

# MÉMOIRE

présenté

à

L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

Comme exigence partielle pour l'obtention du grade de

Maître ès Sciences Appliquées

(M.Sc.A.)

par

Jocelyn Tremblay, B.Sc.

---

SIMULATION D'UN BAIN DE MÉTAL EN FUSION

AVEC CONVECTION NATURELLE

---

septembre 1986



### Mise en garde/Advice

Afin de rendre accessible au plus grand nombre le résultat des travaux de recherche menés par ses étudiants gradués et dans l'esprit des règles qui régissent le dépôt et la diffusion des mémoires et thèses produits dans cette Institution, **l'Université du Québec à Chicoutimi (UQAC)** est fière de rendre accessible une version complète et gratuite de cette œuvre.

Motivated by a desire to make the results of its graduate students' research accessible to all, and in accordance with the rules governing the acceptance and diffusion of dissertations and theses in this Institution, the **Université du Québec à Chicoutimi (UQAC)** is proud to make a complete version of this work available at no cost to the reader.

L'auteur conserve néanmoins la propriété du droit d'auteur qui protège ce mémoire ou cette thèse. Ni le mémoire ou la thèse ni des extraits substantiels de ceux-ci ne peuvent être imprimés ou autrement reproduits sans son autorisation.

The author retains ownership of the copyright of this dissertation or thesis. Neither the dissertation or thesis, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

## RÉSUMÉ

Les équations mathématiques décrivant le transfert de chaleur dans un bain de métal en fusion sont utilisées pour la construction d'un simulateur. L'équation d'énergie est exprimée en terme de l'enthalpie volumique du métal en utilisant la transformée de Kirchhoff de la conductivité thermique. L'utilisation de l'enthalpie et de la transformée de Kirchhoff permet de considérer une variation de la densité, de la chaleur massique et de la conductivité thermique du métal solide, ainsi que du métal liquide, avec la température. Les équations de Navier-Stokes, pour la convection naturelle, sont exprimées à l'aide des variables primitives en utilisant l'approximation de Boussinesq.

L'équation d'énergie est découplée des équations du mouvement. Elle est intégrée en deux dimensions à l'aide de la méthode des éléments finis. Des essais sont effectués avec une matrice de capacité thermique consistante et une matrice de capacité thermique modifiée, ainsi qu'avec une variation quadratique de la transformée de Kirchhoff en fonction de l'enthalpie et une variation linéaire. Les équations du mouvement sont intégrées en deux dimensions à l'aide de la méthode des différences finies par le logiciel SOLA. Une simulation d'un bain d'aluminium liquide entourant une charge d'aluminium solide illustre l'importance de la convection naturelle.

  
Étudiant

  
Directeur de recherche

## REMERCIEMENTS

Je tiens avant tout à remercier les professeurs Rung T. Bui, Ph.D., et André Charette, D.Sc., qui, comme directeur et codirecteur de recherche, m'ont fourni conseils et encouragements tout au long de la préparation de ce diplôme. Je les remercie sincèrement pour toutes les heures qu'ils m'ont consacrées ainsi que pour les nombreuses discussions enrichissantes et stimulantes que j'ai pu avoir avec eux.

Je remercie les membres de l'équipe d'ingénierie des systèmes et en particulier messieurs Guy Simard et Thierry Bourgeois pour tous les conseils judicieux qu'ils m'ont donnés lors de la programmation du simulateur. Un grand merci également à mon épouse pour la mise en page de ce document.

## TABLE DES MATIÈRES

	page
Résumé .....	ii
Remerciements .....	iii
Liste des symboles .....	vi
Liste des tableaux .....	viii
Liste des figures .....	ix
INTRODUCTION .....	2
CHAPITRE 1: LA MÉTHODE DE L'ENTHALPIE .....	6
1.1 L'équation d'énergie en conduction .....	6
1.2 Définition de l'enthalpie volumique et de la transformée de Kirchhoff de la conductivité thermique .....	7
1.3 Construction des polynômes $T(H)$ , $H(T)$ , $\theta(H)$ .....	8
CHAPITRE 2: RÉOLUTION DE L'ÉQUATION ET PREMIERS RÉSUL- TATS .....	14
2.1 Choix des fonctions de pondération et d'inter- polation .....	14
2.2 Formulation intégrale faible .....	14
2.3 Modification de la matrice de capacité thermique .....	17
2.4 Résolution numérique .....	18
2.5 Analyse des résultats .....	20
CHAPITRE 3: INFLUENCE DE LA CONVECTION NATURELLE .....	30
3.1 Considérations générales .....	30
3.2 Détails du calcul .....	32

3.2.1	Détails sur la résolution de l'équation d'énergie .....	33
3.2.2	Détails sur la résolution des équations du mouvement .....	34
3.2.3	Couplage des sous-modèles éléments finis et différences finies .....	36
3.3	Résolution numérique .....	36
3.4	Analyse des résultats .....	37
	CONCLUSION .....	43
	Bibliographie .....	46
	Figures .....	49
	Appendice 1: organisation générale du programme .....	78
	Appendice 2: algorithme de multiplication d'une matrice restructurée $n \times 7$ par un vecteur colonne .....	84
	Annexe 1: listage du programme .....	87
	Annexe 2: développement des termes FVX, FVY et VISY apparaissant dans SOLA .....	163

## LISTE DES SYMBOLES

$c$	la chaleur massique
$Cte$	une constante
$g$	l'accélération gravitationnelle
$H$	l'enthalpie volumique
$k$	la conductivité thermique
$L$	la chaleur latente de fusion
$N$	fonctions d'interpolation
$p$	pression totale
$p_d$	pression dynamique
$p_s$	pression statique
$\vec{q}$	un flux de chaleur
$T$	la température absolue
$T_f$	la température de fusion
$T_l$	la température du liquide
$\bar{T}_{liq}$	la température moyenne du liquide
$T_s$	la température du solide
$\bar{T}_{sol}$	la température moyenne du solide
$T_o$	une température de référence
$u$	la composante horizontale de la vitesse $\vec{V}$
$v$	la composante verticale de la vitesse $\vec{V}$
$\bar{V}$	la vitesse moyenne
$x,y$	coordonnées cartésiennes

Alphabet Grec

$\alpha$	coefficient de différentiation en amont
$\beta$	coefficient de dilatation thermique
$\Gamma^{(e)}$	la frontière de l'élément (e)
$\mu$	viscosité absolue
$\nu$	viscosité cinématique
$\Omega$	le domaine d'intégration
$\Omega^{(e)}$	le domaine de l'élément (e)
$\omega$	facteur de relaxation
$\rho$	la densité
$\rho_0$	la densité évaluée à la température $T_0$
$\theta$	la transformée de Kirchhoff
$\tau$	la durée d'une simulation

Écriture matricielle

$\{ \}$	un vecteur colonne
$\{ \}^T$	un vecteur ligne
$[ ]$	une matrice
$[ ]^T$	la transposée d'une matrice



## LISTE DES TABLEAUX

	page
Tableau 1. Température moyenne du métal solide et du métal liquide (option BQ).....	23
Tableau 2. Température moyenne du métal solide et du métal liquide (option DQ).....	24
Tableau 3. Température moyenne du métal solide et du métal liquide (option BL).....	25
Tableau 4. Température moyenne du métal solide et du métal liquide (option DL).....	26
Tableau 5. Temps de calcul moyen pour solutionner un pas de temps $\Delta t$ de l'équation d'énergie sur un ordinateur VAX-11/780.	27
Tableau 6. Variation temporelle de la température et de la vitesse moyenne du métal liquide calculée par l'option DQ en convection naturelle.....	40

## LISTE DES FIGURES

	page
Figure 1. Schéma du domaine d'intégration .....	49
Figure 2. Relation entre la température et l'enthalpie .....	50
Figure 3. Variation quadratique de la transformée de Kirchhoff de la conductivité thermique en fonction de l'en- thalpie volumique .....	51
Figure 4. Comparaison entre une variation quadratique et une variation linéaire de la transformée de Kirchhoff de la conductivité thermique du solide .....	52
Figure 5. Comparaison entre une variation quadratique et une variation linéaire de la transformée de Kirchhoff de la conductivité thermique du liquide .....	53
Figure 6. (a) Vecteur des fonctions d'interpolation et (b) élément isoparamétrique triangulaire linéaire à trois noeuds, de classe $C^0$ .....	54
Figure 7. (a) Le domaine géométrique et les températures initiales. (b) Le champ enthalpique $H(x,y)$ corres- pondant aux température initiales $T(x,y)$ de la figure (a) .....	55
Figure 8. (a) Représentation des noeuds voisins au noeud $r$ et de la disposition des éléments isoparamétriques triangulaires linéaires. (b) Restructuration des matrices .....	56
Figure 9. Mise en évidence d'un problème causé par la linéa- risation de la transformée de Kirchhoff .....	57
Figure 10. Arrangement des variables $u$ , $v$ et $p$ d'une cellule formée par les éléments triangulaire (k) et (l) ....	58
Figure 11. Surfaces enthalpiques calculées avec l'option BQ (740 noeuds). (a) La condition initiale. (b) Après 1 min. (c) Après 5 min. (d) Après 15 min ...	59
Figure 12. Surfaces enthalpiques calculées avec l'option BQ (740 noeuds). (a) Après 30 min. (b) Après 60 min. (c) Après 120 min. (d) Après 220 min. ....	60
Figure 13. Position de l'interface solide-liquide calculée avec l'option BQ (740 noeuds) .....	61

	page
Figure 14. Isothermes calculées avec l'option BQ (740 noeuds), (a) après 2 min, (b) après 5 min, (c) après 15 min, (d) après 30 min. ....	62
Figure 15. Isothermes calculées après 60 minutes de chauffage en utilisant 740 noeuds et les quatre options (25°C/isotherme) .....	63
Figure 16. Isothermes calculées après 120 minutes de chauffage en utilisant 740 noeuds et les quatre options (25°C/isotherme) .....	64
Figure 17. Isothermes calculées après 180 minutes de chauffage en utilisant 740 noeuds et les quatre options (20°C/isotherme) .....	65
Figure 18. Isothermes calculées après 60 minutes de chauffage en utilisant 112 noeuds et les quatre options (25°C/isotherme) .....	66
Figure 19. Isothermes calculées après 120 minutes de chauffage en utilisant 112 noeuds et les quatre options (25°C/isotherme) .....	67
Figure 20. Isothermes calculées après 180 minutes de chauffage en utilisant 112 noeuds et les quatre options (20°C/isotherme) .....	68
Figure 21. Surfaces enthalpiques calculées en convection naturelle par l'option DQ (740 noeuds). (a) la condition initiale. (b) Après 1 min. (c) Après 30 min. (d) Après 60 min. ....	69
Figure 22. Circulation du fluide au cours des premières secondes .....	70
Figure 23. Circulation du fluide après 5 minutes de chauffage..	71
Figure 24. Circulation du fluide après 30 minutes de chauffage .....	72
Figure 25. Circulation du fluide après 60 minutes de chauffage .....	73
Figure 26. Circulation du fluide après 90 minutes de chauffage .....	74
Figure 27. Circulation du fluide après 115 minutes de chauffage .....	75

	page
Figure 28. Circulation du fluide après 30 minutes de chauffage en utilisant seulement 112 noeuds .....	76
Figure 29. Circulation du fluide après 90 minutes de chauffage en utilisant seulement 112 noeuds .....	77
Figure 30. Organigramme du programme FUSION .....	80
Figure 31. Organigramme du sous-programme BLOC_PRE .....	80
Figure 32. Organigramme du sous-programme BLOC_CALCUL .....	80
Figure 33. Organigramme du sous-programme BLOC_SORTIE .....	80
Figure 34. Organigramme du sous-programme ENERGIE .....	81
Figure 35. Organigramme du sous-programme MOUVEMENT .....	81

## INTRODUCTION

Ce mémoire s'inscrit dans un objectif global de recherche qui consiste à produire un modèle mathématique du four de métal chaud utilisé dans l'industrie de l'aluminium. Le travail présenté ici constitue une des étapes de ce vaste projet, soit l'élaboration d'une méthode simple et efficace pour simuler la fusion d'un bloc de métal solide à l'intérieur d'un bain liquide.

La caractéristique fondamentale du procédé de fusion est la présence d'une frontière mobile, interface entre les phases solide et liquide. Dans la littérature, le problème de la détermination de la position de cette frontière mobile est désignée sous le nom de problème de Stéfan.

Les problèmes de Stéfan pour lesquels il existe des solutions analytiques sont relativement simples et peu nombreux (Rubinstein 1971 et Lunardini 1981). De plus, l'environnement hostile d'un four de fusion rend les prises de mesures compliquées et coûteuses. C'est pourquoi, nous nous tournons vers les méthodes numériques.

Une revue des méthodes numériques pouvant solutionner les problèmes de frontières mobiles peut être trouvée dans Crank (1981) et Ockendon et Hodgkins (1975). Dans la littérature récente, deux méthodes sont souvent employées; (1) la méthode de transformation de coordonnées (Gupta et Kumar 1985), (2) la méthode de l'enthalpie (Pham 1985 et Civan et Sliepcevich 1984). Les travaux présentés ici sont basés sur une formulation enthalpique. Les principaux avantages de cette méthode sont (Ralph III et Klaus-Jürgen 1982):

- i) il n'existe aucune condition devant être satisfaite à la frontière de changement de phase;
- ii) nous n'avons pas besoin de chercher la position de l'interface, cette dernière correspondant tout simplement à une courbe isenthalpique;
- iii) nous n'avons pas besoin de considérer les états séparément, l'état étant déterminé par l'enthalpie du métal;
- iv) il devient tout naturel de considérer la masse volumique et la chaleur massique comme étant des fonctions de la température.

Le domaine géométrique adopté pour le travail est simple mais réaliste puisqu'il est basé sur des données réelles d'un four existant et en opération. Ce four est utilisé pour la fusion de l'aluminium, c'est pourquoi, nous avons choisi ce métal.

Le four réel est d'environ 1 m x 3 m x 11 m, le solide étant déposé dans le fond sur toute la longueur. Dû au problème de temps de calcul, la géométrie utilisée pour le travail sera en deux dimensions.

Le but du mémoire est donc de simuler la fusion d'un bloc d'aluminium solide déposé dans un bain d'aluminium liquide. Voici le plan suivi dans la rédaction de ce mémoire.

- i) Explication de la méthode de l'enthalpie. Élaboration des fonctions à utiliser dans les travaux.
- ii) Intégration de l'équation d'énergie en conduction par la méthode des éléments finis. L'avantage principal du choix de cette méthode est sa facilité d'application à un domaine de géométrie

complexe. Deux modèles seront établis: l'un avec une matrice de capacité thermique consistante et l'autre avec une matrice de capacité thermique modifiée. Nous pourrons y voir l'influence d'une variation linéaire plutôt qu'une variation quadratique de la transformée de Kirchhoff de la conductivité thermique en fonction de l'enthalpie.

- iii) L'équation d'énergie est modifiée pour tenir compte de la convection naturelle. Les équations du mouvement sont intégrées par différences finies à l'aide d'une version modifiée du logiciel SOLA.

Nous aurions pu résoudre l'équation de continuité et les équations de Navier-Stokes par éléments finis mais une telle approche aurait augmenté de beaucoup le temps de calcul. Nous aurions également pu tout résoudre en différences finies mais c'est ce que font déjà plusieurs logiciels commerciaux. Notre intérêt a donc été d'étudier:

- (1) le couplage des deux sous-modèles (équation d'énergie et équations du mouvement) avec le problème de la correspondance des noeuds et
- (2) le comportement de l'équation d'énergie discrétisée par éléments finis avec différentes simplifications.



CHAPITRE 1  
LA MÉTHODE DE L'ENTHALPIE

## CHAPITRE 1

### LA MÉTHODE DE L'ENTHALPIE

Dans ce chapitre, nous établirons la formulation mathématique du problème de Stéfan à deux phases; pour cela, nous devons définir deux fonctions.

Dû à l'importance fondamentale de ces définitions, ce chapitre constitue la base de l'édifice mathématique du mémoire.

#### 1.1 L'équation d'énergie en conduction

Soit  $\Omega$  une région bornée dans l'espace Euclidien  $\mathbb{R}^2$ . Nous supposons que la frontière mobile  $\Gamma$ , interface entre les phases solide-liquide, puisse être décrite en termes d'une fonction régulière  $F(x, y, t)$  telle que nous pouvons définir

$$\Gamma = \{(x, y, t) \in \Omega \times (0, \tau) : F(x, y, t) = 0\}$$

$$\Omega^+ = \{(x, y, t) \in \Omega \times (0, \tau) : F(x, y, t) > 0 \text{ et } T(x, y, t) \geq T_F\} \quad (\text{liquide})$$

$$\Omega^- = \{(x, y, t) \in \Omega \times (0, \tau) : F(x, y, t) < 0 \text{ et } T(x, y, t) \leq T_F\} \quad (\text{solide})$$

$$\Omega^+ \cup \Gamma \cup \Omega^- = \Omega \times (0, \tau)$$

où  $0 \leq t \leq \tau$ , Elliot et Ockendon (1982). La figure 1 schématise la situation.

En utilisant une formulation basée sur la température, nous devons solutionner une équation d'énergie dans la phase solide pour trouver  $T_s$  et une équation d'énergie dans la phase liquide pour trouver  $T_l$ .

$$\begin{aligned} \rho(T_s)c(T_s)\frac{\partial T_s}{\partial t} &= \nabla \cdot (k(T_s)\nabla T_s) \quad , \quad \text{pour } F(x,y,t) < 0 \\ \rho(T_l)c(T_l)\frac{\partial T_l}{\partial t} &= \nabla \cdot (k(T_l)\nabla T_l) \quad , \quad \text{pour } F(x,y,t) > 0 \end{aligned} \quad (1)$$

La dernière équation suppose une absence de mouvements convectifs dans le liquide. Les phases solide et liquide sont séparées par l'interface

définie par l'équation  $F(x,y,t)=0$ .

Les conditions frontières (conditions de Stéfan) permettant de tenir compte du saut d'énergie entre les deux phases sont données par (Özişik 1980):

$$T_s(x,y,t) = T_l(x,y,t) = T_F \quad \text{à } F(x,y,t) = 0$$

et 
$$k_s \nabla T_s \cdot \nabla F - k_l \nabla T_l \cdot \nabla F = -\rho L \frac{\partial F}{\partial t} \quad \text{à } F(x,y,t) = 0$$

Une formulation basée sur l'enthalpie plutôt que sur la température nous permettra d'éliminer simultanément le dédoubleage de l'équation d'énergie et les conditions de Stéfan.

## 1.2 Définition de l'enthalpie volumique et de la transformée de Kirchhoff de la conductivité thermique

L'enthalpie  $H$  est définie comme étant l'enthalpie sensible de la phase solide,

$$H(T) = \int_{T_F}^T \rho(T)c(T)dT \quad \text{pour } T < T_F$$

et la somme des enthalpies sensible et latente dans la phase liquide, (Voller et Cross 1981).

$$H(T) = \int_{T_F}^T \rho(T)c(T)dT + \rho L \quad \text{pour } T > T_F$$

Le zéro de l'enthalpie est défini à la température du solide saturé. Grâce à cette définition, nous pouvons considérer d'une façon tout à fait naturelle,

une variation de la masse volumique et de la chaleur massique avec la température.

De la règle de Liebniz sur la dérivation sous le signe intégral,

$$\frac{\partial H}{\partial t} = \frac{dH}{dT} \cdot \frac{\partial T}{\partial t} = \rho(T)c(T) \frac{\partial T}{\partial t} \quad (2)$$

La transformée de Kirchhoff de la conductivité thermique  $\chi(k)$  permet de tenir compte d'une variation de la conductivité thermique avec la température. Elle se définit ainsi (Elliot et Ockendon 1982):

$$\chi(k) = \theta(T) = \int_{T_r}^T k(T) dT \quad , \quad \forall T$$

En réutilisant la règle de Liebniz, le gradient de la transformée devient

$$\vec{\nabla} \theta = \frac{d\theta}{dT} \cdot \vec{\nabla} T = k(T) \vec{\nabla} T$$

Nous pouvons maintenant écrire

$$\vec{\nabla} \cdot (k \vec{\nabla} T) = \vec{\nabla} \cdot \vec{\nabla} \theta = \nabla^2 \theta \quad (3)$$

En plaçant (2) et (3) dans les équations (1), nous obtenons une seule équation d'énergie en conduction,

$$\frac{\partial H}{\partial t} = \nabla^2 \theta \quad (4)$$

équation valable pour les deux phases, où maintenant toutes les propriétés thermiques varient avec la température.

### 1.3 Construction des polynômes $T(H), H(T), \theta(H)$

Nous devons maintenant trouver une relation mathématique entre la température, l'enthalpie et la transformée de Kirchhoff pour le métal choisi qui est l'aluminium. Pour intégrer la chaleur massique, la masse volumique

et la conductivité thermique en fonction de la température, il faut les exprimer en fonction de cette variable.

Moses (1978) propose les valeurs de la chaleur massique pour l'aluminium pur à l'état solide:

$$c(T) = 0.459T + 767 \quad \text{J/kg} \cdot \text{K} \quad , \quad 250 \leq T \leq 933.15 \quad \text{K}$$

et pour l'état liquide:

$$c(T) = 1.09 \cdot 10^3 \quad \text{J/kg} \cdot \text{K} \quad , \quad 933.15 \leq T \leq 1500 \quad \text{K} .$$

Pour la conductivité thermique, deux polynômes (un pour le solide et l'autre pour le liquide) ont été ajustés par la méthode des moindres carrés sur des valeurs données dans Touloukian (1970).

$$k(T) = -8.65 \cdot 10^{-7}T^2 + 6.26 \cdot 10^{-4}T + 2.26 \quad \text{W/cm} \cdot \text{K} \quad , \quad 250 \leq T \leq 933.15 \quad \text{K}$$

$$k(T) = -1.23 \cdot 10^{-7}T^2 + 5.89 \cdot 10^{-4}T + 0.465 \quad \text{W/cm} \cdot \text{K} \quad , \quad 933.15 \leq T \leq 1500 \quad \text{K} .$$

La densité de l'aluminium liquide est donnée dans Coy et Mateer (1965);

$$\rho(T) = -0.3935T + 2756.8 \quad \text{kg/m}^3 \quad , \quad 933.15 \leq T \leq 1500 \quad \text{K} .$$

Des valeurs de la diffusivité thermique prises dans Touloukian (1973), nous permettent de déduire la densité de l'aluminium solide, à partir de la conductivité thermique et de la chaleur massique;

$$\rho(T) = -2.19 \cdot 10^{-4}T^2 + 0.104T + 2666 \quad \text{kg/m}^3 \quad , \quad 250 \leq T \leq 933.15 \quad \text{K} .$$

L'intégration de  $\rho c dT$  conduit à des résultats sur lesquels on peut ajuster :

- a) un polynôme de degré 4 pour l'enthalpie du solide. Celui-ci peut être réduit à un polynôme de degré 3 en utilisant la méthode

des moindres carrés.

b) un polynôme de degré 3 pour l'enthalpie du liquide. Ici aussi, une réduction est possible et un polynôme de degré 2 représente très bien l'évolution.

Voici les polynômes que nous utiliserons pour convertir les équations d'énergie formulées à l'aide de la température, en une équation basée sur l'enthalpie volumique.

$$\begin{aligned} \text{Pour } 250 \leq T < 933.15 \text{ K} \quad , \quad H(T) &= -0.04T^3 + 652T^2 + 2.04 \cdot 10^6 T - 2.44 \cdot 10^9 \quad \text{J/m}^3 \\ \text{et pour } 933.15 < T \leq 1500 \text{ K} \quad , \quad H(T) &= -214T^2 + 3.00 \cdot 10^6 T - 2.61 \cdot 10^9 + \rho L \quad \text{J/m}^3 \end{aligned}$$

où  $\rho L$  est l'enthalpie latente de fusion:

$$\rho L = 9.53 \cdot 10^8 \quad \text{J/m}^3 .$$

A la température de fusion, l'enthalpie n'est pas définie. Toutes les températures initiales doivent donc être converties en enthalpies initiales à l'aide de ces deux fonctions, en s'assurant qu'il n'y ait aucune température initiale égale à la température de fusion (933.15 K).

