	0	0	20	0	14	15
21	1	0	0	21	0	15	16
22	1	0	0	22	0	13	17
23	1	0	0	23	0	14	18
24	1	0	0	24	0	15	19
25	1	0	0	25	0	16	20
26	1	0	0	26	0	17	18
27	1	0	0	27	0	18	19
28	1	0	0	28	0	19	20
29	1	0	0	29	0	17	21
30	1	0	0	30	0	18	22
31	1	0	0	31	0	19	23
32	1	0	0	32	0	20	24
33	1	0	0	33	0	21	22
34	1	0	0	34	0	22	23
35	1	0	0	35	0	23	24
36	1	0	0	36	0	21	25
37	1	0	0	37	0	22	26
38	1	0	0	38	0	23	27
39	1	0	0	39	0	24	28
40	1	0	0	40	0	25	26
41	1	0	0	41	0	26	27
42	1	0	0	42	0	27	28

-1

SOLC

1	2.	0.	0.
5			
2	3.	0.	0.

```

9
3      5.      0.      0.
13
4      6.      0.      0.
17
5      7.      0.      0.
21
6      9.      0.      0.
25
7      0.      -3.     0.
25
8      0.      -3.     0.
28
-1
1      0.000    -0.035
5 6 6 7 7 8 9 10 10 11 11 12 13 14 14 15 15 16 17 18 18 19 19 20 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
2      0.000    -0.035
21 22 22 23 23 24 25 26 26 27 27 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
-1
1      0.00    300.00    0.00    -3.00
25 26 26 27 27 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
-1
LINM
1
OPTI
200 126 84
      5.00      20.00      2.40      0.25      1.00
STOP

```

**Les déplacements:**

*Déplacements donnés par OPTIPORT*

noeuds	x	y	z	degres de liberte= impose)		
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00 *	0.00000E+00 *	0.00000E+00 *
2	0.40000E+03	0.00000E+00	0.00000E+00	0.00000E+00 *	0.00000E+00 *	0.00000E+00 *
3	0.80000E+03	0.00000E+00	0.00000E+00	0.00000E+00 *	0.00000E+00 *	0.00000E+00 *
4	0.12000E+04	0.00000E+00	0.00000E+00	0.00000E+00 *	0.00000E+00 *	0.00000E+00 *
5	0.00000E+00	0.30000E+03	0.00000E+00	0.10075E+04	-0.16906E+00	-0.38137E+01
6	0.40000E+03	0.30000E+03	0.00000E+00	0.10075E+04	-0.11784E+01	-0.24342E+01
7	0.80000E+03	0.30000E+03	0.00000E+00	0.10075E+04	-0.10635E+01	-0.25104E+01
8	0.12000E+04	0.30000E+03	0.00000E+00	0.10075E+04	-0.94268E+00	-0.31082E+01
9	0.00000E+00	0.60000E+03	0.00000E+00	0.24366E+04	-0.34368E+00	-0.37639E+01
10	0.40000E+03	0.60000E+03	0.00000E+00	0.24366E+04	-0.21614E+01	-0.25878E+01
11	0.80000E+03	0.60000E+03	0.00000E+00	0.24365E+04	-0.19614E+01	-0.26341E+01
12	0.12000E+04	0.60000E+03	0.00000E+00	0.24365E+04	-0.17130E+01	-0.31932E+01
13	0.00000E+00	0.90000E+03	0.00000E+00	0.37658E+04	-0.52535E+00	-0.33303E+01
14	0.40000E+03	0.90000E+03	0.00000E+00	0.37657E+04	-0.29519E+01	-0.22436E+01
15	0.80000E+03	0.90000E+03	0.00000E+00	0.37657E+04	-0.26925E+01	-0.22939E+01
16	0.12000E+04	0.90000E+03	0.00000E+00	0.37656E+04	-0.23078E+01	-0.27407E+01
17	0.00000E+00	0.12000E+04	0.00000E+00	0.48596E+04	-0.70236E+00	-0.26742E+01
18	0.40000E+03	0.12000E+04	0.00000E+00	0.48595E+04	-0.35512E+01	-0.17311E+01
19	0.80000E+03	0.12000E+04	0.00000E+00	0.48595E+04	-0.32552E+01	-0.17962E+01

