METHODE DE PROGRAMMATION

Les Eléments Finis

Synthèse du travail

Par

Renaud CHAMPREDONDE
Quentin DEREKS

Année 2006-2007 **Encadrant : M-M MAUBOURGUET**

SOMMAIRE

1. INTRODUCTION

Dans le cadre de la mineure « Eléments finis », nous avons pour objectif de coder avec le langage Fortran un programme capable de résoudre l'équation de la chaleur. Cette étude se place sur une plaque carré d'arrête a=0,5m avec une production de chaleur constante. Avec les différentes symétries, l'étude est restreinte sur un quart de la plaque. Tout d'abord, nous allons vous présenter la mise en équation de ce problème avec la méthode des éléments finis. Ensuite, nous allons vous commenter l'architecture de notre programme. Cette architecture est décrite dans notre code par l'intermédiaire de nombreux commentaires. Vous pouvez également trouver en pièces jointes à ce rapport le code de notre programme ainsi que deux vidéos illustrant nos résultats avec des éléments à 4 nœuds et des éléments à 9 nœuds.

2. MISE EN EQUATION

Equation de la chaleur avec source : $\rho c \frac{\partial T}{\partial t} - \lambda \Delta T - \phi = 0$

$$\Rightarrow \rho c \frac{\partial T}{\partial t} - \lambda \, div(\vec{grad}\, T) - \phi = 0$$

forme intégrale : $W(T) = \int_\Omega v \left(\rho c \frac{\partial T}{\partial t} - \lambda \, div(\vec{grad}\, T) - \phi\right) d\Omega = 0$

$$= \rho c \int_\Omega v \frac{\partial T}{\partial t}\, d\Omega + \lambda \int_\Omega \vec{grad}\, v \cdot \vec{grad}\, T \, d\Omega - \lambda \int_\Gamma v \, \vec{grad}\, T \cdot \vec{n}\, ds - \phi \int_\Omega v \, d\Omega$$

$$\theta = \begin{pmatrix} T_1 \\ \vdots \\ T_i \\ \vdots \\ T_{NBPT} \end{pmatrix} \qquad \theta_e = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix}$$

$\hookrightarrow$ noeuds de l'élément.

$T_e(P) = N_e(P)\, \theta_e$ — N_e : fonction de forme ; P : un point de l'élément e

$\theta_e = L_e \, \theta$ — L_e : matrice de localisation.

$$\int_\Omega v \frac{\partial T}{\partial t}\, d\Omega = \sum_e \int_{\Omega_e} {}^tV\, {}^tL_e\, {}^tN_e\, N_e\, L_e \frac{\partial \theta}{\partial t}\, d\Omega$$

$$= {}^tV \sum_e \left[{}^tL_e \int_{\Omega_e} \left({}^tN_e N_e \, d\Omega\right) L_e \right] \frac{\partial \theta}{\partial t}$$

$$\vec{grad}\, T_e(P) = D(T_e(P)) = D(N_e(P))\, \theta_e = B_e(P)\, \theta_e = B_e(P)\, L_e\, \theta$$

de la même façon : $\vec{grad}\, v = B_e(P)\, L_e\, V$

$$\Rightarrow \int_\Omega \vec{grad}\, v \cdot \vec{grad}\, T\, d\Omega = \sum_e \int_{\Omega_e} {}^tV\, {}^tL_e\, {}^tB_e\, B_e\, L_e\, \theta \, d\Omega$$

$$= {}^tV \sum_e \left[{}^tL_e \int_{\Omega_e} \left({}^tB_e\, B_e \, d\Omega\right) L_e \right] \theta$$

$$\int_\Gamma v \, \vec{grad}\, T \cdot \vec{n}\, ds = \int_\Gamma v \frac{\partial T}{\partial n}\, ds$$

condition limite : $\lambda \frac{\partial T}{\partial n} = -h(T - T_0) \quad \Rightarrow \frac{\partial T}{\partial n} = -\frac{h}{\lambda}(T - T_0)$

$$\Rightarrow \int_\Gamma v \, \vec{grad}\, T \cdot \vec{n}\, ds = \frac{-h}{\lambda} \int_\Gamma v (T - T_0)\, ds$$

$$= \frac{-h}{\lambda} \int_\Gamma v T\, ds + \frac{h T_0}{\lambda} \int_\Gamma v \, ds$$

$$= \frac{-h}{\lambda}\, {}^tV \sum_e \left[{}^tL_e \int_\Gamma \left({}^tN_e N_e \, ds\right) L_e \right] \theta + \frac{h T_0}{\lambda}\, {}^tV \sum_e \left[{}^tL_e \int_\Gamma {}^tN_e \, ds \right]$$

$$\phi \int_\Omega v \, d\Omega = \phi \, {}^t V \sum_e \left[{}^t L_e \int_\Omega {}^t N_e \, d\Omega \right]$$

donc
$$W(T) = \rho c \, {}^t V \sum_e \left[{}^t L_e \int_{E_e} \left({}^t N_e N_e \, d\Omega \right) L_e \right] \frac{\partial \theta}{\partial t}$$
$$+ \lambda \, {}^t V \sum_e \left[{}^t L_e \int_{E_e} \left({}^t B_e B_e \, d\Omega \right) L_e \right] \theta$$
$$+ h \, {}^t V \sum_e \left[{}^t L_e \int_\Gamma \left({}^t N_e N_e \, dS \right) L_e \right] \theta - h T_0 \, {}^t V \sum_e \left[{}^t L_e \int_\Gamma {}^t N_e \, dS \right]$$
$$- \phi \, {}^t V \sum_e \left[{}^t L_e \int_\Omega {}^t N_e \, d\Omega \right]$$