A la fin des calculs, nous devons pouvoir reconvertir les enthalpies en températures. Pour l'aluminium, l'enthalpie est une fonction monotone croissante en température;  $H$  définit donc d'une façon unique  $T$  (figure 2). Le théorème d'unicité pour la solution faible que nous développerons au chapitre suivant, dépend d'une façon cruciale de la monotonie de l'enthalpie. Les preuves d'existence et de convergence de la méthode de l'enthalpie sont démontrées dans Elliot et Ockendon (1982). La relation pour revenir à la température à partir de l'enthalpie a également été calculée par la méthode

des moindres carrés.

$$\text{Pour } H < 0 \quad , \quad T(H) = 2.77 \cdot 10^{-27} H^3 - 2.01 \cdot 10^{-17} H^2 + 3.13 \cdot 10^{-7} H + 933 \quad K.$$

$$\text{Lorsque } H > 9.53 \cdot 10^8 J/m^3 \quad , \quad T(H) = 2.345 \cdot 10^{-27} H^3 + 2.563 \cdot 10^{-18} H^2 + 3.744 \cdot 10^{-7} H + 571.7 \quad K.$$

$$\text{Lorsque } 0 \leq H \leq \rho L = 9.53 \cdot 10^8 J/m^3 \quad , \quad T(H) = 933.15 \quad K \quad .$$

L'intégration de la conductivité thermique en fonction de la température nous donne une relation entre la transformée de Kirchhoff de la conductivité thermique et T.

$$\text{Pour } 250 \leq T < 933.15 \quad K \quad , \quad \theta(T) = -2.88 \cdot 10^{-5} T^3 + 3.13 \cdot 10^{-2} T^2 + 226T - 2.15 \cdot 10^5 \quad W/m$$

$$\text{et pour } 933.15 < T \leq 1500 \quad K \quad , \quad \theta(T) = -4.10 \cdot 10^{-6} T^3 + 2.95 \cdot 10^{-2} T^2 + 46.5T - 6.57 \cdot 10^4 \quad W/m \quad .$$

$$\text{Lorsque } T = 933.15 \quad K \quad , \quad \theta(T) = 0 \quad .$$

Puisque H définit d'une façon unique T, nous pouvons exprimer la transformée de Kirchhoff en fonction de l'enthalpie. Pour chaque valeur de l'enthalpie, nous calculons la température correspondante; de la valeur de la température ainsi calculée, nous évaluons la transformée de Kirchhoff de la conductivité thermique. Pour des valeurs de températures comprises entre -10°C et 1 000°C, nous avons fait passer deux polynômes: une équation parabolique et une équation linéaire.

$$\text{Pour } H < 0 \quad J/m^3 \quad , \quad \theta(H) = -8.938 \cdot 10^{-15} H^2 + 6.605 \cdot 10^{-5} H - 152.0 \quad W/m \\ \approx 8.33 \cdot 10^{-5} H + 5400 \quad W/m.$$

$$\text{Pour } H > 9.53 \cdot 10^8 \quad J/m^3 \quad , \quad \theta(H) = 3.768 \cdot 10^{-15} H^2 + 2.773 \cdot 10^{-5} H - 2.978 \cdot 10^4 \quad W/m \\ \approx 3.851 \cdot 10^{-5} H - 37200 \quad W/m.$$

$$\text{Pour } 0 \leq H \leq \rho L = 9.53 \cdot 10^8 J/m^3 \quad , \quad \theta(H) = 0 \quad .$$

La figure 3 représente la variation de la transformée de Kirchhoff en fonction de l'enthalpie, en considérant une variation quadratique. La figure 4 permet de comparer la variation linéaire à la variation quadratique pour le solide alors que la figure 5 permet de faire la même comparaison pour le liquide.



CHAPITRE 2  
RÉSOLUTION DE L'ÉQUATION ET PREMIERS RÉSULTATS

## CHAPITRE 2

## RÉSOLUTION DE L'ÉQUATION ET PREMIERS RÉSULTATS

2.1 Choix des fonctions de pondération et d'interpolation

L'équation d'énergie est intégrée par la méthode des éléments finis. Les équations élémentaires sont formulées par la méthode des résidus pondérés utilisant une formulation de type Galerkin comme choix de fonctions de pondération. On utilise des éléments isoparamétriques triangulaires linéaires à trois noeuds pour l'interpolation de l'enthalpie et de la transformée de Kirchhoff.

On retrouve à la figure 6 le vecteur des fonctions d'interpolation  $\{N\}$  de l'enthalpie et de la transformée de Kirchhoff, donné en fonction de l'élément de référence choisi (Dhatt et Touzot 1981).

2.2 Formulation intégrale faible

La méthode de Galerkin requiert que (Huebner et Thornton 1982)

$$\{W(t)\} = \sum_{(e)} \{W(t)\}^{(e)} = \{0\}$$

où

$$\{W(t)\}^{(e)} = \int_{\Omega^{(e)}} \{N\} \left( \frac{\partial H}{\partial t} - \nabla^2 \theta \right) d\Omega .$$

En intégrant par parties,

$$\{W(t)\}^{(e)} = \int_{\Omega^{(e)}} \left( \{N\} \frac{\partial H}{\partial t} + [B]^T \left[ \frac{\partial \theta}{\partial x} \quad \frac{\partial \theta}{\partial y} \right]^T \right) d\Omega - \oint_{\Gamma^{(e)}} \{N\} \frac{\partial \theta}{\partial n} d\Gamma$$

avec

$$[B]^T = \left[ \left\{ \frac{\partial N}{\partial x} \right\} \quad \left\{ \frac{\partial N}{\partial y} \right\} \right] .$$

De la définition de  $\theta$  et de la loi de Fourier, nous pouvons montrer que l'intégrale sur la frontière est nulle lorsque la frontière est isolée ( $\bar{q} = 0$ ). En effet,

$$\frac{\partial \theta}{\partial n} = \vec{\nabla} \theta \cdot \vec{n} = \left( k \frac{\partial T}{\partial x} \vec{i} + k \frac{\partial T}{\partial y} \vec{j} \right) \cdot \vec{n} = -\bar{q} \cdot \vec{n} = 0 \quad .$$

Puisque

$$H^{(e)}(x, y, t) = \{N(x, y)\}^T \{H(t)\}^{(e)}$$

et

$$\theta^{(e)}(x, y, t) = \{N(x, y)\}^T \{\theta(t)\}^{(e)}$$

alors

$$\{W(t)\}^{(e)} = \int_{\Omega^{(e)}} (\{N\}\{N\}^T \left\{ \frac{dH}{dt} \right\}^{(e)} + [B]^T [B] \{\theta\}^{(e)}) d\Omega \quad .$$

Soient:

$$[C]^{(e)} = \int_{\Omega^{(e)}} \{N\}\{N\}^T d\Omega = \frac{\Delta}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$[K']^{(e)} = \int_{\Omega^{(e)}} [B]^T [B] d\Omega$$

$$= \frac{1}{4\Delta} \begin{bmatrix} (y_2 - y_3)^2 + (x_2 - x_3)^2 & ; & (y_2 - y_3)(y_3 - y_1) + (x_2 - x_3)(x_3 - x_1) & ; \\ (y_2 - y_3)(y_3 - y_1) + (x_2 - x_3)(x_3 - x_1) & ; & (y_3 - y_1)^2 + (x_3 - x_1)^2 & ; \\ (y_2 - y_3)(y_1 - y_2) + (x_2 - x_3)(x_1 - x_2) & ; & (y_1 - y_2)(y_3 - y_1) + (x_1 - x_2)(x_3 - x_1) & ; \\ & & (y_2 - y_3)(y_1 - y_2) + (x_2 - x_3)(x_1 - x_2) & \\ & & (y_1 - y_2)(y_3 - y_1) + (x_1 - x_2)(x_3 - x_1) & \\ & & (y_1 - y_2)^2 + (x_1 - x_2)^2 & \end{bmatrix} \quad ,$$

$\Delta$  étant l'aire de l'élément réel (e).

Avec ces définitions,

$$\{W(t)\}^{(e)} = [C]^{(e)} \left\{ \frac{dH}{dt} \right\}^{(e)} + [K']^{(e)} \{\theta(t)\}^{(e)} \quad .$$

Si  $\{\theta\}^{(r)} = \{\theta_i\}^{(r)} = \{a_0^i + a_1^i H_i + a_2^i H_i^2\}^{(r)}$  ,  $i = 1, 2, 3$

$$= \begin{Bmatrix} a_0^1 \\ a_0^2 \\ a_0^3 \end{Bmatrix}^{(e)} + \begin{bmatrix} a_1^1 & 0 & 0 \\ 0 & a_1^2 & 0 \\ 0 & 0 & a_1^3 \end{bmatrix}^{(r)} \{H\}^{(r)} + \begin{bmatrix} a_2^1 & 0 & 0 \\ 0 & a_2^2 & 0 \\ 0 & 0 & a_2^3 \end{bmatrix}^{(e)} \{H^2\}^{(r)}$$

alors  $[K']^{(r)}\{\theta\}^{(r)} = [K]^{(r)}\{H\}^{(r)} + [G]^{(r)}\{H^2\}^{(r)} - \{R\}^{(r)}$

où

$$[K]^{(r)} = [K']^{(e)} \begin{bmatrix} a_1^1 & 0 & 0 \\ 0 & a_1^2 & 0 \\ 0 & 0 & a_1^3 \end{bmatrix}^{(r)}$$

$$[G]^{(e)} = [K']^{(e)} \begin{bmatrix} a_2^1 & 0 & 0 \\ 0 & a_2^2 & 0 \\ 0 & 0 & a_2^3 \end{bmatrix}^{(e)}$$

$$\{R\}^{(e)} = -[K']^{(e)} \begin{Bmatrix} a_0^1 \\ a_0^2 \\ a_0^3 \end{Bmatrix}^{(e)}$$

Donc  $\{W(t)\}^{(e)} = [C]^{(e)}\left\{\frac{dH}{dt}\right\}^{(e)} + [K]^{(e)}\{H(t)\}^{(e)} + [G]^{(e)}\{H^2(t)\}^{(e)} - \{R\}^{(e)}$ .

Lorsqu'une variation linéaire de  $\theta(H)$  est utilisée en guise de simplification, les termes  $a_2^i$  sont tous nuls. D'où:

$$\{W(t)\}^{(e)} = [C]^{(e)}\left\{\frac{dH}{dt}\right\}^{(e)} + [K]^{(e)}\{H(t)\}^{(e)} - \{R\}^{(e)}$$

L'assemblage des matrices élémentaires

$$\{W(t)\} = \sum_{(e)} \{W(t)\}^{(e)} = \{0\}$$

conduit au système d'équations différentielles

$$[C]\left\{\frac{dH}{dt}\right\} + [K]\{H\} + [G]\{H^2\} = \{R\} \quad (5)$$

pour une variation quadratique de  $\theta(H)$ , et

$$[C]\left\{\frac{dH}{dt}\right\} + [K]\{H\} = \{R\} \quad (6)$$

pour une variation linéaire de  $\theta(H)$ . La matrice  $[C]$  est appelée matrice de capacité thermique,  $[K]$  et  $[G]$  matrices de conductivité thermique et  $\{R\}$  vecteur des sollicitations.

### 2.3 Modification de la matrice de capacité thermique

L'application de la méthode de Galerkin produit une matrice de capacité thermique  $[C]$  bandée. La transformation de cette matrice bandée en une matrice diagonale peut offrir deux grands avantages. Le premier est une diminution de l'espace requis pour conserver cette matrice; le second, est la facilité de l'inverser.

Nous utiliserons un procédé simple, déjà utilisé par plusieurs auteurs, pour diagonaliser  $[C]$  (Zienkiewicz 1977, Lapidus et Pinder 1982 et Huebner et Thornton 1982). Les éléments  $c_{ij}$  de la matrice de capacité thermique sont remplacés par les éléments  $c'_{ii}$  que l'on définit ainsi:

$$c'_{ii} = \sum_{j=1}^n c_{ij} \quad , \quad i = 1, \dots, n \quad .$$

Dans la littérature, une matrice ainsi modifiée est appelée matrice groupée alors qu'une matrice bandée (sans modification) est appelée matrice consistante.

Comme il est souvent fait en méthodes numériques, pour justifier l'utilisation d'une matrice de capacité thermique groupée, nous ferons une comparaison des résultats. Par définition, lorsque l'on

résoudra le système d'équations (5) où  $\theta$  varie Quadratiquement en fonction de l'enthalpie avec une matrice de capacité thermique consistante (Bandée), on dira que l'on a choisi l'option BQ. Lorsque ce système sera résolu avec une matrice de capacité thermique groupée (Diagonale), on parlera de l'option DQ. D'une façon tout à fait similaire, lorsque l'on résoudra le système d'équations (6) où  $\theta$  varie Linéairement avec l'enthalpie, on parlera de l'option BL lorsque la matrice de capacité thermique sera Bandée et de l'option DL lorsque  $[C]$  sera groupée (Diagonale).

#### 2.4 Résolution numérique

Différents essais ont été effectués avec la géométrie de la figure 7a, représentant une demi-section d'un bain. Le volume de contrôle est isolé sur les côtés et dans le fond. La température de surface est maintenue à 900°C. Initialement l'aluminium liquide est à 700°C et l'aluminium solide à 30°C.

Puisque les calculs ne se font pas avec le champ de température mais bien avec un champ d'enthalpie, le programme transforme les températures en enthalpies dès le début de l'exécution. A la toute fin, il reconvertit les enthalpies en températures. La figure 7b présente la condition initiale pour le début des calculs. Le plateau inférieur est à la valeur  $H(30^\circ\text{C})$  alors que le plateau supérieur est à la valeur  $H(700^\circ\text{C})$ , une valeur énergétique de beaucoup supérieure au premier. En fait, cette "énorme" différence entre ces valeurs d'enthalpie oblige le simulateur à devoir supporter une très forte contrainte sur un tout

petit déplacement. Le dernier saut correspond au changement d'enthalpie pour passer de 700°C à 900°C, la condition frontière imposée.

Le domaine rectangulaire est subdivisé en  $m$  rangées de noeuds sur  $\ell$  colonnes,  $m$  étant inférieur ou égal à  $\ell$ , le nombre total de noeuds  $n$  étant égal au produit  $m \times \ell$ . La figure 8a montre une partie du maillage de ces  $n$  noeuds par des éléments triangulaires linéaires de type Lagrange (figure 6). Comme le montre la figure 8a, le noeud  $r$  est en relation directe avec seulement six autres noeuds, c'est-à-dire les noeuds  $r-m-1$ ,  $r-m$ ,  $r-1$ ,  $r+1$ ,  $r+m$  et  $r+m+1$ . Les matrices  $[C]$ ,  $[K]$  et  $[G]$  des équations (5) et (6) ne peuvent donc avoir au plus que sept diagonales non nulles. En effet, de part et d'autre des trois diagonales principales, il y a  $(m-2)$  diagonales de "0" (figures 8b). En tout,  $2(m-2)n$  éléments nuls à conserver inutilement si nous conservons une structure bandée conventionnelle. Nous réécrivons donc ces matrices selon la représentation  $n \times 7$  de la figure 8b. L'appendice 2 donne l'algorithme de multiplication de ces matrices  $n \times 7$  par un vecteur colonne.

Lorsque la matrice de capacité thermique est modifiée (option DL ou DQ), son inverse est conservé dans un vecteur  $n \times 1$ . Les équations (5) et (6) sont multipliées à gauche par cette matrice de capacité thermique inversée. L'intégration se fait alors par un Runge-Kutta d'ordre 5 (James, Smith et Wolford 1977).

Lorsque la matrice de capacité thermique est consistante (option BL ou BQ), son inverse n'est pas calculé. Le système d'équations différentielles (5) pour l'option BQ est solutionné par la méthode de prédiction-

correction de Hamming que l'on retrouve dans Carnahan, Luther et Wilkes (1969). La correction est appliquée jusqu'à ce que l'écart relatif maximum soit inférieur à 2%. On trouvera à l'appendice 1 le calcul des trois premiers résultats utilisés pour la prédiction. Le système (6) pour l'option BL est intégré par la méthode d'Euler semi-implicite de Crank-Nicolson (Dhatt et Touzot 1981). Dans les deux cas (BL ou BQ), les conditions aux limites sont introduites par la méthode du terme diagonal dominant telle que décrite dans Dhatt et Touzot (1981) et le système est solutionné par la méthode d'élimination de Gauss pour matrices bandées dont l'algorithme est donné dans Brebbia et Ferrante (1978).

Le domaine a été couvert par  $20 \times 37 = 740$  noeuds et par  $8 \times 14 = 112$  noeuds. Pour chacune de ces discrétisations, l'équation d'énergie en conduction a été intégrée par les options BQ, DQ, BL et DL.

## 2.5 Analyse des résultats

La transformée de Kirchhoff variant quadratiquement avec l'enthalpie, l'équation (5) décrit plus fidèlement le problème que l'équation (6). Aucune simplification n'étant faite sur la matrice de capacité thermique lorsque l'on intègre (5) avec l'option BQ, nous l'utiliserons comme référence.

L'équation d'énergie étant solutionnée par une formulation enthalpique, il est intéressant de voir comment la valeur d'enthalpie initiale du solide est augmentée tout au long de la simulation. Les figures 11 et



12 montrent des surfaces enthalpiques calculées avec 740 noeuds par l'option BQ. Nous observons une augmentation de la surface du solide lors des 15 premières minutes. En fait, la bande de liquide se trouvant à l'interface solide-liquide fournit beaucoup d'énergie au solide se trouvant initialement à 630°C en dessous de la température de fusion. Cette baisse d'énergie produit une solidification partielle du liquide. La figure 13 présente l'évolution de la frontière mobile calculée par l'option BQ.

La figure 14 permet d'observer l'établissement du gradient de température dans le bain. Les trois figures suivantes permettent de comparer les résultats obtenus par les quatre options. Ces figures montrent clairement que la modification de la matrice de capacité thermique n'a pas influencé d'une façon significative la qualité des résultats. Par contre, nous remarquons une nette distinction entre les résultats obtenus par les options BQ et DQ avec ceux obtenus par les options BL et DL. Par exemple, à la figure 16, les isothermes de 675 à 725°C sont beaucoup plus rapprochées dans les cas BL et DL que dans les cas BQ et DQ. Nous pouvons expliquer ceci en étudiant la figure 9. Cette figure est en quelque sorte un agrandissement des droites des figures 4 et 5, près de l'origine. Lorsque l'enthalpie passe d'une valeur positive à une valeur négative, la transformée de Kirchhoff de la conductivité thermique subit une légère augmentation. Puisque le gradient de la transformée de Kirchhoff n'est autre que le flux de chaleur, ceci produit un transfert de chaleur du solide vers le liquide, ce qui est physiquement impossible. Cette anomalie est d'origine

mathématique et n'a aucune signification physique; ce flux est produit artificiellement par le fait que la fonction linéaire reliant la transformée de Kirchhoff à l'enthalpie n'est pas continue. Le flux de chaleur ainsi créé tend à faire diminuer le transfert de chaleur naturel se dirigeant d'un point chaud vers un point froid, près de l'interface solide-liquide. En effet, dans la partie du solide se trouvant près de cette interface, il s'est produit une inversion dans le gradient de température. Cette inversion n'est malheureusement pas visible sur les figures 15 à 17. Les figures 18 à 20 permettent de tirer les mêmes conclusions pour un nombre de noeuds de beaucoup inférieur. Il est intéressant de remarquer la corrélation entre les résultats obtenus en utilisant 740 noeuds et ceux en utilisant 112 noeuds.

Tout au long de la simulation, le pas de temps  $\Delta t$  maximum que nous pouvons utiliser pour qu'il n'y ait pas d'oscillations varie. Au début de la simulation, lorsque la contrainte enthalpique est très grande, il est de l'ordre de 0,1 s. Vers la fin de la simulation, il est de l'ordre de 0,5 s. Ces valeurs sont tirées de différents tests numériques faits en parallèle avec la simulation. Lorsque le pas pouvait être augmenté, la valeur  $\Delta t$  du simulateur était augmentée. La résolution de l'équation (5) par l'option BQ avec 740 noeuds a pris 3 jours 16 heures CPU sur un ordinateur VAX-11/785. Le tableau 5 permet de comparer le temps de calcul des différentes options sur un ordinateur VAX-11/780. Le programme est écrit en Fortran-77 et

## Option BQ

temps (min)	$\bar{T}_{\text{liq}} (^{\circ}\text{C})$	$\bar{T}_{\text{sol}} (^{\circ}\text{C})$	% de noeuds dans le solide
0	714.7	30.0	32.2
1	736.0	238.9	40.5
5	748.6	370.3	45.5
10	759.2	445.3	47.0
15	767.8	497.4	47.8
20	773.2	534.5	47.7
30	779.4	582.4	45.4
40	782.9	613.0	42.3
50	785.4	632.7	38.9
60	787.8	645.1	35.7
70	789.6	652.4	32.2
80	791.1	656.2	28.6
90	793.0	658.1	25.4
100	794.6	658.9	22.2
110	796.3	659.1	19.2
120	797.9	659.2	16.2
130	800.0	659.3	13.8
140	803.2	659.3	12.0
150	805.4	659.3	9.7
160	807.8	659.4	7.6
170	811.3	659.5	6.2
180	814.5	659.5	4.6
190	817.5	659.6	2.8
200	822.1	659.6	1.9
210	826.7	660.0	0.5
220	834.0	---	---
230	841.9	---	---

Tableau 1. Température moyenne du métal solide et du métal liquide. Option BQ (740 noeuds).

## Option DQ

temps (min)	$\bar{T}_{\text{liq}} (^{\circ}\text{C})$	$T_{\text{sol}} (^{\circ}\text{C})$	% de noeuds dans le solide
0	714.7	30.0	32.2
1	732.5	238.8	40.7
5	748.0	369.8	45.5
10	758.2	444.7	47.0
15	767.1	496.5	47.7
60	789.1	645.8	36.1
90	794.1	659.1	25.7
120	799.4	660.0	16.9
135	802.9	660.0	13.4
150	806.4	660.0	10.1
165	810.3	660.0	7.3
180	814.9	660.0	4.9
195	819.8	660.0	2.6
210	826.2	660.0	0.8
215	829.3	660.0	0.5
220	832.4	---	---
225	836.6	---	---
230	840.5	---	---

Tableau 2. Température moyenne du métal solide et du métal liquide. Option DQ (740 noeuds).

## Option BL

temps (min)	$\bar{T}_{\text{liq}} (^{\circ}\text{C})$	$\bar{T}_{\text{sol}} (^{\circ}\text{C})$	% de noeuds dans le solide
0	714.7	30.0	32.2
1	736.4	237.2	40.4
5	749.2	368.5	45.4
10	760.4	443.8	46.9
15	769.1	495.6	47.4
20	775.2	533.8	47.4
25	780.9	563.6	47.4
60	799.3	642.3	37.7
90	807.1	652.0	28.2
120	811.7	652.9	20.5
130	813.3	652.8	18.4
140	814.3	651.8	16.2
150	816.3	653.2	14.2
160	819.0	654.8	12.6
170	820.3	654.5	10.7
180	821.2	649.3	8.9
190	824.2	654.7	7.7
200	825.8	654.5	6.1
210	827.9	651.9	4.7
220	829.7	648.9	3.4
230	831.1	650.1	2.0
235	833.3	651.6	1.6
240	835.0	656.3	1.2
245	837.1	657.6	0.7

Tableau 3. Température moyenne du métal solide et du métal liquide. Option BL (740 noeuds).

## Option DL

temps (min)	$\bar{T}_{\text{liq}}(^{\circ}\text{C})$	$\bar{T}_{\text{sol}}(^{\circ}\text{C})$	% de noeuds dans le solide
0	714.7	30.0	32.2
1	732.6	237.5	40.5
5	748.4	368.3	45.4
60	800.0	643.2	37.8
90	807.9	652.4	28.8
120	813.3	654.0	21.2
150	817.2	653.5	14.6
170	821.5	655.3	11.2
180	822.7	654.0	9.5
190	825.1	655.9	8.1
200	826.2	653.6	6.5
210	829.1	654.5	5.4
220	831.9	657.6	4.2
230	834.1	657.1	3.1
240	836.5	656.1	2.0

Tableau 4. Température moyenne du métal solide et du métal liquide. Option DL (740 noeuds).

Tableau 5. Temps de calcul moyen pour solutionner un pas de temps  $\Delta t$  de l'équation d'énergie sur un ordinateur VAX-11/780. (Equations 5 et 6).

(a) En utilisant 740 noeuds:

Option	temps (s)
BQ	14.58
DQ	1.59
BL	5.14
DL	1.30

(b) En utilisant 112 noeuds:

Option	temps (s)
BQ	0.61
DQ	0.23
BL	0.25
DL	0.19

compilé avec l'option d'optimisation.

La modification de la matrice de capacité thermique étant un procédé qui diminue de beaucoup le temps de calcul sans trop perturber la solution, il s'avère profitable d'utiliser l'option DQ pour solutionner l'équation d'énergie. L'option DL pourrait être utilisée si nous subdivisions la variation quadratique de la transformée de Kirchhoff pour  $H$  négatif en deux ou plusieurs droites au lieu d'une seule. Cependant, pour un temps de calcul légèrement supérieur, nous avons préféré conserver la représentation quadratique avec l'option DQ.



## CHAPITRE 3

### INFLUENCE DE LA CONVECTION NATURELLE

## CHAPITRE 3

## INFLUENCE DE LA CONVECTION NATURELLE

Les chapitres précédents ont été basés sur l'hypothèse que le fluide (l'aluminium liquide) était en équilibre mécanique sans se trouver pour autant en équilibre thermique. La condition d'équilibre mécanique

$$\vec{\nabla} p = \rho \vec{g}$$

peut ne pas être remplie si la densité du fluide  $\rho$  est une fonction de la température  $T_i(x, y, t)$  comme c'est le cas dans ce travail. En effet, l'équilibre peut être perturbé entraînant ainsi des courants ayant tendance à brasser le fluide pour y établir une température constante.