20	0.12000E+04	0.12000E+04	0.00000E+00	0.48595E+04	-0.27389E+01	-0.20468E+01
21	0.00000E+00	0.15000E+04	0.00000E+00	0.56524E+04	-0.85715E+00	-0.17397E+01
22	0.40000E+03	0.15000E+04	0.00000E+00	0.56524E+04	-0.39618E+01	-0.11674E+01
23	0.80000E+03	0.15000E+04	0.00000E+00	0.56523E+04	-0.36469E+01	-0.11615E+01
24	0.12000E+04	0.15000E+04	0.00000E+00	0.56523E+04	-0.30243E+01	-0.13254E+01
25	0.00000E+00	0.18000E+04	0.00000E+00	0.60913E+04	-0.96723E+00	-0.14098E+01
26	0.40000E+03	0.18000E+04	0.00000E+00	0.60912E+04	-0.41881E+01	-0.25335E+00
27	0.80000E+03	0.18000E+04	0.00000E+00	0.60911E+04	-0.38652E+01	-0.51395E+00
28	0.12000E+04	0.18000E+04	0.00000E+00	0.60910E+04	-0.31843E+01	0.89204E-01

*déplacements obtenus avec le SAP2000*

SAP2000 v7.40 File: EXPTHESE Ton-cm Units PAGE 1

6/20/03 16:30:4

perso

JOINT DISPLACEMENTS

JOINT LOAD	U1	U2	U3	R1	R2	R3
1 LOAD1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2 LOAD1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3 LOAD1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4 LOAD1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5 LOAD1	1007.9411	0.0000	-0.1655	0.0000	3.8149	0.0000
6 LOAD1	1007.9417	0.0000	-1.1784	0.0000	2.4349	0.0000
7 LOAD1	1007.9343	0.0000	-1.0635	0.0000	2.5111	0.0000
8 LOAD1	1007.9207	0.0000	-0.9462	0.0000	3.1091	0.0000
9 LOAD1	2437.5962	0.0000	-0.3366	0.0000	3.7652	0.0000
10 LOAD1	2437.5527	0.0000	-2.1615	0.0000	2.5885	0.0000
11 LOAD1	2437.5261	0.0000	-1.9614	0.0000	2.6349	0.0000
12 LOAD1	2437.5156	0.0000	-1.7200	0.0000	3.1942	0.0000
13 LOAD1	3767.2930	0.0000	-0.5148	0.0000	3.3315	0.0000
14 LOAD1	3767.2266	0.0000	-2.9519	0.0000	2.2442	0.0000
15 LOAD1	3767.1855	0.0000	-2.6925	0.0000	2.2946	0.0000
16 LOAD1	3767.1707	0.0000	-2.3183	0.0000	2.7415	0.0000
17 LOAD1	4861.6001	0.0000	-0.6883	0.0000	2.6751	0.0000
18 LOAD1	4861.5146	0.0000	-3.5512	0.0000	1.7316	0.0000
19 LOAD1	4861.4624	0.0000	-3.2552	0.0000	1.7967	0.0000
20 LOAD1	4861.4414	0.0000	-2.7530	0.0000	2.0475	0.0000
21 LOAD1	5654.7476	0.0000	-0.8396	0.0000	1.7404	0.0000

22	LOAD1	5654.6699	0.0000	-3.9618	0.0000	1.1677	0.0000
23	LOAD1	5654.6235	0.0000	-3.6469	0.0000	1.1619	0.0000
24	LOAD1	5654.6152	0.0000	-3.0419	0.0000	1.3258	0.0000
25	LOAD1	6093.8213	0.0000	-0.9461	0.0000	1.4104	0.0000
26	LOAD1	6093.6631	0.0000	-4.1881	0.0000	0.2535	0.0000
27	LOAD1	6093.5591	0.0000	-3.8652	0.0000	0.5142	0.0000
28	LOAD1	6093.4990	0.0000	-3.2054	0.0000	-0.0891	0.0000

