

4.1 Introduction :

Pour résoudre le problème de vibrations des plaques, des suppositions et des approximations sont faites à l'aide de la version $h-p$ de la méthode des éléments finis pour rendre le système continu avec une infinité de degrés de liberté à un système discret avec un nombre fini de degrés de liberté. Un programme écrit en Fortran, basé sur une formulation mixte de la version h et la version p , de sorte que la structure soit discrétisée en un nombre d'éléments (version h), et chaque élément est enrichi par des polynômes d'interpolations de degrés p (version p). Le programme fait aussi, l'assemblage des matrices élémentaires dans les matrices globales, l'élimination des degrés de liberté restreints, et le calcul des paramètres de fréquences.

L'ordre de numérotation des nœuds, côtés, faces triangulaires, faces rectangulaires, de l'élément pentaèdre utilisé est montré dans la figure (3.1) :

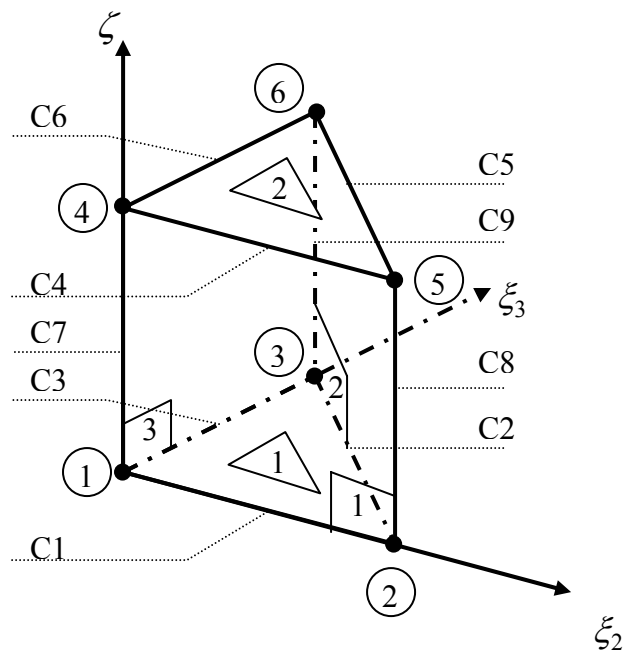
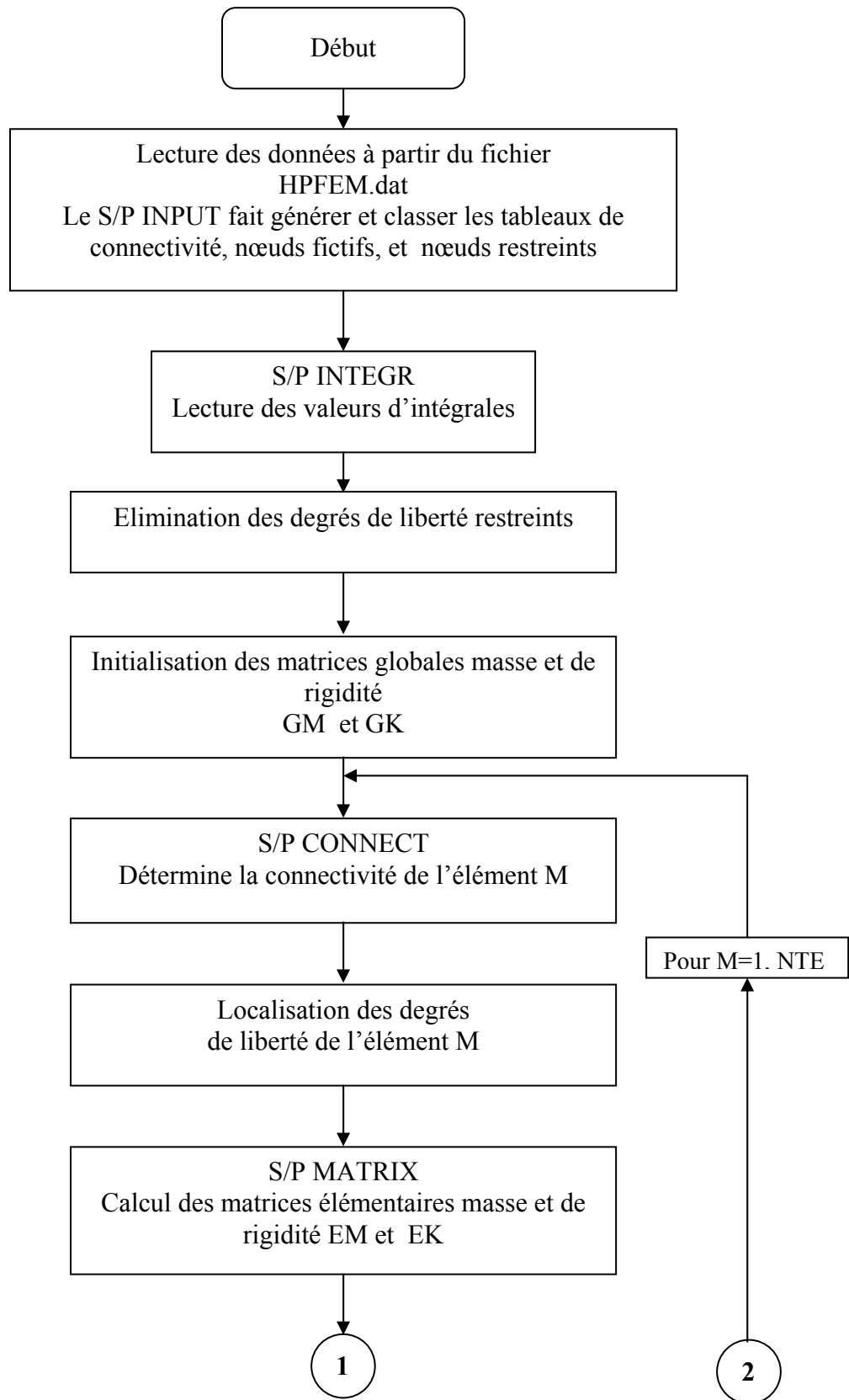