### 3.1 Considérations générales

Au taux d'accumulation d'énergie interne de l'équation d'énergie telle que considérée précédemment,

$$\frac{\partial H}{\partial t} = \nabla^2 \theta ,$$

doit être ajouté un terme d'advection de l'énergie lorsque l'équilibre mécanique du fluide est rompu. L'équation d'énergie devient donc

$$\frac{DH}{Dt} = \nabla^2 \theta \quad (7)$$

avec la dérivée matérielle  $\frac{D}{Dt}$  ainsi définie:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} .$$

Les équations du mouvement permettent de calculer le champ de vitesse à utiliser pour évaluer l'équation d'énergie. Pour un fluide Newtonien incompressible soumis à un champ de pesanteur uniforme, l'équation de continuité en deux dimensions s'écrit:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (8)$$

et les équations de Navier-Stokes sont

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \mu \nabla^2 u$$

dans la direction x et

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \mu \nabla^2 v - \rho g$$

dans la direction y. En utilisant l'approximation de Boussinesq où (Gray et Giorgini 1975)

- i) la densité est supposée constante sauf dans le terme de la poussée d'Archimède,
  - ii) toutes les autres propriétés du fluide sont supposées constantes,
- il est possible de réécrire ces équations différemment. La densité apparaissant dans la force d'Archimède est approchée par

$$\rho \approx \rho_0 - \rho_0 \beta (T - T_0) = \rho_0 [1 - \beta (T - T_0)]$$

où  $\rho_0$  est la densité évaluée à la température  $T_0$  et  $\beta$  est un coefficient de dilatation thermique. Ce coefficient variant très peu avec la température, il sera considéré constant. La température de référence  $T_0$  sera la température moyenne du liquide,  $T_{\text{moy}}$ , calculée à chaque pas de temps,  $\Delta t$ . Les équations de Navier-Stokes peuvent donc être réécrites comme suit:

$$\frac{Du}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + \nu \nabla^2 u \quad , \quad \frac{Dv}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial y} + \nu \nabla^2 v - [1 - \beta (T - T_0)] g$$

avec

$$\nu = \mu / \rho_0 \quad .$$

La pression totale est la somme de la pression statique et de la pression dynamique

$$p = p_{\text{statique}} + p_{\text{dynamique}} = p_s + p_d$$

où la pression statique est de la forme

$$p_s = cte - \rho_0 g y$$

D'où:

$$\frac{\partial p}{\partial x} = \frac{\partial p_d}{\partial x} \quad \text{et} \quad \frac{\partial p}{\partial y} = -\rho_0 g + \frac{\partial p_d}{\partial y}$$

En reportant ces valeurs dans les équations de Navier-Stokes, nous obtenons finalement

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho_0} \frac{\partial p_d}{\partial x} + \nu \nabla^2 u \quad (9a)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho_0} \frac{\partial p_d}{\partial y} + \nu \nabla^2 v + \beta(T - T_0)g \quad (9b)$$

Maintenant nous devons intégrer (7), les vitesses étant fournies par les équations (9) soumises à la contrainte de l'équation (8).

### 3.2 Détails du calcul

L'intégration des équations (7), (8) et (9) se fait selon l'algorithme suivant. On l'utilise à chaque pas de temps  $\Delta t$ .

<i>Début</i>	(temps = t)
calcul de $H_{t+\Delta t}$ à partir de $H_t$ et $V_t$	(équation (7))
conversion de $H_{t+\Delta t}$ en $T_{t+\Delta t}$	(figure 2)
calcul de $V_{t+\Delta t}$ à partir de $V_t$ et $T_{t+\Delta t}$	(équations (8) et (9))
<i>Fin</i>	(temps = t + $\Delta t$ )

$H$  représente l'enthalpie,  $T$  la température et  $v$  les vitesses. Ce couplage entre l'équation d'énergie (7) et les équations du mouvement (8) et (9) doit donner de bons résultats si

- i) le pas de temps d'intégration  $\Delta t$  est petit et
- ii) les vitesses et les enthalpies ne changent pas rapidement avec le temps.

### 3.2.1 Détails sur la résolution de l'équation d'énergie

Le terme d'advection de l'énergie

$$\rho c \left( u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right) = u \rho c \frac{\partial T}{\partial x} + v \rho c \frac{\partial T}{\partial y} = u \frac{\partial H}{\partial x} + v \frac{\partial H}{\partial y}$$

peut être réécrit à l'aide des fonctions d'interpolation:

$$\{N\}^T \{u\} \frac{\partial}{\partial x} \{N\}^T \{H\} + \{N\}^T \{v\} \frac{\partial}{\partial y} \{N\}^T \{H\} .$$

Il est donc possible de reformuler l'équation d'énergie où la transformée de Kirchhoff varie quadratiquement avec l'enthalpie comme suit:

$$[C] \left\{ \frac{dH}{dt} \right\} + ([K] + [K_V]) \{H\} + [G] \{H^2\} = \{R\}$$

avec

$$[K_V] = \sum_{(c)} [K_V]^{(c)}$$

$$[K_V]^{(c)} = \int_{\Omega^{(c)}} \{N\} \{N\}^T \left( \{u\}^{(c)} \left\{ \frac{\partial N}{\partial x} \right\}^T + \{v\}^{(c)} \left\{ \frac{\partial N}{\partial y} \right\}^T \right) d\Omega$$

$$\begin{aligned}
&= \frac{1}{24} \begin{bmatrix} (y_2 - y_3)(2u_1 + u_2 + u_3) + (x_3 - x_2)(2v_1 + v_2 + v_3) \\ (y_2 - y_3)(u_1 + 2u_2 + u_3) + (x_3 - x_2)(v_1 + 2v_2 + v_3) \\ (y_2 - y_3)(u_1 + u_2 + 2u_3) + (x_3 - x_2)(v_1 + v_2 + 2v_3) \end{bmatrix} \\
&\quad \begin{bmatrix} (y_3 - y_1)(2u_1 + u_2 + u_3) + (x_1 - x_3)(2v_1 + v_2 + v_3) \\ (y_3 - y_1)(u_1 + 2u_2 + u_3) + (x_1 - x_3)(v_1 + 2v_2 + v_3) \\ (y_3 - y_1)(u_1 + u_2 + 2u_3) + (x_1 - x_3)(v_1 + v_2 + 2v_3) \end{bmatrix} \\
&\quad \begin{bmatrix} (y_1 - y_2)(2u_1 + u_2 + u_3) + (x_2 - x_1)(2v_1 + v_2 + v_3) \\ (y_1 - y_2)(u_1 + 2u_2 + u_3) + (x_2 - x_1)(v_1 + 2v_2 + v_3) \\ (y_1 - y_2)(u_1 + u_2 + 2u_3) + (x_2 - x_1)(v_1 + v_2 + 2v_3) \end{bmatrix}
\end{aligned}$$

$u_i$  = la composante horizontale de la vitesse au noeud  $i$  de l'élément (e)

$v_i$  = la composante verticale de la vitesse au noeud  $i$  de l'élément (e).

### 3.2.2. Détails sur la résolution des équations du mouvement

Le logiciel SOLA est utilisé pour intégrer les équations (8) et (9). L'algorithme de base de ce logiciel est présenté dans le rapport de Hirt, Nichols et Romero (1975); il ne sera pas répété ici. Cependant, une modification lui a été apportée pour tenir compte de la force de corps (poussée d'Archimède) dans chaque cellule, par l'intermédiaire du terme  $v_{i,j}^{n+1}$ . C'est en effet dans le calcul de ce terme qu'on doit faire intervenir l'approximation de Boussinesq. La température moyenne du liquide  $T_{\text{moy}}^{n+1}$  au temps  $n+1$  est d'abord calculée. La valeur de la température  $T_{i,j}^{n+1}$  de la cellule (i,j) est fournie par interpolation des valeurs enthalpiques. En portant la différence de températures  $T_{i,j}^{n+1} - T_{\text{moy}}^{n+1}$  dans l'équation (9b) et en écrivant cette équation comme dans le rapport de Hirt, Nichols et Romero, nous obtenons

$$v_{i,j}^{n+1} = v_{i,j}^n + \delta t \left\{ \frac{1}{\delta y} (P_{i,j}^n - P_{i,j+1}^n) + \beta g (T_{i,j}^{n+1} - T_{\text{moy}}^{n+1}) - FVX - FVY + VISY \right\}$$

Les termes FVX, FVY et VISY sont développés à l'annexe 2.

L'accélération gravitationnelle  $g$  est prise comme étant  $9.81 \text{ m/s}^2$ . La valeur choisie de  $\beta$  est celle qui génère une valeur de  $\rho$  calculée par

$$\rho \approx \rho_0 - \rho_0 \beta (T - T_0) = \rho_0 [1 - \beta (T - T_0)]$$

(pour  $T_0$  compris entre  $660^\circ\text{C}$  et  $900^\circ\text{C}$ )

la plus près possible de la valeur exacte de  $\rho$  donnée par

$$\rho(T) = -0.3935T + 2756.8 \quad \text{kg/m}^3 \quad , \quad 933.15 \leq T \leq 1500 \quad \text{K}$$

la valeur ainsi trouvée est:

$$\beta = 1.65 \cdot 10^{-4} \quad \text{K}^{-1} \quad .$$

A  $700^\circ\text{C}$ , la valeur de la viscosité absolue est de  $2.9 \times 10^{-3} \text{ Pa s}$  (Lyon 1952) et la densité est de  $2\,374 \text{ kg/m}^3$ . La valeur de la viscosité cinématique  $\nu$  que nous utiliserons sera

$$\frac{2.9 \times 10^{-3} \text{ Pa s}}{2\,374 \text{ kg/m}^3} = 1.22 \times 10^{-6} \text{ m}^2/\text{s} \quad .$$

Nous supposerons cette valeur constante, même si en réalité elle varie quelque peu avec la température.

Différents coefficients de différentiation en amont  $\alpha$  ainsi que différents facteurs de relaxation  $\omega$  ont été comparés. Ces coefficients sont utilisés comme entrées pour SOLA. Le coefficient de différentiation en amont permet une meilleure stabilité numérique alors que le facteur de relaxation fait accélérer la convergence. Voici les valeurs qui ont donné les meilleures convergences lors des simulations:

$$\omega = 1.8 \quad \alpha = 0.3 \quad .$$

### 3.2.3 Couplage des sous-modèles éléments finis et différences finies.

La résolution des équations (8) et (9) par différences finies requiert une subdivision du domaine d'intégration en cellules. La figure 10 montre comment seront construites ces cellules à partir des éléments utilisés pour solutionner l'équation (7). Deux éléments (k) et (l) bornés par les quatre noeuds r, r+1, r+m, r+m+1 délimiteront une cellule.

Lorsque nous intégrerons l'équation (7) en éléments finis, nous aurons besoin des composantes des vitesses aux noeuds des éléments (k) et (l). La composante u au noeud r se calculera ainsi:

$$u_r = \frac{u_{i-1,j} + u_{i-1,j-1}}{2} \quad .$$

La composante v au noeud r se calculera de la même façon:

$$v_r = \frac{v_{i-1,j-1} + v_{i,j-1}}{2} \quad .$$

Les températures à l'intérieur d'une cellule sont calculées grâce aux fonctions d'interpolation. Les pressions p étant utilisées uniquement dans les équations (9), il sera donc inutile de calculer la pression aux noeuds des éléments finis.

### 3.3 Résolution numérique

Une fois de plus, nous utiliserons la géométrie et les conditions initiales de la figure 7a. Le domaine d'intégration sera couvert par  $20 \times 37 = 740$  noeuds définissant  $2(19 \times 36) = 1368$  éléments triangulaires linéaires. L'équation d'énergie (équation 7) sera résolue à l'aide de



l'option DQ. Les équations du mouvement (équations 8 et 9) seront intégrées par le logiciel SOLA en différences finies. Sur chaque pas d'intégration  $\Delta t$ , SOLA itérera jusqu'à ce que l'équation de continuité (8) converge à une valeur inférieure à  $10^{-4} \text{ s}^{-1}$ .

### 3.4 Analyse des résultats

Comme il a été fait dans le chapitre sur la conduction, nous présentons à la figure 21 quelques surfaces enthalpiques. Il est intéressant d'observer que le sous-modèle en éléments finis utilisé pour intégrer (7) supporte bien le champ de vitesse. Le champ enthalpique ne possède aucune oscillation numérique (tout comme le modèle non convectif) tout au long de la simulation. Une fois encore la surface du solide prend un peu d'expansion lors des premières minutes de chauffage. Cette fois-ci cependant, l'expansion est moins importante dans le bas du bain (loin de la source de chaleur) qu'en conduction pure. C'est là une influence directe et prévisible de la convection naturelle. En effet, c'est justement dans cette zone qu'est générée la poussée engendrant le mouvement du fluide.

Considérons la région liquide se trouvant entre l'extrémité gauche du bain et le côté gauche du solide. Cette région étant initialement isotherme ( $700^\circ\text{C}$ ), sa densité initiale se trouvait uniforme. Au tout début de la simulation, l'influence du solide (très froid) produit un gradient thermique horizontal<sup>1</sup> dans le liquide sur quelques centimètres. Ces différences de températures

<sup>1</sup> L'influence du solide est beaucoup plus importante pour cette région que l'influence de la condition frontière imposée (voir les isothermes des figures 23 et 24).

(différences de densité) produisent une force motrice (force de corps). N'ayant que les forces d'inertie et de viscosité pour ralentir le mouvement, cette force est assez grande pour engendrer une circulation dans toute cette région (figure 22). Ce mouvement apporte une énergie supplémentaire à la partie gauche de la frontière mobile, diminuant ainsi la solidification initiale du liquide.

Après 5 minutes de chauffage (figure 23) la région décrite précédemment a diminué en superficie. La surface à  $900^{\circ}\text{C}$  a réchauffé une vingtaine de centimètres de liquide et la solidification a déplacé la frontière mobile encore plus vers la gauche. Le gradient horizontal de température a diminué dû à un refroidissement du liquide produisant ainsi une poussée plus faible que lors des premières secondes (figure 23). Après 30 minutes de chauffage (figure 24) le gradient thermique horizontal est pratiquement nul. En effet le coin inférieur gauche initialement à  $700^{\circ}\text{C}$  est maintenant à environ  $670^{\circ}\text{C}$ . La force de corps est maintenant beaucoup plus faible à la frontière solide-liquide qu'à la première minute de chauffage. Nous remarquons également sur cette figure que le liquide est stratifié horizontalement en température<sup>1</sup>; c'est là un effet de la convection naturelle: un brassage du fluide pour homogénéiser la température. Le liquide est donc progressivement (sans palier) plus dense dans le fond et plus léger sur le dessus. Combinant cet état

<sup>1</sup> Les pentes des isothermes de la partie liquide des figures 24 à 27 sont différentes des pentes des mêmes isothermes calculées en conduction (figures 14 à 17).

stable à la faible force de corps, nous obtenons un mouvement lent avec plusieurs petites zones de circulation indiquant que le fluide tend vers l'équilibre mécanique (figures 24 à 27).

Lorsque l'enthalpie moyenne d'une cellule est inférieure à la chaleur latente de fusion, le sous-modèle en différences finies utilisé pour intégrer (8) et (9) considère que cette cellule est uniquement constituée de solide. Tout au long de la simulation, ces cellules vont devenir des cellules constituées uniquement de liquide. Lorsqu'une ou plusieurs cellules passent de l'état solide à l'état liquide, il apparaît subitement dans le liquide en mouvement une zone immobile, région correspondant à la ou des cellules venant de passer à l'état liquide. Lorsque ceci se produit, le nombre d'itérations utilisé pour que l'équation de continuité soit satisfaite est beaucoup plus important. Ceci produit de grandes variations dans le temps de calcul nécessaire pour calculer un pas de temps  $\Delta t$ . En moyenne, pour intégrer un  $\Delta t$  en convection naturelle par l'option DQ, il a fallu 2,53 s CPU sur un ordinateur VAX-11/780. Un second effet du passage d'une cellule de l'état solide à l'état liquide est une petite variation sur la vitesse moyenne  $\bar{V}$  du fluide, par exemple à 40 et à 70 minutes (tableau 6).

Malgré le fait que la vitesse moyenne du fluide n'est que de quelques millimètres par seconde, l'influence de la convection naturelle n'est vraiment pas négligeable sur le temps requis pour une fusion complète. Il est en effet passé de 220 minutes à 110 minutes. C'est un temps comparable à celui qu'on obtient à partir d'un four à fusion ayant une géométrie similaire à celle de la

temps (min)	$\bar{T}_{\text{liq}}$ (°C)	$\bar{T}_{\text{sol}}$ (°C)	$V_{\text{max}}$ (mm/s)	$\bar{V}$ (mm/s)	% de noeuds dans le solide
0	714.7	30.0	0.0	0.0	32.2
0.5	725.8	193.4	59.6	8.47	38.8
1	732.8	232.5	70.8	12.63	40.0
5	735.6	364.5	42.4	7.97	43.1
10	738.5	445.5	26.6	4.88	43.5
20	744.1	536.9	14.0	3.20	41.8
30	749.2	588.4	10.4	2.67	37.7
40	752.9	620.4	14.7	3.02	32.3
50	758.4	641.3	13.5	2.01	27.8
60	762.4	652.8	14.0	2.66	21.9
70	766.3	657.9	22.5	3.81	16.6
80	770.9	660.0	15.0	2.51	11.5
90	775.5	660.0	10.5	1.38	6.2
100	783.5	660.0	13.1	1.52	3.0
105	788.1	660.0	16.0	1.86	1.1
110	793.8	---	1.3	0.24	---
115	800.3	---	0.3	0.02	---

Tableau 6. Variation temporelle de la température et de la vitesse moyenne du métal liquide calculée par l'option DQ en convection naturelle. (740 noeuds)

figure 7a (Stevens 1986).

Les calculs précédents ont été repris avec une discrétisation de 112 noeuds (figures 28 et 29). Aucun résultat valable n'a été obtenu même en essayant différents facteurs de relaxation et de coefficients de différentiation en amont. En fait, les cellules ainsi produites sont trop grandes.

CONCLUSION

## CONCLUSION

La méthode de l'enthalpie et la transformée de Kirchhoff ont été utilisées simultanément pour solutionner un problème de frontière mobile avec propriétés thermiques variables. Des calculs faits avec une variation linéaire plutôt que quadratique de la transformée de Kirchhoff de la conductivité thermique en fonction de l'enthalpie montrent qu'il est important d'avoir une fonction continue sur tout le domaine d'intégration. Il est également montré qu'une modification de la matrice de capacité thermique permet de diminuer d'une façon appréciable le temps de calcul sans perturber la qualité des résultats.

Pour tenir compte de la convection naturelle du liquide, le logiciel SOLA en différences finies a été couplé au modèle en éléments finis. Le brassage engendré par la convection a produit un mouvement lent du fluide tout en diminuant d'un facteur deux le temps requis pour obtenir une fusion complète.

Les résultats obtenus concordent avec des données expérimentales prises sur un four industriel de dimensions semblables. Ceci nous permet donc de conclure que le découplage entre l'équation d'énergie intégrée par la méthode des éléments finis et les équations du mouvement intégrées par SOLA en différences finies a donné d'excellents résultats.

Le programme construit pour effectuer ces calculs offre la

possibilité de considérer différentes géométries et conditions initiales; cependant, la dimension maximale des cellules ne devrait jamais dépasser 5 cm x 5 cm pour obtenir des résultats satisfaisants. Ce programme se retrouve à l'annexe 1.



## BIBLIOGRAPHIE

## BIBLIOGRAPHIE

- BREBBIA, C.A., FERRANTE, A.J., Computational methods for the solution of engineering problems, London, Pentech Press, 1978, chapitre 2.
- CARNAHAN, B., LUTHER, H.A., WILKES, J.O., Applied numerical methods, John Wiley, 1969, pp 390-392.
- CIVAN, F., SLIEPCEVICH, C.M., "Efficient numerical solution for enthalpy formulation of conduction heat transfer with phase change", International Journal of heat and Mass Transfer, v. 27, numéro 8, 1984, pp 1428-1430.
- COY, W.J., MATEER, R.S., Trans. A.S.M., vol. 58, 1965, pp 99-102.
- CRANK, J., Numerical methods in heat transfer, John Wiley, 1981, chap. 9.
- DHATT, G., TOUZOT, G., Une présentation de la méthode des éléments finis, Les presses de l'Université Laval, 1981, 543p.
- ELLIOT, C.M., OCKENDON, J.R., Weak and variational methods for moving boundary problems, Research notes in Mathematics, Pitman Advanced Publishing Program, 1982, 213p.
- GRAY, D.D., GIORGINI, A., "The validity of the Boussinesq approximation for liquids and gases", Int. J. Heat Mass Transfer, 1975, vol.19, pp 545-551.
- GUPTA, R.S., KUMAR, A., "Treatment of multi-dimensional moving boundary problems by coordinate transformation", International Journal of Heat and Mass Transfer, Vol. 28, numéro 7, 1985, pp 1355-1366.
- HIRT, C.W., NICHOLS, B.D., ROMERO, N.C., "A numerical solution algorithm for transient fluid flows", Los Alamos Scientific Laboratory, Los Alamos, NM, Rapport UC-34 et UC-79d, avril 1975.
- HUEBNER, K.H., THORNTON, E.A., The finite element method for engineers, John Wiley and Sons, seconde édition, 1982, 623p.
- JAMES, M.L., SMITH, G.M., WOLFORD, J.C., Applied numerical methods for digital computation, seconde édition, Harper and Row, 1977, p. 405.
- LAPIDUS, L., PINDER, G.F., Numerical solution of partial differential equations in science and engineering, John Wiley and Sons, 1982, 677p.
- LUNARDINI, V.J., Heat Transfer in Cold Climates, Van Nostrand Reinhold, New York, 1981.

LYON, R., Liquid metals handbook, seconde édition, Washington, D.C., Atomic energy commission and department of the navy, 1952, page 41.

MOSES, A.J., The practicing scientist's handbook, Van Nostrand Reinhold, 1978, pp 204-205.

OCKENDON, J.R., HODGKINS, W.R., Moving boundary problems in heat flow and diffusion, résumés des communications tenus à l'Université d'Oxford du 25 au 27 mars 1974, Les presses de l'Université d'Oxford, 1975, 299p.

ÖZIŞIK, M.N., Heat conduction, Wiley-Interscience publication, 1980, 687p.

PHAM, Q.T., "A fast, unconditionally stable finite-difference scheme for heat conduction with phase change", International Journal of heat and Mass Transfer, vol. 28, numéro 11, 1985, pp 2079-2084.

RALPH III, W.D., KLAUS-JÜRGEN, B., "An efficient algorithm for analysis of non linear heat transfer with phase changes", International Journal of Heat and Mass Transfer, Vol. 18, 1982, pp 119-134.

RUBINSTEIN, L.I., The Stefan problem, American mathematical Society, Providence, 1971.

STEVENS, W., Lettre de W. Stevens d'Alcan International Ltée, Jonquièrre, à l'Université du Québec à Chicoutimi, juin 1986; document interne.

TOULOUKIAN, Y.S., Thermal conductivity: Metallic elements and alloys, IFI/Plenum, 1970, page 9.

TOULOUKIAN, Y.S., Thermal Diffusivity, IFI/Plenum, 1973, page 2.

VOLLER, V., CROSS, M., "Accurate solutions of moving boundary problems using the enthalpy method", Int. J. Heat Mass Transfer, 1981, Vol. 24, pp 545-556.

ZIENKIEWICZ, O.C., The finite element method, McGraw-Hill, 3<sup>o</sup> édition, 1977, 787p.

## FIGURES

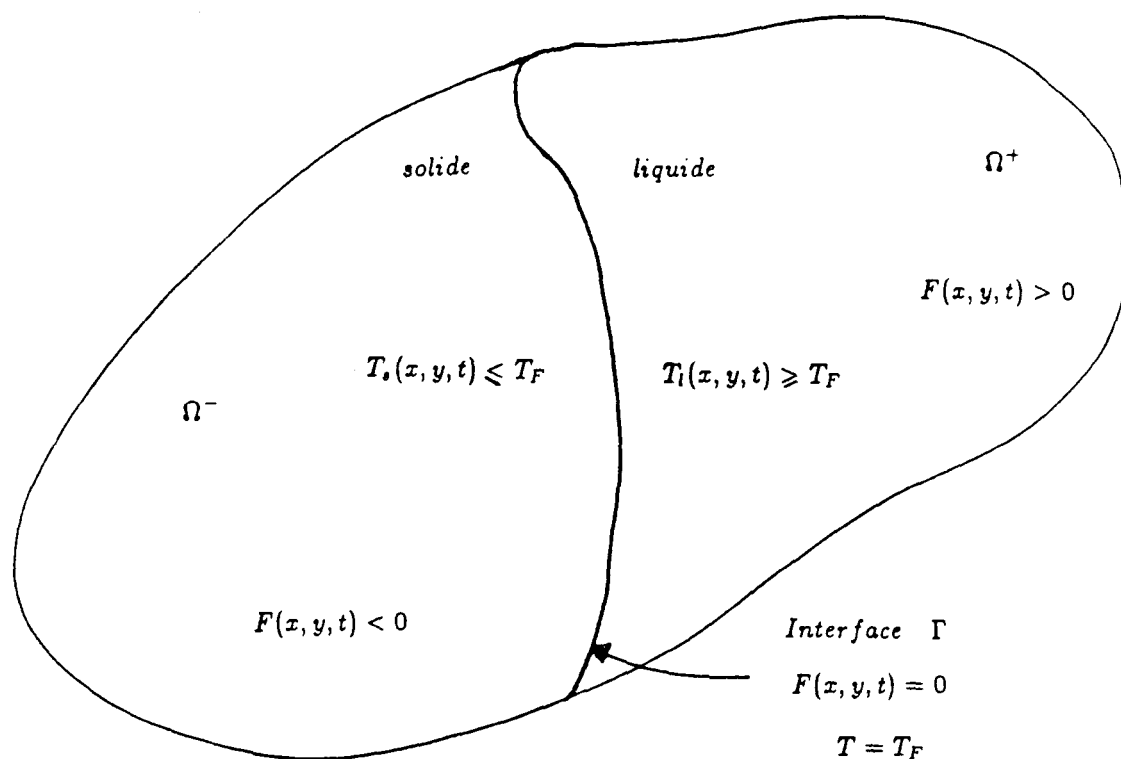


Figure 1. Schéma du domaine d'intégration  $\Omega \times (0, \tau) = \Omega^+ \cup \Gamma \cup \Omega^-$ .