**Efforts internes de la première itération : (comparaison avec résultats donnés par SAP 2000)**

**Effort normal N:**

Valeurs de l'effort normal dans les éléments (tonne)				
N° d'élément	noeud1 PORT	noeud1 SAP	noeud2 PORT	noeud2 SAP
1	-1.35E+01	-13.26	-1.35E+01	-13.23
2	-9.43E+01	-94.28	-9.43E+01	-94.26
3	-8.51E+01	-85.1	-8.51E+01	-85.07
4	-7.54E+01	-75.71	-7.54E+01	-75.68
5	3.36E-02	0.03	3.36E-02	0.03
6	-4.41E-01	-0.44	-4.41E-01	-0.44
7	-8.17E-01	-0.82	-8.17E-01	-0.82
8	-1.40E+01	-13.7	-1.40E+01	-13.67
9	-7.87E+01	-78.66	-7.86E+01	-78.63
10	-7.18E+01	71.84	-7.18E+01	71.81
11	-6.16E+01	-61.92	-6.16E+01	-61.89
12	-2.61E+00	-2.61	-2.61E+00	-2.61
13	-1.60E+00	-1.6	-1.60E+00	-1.6
14	-6.37E-01	-0.64	-6.37E-01	-0.64
15	-1.45E+01	-14.27	-1.45E+01	-14.24
16	-6.33E+01	-63.25	-6.32E+01	-63.22
17	-5.85E+01	-58.5	-5.85E+01	-58.47
18	-4.76E+01	-47.88	-4.76E+01	-47.85
19	-3.98E+00	-3.98	-3.98E+00	-3.98
20	-2.46E+00	-2.46	-2.46E+00	-2.46
21	-9.01E-01	-0.9	-9.01E-01	-0.9
22	-1.42E+01	-13.89	-1.41E+01	-13.86
23	-4.80E+01	-47.96	-4.79E+01	-47.93
24	-4.50E+01	-45.03	-4.50E+01	-45
25	-3.45E+01	-34.79	-3.45E+01	-34.76
26	-5.11E+00	-5.11	-5.11E+00	-5.11
27	-3.13E+00	-3.13	-3.13E+00	-3.13
28	-1.26E+00	-1.26	-1.26E+00	-1.26
29	-1.24E+01	-12.12	-1.24E+01	-12.09
30	-3.29E+01	-32.86	-3.28E+01	-32.83
31	-3.13E+01	-31.35	-3.13E+01	-31.32
32	-2.28E+01	-23.13	-2.28E+01	-23.1
3	-4.66E+00	-4.66	-4.66E+00	-4.66
34	-2.78E+00	-2.78	-2.78E+00	-2.78
35	-4.80E-01	-0.48	-4.80E-01	-0.48
36	-8.82E+00	-8.54	-8.79E+00	-8.51

37	-1.81E+01	-18.12	-1.81E+01	-18.09
38	-1.75E+01	-17.47	-1.74E+01	-17.44
39	-1.28E+01	-13.09	-1.28E+01	-13.06
40	-9.48E+00	-9.48	-9.48E+00	-9.48
41	-6.27E+00	-6.27	-6.27E+00	-6.27
42	-3.60E+00	-3.6	-3.60E+00	-3.6