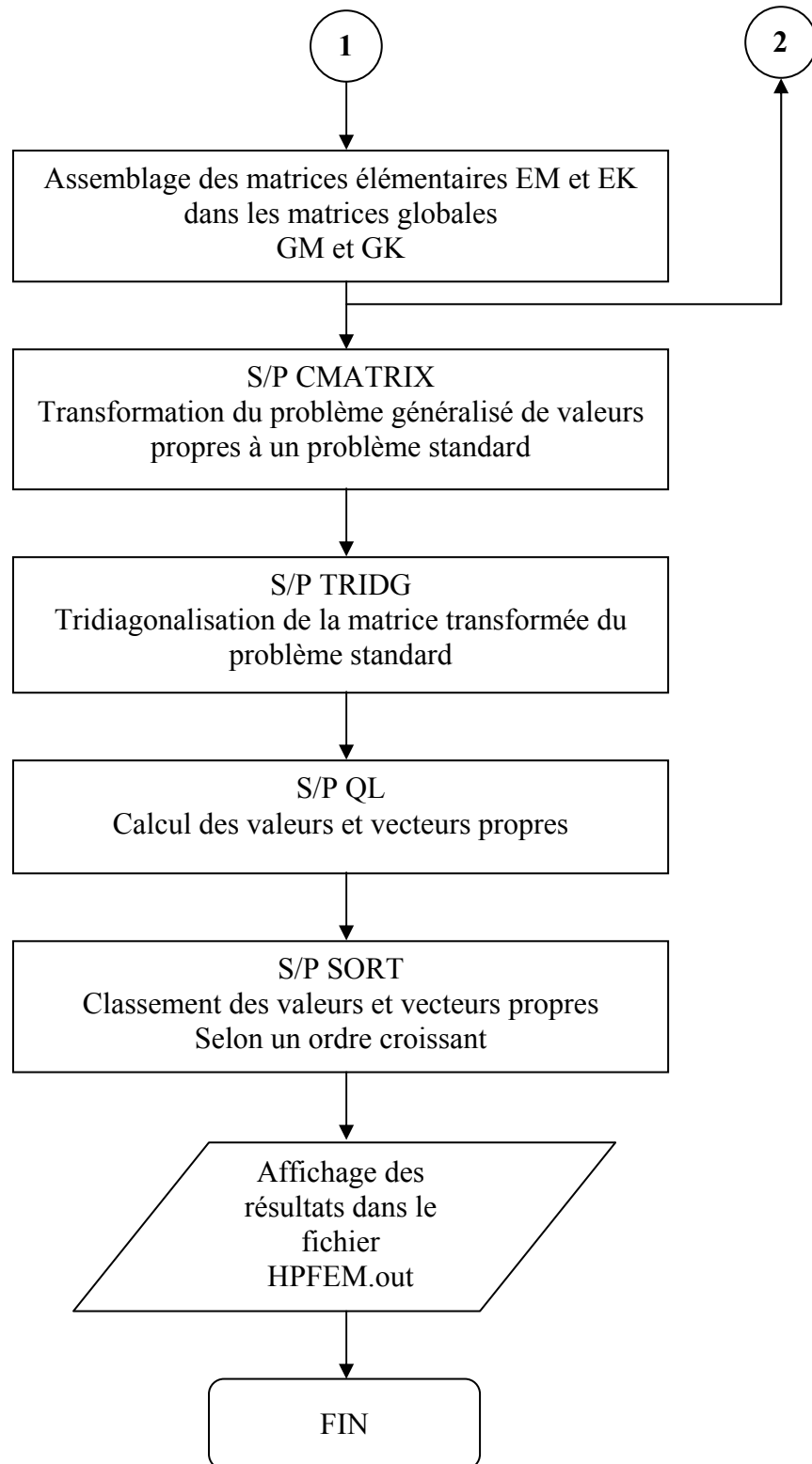
FIG 4.1 : Ordre de numérotation des nœuds, cotés, faces triangulaires, faces rectangulaires, de l'élément pentaèdre.