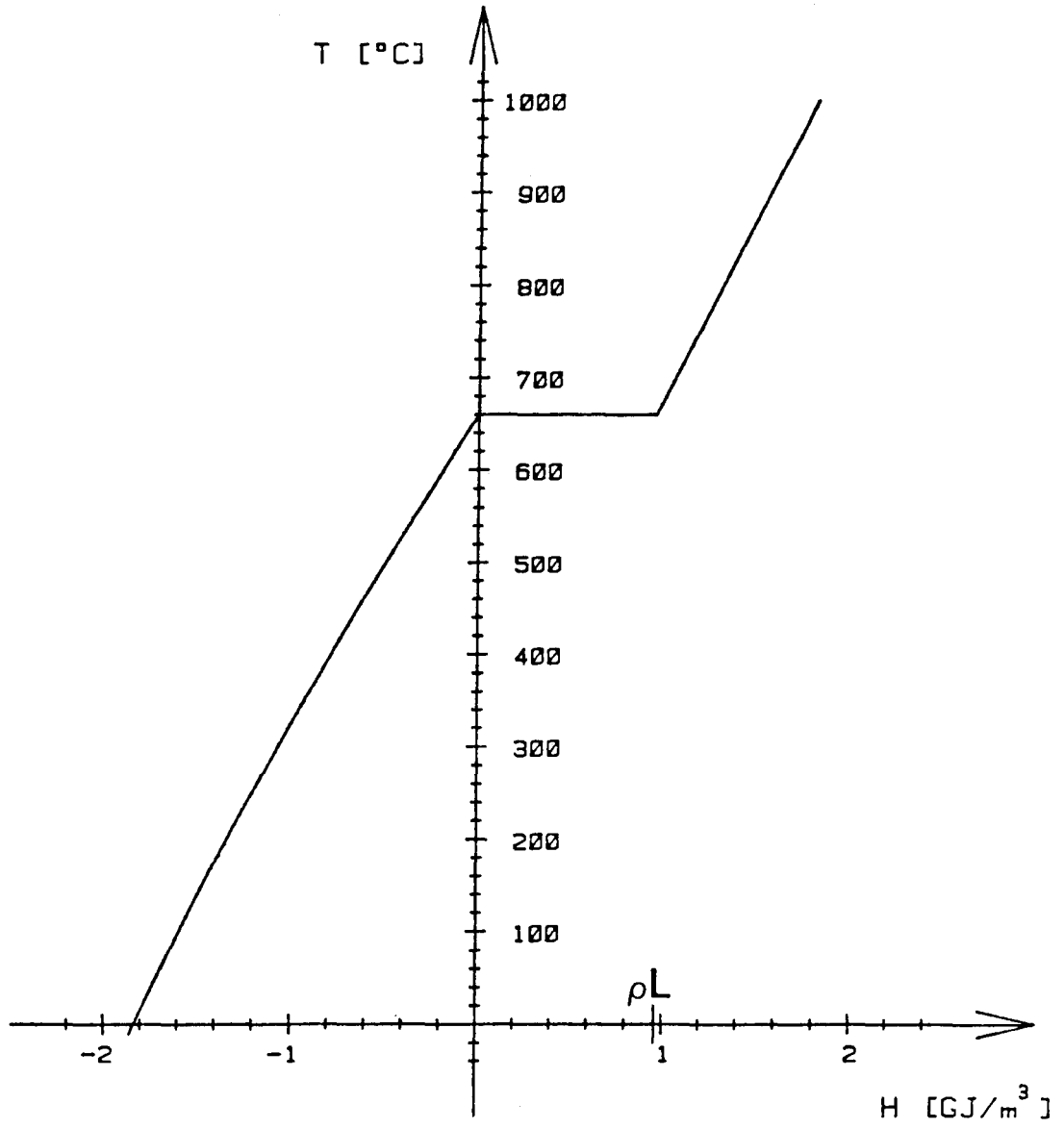


Figure 2. Relation entre la température et l'enthalpie.

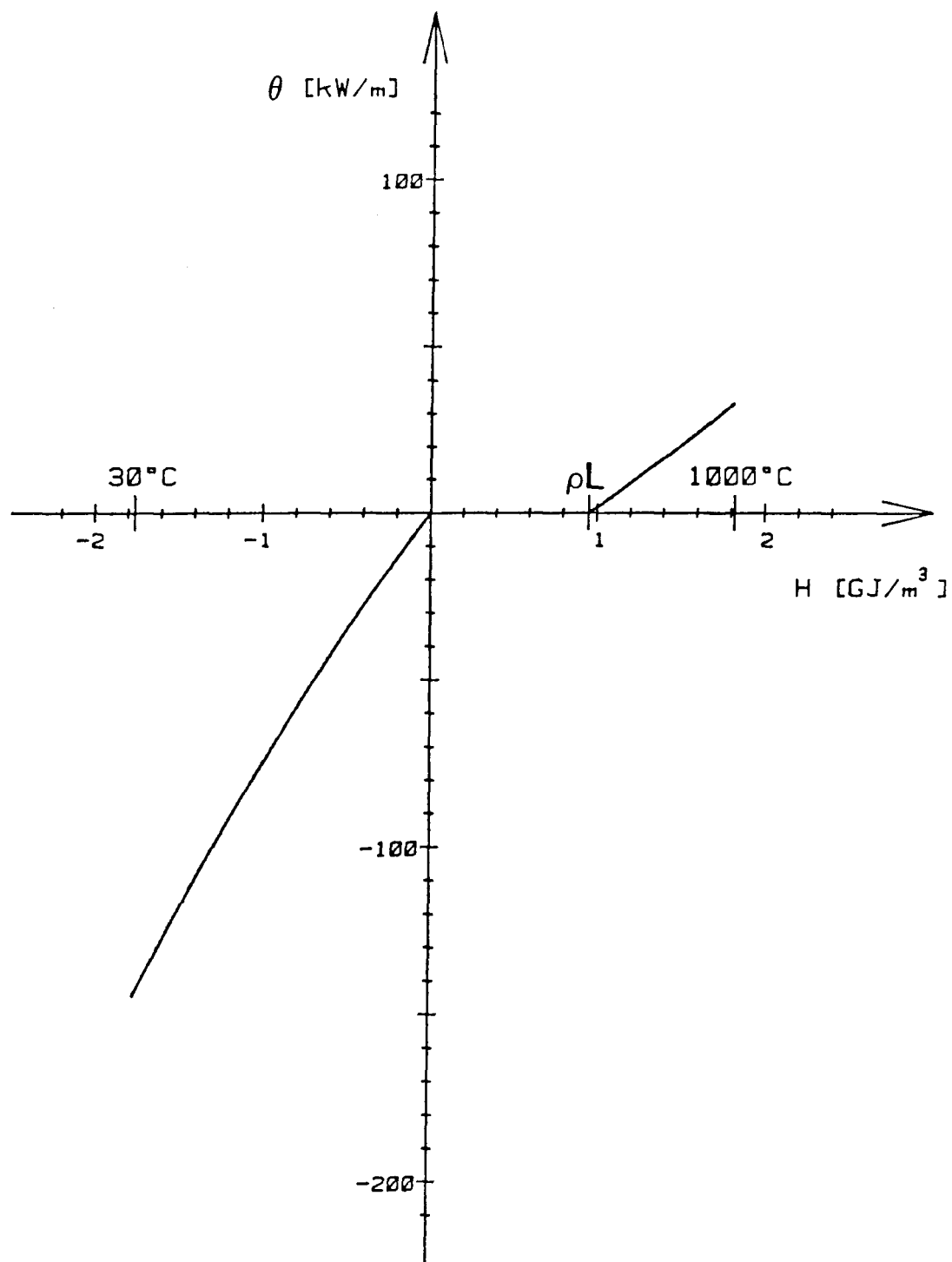


Figure 3. Variation quadratique de la transformée de Kirchhoff de la conductivité thermique en fonction de l'enthalpie volumique.

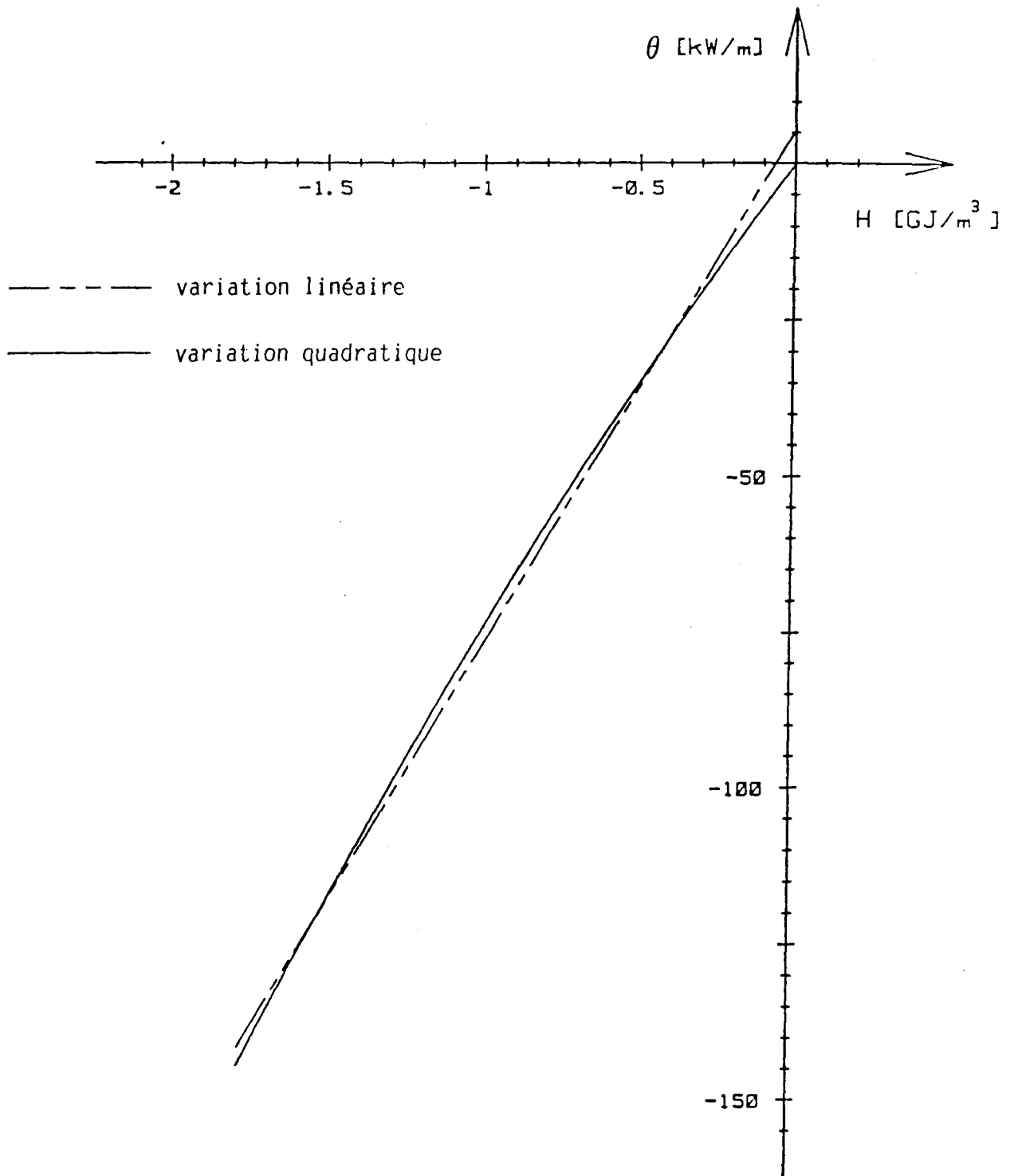


Figure 4. Comparaison entre une variation quadratique et une variation linéaire de la transformée de Kirchhoff de la conductivité thermique du solide.



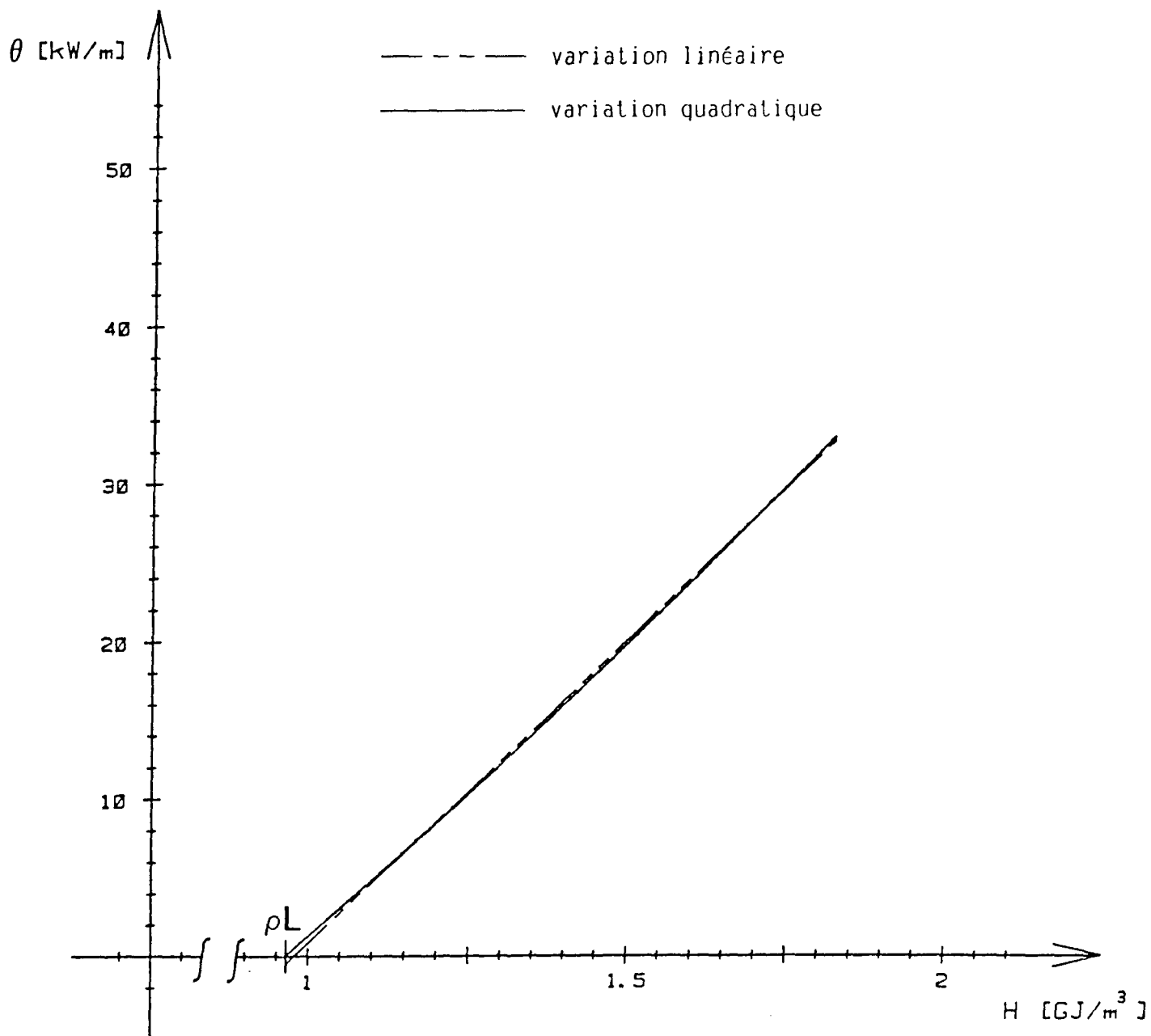
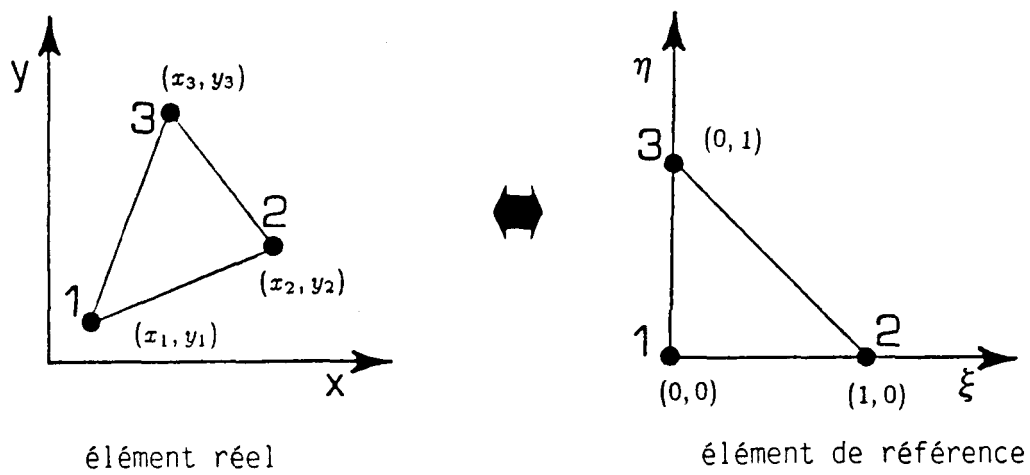


Figure 5. Comparaison entre une variation quadratique et une variation linéaire de la transformée de Kirchhoff de la conductivité thermique du liquide.

$$\{N\} = \begin{Bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{Bmatrix}$$

(a)



$$x(\xi, \eta) = \{N(\xi, \eta)\}^T \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

$$y(\xi, \eta) = \{N(\xi, \eta)\}^T \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix}$$

(b)

Figure 6. (a) Vecteur des fonctions d'interpolation  
 (b) Élément isoparamétrique triangulaire  
 linéaire à trois noeuds, de classe  $C^0$ .

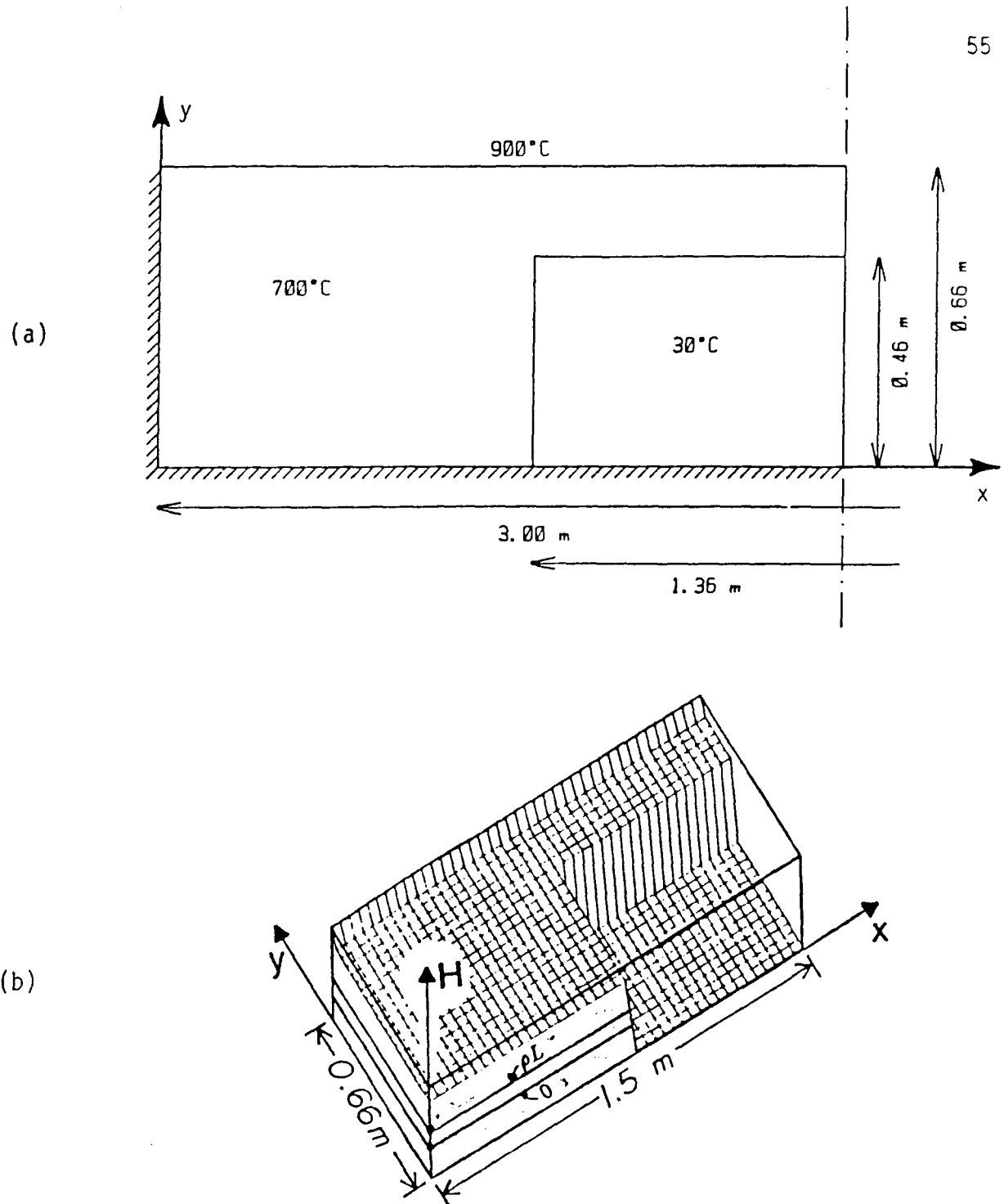
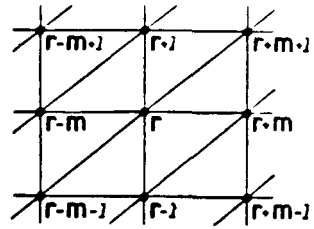
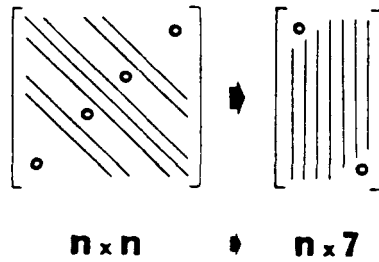


Figure 7. (a) Le domaine géométrique et les températures initiales.  
 (b) Le champ enthalpique  $H(x,y)$  correspondant aux températures initiales  $T(x,y)$  de la figure (a).



(a)



(b)

Figure 8. (a) Représentation des noeuds voisins au noeud  $r$  et de la disposition des éléments isoparamétriques triangulaires linéaires. (b) Restructuration des matrices  $[C]$ ,  $[K]$  et  $[G]$  pour diminuer l'espace mémoire requise et minimiser les opérations lors des produits matriciels.

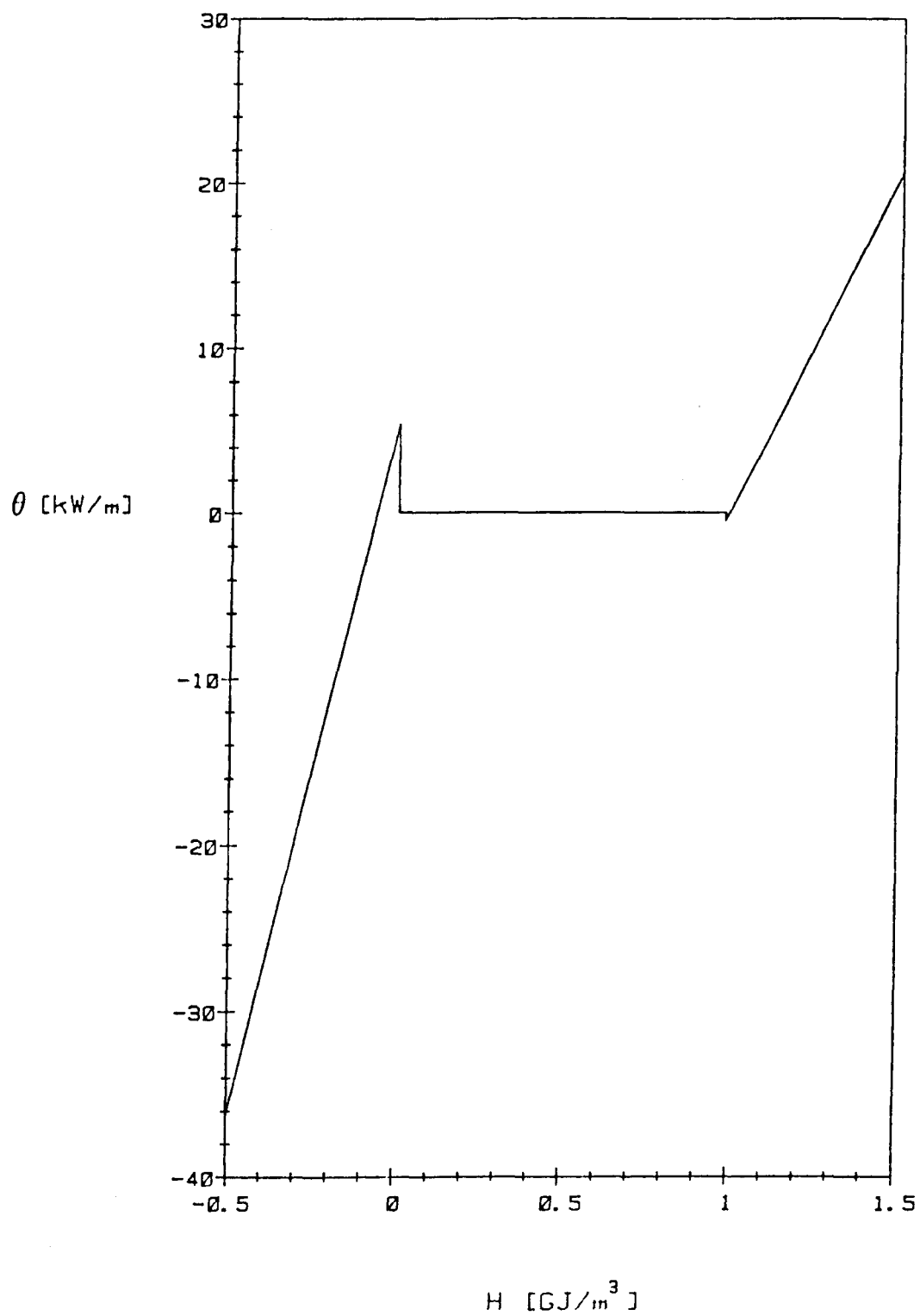


Figure 9. Mise en évidence d'un problème causé par la linéarisation de la transformée de Kirchhoff.