**Tableau4.2** : valeurs des efforts normaux dans les éléments (résultats de: OPTIPORT &SAP2000)

**Effort tranchant:**

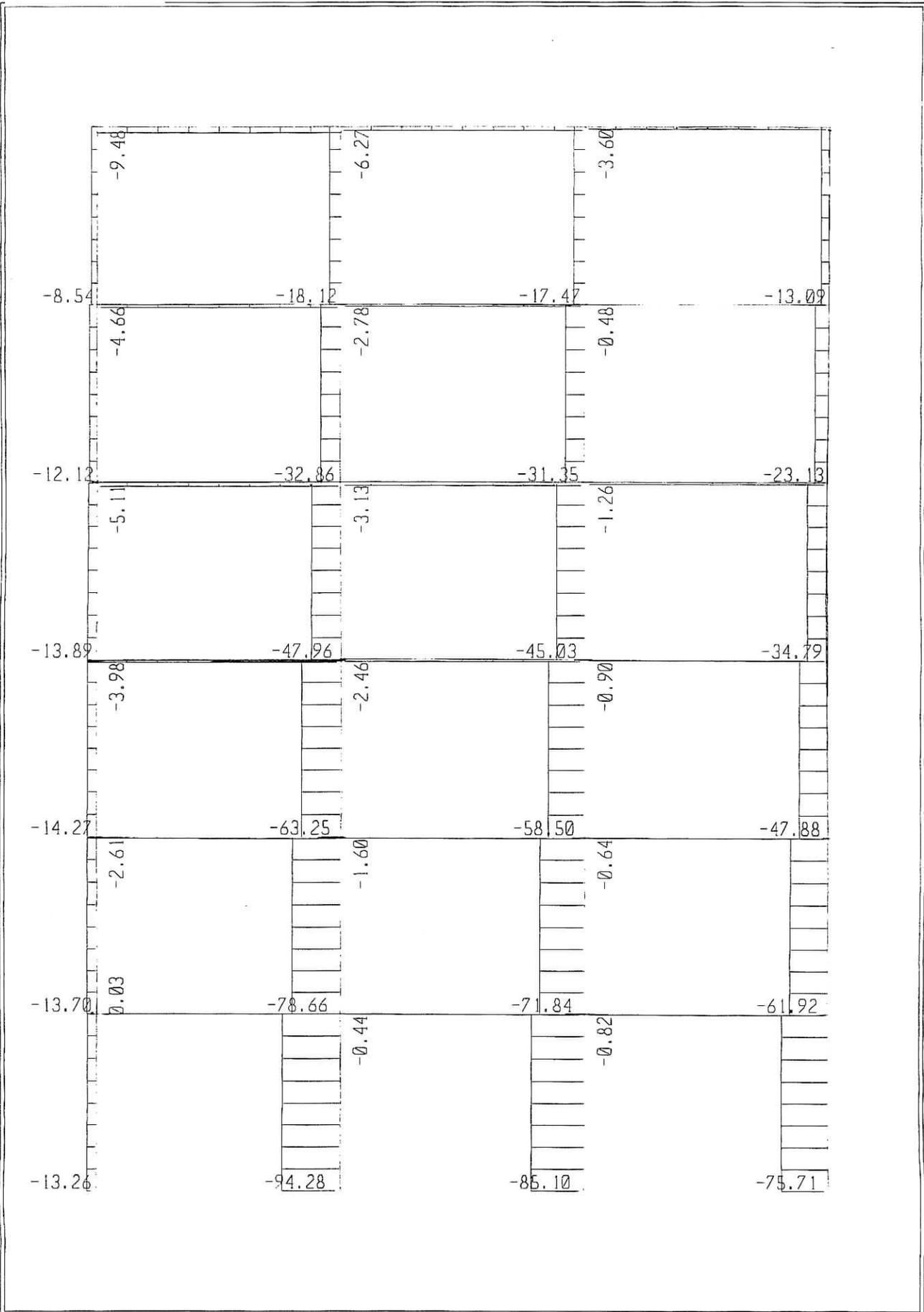
Valeurs de l'effort tranchant éléments (tonne)				
N° d'élément	noeud1 PORT	noeud1 SAP	noeud2 PORT	noeud2 SAP
1	0.61930E+01	6.19	0.61930E+01	6.19
2	0.91359E+01	9.14	0.91359E+01	9.14
3	0.89733E+01	8.97	0.89733E+01	8.97
4	0.76978E+01	7.70	0.76978E+01	7.70
5	0.47277E+00	0.47	0.14510E+02	14.51
6	-0.10844E+01	-1.00	0.12953E+02	12.95
7	-0.27563E+00	-0.28	0.13762E+02	13.76
8	0.41594E+01	4.16	0.41594E+01	4.16
9	0.96106E+01	9.61	0.96106E+01	9.61
10	0.93489E+01	9.35	0.93489E+01	9.35
11	0.68811E+01	6.88	0.68811E+01	6.88
12	0.59239E+00	0.59	0.14630E+02	14.63
13	-0.75118E+00	-0.75	0.13286E+02	13.29
14	-0.24483E-01	-0.02	0.14013E+02	14.01