4.2 Environnement de la programmation :

Le matériel utilisé pour l'exécution de ce travail est un micro-ordinateur ATTILON-AMD 1700+, 256 M-RAM. Le langage utilisé est le Fortran 90 version 4.0 de l'année 1994-1995.

4.3 Organigramme :





4.4 Description du programme:

Le programme élaboré calcule les paramètres de fréquences tridimensionnels des plaques vibrantes. Il se compose d'un programme principal qui fait appel aux sous programmes suivants :

- S/P INPUT
- S/P INTEGR
- S/P CONNECT
- S/P MATRIX
- S/P CMATRIX
- S/P TRID
- S/P QL
- S/P SORT

4.4.1 S/P INPUT :

Le sous programme INPUT permet de lire les différentes données qui caractérisent la géométrie et les propriétés physiques du matériau.

La lecture se fait à partir d'un fichier déjà existant appelé HPFEM.dat selon l'ordre suivant :

- NTE : Nombre total d'éléments.
- NTN : Nombre total de nœuds.
- NTC : Nombre total de côtés.
- NTFT : Nombre total de faces triangulaires.
- NTFR : Nombre total de faces rectangulaires.
- NP : Degré du polynôme d'interpolation.
- NMODE : Nombre des modes propres voulus.
- H : Epaisseur de la plaque
- XNU : Coefficient de Poisson.
- NNR : nombre de nœuds restreints.
- NCR : nombre de côtés restreints.
- NFTR : nombre de faces triangulaires restreintes.
- NFRR : nombre de faces rectangulaires restreintes.
- X(i),Y(i) : Coordonnées des nœuds réels.

Il faut introduire pour chaque élément son tableau de connectivité (nœuds réels, cotés, faces triangulaires, faces rectangulaires, et intérieur). Ce classement se fait en fonction du maillage de la plaque et du degré du polynôme d'interpolation.