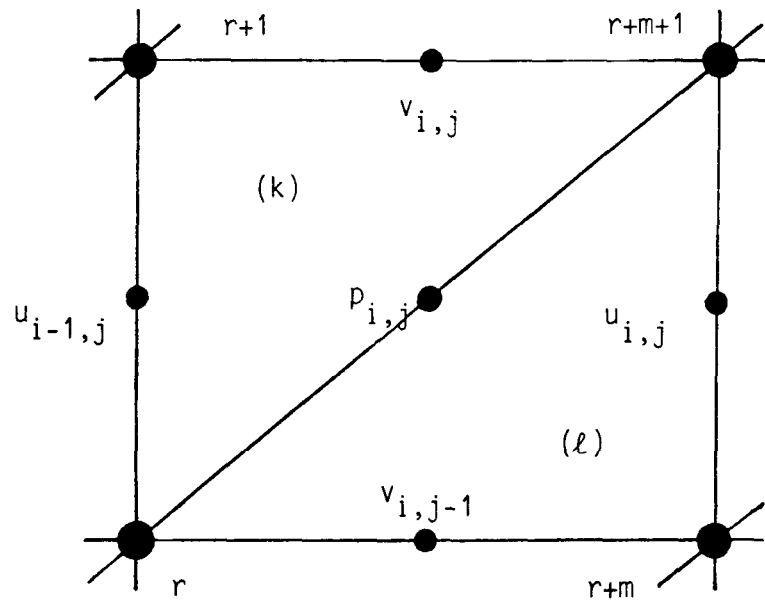


Figure 10. Arrangement des variables  $u$ ,  $v$  et  $p$  d'une cellule formée par les éléments triangulaires  $(k)$  et  $(\ell)$ .

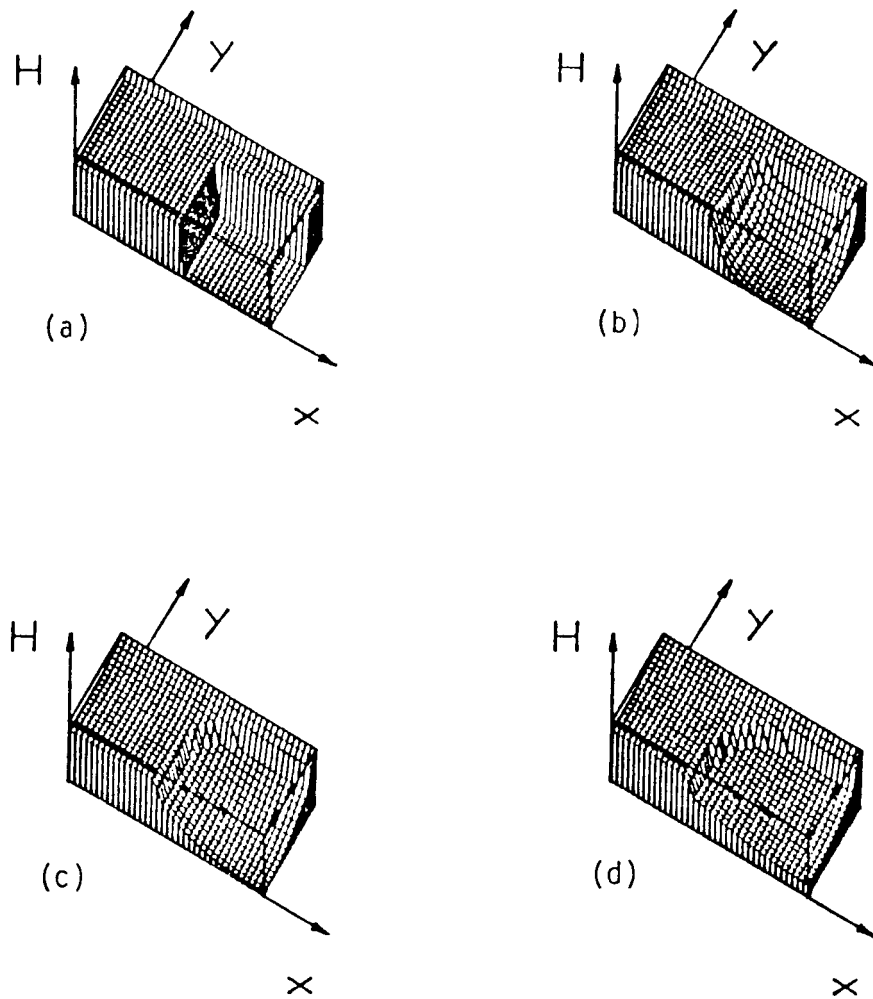


Figure 11. Surfaces enthalpiques calculées avec l'option BQ (740 noeuds). (a) La condition initiale. (b) Après 1 min. (c) Après 5 min. (d) Après 15 min.

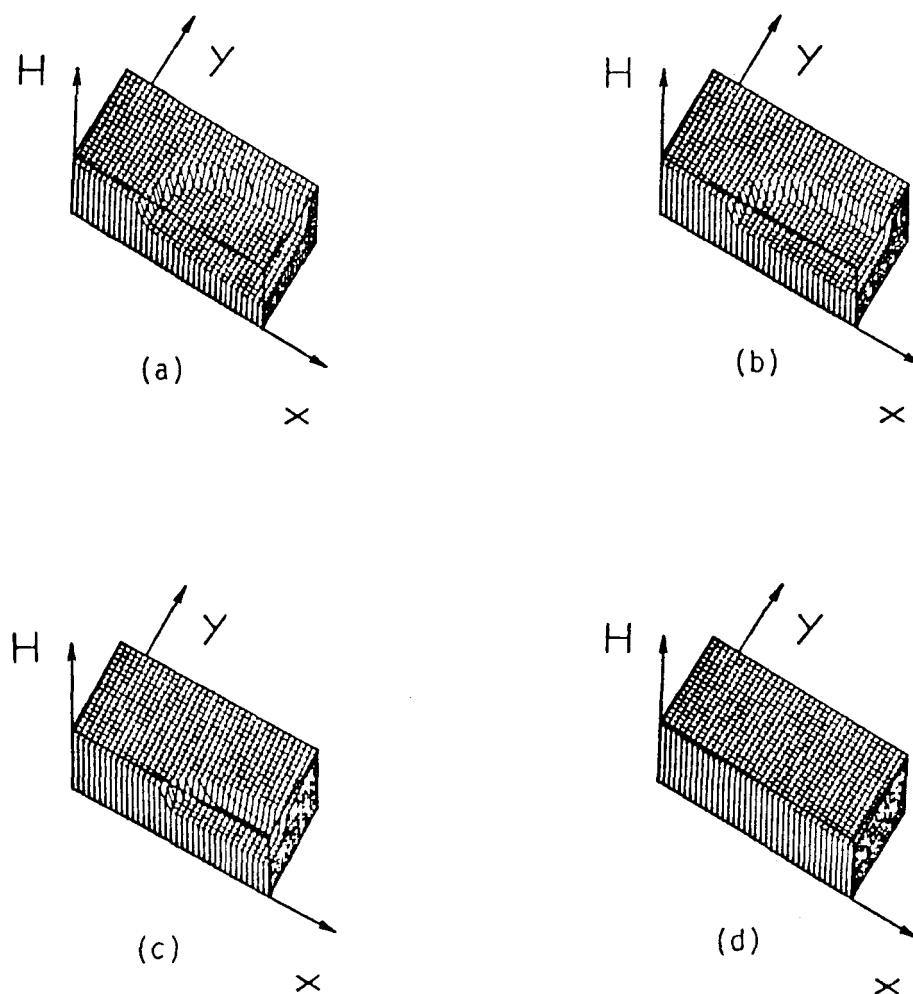


Figure 12. Surfaces enthalpiques calculées avec l'option BQ (740 noeuds). (a) Après 30 min. (b) Après 60 min. (c) Après 120 min. (d) Après 220 min.



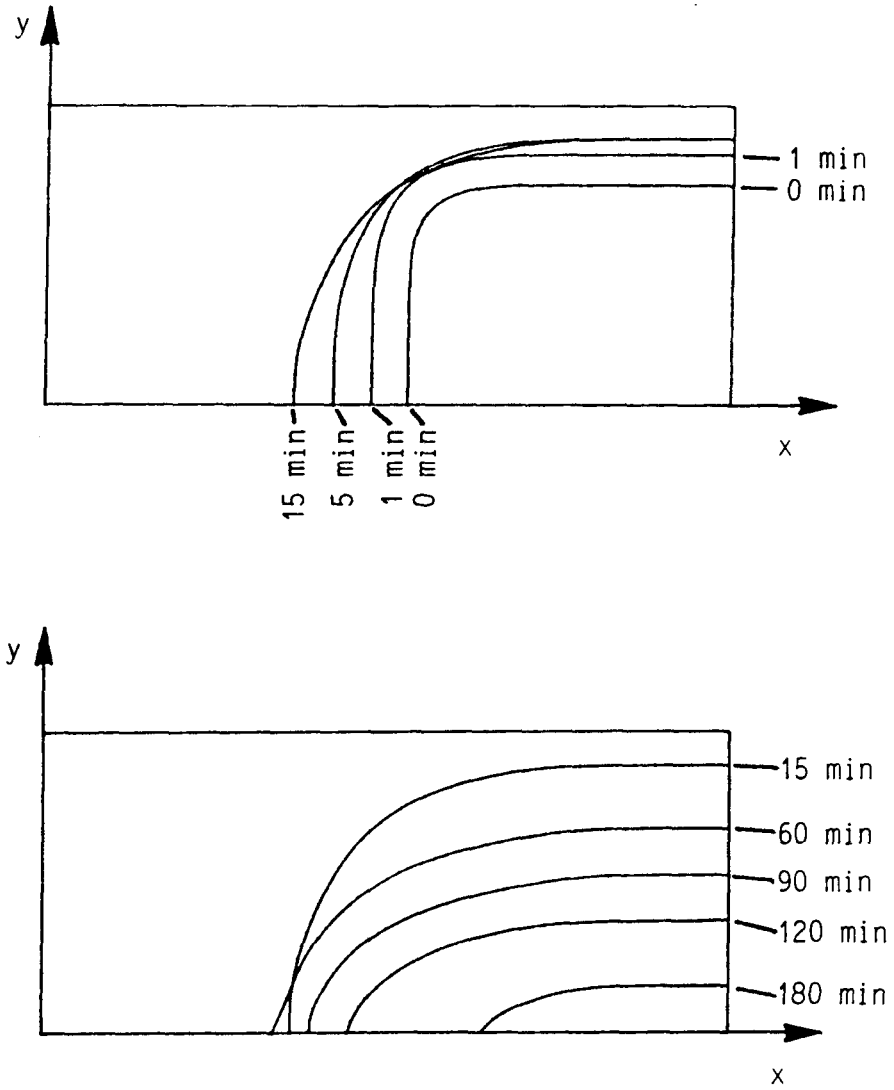


Figure 13. Position de l'interface solide-liquide calculée avec l'option BQ (740 noeuds).

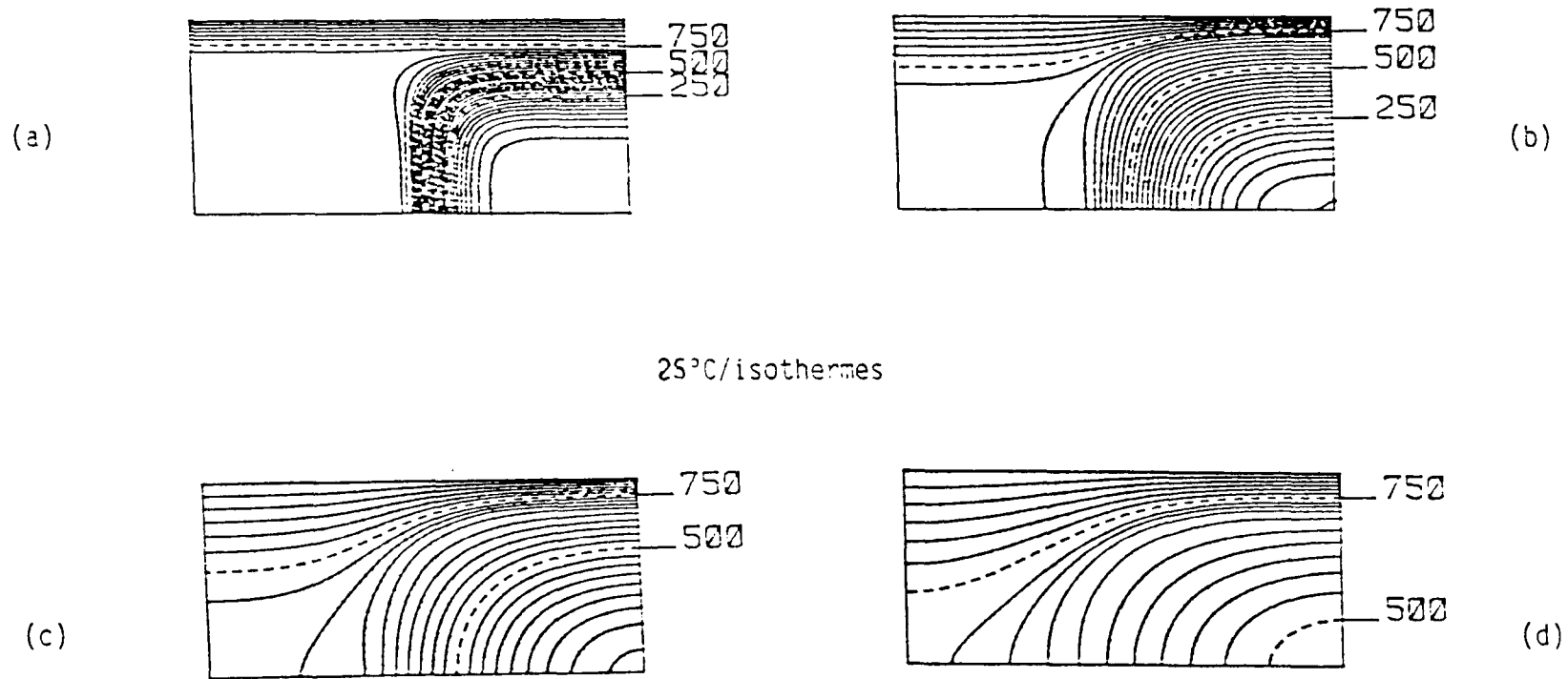


Figure 14. Isothermes calculées avec l'option BQ (740 noeuds), (a) après 2 min, (b) après 5 min, (c) après 15 min, (d) après 30 min.

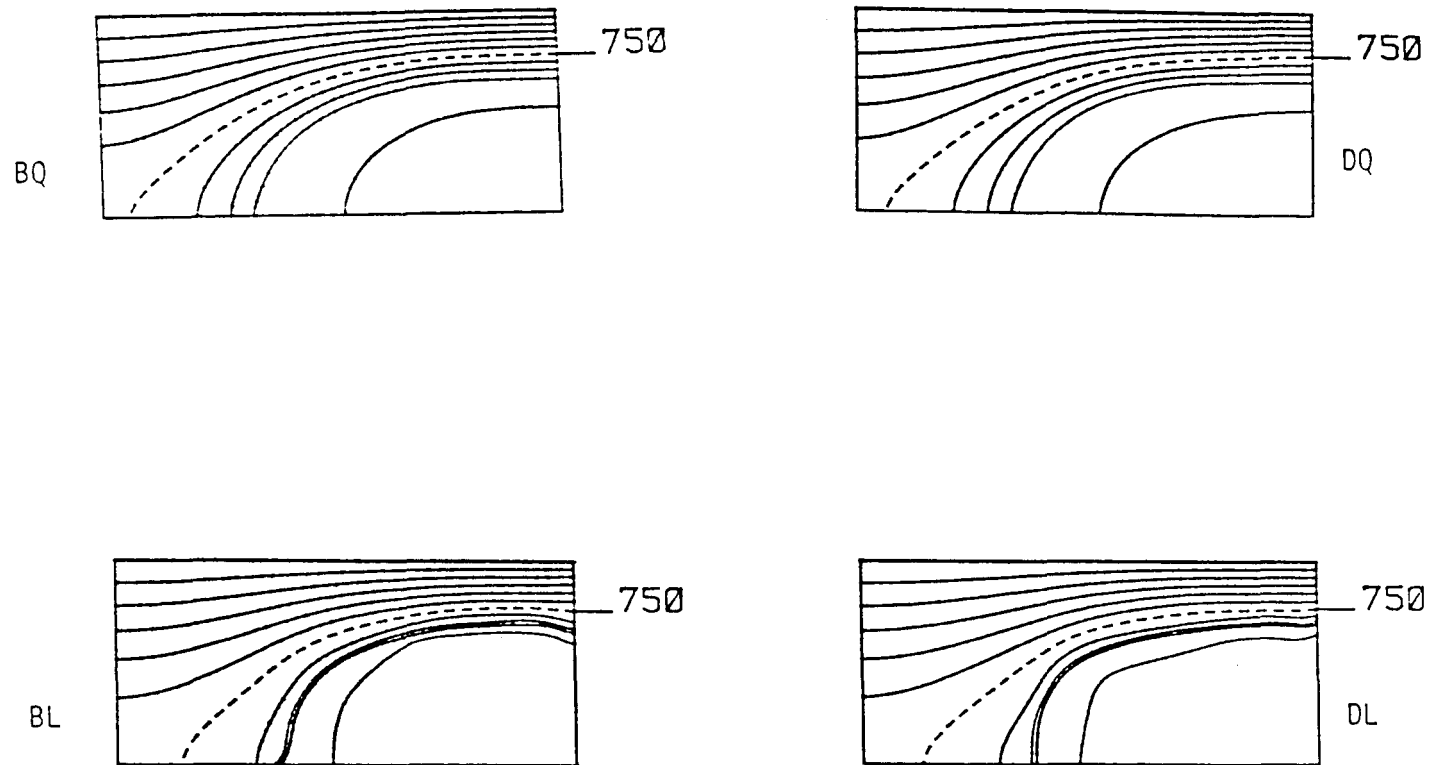


Figure 15. Isothermes calculées après 60 minutes de chauffage en utilisant 740 noeuds et les quatre options (25°C/isotherme).

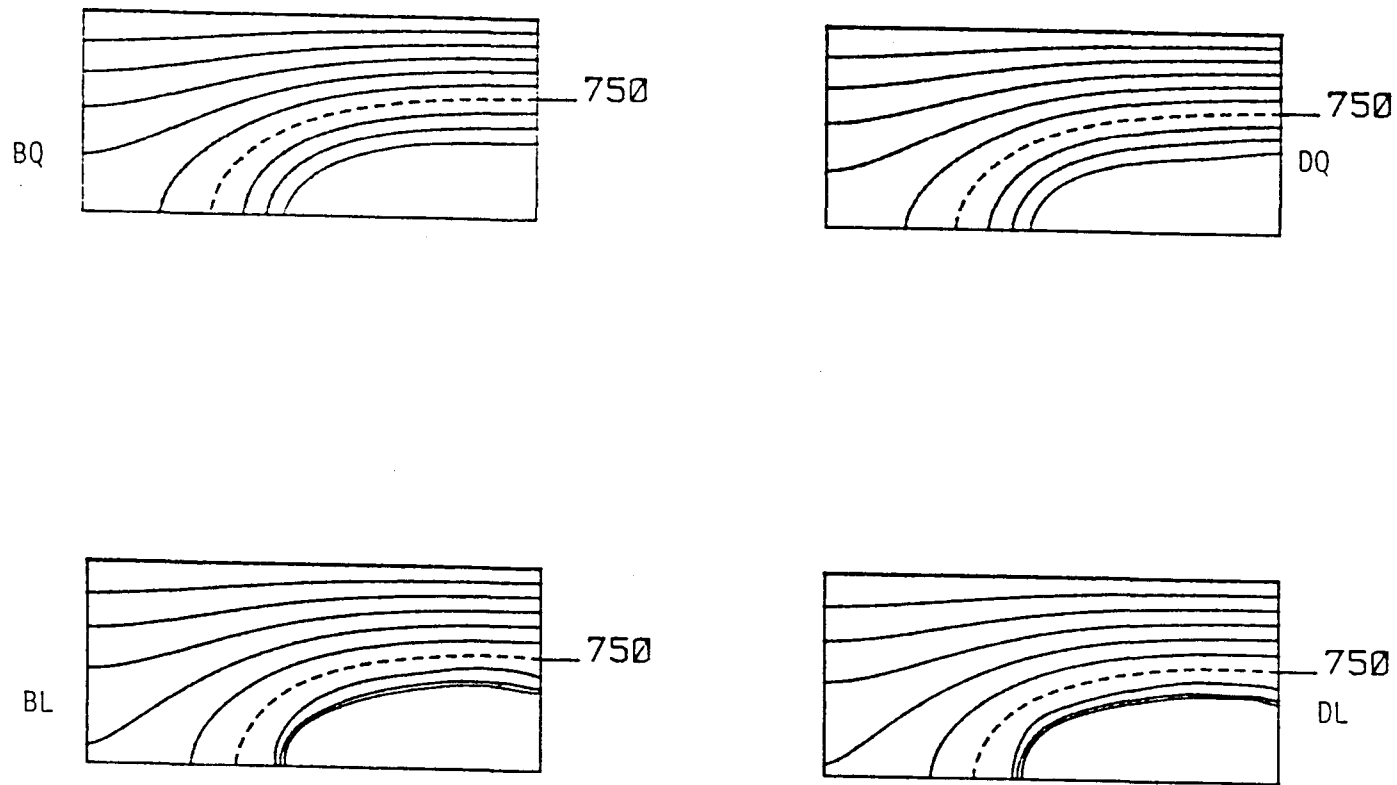


Figure 16. Isothermes calculées après 120 minutes de chauffage en utilisant 740 noeuds et les quatre options (25°C/isotherme).

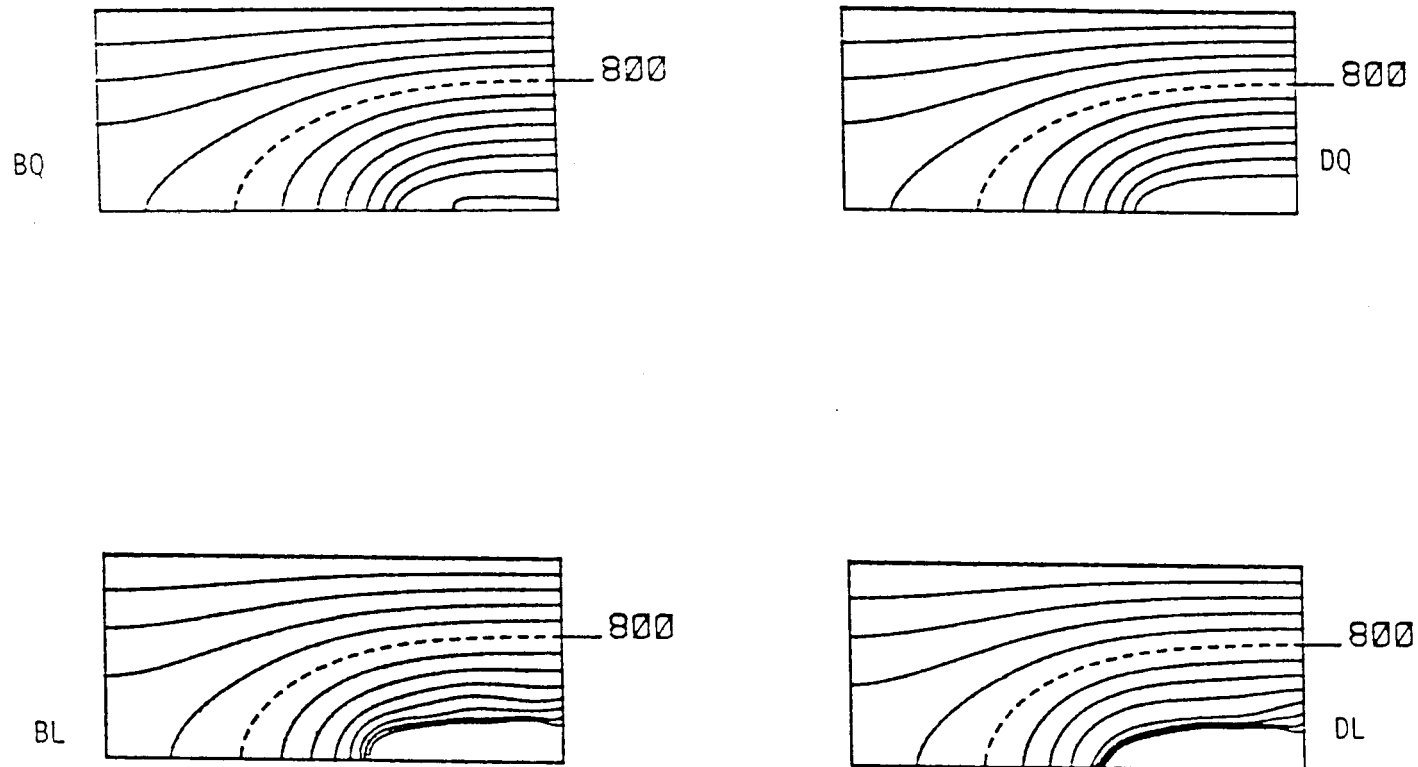


Figure 17. Isothermes calculées après 180 minutes de chauffage en utilisant 740 noeuds et les quatre options (20°C/isotherme).