15	0.37695E+01	33.77	0.37695E+01	33.77
16	0.85964E+01	8.60	0.85964E+01	8.60
17	0.83899E+01	8.39	0.83899E+01	8.39
18	0.62443E+01	6.24	0.62443E+01	6.24
19	-0.34466E+00	-0.34	0.13693E+02	13.69
20	-0.15722E+01	-1.57	0.12465E+02	12.47
21	-0.97492E+00	-0.97	0.13063E+02	13.06
22	0.27477E+01	2.75	0.27477E+01	2.75
23	0.70777E+01	7.08	0.70777E+01	7.08
24	0.68312E+01	6.83	0.68312E+01	6.83
25	0.53435E+01	5.34	0.53435E+01	5.34
26	-0.17495E+01	-1.75	0.12288E+02	12.29
27	-0.27843E+01	-2.78	0.11253E+02	11.25
28	-0.24041E+01	-2.4	0.11633E+02	11.63
29	0.18594E+01	1.86	0.18594E+01	1.86
30	0.50925E+01	5.09	0.50925E+01	5.09
31	0.49661E+01	4.97	0.49661E+01	4.97
32	0.40820E+01	4.08	0.40820E+01	4.08
3	-0.35489E+01	-3.55	0.10489E+02	10.49
34	-0.42221E+01	-4.22	0.98153E+01	9.82