Il faut aussi introduire des tableaux des degrés de liberté restreints pour :

1. nœuds réels restreints.
2. cotés restreints.
3. faces triangulaires restreintes.
4. faces rectangulaires restreintes.

4.4.2 S/P INTEGR :

Les intégrales données dans les équations (2.35) et (2.39), sont calculées par un logiciel Mathématique MATLAB 6.0. Les valeurs de ces intégrales sont classées et stockées dans un seul fichier (integr.dat) sous forme de vecteur. Le nombre de fonctions de forme correspond a un degré de polynôme $p = 8$ est $m=176$. Le sous programme INTEGR prend toutes les valeurs de chaque forme d'intégrale à partir du fichier (integr.dat) et les stocke dans des matrices qui seront utilisées pour le calcul des matrices masse et de rigidité élémentaires.

4.4.3 S/P CONNECT (M) :

Ce sous-programme assure la connectivité de l'élément pentaèdre M avec les éléments adjacents dans le maillage en fonction du degré d'interpolation. En effet le sous programme CONNECT assure la connectivité entre les nœuds, côtés, et faces, pour tout les éléments du maillage.

4.4.4 S/P MATRIX (M) :

Une fois que la lecture des données nécessaires pour le calcul des matrices élémentaires telles que les intégrales et les coordonnées des nœuds, le sous programme MATRIX calcule les matrices élémentaires à l'aide des équations (2.34) et (2.40).

4.4.5 Elimination des DDL restreints :

Dans le fichier des données le degré de liberté prend deux valeurs soit la valeur 1 (degré de liberté fixe), soit la valeur 0 (degré de liberté libre). Une boucle avec un test d'élimination des degrés des libertés fixes est utilisée dans le programme principal.

4.4.6 Assemblage :

L'assemblage des matrices élémentaires **EM** et **EK** dans les matrices globales **GM** et **GK** se fait dans le programme principal. Pour cela les matrices globales sont initialisées à zéro.

La connectivité nécessaire pour l'assemblage est assurée par le sous programme CONNECT. Les éléments des matrices élémentaires correspondants aux degrés de liberté restreints ne seront pas assemblés dans les matrices globales.

4.4.7 S/P CMATRIX:

Le sous programme **CMATRIX** consiste à transformer le problème généralisé aux valeurs propres de la forme:

$$[\mathbf{K} - \omega^2 \mathbf{M}] \mathbf{x} = 0 \quad (4.1)$$

à un problème standard de la forme:

$$[\mathbf{A} - \omega^2 \mathbf{I}] \mathbf{x} = 0 \quad (4.2)$$

où **I** est la matrice identité.

La procédure utilisée est la factorisation de *Cholesky* de la matrice masse donnée par :

$$\mathbf{M} = \mathbf{L} \cdot \mathbf{L}^T \quad (4.3)$$

Où **L** est une matrice triangulaire inférieure.

Un vecteur **y** est choisi de sorte que :

$$\mathbf{y} = \mathbf{L}^T \mathbf{x} \quad (4.4)$$

Par substitution de (4.3) et (4.4) dans (4.1), et après quelques transformations nous trouvons l'équation suivante :

$$[\mathbf{L}^{-1} \cdot \mathbf{K} \cdot \mathbf{L}^{-T} - \omega^2 \mathbf{I}] \mathbf{y} = 0 \quad (4.5)$$

L'équation (45) est bien de la forme donnée dans (4.2) avec :

$$\mathbf{A} = \mathbf{L}^{-1} \cdot \mathbf{K} \cdot \mathbf{L}^{-T} \quad (4.6)$$

Afin de déterminer la matrice A donnée par (4.6), et pour éviter le calcul de l'inverse très coûteux numériquement, deux substitutions en arrière sont utilisées. La relation (4.6) peut être écrite sous la forme :

$$K = L.A.L^T \quad (4.7)$$

La première substitution considère un vecteur Z tel que :

$$Z = A.L^T \quad (4.8)$$

Le vecteur Z est obtenu par substitution en arrière en résolvant équation :

$$L.Z = K \quad (4.9)$$

A partir de (4.8), et puisque A est symétrique, la relation (4.9) s'exprime comme suit :

$$L.A = Z^T \quad (4.10)$$

Et enfin une deuxième substitution en arrière de (4.10) est effectuée pour la détermination de la matrice A .