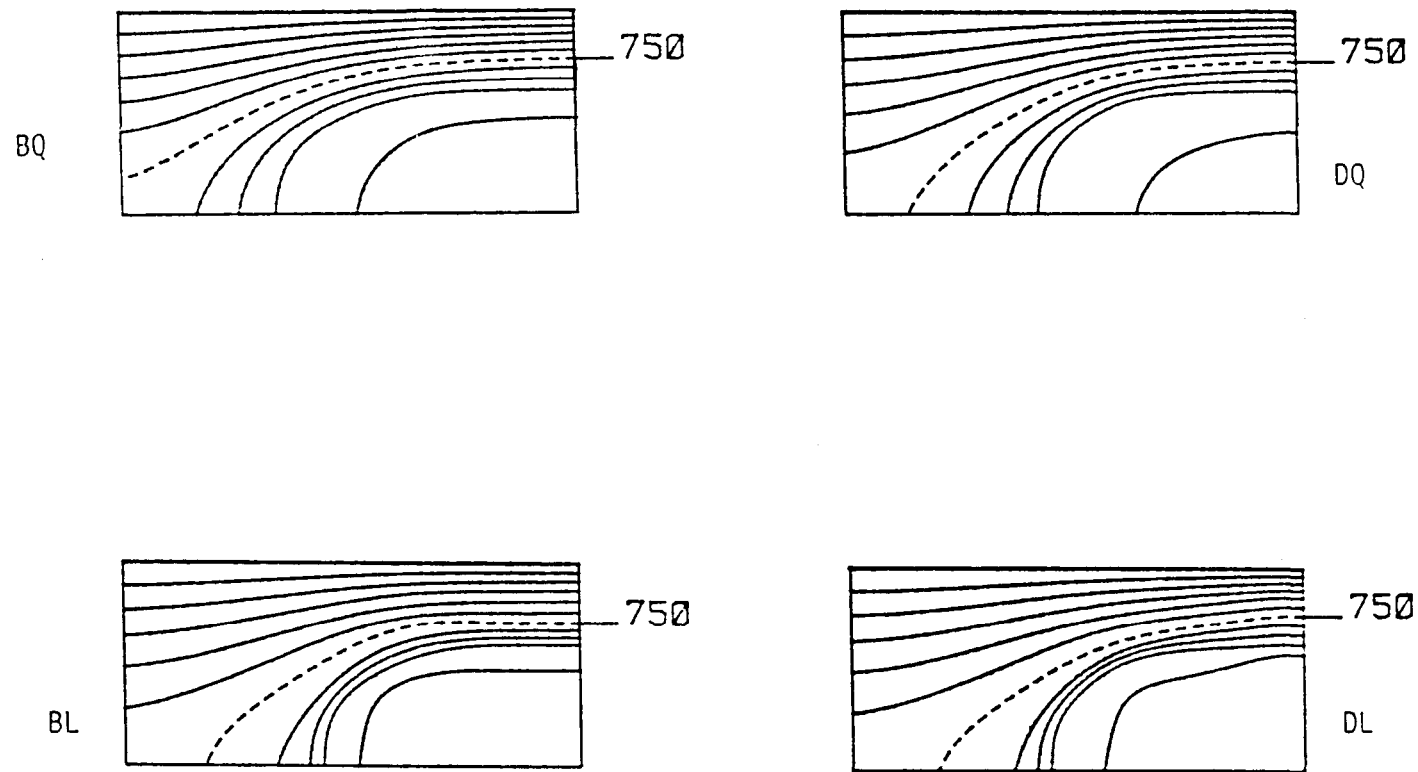


Figure 18. Isothermes calculées après 60 minutes de chauffage en utilisant 112 noeuds et les quatre options (25°C/isotherme).

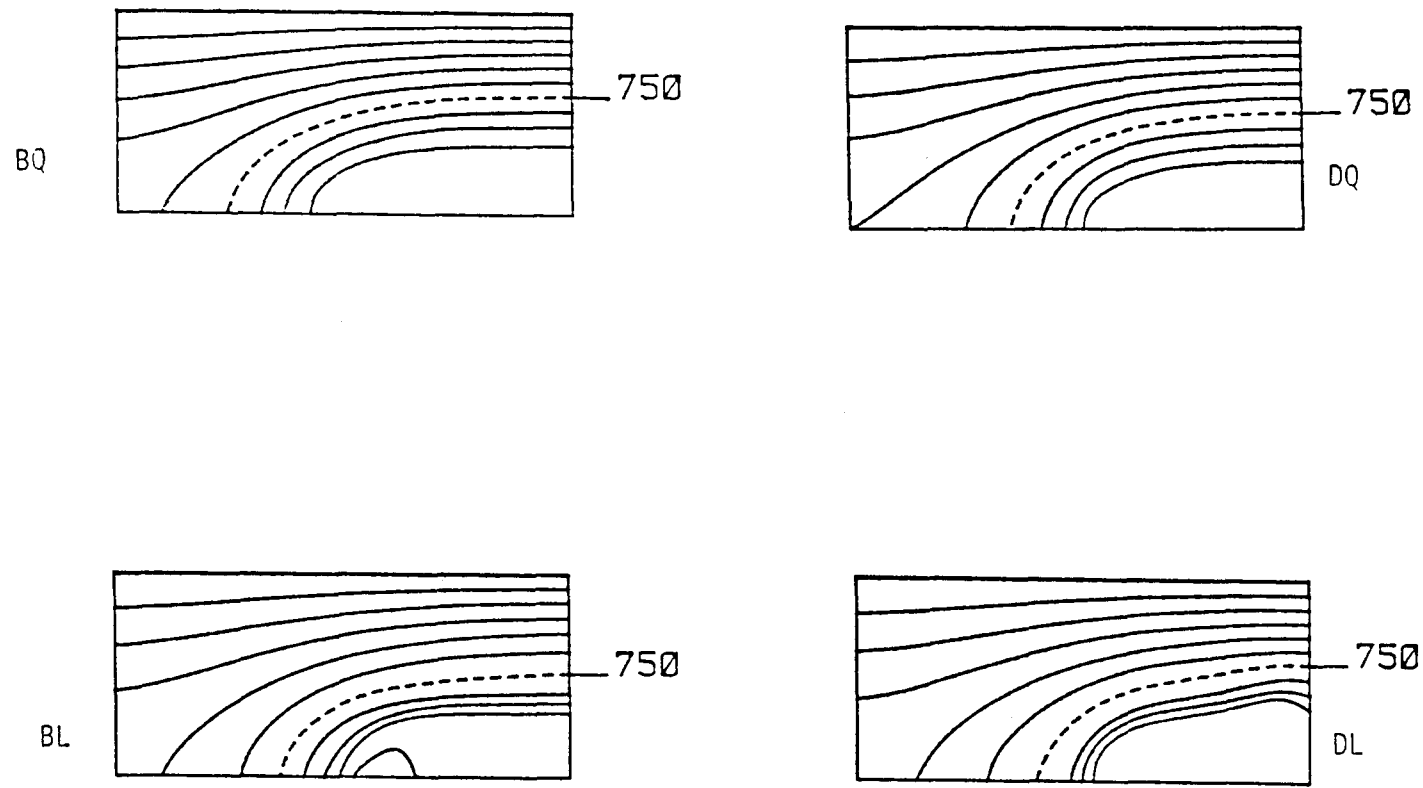


Figure 19. Isothermes calculées après 120 minutes de chauffage en utilisant 112 noeuds et les quatre options (25°C/isotherme).

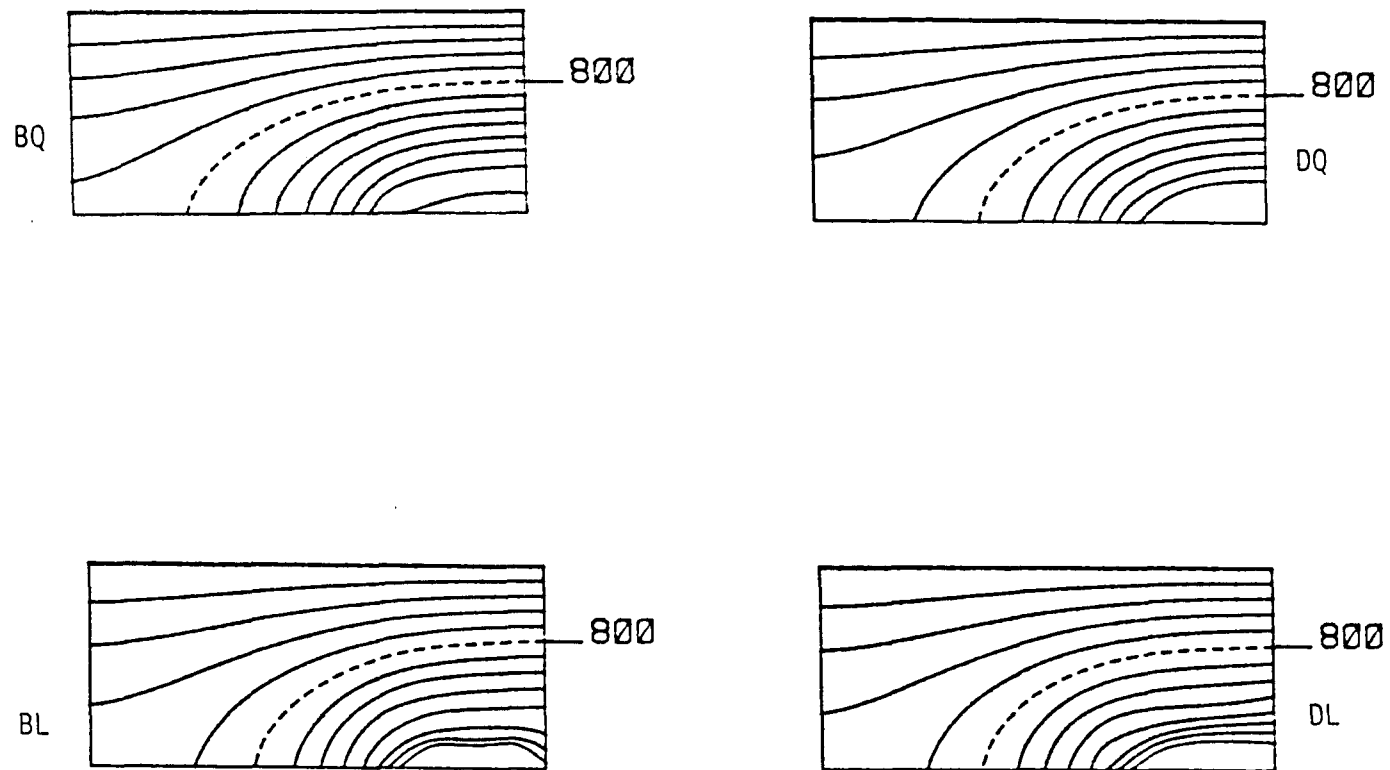


Figure 20. Isothermes calculées après 180 minutes de chauffage en utilisant 112 noeuds et les quatre options (20°C/isotherme).



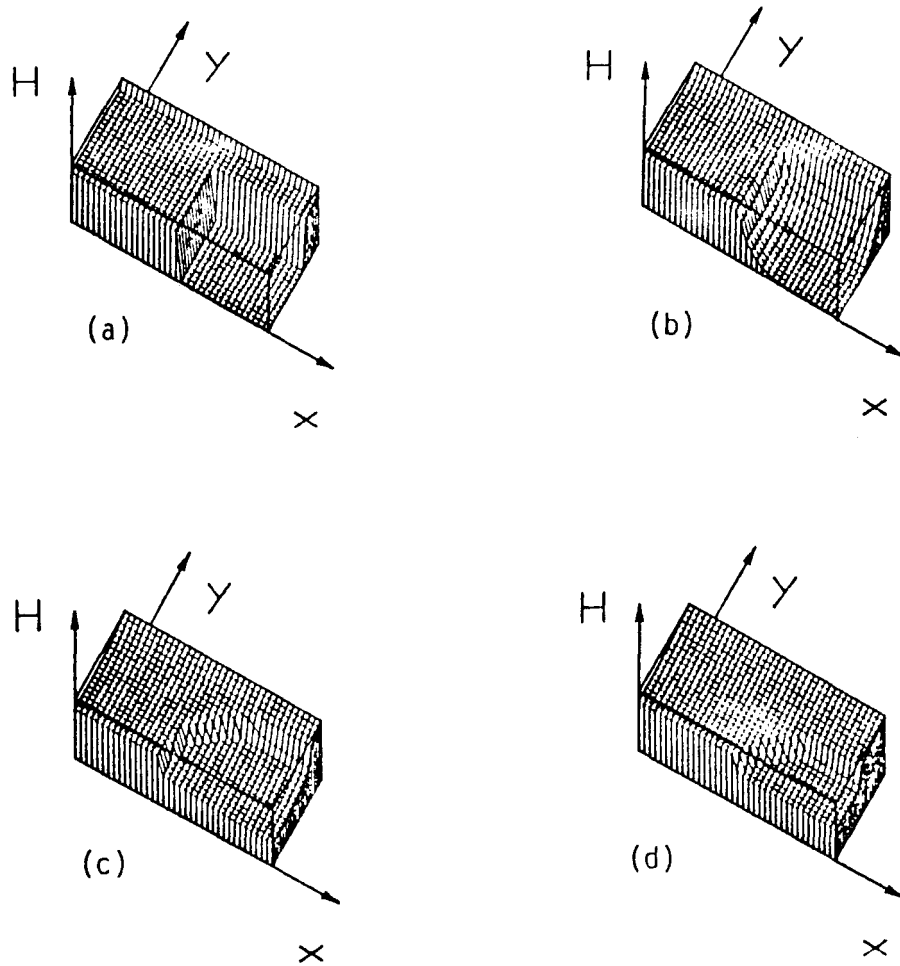
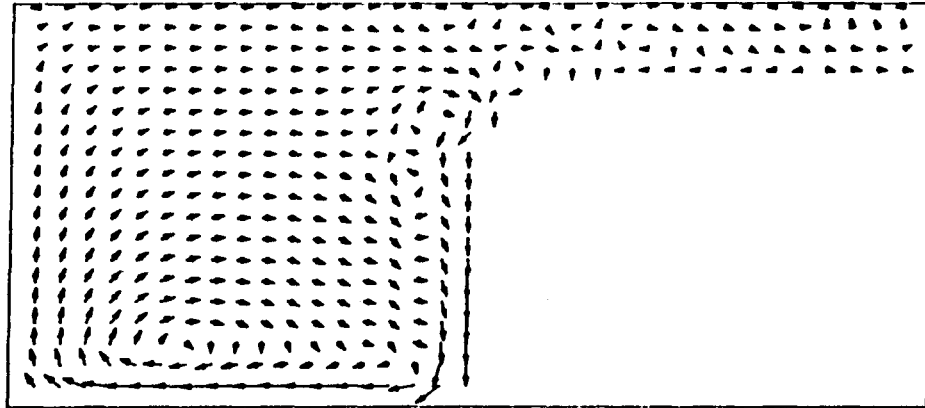
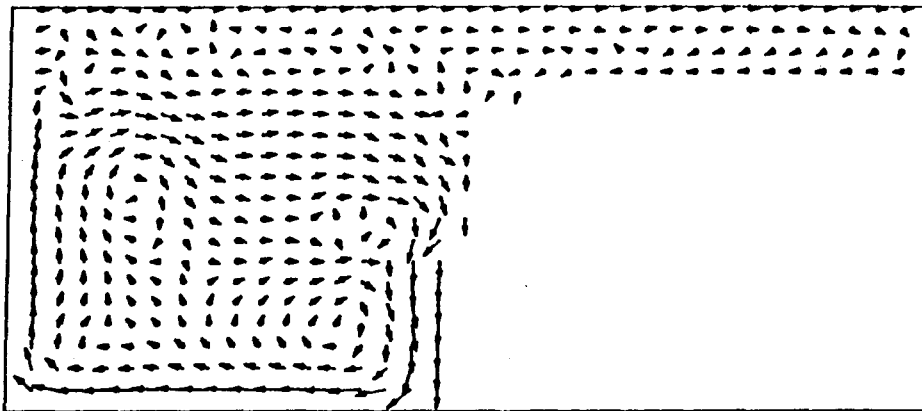


Figure 21. Surfaces enthalpiques calculées en convection naturelle par l'option DQ (740 noeuds). (a) La condition initiale. (b) Après 1 min. (c) Après 30 min. (d) Après 60 min.

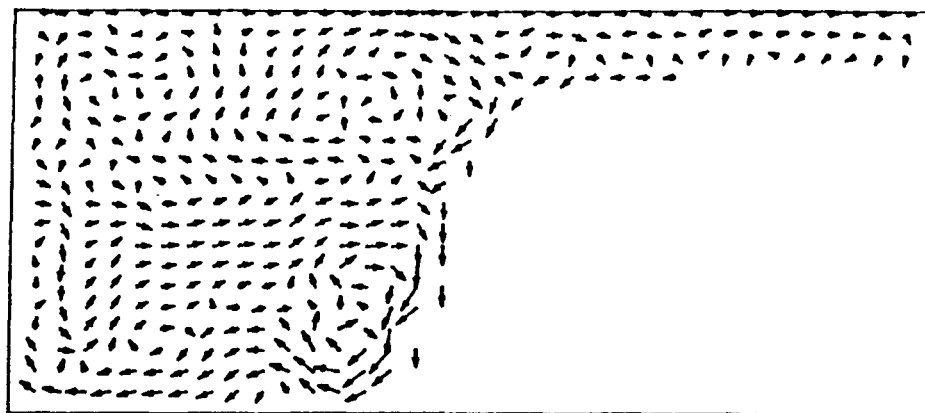


(a)

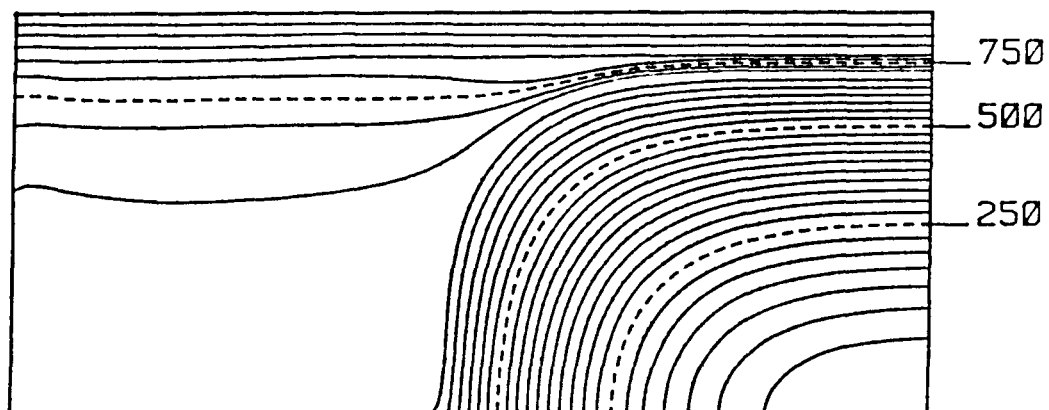


(b)

Figure 22. Circulation du fluide au cours des premières secondes.  
(a)  $t=30$  sec;  $v_{\max}=59.6$  mm/s. (b)  $t=1$  min;  $v_{\max}=70.8$  mm/s.



(a)



(b)

Figure 23. Circulation du fluide après 5 minutes de chauffage.  
(a) Champ de vitesse;  $v_{\max} = 42.2$  mm/s. (b) Isothermes associées à (a); 25°C/isotherme.

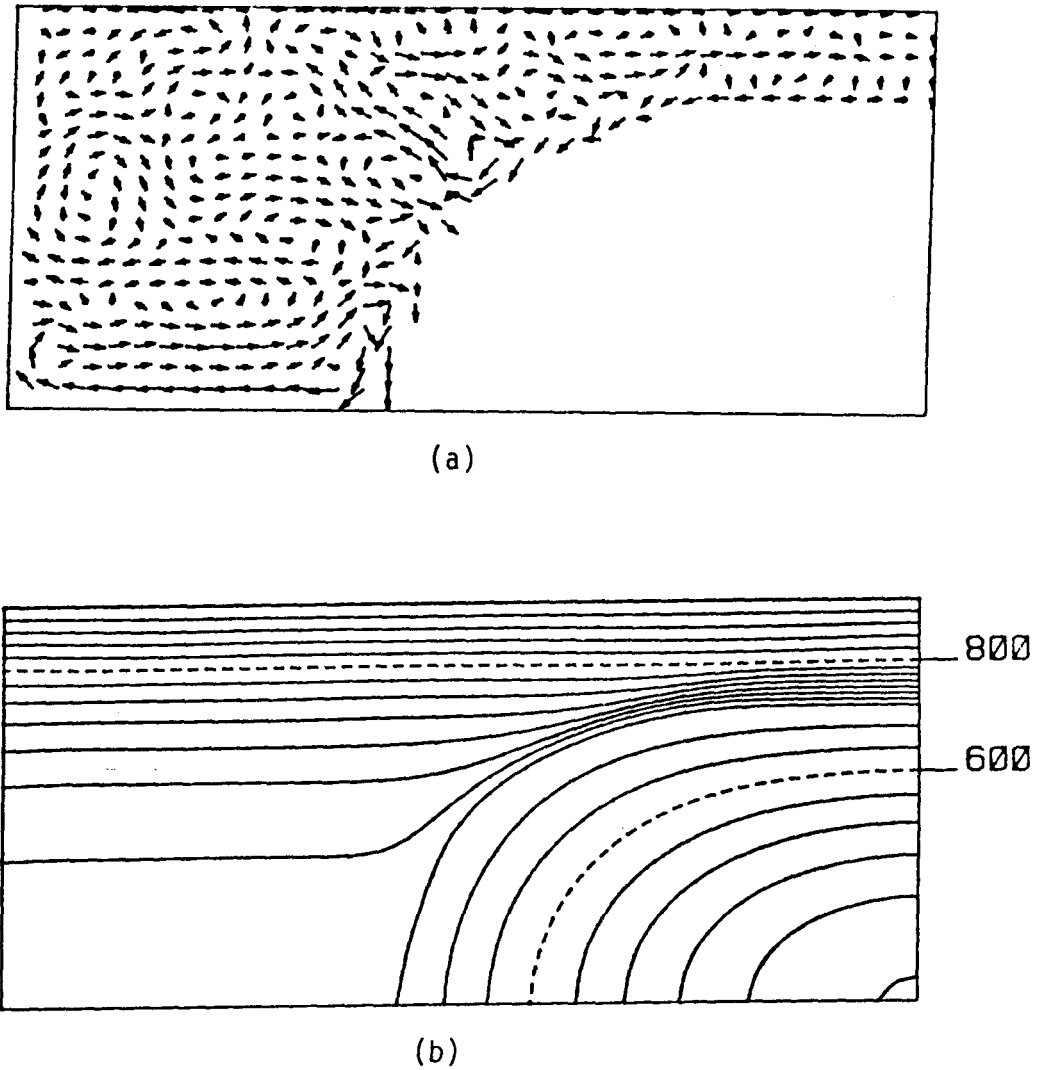
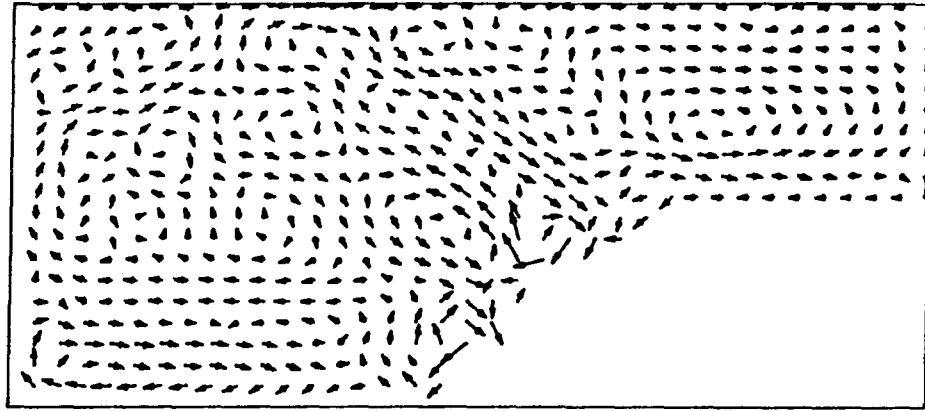
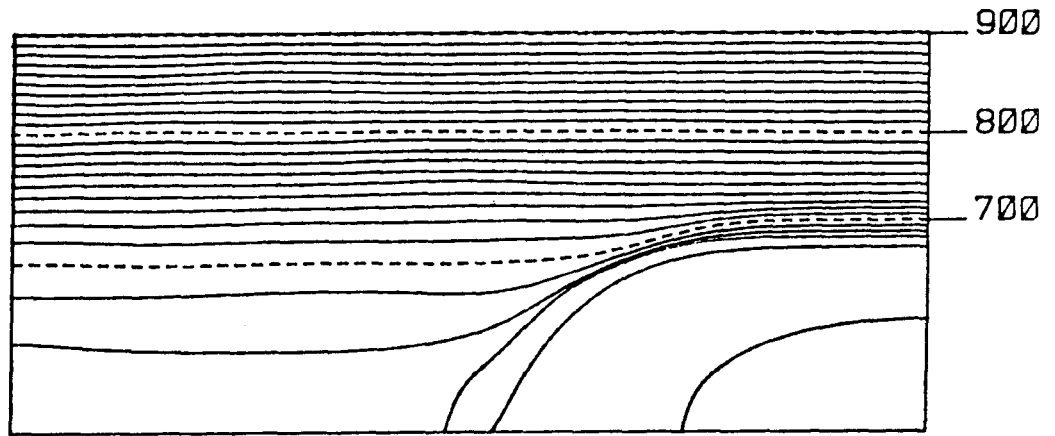


Figure 24. Circulation du fluide après 30 minutes de chauffage.  
(a) Champ de vitesse;  $v_{\max} = 10.4$  mm/s. (b) Isothermes associées à (a);  $20^\circ\text{C}$ /isotherme.