35	-0.40307E+01	-4.03	0.10007E+02	10.1
36	-0.47707E+00	-0.48	-0.47707E+00	-0.48
37	0.32098E+01	3.21	0.32098E+01	3.21
38	0.26655E+01	2.67	0.26655E+01	2.67
39	0.36018E+01	3.60	0.36018E+01	3.60
40	-0.57923E+01	-5.51	0.11245E+02	11.53
41	-0.68460E+01	-6.56	0.10191E+02	10.47
42	-0.72549E+01	-6.97	0.97825E+01	10.06

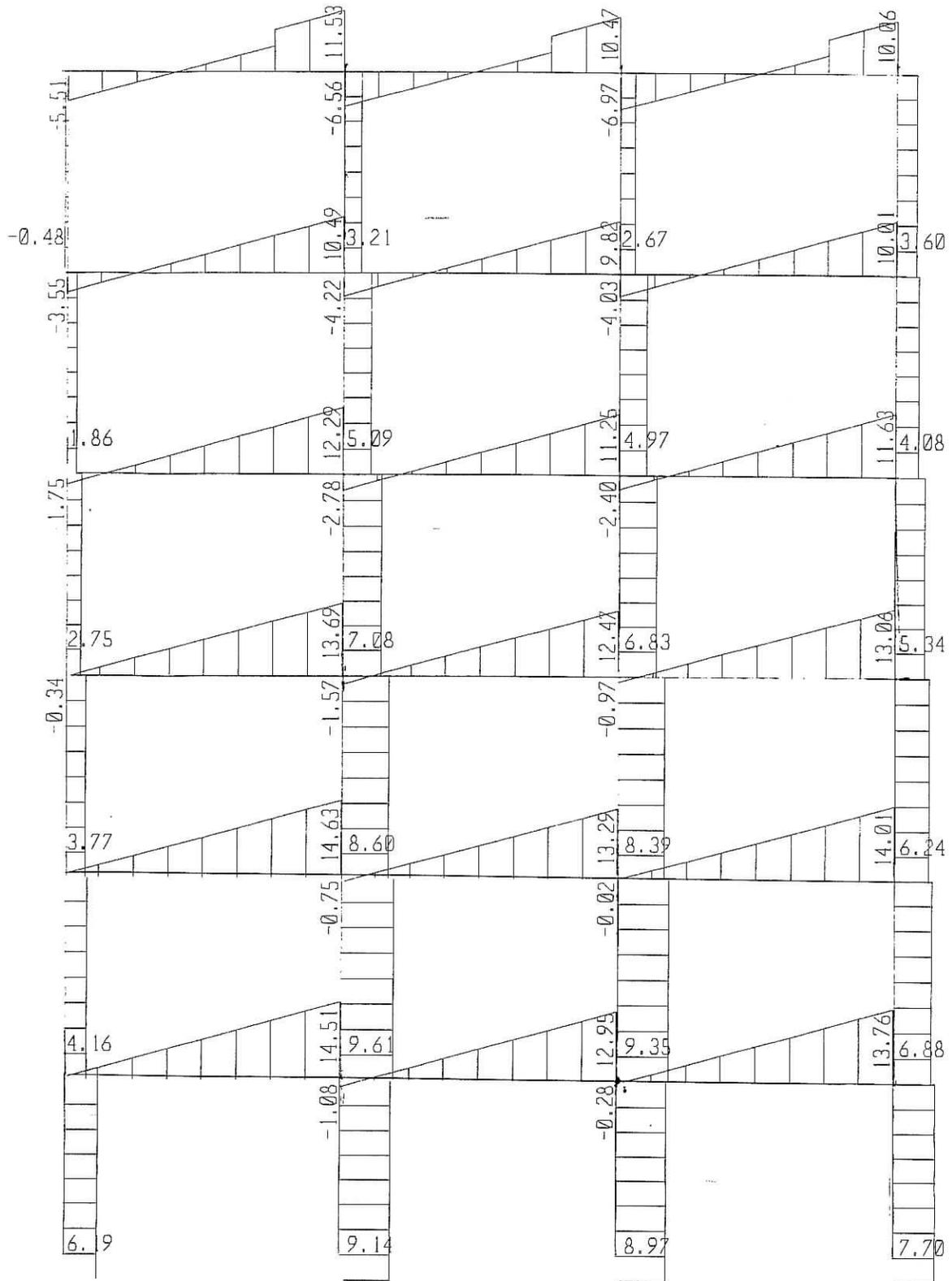
**Tableau4.3** : valeurs des efforts normaux dans les éléments (résultats de: OPTIPOINT & SAP2000)