$$\Rightarrow \rho c \sum_e \left[{}^t L_e \int_{E_e} \left({}^t N_e N_e \, d\Omega \right) L_e \right] \frac{\partial \theta}{\partial t}$$
$$+ \left[\lambda \sum_e \left[{}^t L_e \int_{E_e} \left({}^t B_e B_e \, d\Omega \right) L_e \right] + h \sum_e \left[{}^t L_e \int_\Gamma \left({}^t N_e N_e \, dS \right) L_e \right] \right] \theta$$
$$- \left[h T_0 \sum_e \left[{}^t L_e \int_\Gamma {}^t N_e \, dS \right] + \phi \sum_e \left[{}^t L_e \int_\Omega {}^t N_e \, d\Omega \right] \right]$$
$$= 0$$

$$AM = \sum_e \left[{}^t L_e \int_{E_e} \left({}^t N_e N_e \, d\Omega \right) L_e \right]$$
$$AK = \sum_e \left[{}^t L_e \int_{E_e} \left({}^t B_e B_e \, d\Omega \right) L_e \right]$$
$$AK\_2 = \sum_e \left[{}^t L_e \int_\Gamma \left({}^t N_e N_e \, dS \right) L_e \right]$$
$$BQ = \phi \sum_e \left[{}^t L_e \int_\Omega {}^t N_e \, d\Omega \right]$$
$$BQ\_2 = T_0 \sum_e \left[{}^t L_e \int_\Gamma {}^t N_e \, dS \right]$$

$$\rho c \, AM \, \frac{\partial \theta}{\partial t} + (\lambda \, AK + h \, AK\_2) \theta = BQ + h \, BQ\_2$$

Intégration temporelle (Crank-Nicholson)

$$\rho c \, AM \, \frac{\theta^{n+1} - \theta^n}{DT} + (\lambda \, AK + h \, AK\_2) \frac{\theta^{n+1} + \theta^n}{2} = BQ + h \, BQ\_2$$

$$\Rightarrow A \theta^{n+1} = B \qquad \text{avec} \qquad A = \frac{\rho c \, AM}{DT} + \frac{\lambda AK + h AK\_2}{2}$$
$$B = \left(\frac{\rho c \, AM}{DT} - \frac{\lambda AK + h AK\_2}{2} \right) \theta^n + BQ + h \, BQ\_2$$

3. COMMENTAIRES DE L'ARCHITECTURE DU PROGRAMME

Tout d'abord, notre programme se divise en un programme principal (main) et trois sous-programmes. Dans la première partie du programme, nous allons à la recherche des différentes données nécessaires à la suite du programme (les données physiques et le maillage), nous calculons également certaines données numériques nécessaires à la méthode des éléments finis (largeur de bande, nombre d'éléments totaux). Nous établissons aussi le dimensionnement des différents objets utilisés par la suite du programme ainsi que la définition de la condition initiale sur la température. Ces objets peuvent être utilisés dans les sous-programmes grâce à la création d'un module « définition » qu'il faut appeler à chaque fois.

Une fois toutes données prises, nous passons au calcul proprement dit (sous-programme « calcul »). Dans cette partie, nous définissons avant tout la matrice W contenant les poids et les ξ pour la méthode des points de Gauss. Ensuite nous entrons dans les boucles. Le calcul se découpe en deux étapes : le calcul des contributions et le calcul des conditions aux limites. En effet, l'un demande une double intégration sur les points de Gauss et l'autre une seule. Dans ces différentes phases, on appelle à plusieurs repris la fonction FUNCT4 ou FUNCT9 (ça dépend du choix de maillage fait par l'utilisateur). Ces deux fonctions sont définies dans le sous-programme « Fonctions4&9 », elles permettent de calculer les facteurs de forme ainsi que la jacobienne respectivement pour un maillage à 4 nœuds par éléments et pour un maillage à 9 nœuds par éléments. Une fois les matrices A et B calculées (fin des boucles sauf celle sur le temps), nous appelons la fonction « Gauss » permettant de résoudre le système AX=B. Il reste alors à écrire les résultats dans un fichier sous le format Tecplot afin de pouvoir les visualiser.

4. CONCLUSIONS

Au cours de cette étude qui constitue une première approche à la méthode de programmation en éléments finis, nous avons pu aborder les différentes parties qui s'inscrivent dans l'élaboration d'un programme avec cette méthode. Après avoir posé le problème, nous avons pus mettre en équation notre problème avec notamment l'écriture de la formulation intégrale qui permet de ne pas perdre le fil du cheminement de la programmation.

Le programme élaboré a montré une bonne simulation par cette pour méthode pour le maillage à quatre nœuds par élément. La température modélisée augmente progressivement et de façon cohérente avec nos conditions initiales et aux limites. Cependant, notre programme s'est révélé ne pas être aussi performant pour le maillage à neuf nœuds. En effet la vidéo effectuée avec le logiciel Tecplot a montré l'incohérence de l'évolution de la température qui présente dés les premiers pas de temps une augmentation de température au milieu du domaine. La source de cette erreur n'a pu être identifiée, mais nous supposons une mauvaise gestion des conditions aux limites pour le maillage à neuf nœuds.

En conclusion, cette première approche de la méthode de programmation en éléments finis nous a permis de comprendre le cheminement de la résolution d'équations de ce type de problème. Cette initiation nous autorisera une meilleure compréhension des codes de calcul basés sur cette méthode tel que le puissant logiciel TELEMAC pour mieux appréhender les résultats et connaître la démarche à aborder pour l'utilisation de ceux-ci.