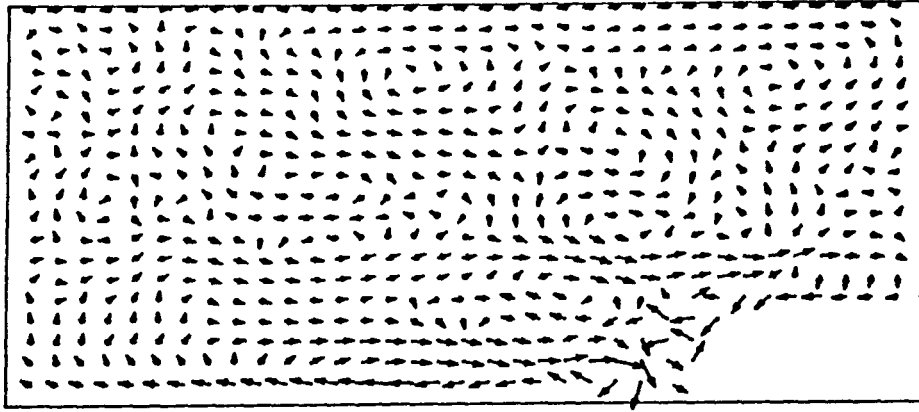


(a)

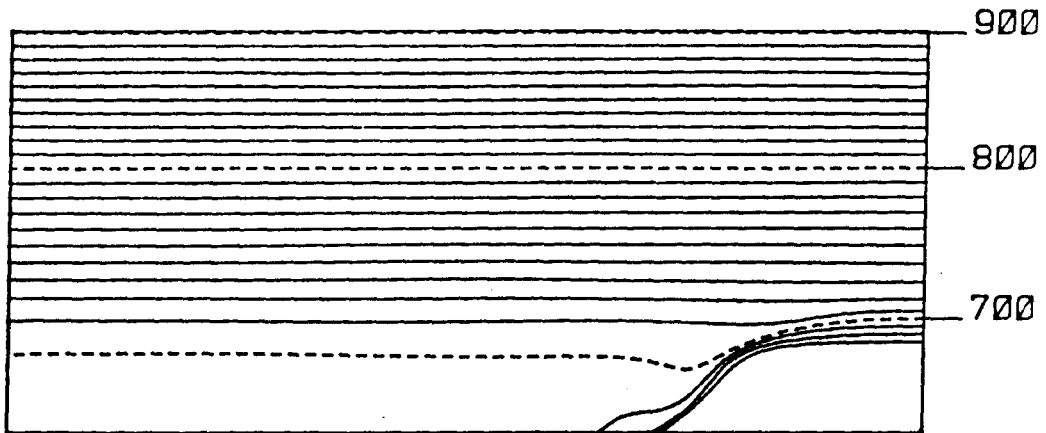


(b)

Figure 25. Circulation du fluide après 60 minutes de chauffage.  
 (a) Champ de vitesse;  $v_{\max} = 14.0 \text{ mm/s}$ . (b) Isothermes associées à (a);  $10^\circ\text{C}/\text{isotherme}$ .

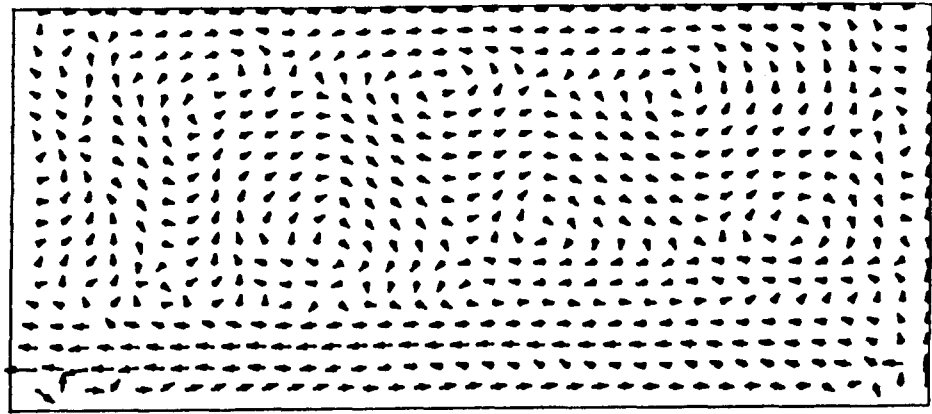


(a)

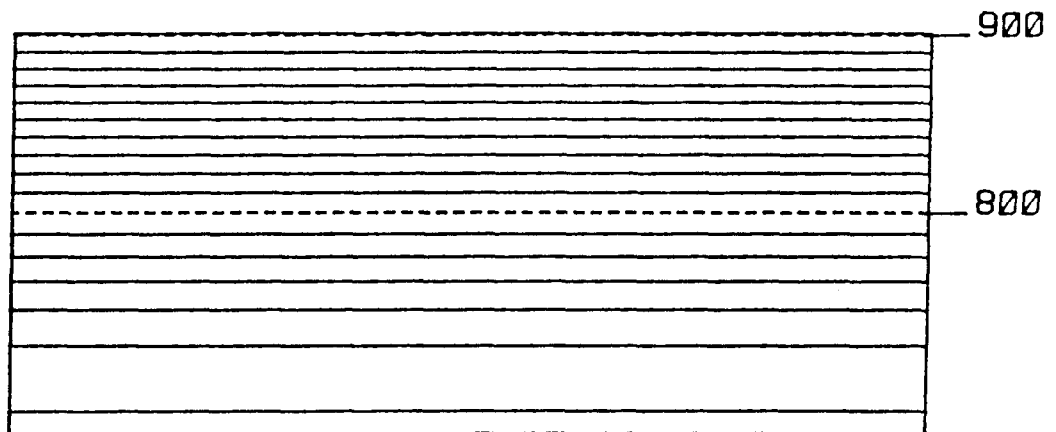


(b)

Figure 26. Circulation du fluide après 90 minutes de chauffage.  
 (a) Champ de vitesse;  $v_{\max} = 10.5 \text{ mm/s}$ . (b) Isothermes associées à (a);  $10^\circ\text{C}/\text{isotherme}$ .

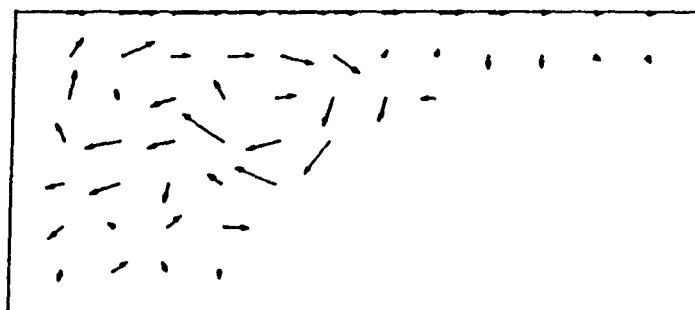


(a)

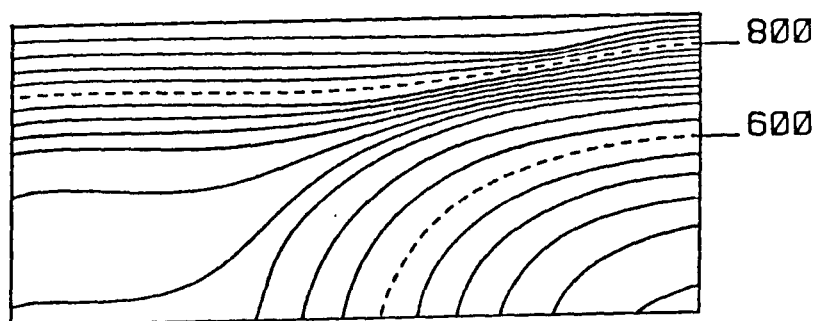


(b)

Figure 27. Circulation du fluide après 115 minutes de chauffage.  
(a) Champ de vitesse;  $v_{\max} = 0.3$  mm/s. (b) Isothermes associées à (a);  $10^{\circ}\text{C}/\text{isotherme}$ .



(a)



(b)

Figure 28. Circulation du fluide après 30 minutes de chauffage en utilisant seulement 112 noeuds (nombre insuffisant). (a) Champ de vitesse;  $v_{\max} = 13.3 \text{ mm/s}$ . (b) Isothermes associées à (a);  $20^\circ\text{C}/\text{isotherme}$ .



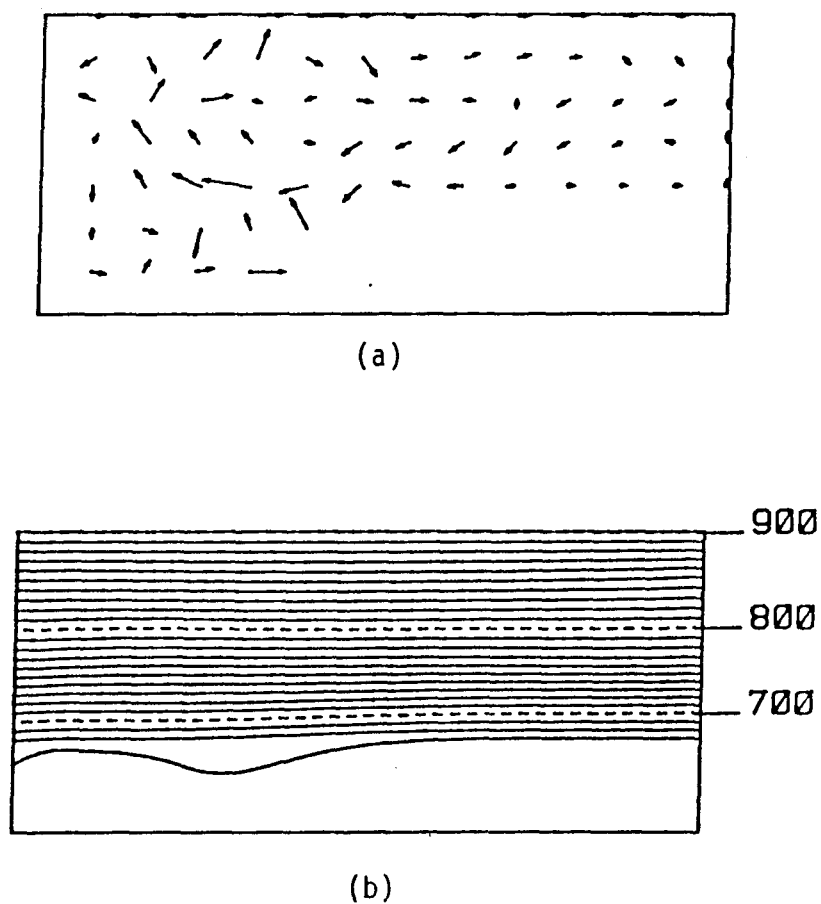


Figure 29. Circulation du fluide après 90 minutes de chauffage en utilisant seulement 112 noeuds (nombre insuffisant). (a) Champ de vitesse;  $v_{max} = 9.6$  mm/s. (b) Isothermes associées à (a);  $10^{\circ}\text{C}/\text{isotherme}$ .

APPENDICE 1  
ORGANISATION GÉNÉRALE DU PROGRAMME

## APPENDICE 1

### ORGANISATION GÉNÉRALE DU PROGRAMME

FUSION est un programme de 118 300 bytes écrit en FORTRAN-77. Une version compatible avec les ordinateurs de la série VAX-11 se retrouve à l'annexe 1. Ce programme permet à l'utilisateur de simuler la fusion d'un bloc d'aluminium solide immergé dans un bain d'aluminium liquide. Le bain est de forme rectangulaire. Il est isolé sur les murs et dans le fond; une température est imposée sur le dessus. La charge solide peut être déposée n'importe où pourvu qu'elle soit submergée dans le bain. L'utilisateur peut aussi ne placer que du liquide ou du solide comme point de départ. À la sortie, le programme FUSION fournit:

- i) le champ de température,
- ii) le champ de vitesse et de pression dynamique,
- iii) le champ enthalpique,
- iv) la forme de l'amas solide.

Le programme FUSION est construit en trois parties (figure 30).

#### Partie 1: BLOC\_PRE

Le bloc de préparation (figure 31) est exécuté avant le début des calculs. POSITIONS est utilisé pour calculer les coordonnées des noeuds du maillage en éléments finis. INIT (figure 31) fait l'initialisation des variables et calcule la matrice de capacité thermique, ENTHALPIE convertit les températures initiales en

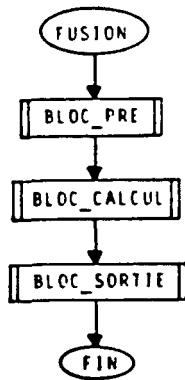


Figure 30

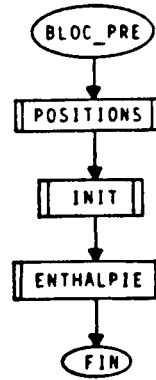


Figure 31

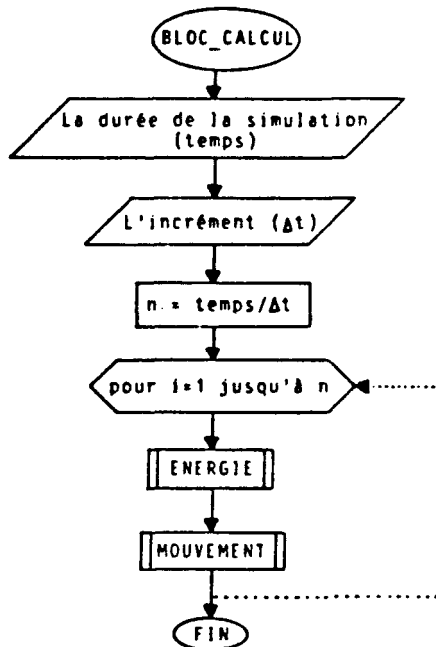


Figure 32

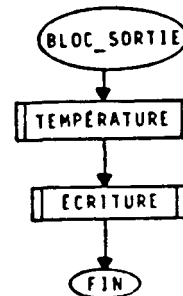


Figure 33

Figure 30. Organigramme du programme FUSION. Figure 31. Organigramme du sous-programme BLOC PRE. Figure 32. Organigramme du sous-programme BLOC\_CALCUL. Figure 33. Organigramme du sous-programme BLOC\_SORTIE.

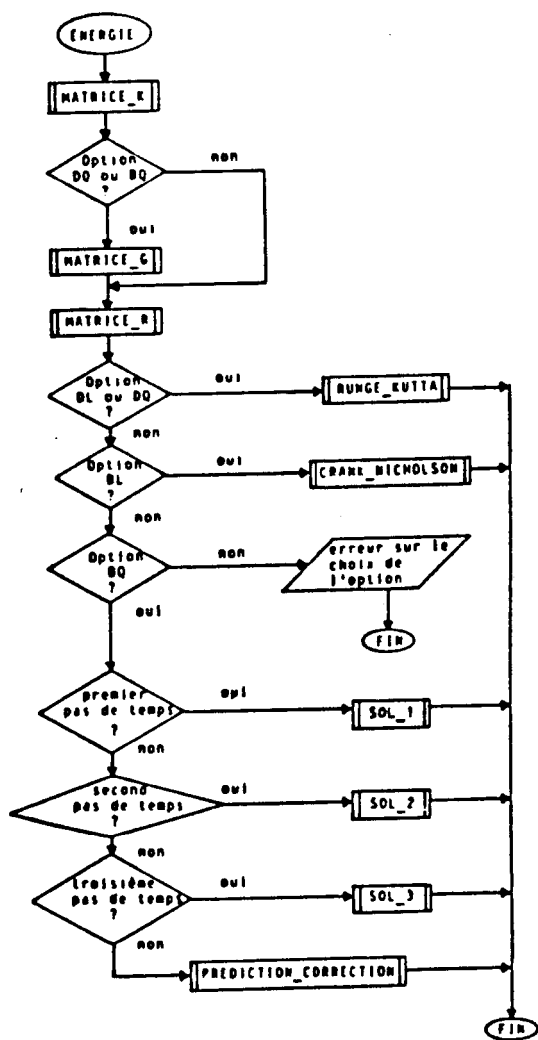


Figure 34

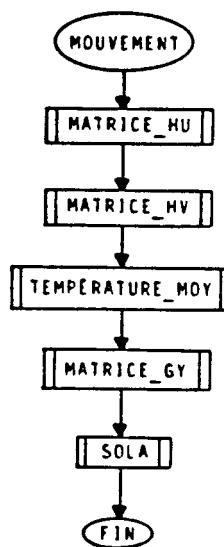


Figure 35

Figure 34. Organigramme du sous-programme ENERGIE.  
 Figure 35. Organigramme du sous-programme MOUVEMENT.

enthalpies initiales.

### Partie 2: BLOC\_CALCUL

Le bloc des calculs (figure 32) est utilisé pour solutionner l'équation d'énergie (figure 34) et les équations du mouvement (figure 35).

ENERGIE (figure 34) commence par assembler les matrices K, G et R; par la suite, il intègre l'équation d'énergie. Lorsque l'utilisateur a choisi l'option BQ, l'intégration du premier pas de temps (figure 34) se fait par la méthode d'Euler explicite. L'intégration du second pas de temps se fait par une méthode de prédiction-correction d'ordre 2, et le troisième, par une méthode de prédiction-correction d'ordre 3. Le reste des intégrations se fait par la méthode de prédiction-correction de Hamming.

MOUVEMENT (figure 35) commence par interpoler l'enthalpie des noeuds en différences finies (MATRICE\_HU et MATRICE\_HV de la figure 35). Par la suite, MOUVEMENT calcule la température moyenne du liquide utilisée dans l'approximation de Boussinesq. Ensuite, MOUVEMENT procède à l'évaluation de la force de corps par l'intermédiaire du sous-programme MATRICE\_GY de la figure 35. Pour terminer, MOUVEMENT résout les équations du mouvement et transforme les valeurs des vitesses ainsi calculées en vitesses nodales, utilisées par le sous-modèle en éléments finis.

### Partie 3: BLOC\_SORTIE

Ce bloc (figure 33) transforme les enthalpies en températures

et écrit les résultats obtenus sur disque.

## APPENDICE 2

ALGORITHME DE MULTIPLICATION D'UNE MATRICE  
RESTRUCTURÉE  $n \times 7$  PAR UN VECTEUR COLONNE



Le domaine en éléments finis est couvert par P noeuds dans la direction x et par M noeuds dans la direction y, le produit  $P \times M = N$  représentant le nombre total de noeuds. Alors le produit

$$[K]_{n \times 7} (H)_{n \times 1} = (B)_{n \times 1}$$

est donné par l'algorithme (où  $n=N$ ) :

```

_____
I1 = M + 1
I2 = M + 2
B(1) = K(1,4)*H(1) + K(1,5)*H(2) + K(1,6)*H(I1)
1      + K(1,7)*H(I2)
J = 1
DO I=2,M
    J1 = J + 1
    J2 = J + 2
    JI1 = J + I1
    JI2 = J + I2
    B(I) = K(I,3)*H(J) + K(I,4)*H(J1) + K(I,5)*H(J2)
1        + K(I,6)*H(JI1) + K(I,7)*H(JI2)
    J = J + 1
END DO
I3 = 2 * M + 1
I4 = I3 + 1
B(I1) = K(I1,2)*H(1) + K(I1,3)*H(M) + K(I1,4)*H(I1)
1      + K(I1,5)*H(I2) + K(I1,6)*H(I3) + K(I1,7)*H(I4)
L = N - I1
J = 1
DO I=I2,L
    J2 = J + 1

```

```

      J3 = J + M
      J4 = J3 + 1
      J5 = J4 + 1
      J6 = J5 + M - 1
      J7 = J6 + 1
      B(I) = K(I,1)*H(J) + K(I,2)*H(J2) + K(I,3)*H(J3)
1          + K(I,4)*H(J4) + K(I,5)*H(J5) + K(I,6)*H(J6)
2          + K(I,7)*H(J7).
      J = J + 1
END DO
L = L + 1
J2 = J + 1
J3 = J + M
J4 = J3 + 1
J5 = J4 + 1
J6 = N
B(L) = K(L,1)*H(J) + K(L,2)*H(J2) + K(L,3)*H(J3)
1      + K(L,4)*H(J4) + K(L,5)*H(J5) + K(L,6)*H(J6)
L = L + 1
N1 = N - 1
I1 = N - 2 * M
DO I=L,N1
      I2 = I1 + 1
      J3 = I1 + M
      J4 = J3 + 1
      J5 = J4 + 1
      B(I) = K(I,1)*H(I1) + K(I,2)*H(I2) + K(I,3)*H(J3)
1          + K(I,4)*H(J4) + K(I,5)*H(J5)
      I1 = I1 + 1
END DO
J1 = N - M - 1
J2 = J1 + 1
B(N) = K(N,1)*H(J1) + K(N,2)*H(J2) + K(N,3)*H(N1)
1      + K(N,4)*H(N)

```

---

ANNEXE 1

LISTAGE DU PROGRAMME

```

PROGRAM FUSION
C
C=====C
C
C   CALCUL DE LA DISTRIBUTION DE TEMPERATURE DANS
C   UN BAIN D'ALUMINIUM EN FUSION.
C
C   LE DOMAINE EST RECTANGULAIRE, ISOLE SUR LES
C   COTES ET DANS LE FOND. UNE TEMPERATURE EST
C   IMPOSEE SUR LE DESSUS.
C
C   L'EQUATION D'ENERGIE EST SOLUTIONNEE PAR LA
C   METHODE DES ELEMENTS FINIS DE TYPE GALERKIN
C   UTILISANT DES ELEMENTS TRIANGULAIRES LINEAIRES.
C
C   LES EQUATIONS DU MOUVEMENT SONT INTEGREES EN
C   DIFFERENCES FINIES PAR LE LOGICIEL SOLA.
C
C=====C
C
C   Ecrit par: Jocelyn Tremblay, etudiant (M.R.S.)
C   -----
C
C   sous la supervision des professeurs:
C   R.T. Bui et A. Charette
C   -----
C
C   de l'Universite du Quebec a Chicoutimi.
C
C=====C
C
DO I=1,2
  PRINT *,CHAR(12)
END DO
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
PRINT *,'
DO I=1,5
  PRINT *
END DO
C
C-----
C
CALL CPU
C
C-----
C
CALL BLOC_PRE ( M , NELEM )

```

```
CALL BLOC_CALCUL ( M , NELEM )
CALL BLOC_SORTIE ( M )
```

89

```
C
C-----
C
```

```
CALL CPU
WRITE (6,10)
```

```
C
C-----
C
```

```
10      FORMAT (3(/), ' *****',/,
+          ' ***  FIN DU PROGRAMME FUSION  ***',/,
+          ' *****',3(/))
```

```
C
C-----
C
```

```
END
```

```
SUBROUTINE BLOC_PRE ( M , NELEM )
```

```
C
C=====
```

```
C
C      ROUTINE DE PREPARATION POUR LES CALCULS
```

```
C
C=====
```

```
C
C      IMPLICIT REAL*4 (A-H,O-Z)
C      IMPLICIT INTEGER*4 (I-N)
C      REAL*4 T(740)
C      INTEGER*4 FLAG
```

```
C
C-----
C
```

```
FLAG = 0
CALL POSITIONS ( M )
CALL INIT ( T , FLAG , NELEM , M )
IF (FLAG .EQ. 1) GOTO 10
CALL ENTHALPIE ( T , M )
```

```
C
C-----
C
```

```
10      RETURN
END
```

```
SUBROUTINE BLOC_CALCUL ( M , NELEM )
```

```
C
C=====
```

```
C
C      ROUTINE CALCULANT L'ENTHALPIE ET LE CHAMP DE VITESSE
C      A CHAQUE ITERATION.
```

```
C
C=====
```

```
C
C      IMPLICIT REAL*4 (A-H,O-Z)
```

```
IMPLICIT INTEGER*4 (I-N)
INTEGER*4 CONVECTION
```

90

```
C
C-----
C
OPEN (1,FILE='DONNEES.DAT',STATUS='OLD')
READ (1,*) XL,YL,KX,KY
READ (1,*) TEMPS
READ (1,*) DT
READ (1,*) I,TSOL,TLIQ,TFRONT
READ (1,*) CONVECTION
CLOSE (1)
N = JNINT((TEMPS / DT))
```

```
C
C-----
C
DO I=1,N
  L = I
  CALL ENERGIE ( M , NELEM , CONVECTION , DT , L )
  IF (CONVECTION .EQ. 1) CALL MOUVEMENT ( M , DT )
END DO
```

```
C
C-----
C
RETURN
END
```

```
SUBROUTINE BLOC_SORTIE ( M )
```

```
C
C=====
C
TRANSFORMATION DES ENTHALPIES EN TEMPERATURES.
C
ECRITURE DES RESULTATS.
C
C=====
C
IMPLICIT REAL*4 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL *4 T(740)
```

```
C
C-----
C
CALL TEMPERATURE ( T , M )
CALL ECRITURE ( T , M )
```

```
C
C-----
C
RETURN
END
```

```
SUBROUTINE POSITIONS ( M )
```

```
C
C=====
C
```

C COORDONNEES DES NOEUDS DU MAILLAGE EN ELEMENTS FINIS.