***Moment fléchissant:***

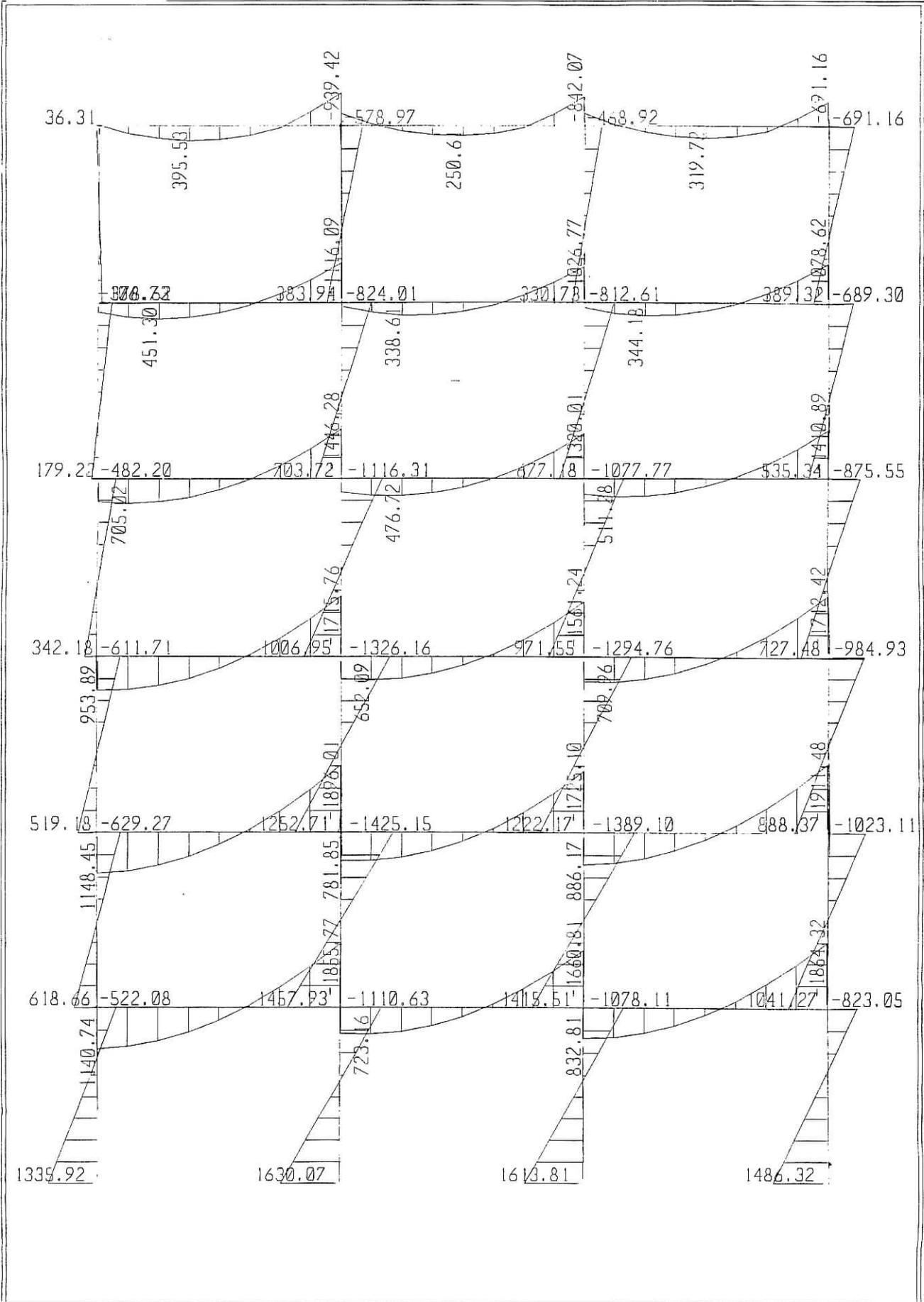
Valeurs du moment fléchissant (t.cm)				
N° d'élément	noeud1 PORT	noeud1 SAP	noeud2 PORT	noeud2 SAP
1	0.13357E+04	1335.92	-0.52215E+03	-522.08
2	0.16300E+04	1630.07	-0.11107E+04	-1110.63
3	0.16138E+04	1613.81	-0.10782E+04	-1078.11
4	0.14862E+04	1486.32	-0.82313E+03	-823.05
5	0.11407E+04	1140.74	-0.18559E+04	-1855.77
6	0.71285E+03	712.79	-0.16609E+04	-1660.81
7	0.83288E+03	832.81	-0.18644E+04	-1864.32
8	0.61859E+03	618.66	-0.62922E+03	-629.27
9	0.14580E+04	1457.93	-0.14252E+04	-1425.15
10	0.14155E+04	1415.51	-0.13891E+04	-1389.10
11	0.10412E+04	1041.27	-0.10231E+04	-1023.11
12	0.11484E+04	1148.45	-0.18961E+04	-1896.01
13	0.78189E+03	781.85	-0.17251E+04	-1696.43
14	0.88621E+03	886.17	-0.19115E+04	-1911.48
15	0.51917E+03	519.18	0.37695E+01	-611.71
16	0.12527E+04	1252.71	-0.13262E+04	-1326.16
17	0.12222E+04	1222.17	-0.12948E+04	-1294.76
18	0.88837E+03	888.37	-0.98491E+03	-984.93
19	0.95383E+03	953.89	-0.17158E+04	-1715.76
20	0.61737E+03	617.35	-0.15613E+04	-1561.24
21	0.70511E+03	705.08	-0.17124E+04	-1712.42
22	0.34217E+03	342.18	-0.48213E+03	-482.20
23	0.10070E+04	1006.95	-0.11163E+04	-1091.83
24	0.97158E+03	971.55	-0.10778E+04	-1077.77
25	0.72751E+03	727.48	-0.87553E+03	-875.55
26	0.66137E+03	61.41	-0.14463E+04	-1446.28
27	0.37377E+03	373.76	-0.13200E+04	-1320.01
28	0.43497E+03	434.95	-0.14109E+04	-1410.89
29	0.17924E+03	179.22	-0.37859E+03	-378.62
30	0.70374E+03	703.72	-0.82400E+03	-824.01
31	0.67721E+03	677.18	-0.81261E+03	-812.61