4.4.8 S/P TRID :

Le sous programme **TRID** permet d'obtenir une matrice tridiagonale symétrique au bout d'un nombre fini de transformations semblables, en utilisant la procédure de **Householder** qui consiste à faire apparaître des zéros en dehors de la bande tridiagonale, colonne par colonne, donc simultanément ligne par ligne, à l'aide des matrices élémentaires de *householder*.

La méthode de *householder* utilise une séquence de transformations orthogonales sur la matrice A , données par:

$$A_{r+1} = H_r . A_r . H_r \quad (4.11)$$

où :

$$A_1 = A \quad (4.12)$$

Et la matrice H_r est donnée par :

$$H_r = I - 2.\mu.z.z^T \quad (4.13)$$

Où μ est un scalaire, et z est un vecteur défini par :

$$z_i = \begin{cases} 0 & i < r+1 \\ a_{r,r+1} + S & i = r+1 \\ a_{ri} & i > r+1 \end{cases} \quad (4.14)$$

Le vecteur constante S est donnée par :

$$S^2 = \sum_{i=r+1}^n a_{ri}^2 \quad (4.15)$$

S est pris du même signe que l'élément $a_{r,r+1}$.

Le vecteur z dans l'expression (4.13) vérifie la multiplication vectorielle suivante :

$$z^T \cdot z = \frac{1}{\mu} \quad (4.16)$$

La substitution de (4.14) et (4.15) dans (4.16) donne :

$$\mu = \frac{1}{2 \cdot S(a_{r,r+1} + S)} \quad (4.17)$$

La matrice de *householder* de la première transformation est de la forme :

$$H_1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & & & & \\ 0 & H_1^{(n-1)} & & & \\ \dots & & & & \\ 0 & & & & \end{bmatrix} \quad (3.18)$$

Après une transformation *householder*, A_2 aura la forme :

$$A_2 = \begin{bmatrix} x & x & 0 & \dots & 0 \\ x & x & x & \dots & x \\ 0 & x & . & & \\ \dots & \dots & & . & \\ 0 & x & & & . \end{bmatrix} \quad (4.19)$$

En conséquence, H est symétrique, orthogonale, et conserve les valeurs propres, et au bout de (n-2) étapes la matrice A_{n-1} est tridiagonale.

4.4.9 S/P QL :

Ce sous programme permet de déterminer les valeurs et vecteurs propres de la matrice tridiagonale A . Il repose sur l'algorithme de la méthode QL qui consiste à décomposer la matrice A en un produit d'une matrice orthogonale Q et une matrice triangulaire inférieure L .

La technique utilisée dans cette méthode utilise des transformations unitaires définies par :

$$A_{r+1} = L_r \cdot Q_r \quad (4.20)$$

de sorte que :

$$A_r = Q_r \cdot L_r \quad (4.21)$$

Où Q_r est une matrice orthogonale et L_r est une matrice triangulaire inférieure.

Lorsque $r \rightarrow \infty$, A_{r+1} converge vers la matrice diagonale des valeurs propres, et Q tend vers la matrice des vecteurs propres.

Pour une matrice A quelconque d'ordre n , une itération de l'algorithme QL a un coût de calcul proportionnel à n^3 mais pour une matrice tridiagonale, ce coût est proportionnel à n .

4.4.10 S/P SORT :

Le sous programme SORT permet de classer les valeurs propres et vecteurs propres associés dans un ordre croissant.

4.4.11 AFFICHAGE DES RESULTATS :

Après détermination et classement des valeurs propres et vecteurs propres, les résultats sont affichés dans un fichier de sortie appelé HPFEM.OUT.