SAP2000 v7.40 - File:expthese - Axial Force Diagram (LOAD1) - Ton-cm Units



SAP2000 v7.40 - File:exphese - Shear Force 2-2 Diagram (LOAD1) - Ton-cm Units



SAP2000 v7.40 - File:expheuse - Moment 3-3 Diagram (LOAD1) - Ton-cm Units

32	0.53536E+03	535.34	-0.68925E+03	-689.30
3	0.27184E+03	271.89	-0.11161E+04	-1116.09
34	0.91867E+02	91.87	-0.10268E+04	-1026.77
35	0.11659E+03	116.57	-0.10786E+04	-1078.62
36	-0.10675E+03	-106.73	-0.47707E+00	36.31
37	0.38397E+03	383.94	-0.57896E+03	-578.97
38	0.33074E+03	330.73	-0.46889E+03	-468.92
39	0.38938E+03	389.32	-0.69117E+03	-691.16
40	-0.36368E+02	-36.31	0.11245E+02	-939.42
41	-0.36047E+03	-360.46	-0.84205E+03	-842.07
42	-0.37316E+03	-373.15	-0.69117E+03	-691.16

**Tableau4.4** : valeurs des moments fléchissants dans les éléments (résultats de: OPTIPORT & SAP2000)

Nous constatons que les valeurs des efforts internes obtenues par le programme OPTIPORT et celles obtenues par le SAP2000 sont très voisines, donc les contraintes dans les éléments sont justes.

**Valeurs des dimensions des éléments de la dernière itération:**

élément	Hauteur (cm)	Largeur (cm)	élément	Hauteur (cm)	Largeur (cm)
1	18.7792295	11.7328898	22	5.76894943	5.00000000
2	16.7256052	16.5873349	23	17.0660287	14.8813194
3	16.0835007	15.7386164	24	16.0161553	14.4694277
4	15.7861741	14.2550178	25	12.6907143	10.9465974
5	20.0000000	12.7915036	26	15.7502822	11.5874963
6	19.6794676	13.4469705	27	20.0000000	14.3541015
7	20.0000000	9.78339880	28	16.6592348	8.23862598
8	5.65362080	5.00000000	29	5.23809507	5.00000000
9	18.8780168	15.6986077	30	14.0410218	14.5919949
10	18.8364518	16.4047145	31	15.2399329	13.8543901
11	13.7902522	10.4584773	32	10.7290147	7.42373117
12	17.6418067	10.5229829	33	15.8949392	11.4980616
13	20.0000000	19.4737795	34	14.9286377	11.6010930
14	19.3871487	8.90741402	35	17.5246102	9.91553571
15	9.29234810	6.32403548	36	5.04651739	6.02014468
16	17.6671317	12.4568889	37	10.0553132	9.94007110
17	17.5129937	17.7443514	38	6.47434064	5.89549988
18	12.8888388	11.7540509	39	13.8814655	14.4585599

19	17.8424404	10.4040508	40	15.3807904	10.3005320
20	20.0000000	16.2881717	41	12.7877920	9.14400741
21	18.9594651	8.90086323	42	15.0325682	10.4771490

**Tableau 4.5:** valeurs des dimensions des éléments de la dernière itération.

**Valeurs des contraintes:**

élément	Première itération	Dernière itération
1	64.6588	2.3981
2	82.0310	2.3374
3	80.8860	2.4000
4	74.3811	2.3794
5	89.1339	2.3999
6	79.8137	2.4013
7	89.5796	2.3999
8	30.7285	1.7314
9	73.1412	2.3389
10	70.8425	2.2742
11	52.4912	2.3997
12	91.1496	2.3999
13	82.9510	2.4001
14	91.8448	2.3999
15	29.9011	2.0787
16	66.1868	2.3999
17	64.5139	2.2677
18	49.2204	2.3893
19	82.5396	2.4000
20	75.1245	2.4000
21	82.3060	2.4000
22	23.6616	1.6620
23	55.4980	2.2374
24	53.5604	2.1163
25	43.4484	2.3998

26	69.6407	2.3999
27	63.5746	2.4000
28	67.8526	2.4000
29	18.6192	1.5608
30	40.8568	2.1090
31	40.2857	2.3928
32	34.0382	2.2471
33	53.7664	2.3998
34	49.4865	2.4000
35	51.8798	2.3929
36	5.5415	1.1292
37	28.5042	2.3964
38	23.2371	2.0580
39	33.7509	2.1037
40	45.4929	2.4000
41	40.7556	2.3999
42	33.3832	2.4000

**Tableau 4.5:** valeurs des contraintes de la première et la dernière itération