C  
C-----  
C

```

IMPLICIT INTEGER*4 (I-N)
IMPLICIT REAL*4 (A-H,O-Z)
COMMON /POSI/ X(740),Y(740)
COMMON /COORD/ XL,YL,KX,KY
    
```

C  
C-----  
C

```

OPEN (1,FILE='DONNEES.DAT',STATUS='OLD')
READ (1,*) XL,YL,KX,KY
CLOSE (1)
M = KX * KY
IF ( M .GT. 740 ) THEN
    WRITE ( 5 , 10 )
    STOP
END IF
    
```

C  
C-----  
C

```

XX = 0.0
N = 1
DX = XL / FLOAT(KX-1)
DY = YL / FLOAT(KY-1)
    
```

C  
C-----  
C

```

DO 300 I=1,KX
YY = 0.0
DO 200 J=1,KY
X(N) = XX
Y(N) = YY
N = N + 1
YY = YY + DY
200 CONTINUE
XX = XX + DX
300 CONTINUE
    
```

C  
C-----  
C

```

10 FORMAT (3(/),1X,' **** LA DIMENSION DES MATRICES N'EST PAS
+ ASSEZ GRANDE. ',3(/),1X,' **** FIN DE L'EXECUTION.',3(/))
100 FORMAT(6X,I3,9X,F7.4,9X,F7.4)
CLOSE (1)
    
```

C  
C-----  
C

```

RETURN
END
    
```

SUBROUTINE INIT ( T , FLAG , NELEM , M )

C  
C-----  
C

```

INITIALISATION POUR LES CALCULS.
CALCUL DE LA MATRICE GLOBALE (C)
    
```

```

C
C
C-----
C
IMPLICIT REAL *4 (A-H,O-Z)
IMPLICIT INTEGER *4 (I-N)
REAL*4 T(740),C(740,7)
INTEGER *4 FLAG,N(3)
CHARACTER*2 OPTION
COMMON /POSI/ X(740),Y(740)
COMMON /ENTHA/ H(740) /NAV/ XPUT(6)
COMMON /UVP/ U(38,21),V(38,21),P(38,21)
COMMON /MATVIT/ VIT(2*740)
COMMON /MATCINV/ CINV(740) /MATC/ C
COMMON /CONNEX/ IBAN,SURF,N1(1368),N2(1368),N3(1368)
COMMON /CHOIX/ OPTION
DATA ZERO/0.0/ , UN/1.C/

```

```

C
C-----
C
OPEN (1,FILE='DOMNEES.DAT',STATUS='OLD')
READ (1,*) XL , YL , KX , KY
READ (1,*) TEMPS
READ (1,*) DT
READ (1,*) I , TSOL , TINIT , TSURF
READ (1,*) ICONVEC
READ (1,*) OMEGA , ALPHA
READ (1,*) WL , WR
READ (1,5) OPTION
CLOSE (1)
XPUT(3) = OMEGA
XPUT(4) = ALPHA
XPUT(5) = WL
XPUT(6) = WR
IF (OPTION .NE. 'BL' .AND. OPTION .NE. 'BQ' .AND.
+   OPTION .NE. 'DL' .AND. OPTION .NE. 'DQ') THEN
    WRITE (6,7)
    STOP
END IF
TSOL = TSOL + 273.15
TINIT = TINIT + 273.15
TSURF = TSURF + 273.15
IF (I .EQ. 2) GO TO 100

```

```

C
C-----
C
FLAG = 1
OPEN (2,FILE='H.DAT',STATUS='OLD')
C--- LECTURE DES ENTHALPIES INITIALES ---
DO I=1,KX
    READ (2,20)
    DO J=1,KY
        L = (I-1)*KY + J
        READ (2,*) H(L),VIT(2*L-1),VIT(2*L)
    END DO
END DO
CLOSE (2)
C--- LECTURE DES VITESSES ET DES PRESSIONS INITIALES
OPEN (3,FILE='UVP.DAT',STATUS='OLD')
RHO = 2.374E+03
I2 = KX + 1
J2 = KY + 1
DO I=1,I2

```



```

      READ (3,20)
      DO J=1,J2
        READ (3,*) U(I,J),V(I,J),W
        P(I,J) = W / RHO
      END DO
    END DO
  CLOSE (3)

C
C-----
C
20   FORMAT (/)
     GO TO 200

C
C-----
C
100  DO I=1,M
      T(I) = TINIT
    END DO
    DO I=1,M
      IF (ABS( Y(I)-YL ) .LT. 0.0005) T(I) = TSURF
    END DO

C
C-----
C
      OPEN (5,FILE='SOLIDE.DAT',STATUS='UNKNOWN')
      DO I=1,M
        READ (5,*,END=199) J
        T(J) = TSOL
      END DO
198  CLOSE (5)

C
C-----
C
      L = 2 * M
      DO I=1,L
        VIT(I) = ZERO
      END DO
      I2 = KX + 1
      J2 = KY + 1
      DO I=1,I2
        DO J=1,J2
          U(I,J) = 0.0
          V(I,J) = 0.0
          P(I,J) = 0.0
        END DO
      END DO

C
C-----
C
200  IF (OPTION .EQ. 'DL' .OR. OPTION .EQ. 'DC') THEN
      DO I=1,M
        CINV(I) = ZERO
      END DO
    ELSE
      DO I=1,M
        DO J=1,7
          C(I,J) = ZERO
        END DO
      END DO
    END IF

C
C-----
C

```

C LE MAILLAGE

C

C-----

C

```

DX = XL / FLOAT(KX-1)
DY = YL / FLOAT(KY-1)
SURF = ( DX * DY ) / 2.0
NCOM = 1
DO I=1,(KX-1)
  DO J=1,(KY-1)
    NN1 = (I-1)*KY + J
    NN2 = (I-1)*KY + J+1
    NN3 = I * KY + J
    NN4 = I * KY + J+1
    N1(NCOM) = NN1
    N2(NCOM) = NN3
    N3(NCOM) = NN4
    NCOM = NCOM + 1
    N1(NCOM) = NN1
    N2(NCOM) = NN4
    N3(NCOM) = NN2
    NCOM = NCOM + 1
  END DO
END DO

```

C

C-----

C

```
NELEM = NCOM - 1
```

C

C-----

C

```
IBAN = KY - 2
```

C

C-----

C

```

IF (OPTION .EQ. 'DL' .OR. OPTION .EQ. 'DC') THEN
  S = SURF / 3.0
  DO I=1,NELEM
    NN1 = N1(I)
    NN2 = N2(I)
    NN3 = N3(I)
    CINV(NN1) = CINV(NN1) + S
    CINV(NN2) = CINV(NN2) + S
    CINV(NN3) = CINV(NN3) + S
  END DO
  DO I=1,M
    CINV(I) = UN / CINV(I)
  END DO
END IF

```

C

C-----

C

```

IF (OPTION .EQ. '2L' .OR. OPTION .EQ. 'BC') THEN
  DO I=1,NELEM
    NN1 = N1(I)
    NN2 = N2(I)
    NN3 = N3(I)
    C(NN1,4) = C(NN1,4) + SURF / 6.0
    C(NN2,4) = C(NN2,4) + SURF / 6.0
    C(NN3,4) = C(NN3,4) + SURF / 6.0
    L = I
    CALL NOEUDS ( N , L )
    DO I1=1,3

```

```

        J = N(I1)
        DO I2=1,3
            IF (I1 .EQ. I2) GOTO 11
            K = N(I2) - J + 4
            IF (K .GT. 7) K = K - I2*AN
            IF (K .LT. 1) K = K + I2*AN
            C(J,K) = C(J,K) + SURF / 12.0
11      END DO
        END DO
    END DO
END IF

C
C-----
C
5      FORMAT (A3)
7      FORMAT (/, '      C-DIX D'OPTION NON PERMISE',/,
+      '      ARRET DE L'EXECUTION DANS INIT',3(/))

C
C-----
C
        RETURN
        END
    
```

```

        SUBROUTINE ENTHALPIE ( T , M )
C
C=====
C
C      CALCUL DE L'ENTHALPIE A PARTIR DE LA TEMPERATURE.
C
C=====
C
        IMPLICIT REAL*4 (A-H,O-Z)
        IMPLICIT INTEGER*4 (I-N)
        REAL*4 T(740)
        COMMON /ENTHA/ H(740),H0(740),H1(740),H2(740),H3(740)
C
C-----
C
        DO I=1,M
            IF (T(I) .LT. 933.15) THEN
                H(I) = -0.04 * T(I)*T(I)*T(I) + 652.0 * T(I)*T(I) +
+                2.04E6 * T(I) - 2.44E9
                GOTO 10
            END IF
            IF (T(I) .GT. 933.15) THEN
                H(I) = -214.0 * T(I)*T(I) + 3.0E6 * T(I) -
+                2.51E9 + 9.53E8
                GOTO 10
            END IF
            PRINT *
            PRINT *, ' ***** ERREUR ***** '
            PRINT *, ' IL N'Y A PAS DE VALEUR D'ENTHALPIE '
            PRINT *, ' CORRESPONDANT A 660 "C . '
            PRINT *, ' ***** '
            PRINT *
            WRITE (6,20) I
            STOP
10      END DO
        FORMAT (4X, 'ERREUR DECELEE AU NOEUD : ',I3,/,/)
    
```

RETURN  
END

SUBROUTINE TEMPERATURE ( T , M )

```

C
C=====
C
C      CALCUL DE LA TEMPERATURE A PARTIR DE L'ENTHALPIE.
C
C=====
C
C      IMPLICIT REAL*4 (A-H,D-Z)
C      IMPLICIT INTEGER*4 (I-N)
C      REAL*4 T(740)
C      COMMON /ENTHA/ H(740)
C
C-----
C
C      DO I=1,M
C        IF (H(I) .LT. 0.0) THEN
C          T(I) = 2.77E-27 * H(I)*H(I)*H(I) - 2.01E-17*H(I)*H(I) +
+          3.13E-7 * H(I) + 932.5
C          GOTO 10
C        END IF
C        IF (H(I) .GT. 9.53E8) THEN
C          T(I) = 2.345E-27 * H(I)*H(I)*H(I) + 2.563E-18*H(I)*H(I) +
+          3.744E-7 * H(I) + 570.7
C          GOTO 10
C        END IF
C        T(I) = 933.15
C      END DO
10
C-----
C
C      RETURN
C      END

```

SUBROUTINE ECRITLRE ( T , M )

```

C
C=====
C
C      ECRITURE DES TEMPERATURES NODALES DANS LE
C      FICHER : TEMPERATURES.DAT
C
C      ECRITURE DE L'ENTHALPIE ET DES VITESSES NODALES
C      DANS LE FICHER : H.DAT
C
C      ECRITURE DES VITESSES ET DES PRESSIONS DANS
C      LE FICHER : UVP.DAT
C
C=====

```

```

C
  IMPLICIT REAL *4 (A-H,O-Z)
  IMPLICIT INTEGER *4 (I-N)
  REAL *4 T(740)
  INTEGER*4 FICTIF(740)
  CHARACTER*1 D(5)
  CHARACTER*22 V
  COMMON /ENTHA/ H(740)
  COMMON /CODRD/ XL,YL,KX,KY
  COMMON /MATVIT/ VIT(2*740)
  COMMON /UVP/ UU(38,21),UV(38,21),P(38,21)
  COMMON /NDEUDSF/ N,NOUVN(740),FICTIF

```

```

C
C-----
C
  OPEN (1,FILE="DONNEES.DAT",STATUS="OLD")
  DO I=1,8
    READ (1,*)
  END DO
  READ (1,*) SXG,SXD
  CLOSE (1)

```

```

C
C-----
C
  DO I=1,M
    T(I) = T(I) - 273.15
  END DO

```

```

C
C-----
C
  OPEN (1,FILE="TEMPERATURES.DAT",STATUS="UNKNOWN")
  DO I=1,KX
    WRITE(1,10) (T((I-1)*KY+J),J=1,KY)
  END DO
  DO I=1,5
    WRITE(1,*)
  END DO

```

```

C
C-----
C
  IF (ABS(SXD-XL) .LT. 0.0002) THEN
    D(1) = '+'
    D(2) = '-'
    D(3) = 'I'
    D(4) = '\'
    D(5) = ' '
    CALL DCMAINE ( M )
    V = D(1)
    DO I=2,KY-1
      L = I-1
      IF (FICTIF(I).NE.0 .AND. FICTIF(I-1).EQ.0) THEN
        V = V (:L) // D(1)
      ELSE
        V = V (:L) // D(2)
      END IF
    END DO
    V = V (:KY-1) // D(1)
    WRITE (1,*) V
    DO I=2,KX-1
      L = (I-1)*KY + 1
      IF (FICTIF(L) .NE. 0 .AND. FICTIF(L+KY) .EQ. 0) THEN
        V = D(1)
      ELSE

```

```

      V = D(3)
      END IF
      DO J=2,KY-1
        L = (I-1)*KY + J
        IF (FICTIF(L+KY).NE.0 .AND. FICTIF(L+KY-1).NE.0)
+           V = V (:J-1) // D(5)
        IF (FICTIF(L).EQ. 0) THEN
          V = V (:J-1) // D(5)
          GOTO 500
        END IF
        IF (FICTIF(L-1).EQ.0 .AND. FICTIF(L+KY).EQ.0 .AND.
+         FICTIF(L+KY+1).NE.0) THEN
          V = V (:J-1) // D(4)
          GOTO 600
        END IF
        IF (FICTIF(L-1-KY).NE.0 .AND. FICTIF(L-1).EQ.0 .AND.
+         FICTIF(L+KY).EQ.0) THEN
          V = V (:J-1) // D(1)
          GOTO 500
        END IF
        IF (FICTIF(L-1).NE.0 .AND. FICTIF(L+KY).EQ.0
+         .AND. FICTIF(L+KY+1).NE.0) THEN
          V = V (:J-1) // D(1)
          GOTO 500
        END IF
        IF (FICTIF(L-1).NE.0 .AND. FICTIF(L+KY-1).EQ.0 .AND.
+         FICTIF(L+KY).NE.0 .AND. FICTIF(L).LT.0) THEN
          V = V (:J-1) // D(1)
          GOTO 600
        END IF
        IF (FICTIF(L-KY).NE.0 .AND. FICTIF(L-1).EQ.0 .AND.
+         FICTIF(L+KY).NE.0) THEN
          V = V (:J-1) // D(3)
          GOTO 600
        END IF
        IF (FICTIF(L-KY-1).EQ.0 .AND. FICTIF(L+KY).EQ.0 .AND.
+         FICTIF(L+KY+1).EQ.0) THEN
          V = V (:J-1) // D(1)
          GOTO 500
        END IF
        IF (FICTIF(L-1).NE.0 .AND. FICTIF(L+1).NE.0 .AND.
+         FICTIF(L+KY).EQ.0) V = V (:J-1) // D(2)
500      END DO
600      IF (FICTIF(L+1).EQ.0 .AND. FICTIF(L+1-KY).NE.0) THEN
        V = V (:KY-1) // D(1)
      ELSE
        V = V (:KY-1) // D(3)
      END IF
      WRITE (1,*) V
    END DO
    V = D(1)
    DO I=M-KY+2,M-1
      L = I - M + KY - 1
      IF (FICTIF(I).NE.0 .AND. FICTIF(I-1).EQ.0) THEN
        V = V (:L) // D(1)
      ELSE
        V = V (:L) // D(2)
      END IF
    END DO
    V = V (:KY-1) // D(1)
    WRITE (1,*) V
  END IF

```

```

C-----
C
      DO I=1,5
        WRITE(1,*)
      END DO
      CLOSE (1)
      DO I=1,3
        PRINT *
      END DO
      PRINT #,' ==> FICHER TEMPERATURES.DAT CPEE <== '

```

```

C-----
C
      OPEN (2,FILE='H.DAT',STATUS='UNKNOWN')
      DO I=1,KX
        WRITE (2,20)
        DO J=1,KY
          L = (I-1)*KY + J
          WRITE (2,30) H(L),VIT(2*L-1),VIT(2*L)
        END DO
      END DO
      CLOSE (2)
      PRINT #,' ==> FICHER H.DAT CPEE <== '

```

```

C-----
C
      OPEN (3,FILE='UVP.DAT',STATUS='UNKNOWN')
      RHO = 2.374E+03
      I2 = KX + 1
      J2 = KY + 1
      DO I=1,I2
        WRITE (3,20)
        DO J=1,J2
          W = P(I,J) * RHO
          WRITE (3,30) UU(I,J),UV(I,J),W
        END DO
      END DO
      CLOSE (3)
      PRINT #,' ==> FICHER UVP.DAT CREE <== '

```

```

C-----
C
10      FORMAT(2(/),50F6.1)
20      FORMAT(/)
30      FORMAT (3(1X,E13.6))
40      FORMAT (1X,F7.2)

```

```

C-----
C
      RETURN
      END

```

SUBROUTINE DOMAINE ( M )

```

C-----
C
C      DONNE DES NUMEROTS NEGATIFS AUX NOEUDS SITUES A LA
C      FRONTIERE DU LIQUIDE.
C

```

```

C=====
C
      IMPLICIT INTEGER*4 (I-N)
      IMPLICIT REAL*4 (A-H,O-Z)
      INTEGER*4 FICTIF(740)
      COMMON /ENTHA/ H(740)
      COMMON /NOEUDSF/ N,NOUVI.(740),FICTIF
      COMMON /CBDRD/ XL,YL,KX,KY
      COMMON /PSI/ X(740),Y(740)

C
C-----
C
      DO I=1,M
        FICTIF(I) = I
      END DO

C
C-----
C
      I1 = M - KY + 1
      J = I1
      IF (H(J) .GE. 9.53E8) GOTO 40
      NS = 0
10     IF (J .LE. KY) GOTO 20
      IF (H(J-KY) .GE. 9.53E8 .OR. H(J+1) .GE. 9.53E8) THEN
        FICTIF(J) = -FICTIF(J)
        GOTO 20
      END IF
      FICTIF(J) = 0
      J = J - KY
      NS = NS + 1
      GOTO 10
20     I1 = I1 + 1
      M1 = M - 1
      DO I=I1,M1
        J = I
        IF (H(J) .GE. 9.53E8) GOTO 40
        NSC = 0
        DO K=1,NS
          IF (J .LE. KY) GOTO 30
          IF (H(J-KY) .GE. 9.53E8 .OR. H(J+1) .GE. 9.53E8) THEN
            FICTIF(J) = -FICTIF(J)
            GOTO 30
          END IF
          FICTIF(J) = 0
          J = J - KY
          NSC = NSC + 1
        END DO
30     NS = NSC
      END DO

C
C-----
C
40     N = 0
      DO I=1,M
        IF (FICTIF(I) .NE. 0) THEN
          N = N + 1
          NOUVN(N) = I
          IF (FICTIF(I) .LT. 0 .OR.
+         ABS(X(I)) .LT. 0.0001 .OR.
+         ABS(X(I)-XL) .LT. 0.0001 .OR.
+         ABS(Y(I)) .LT. 0.0001 .OR.
+         ABS(Y(I)-YL) .LT. 0.0001) THEN
            FICTIF(I) = -N
          END IF
        END IF
      END DO

```



```

        ELSE
          FICTIF(I) = N
        END IF
      END IF
    END DO

```

```

C
C-----
C
      RETURN
      END

```

SUBROUTINE NOEUDS ( N , L )

```

C
C=====
C
      PLACE LES NUMEROS DES NOEUDS DE L'ELEMENT *L DANS
      LE VECTEUR N.
C=====
C

```

```

      IMPLICIT INTEGER*4 (I-N)
      IMPLICIT REAL*4 (A-H,O-Z)
      INTEGER*4 L,N(3)
      COMMON /CDNEC/ IBAN,SURF,N1(1368),N2(1368),N3(1368)

```

```

C
C-----
C
      N(1) = N1(L)
      N(2) = N2(L)
      N(3) = N3(L)

```

```

C
C-----
C
      RETURN
      END

```

SUBROUTINE ENERGIE ( M , NELEM , CONVECTION , DT , L )

```

C
C=====
C
      RESOLUTION DE L'EQUATION D'ENERGIE.
C=====
C

```

```

      INTEGER*4 M , NELEM , CONVECTION
      REAL*4 DT
      CHARACTER*2 OPTION
      COMMON /CHOIX/ OPTION

```

```

C
C-----
C
      CALL MATRK ( M , NELEM , CONVECTION )
      IF (OPTION.EQ.'BC'.OR.OPTION.EQ.'DQ') CALL MATRC (M,NELEM)
      CALL MATPRH ( M , NELEM )

```

C

```

C-----
C
IF (OPTION .EQ. 'DL' .OR. OPTION .EQ. 'DD' ) THEN
  CALL RUNGEKUT ( M , DT )
END IF
IF (OPTION.EQ.'BL') CALL CRANKNICHOL ( M , DT )
IF (OPTION.EQ.'BC') THEN
  IF (L .EQ. 1) CALL SOL1 ( M , DT )
  IF (L .EQ. 2) CALL SOL2 ( M , DT )
  IF (L .EQ. 3) CALL SOL3 ( M , DT )
  IF (L .GE. 4) CALL PREDICOR ( M , DT , L )
END IF

```

```

C
C-----
C
RETURN
END

```

```

SUBROUTINE MATPK ( M , NELEM , CONVECTION )

```

```

C-----
C
C      CALCUL DE      (K) = (K ) + (K )
C                      C      V
C-----
C

```

```

C
IMPLICIT REAL *4 (A-H,O-Z)
IMPLICIT INTEGER *4 (I-N)
REAL*4 K(740,7)
INTEGER*4 CONVECTION
CHARACTER*2 OPTION
COMMON /CHOIX/ OPTION
COMMON /MATK/ K /MATVIT/ V(2*740)
COMMON /MATCINV/ CINV(740)
COMMON /ENTHA/ H(740) /POSI/ X(740),Y(740)
COMMON /CONNEC/ IBAN,SURF,N1(1368),N2(1368),N3(1368)
DATA ZERO/0.E0/,LN/1.E0/,DFUX/2.E0/,DOUZE/1.2E1/
DATA QUATRE/4.0E0/,VINGTQUAT/24.0E0/

```

```

C-----
C
DO I=1,M
  DO J=1,7
    K(I,J) = ZERO
  END DO
END DO

```

```

C-----
C
S = UN / ( QUATRE * SURF )
DO I=1,NELEM
  NN1 = N1(I)
  NN2 = N2(I)
  NN3 = N3(I)
  X1 = X(NN1)
  X2 = X(NN2)
  X3 = X(NN3)

```

```

Y1 = Y(NN1)
Y2 = Y(NN2)
Y3 = Y(NN3)
IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
    B1 = 3.851E-5
    B2 = 3.851E-5
    B3 = 2.851E-5
ELSE
    B1 = 2.773E-5
    B2 = 2.773E-5
    B3 = 2.773E-5
END IF
IF ( H(NN1) .GE. ZERO .AND. H(NN1) .LE. 9.53E8 ) B1 = ZERO
IF ( H(NN2) .GE. ZERO .AND. H(NN2) .LE. 9.53E8 ) B2 = ZERO
IF ( H(NN3) .GE. ZERO .AND. H(NN3) .LE. 9.53E8 ) B3 = ZERO
IF (H(NN1) .LT. ZERO) THEN
    IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
        B1 = 8.33E-5
    ELSE
        B1 = 6.605E-5
    END IF
END IF
IF (H(NN2) .LT. ZERO) THEN
    IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
        B2 = 8.33E-5
    ELSE
        B2 = 6.605E-5
    END IF
END IF
IF (H(NN3) .LT. ZERO) THEN
    IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
        B3 = 8.33E-5
    ELSE
        B3 = 6.605E-5
    END IF
END IF
K(NN1,4) = K(NN1,4) + S * B1 *
+ ((Y2-Y3)*(Y2-Y3) + (X2-X3)*(X2-X3))
K(NN2,4) = K(NN2,4) + S * B2 *
+ ((Y3-Y1)*(Y3-Y1) + (X3-X1)*(X3-X1))
K(NN3,4) = K(NN3,4) + S * B3 *
+ ((Y1-Y2)*(Y1-Y2) + (X1-X2)*(X1-X2))
J = NN2 - NN1 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
K(NN1,J) = K(NN1,J) + S * B2 *
+ ((Y2-Y3)*(Y3-Y1) + (X2-X3)*(X3-X1))
J = NN3 - NN1 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
K(NN1,J) = K(NN1,J) + S * B3 *
+ ((Y2-Y3)*(Y1-Y2) + (X2-X3)*(X1-X2))
J = NN1 - NN2 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
K(NN2,J) = K(NN2,J) + S * B1 *
+ ((Y3-Y1)*(Y2-Y3) + (X3-X1)*(X2-X3))
J = NN3 - NN2 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
K(NN2,J) = K(NN2,J) + S * B3 *
+ ((Y3-Y1)*(Y1-Y2) + (X3-X1)*(X1-X2))
J = NN1 - NN3 + 4

```

```

      IF (J .GT. 7) J = J - IBAN
      IF (J .LT. 1) J = J + IBAN
      K(NN3,J) = K(NN3,J) + S * B1 *
+      ((Y1-Y2)*(Y2-Y3) + (X1-X2)*(X2-X3))
      J = NN2 - NN3 + 4
      IF (J .GT. 7) J = J - IBAN
      IF (J .LT. 1) J = J + IBAN
      K(NN3,J) = K(NN3,J) + S * B2 *
+      ((Y1-Y2)*(Y3-Y1) + (X1-X2)*(X3-X1))
      END DO

```

```

C
C-----
C

```

```

IF (CONVECTION .EQ. 0) GOTO 1000
S = 1.0 / 24.0
DO I=1,NELEM
  NN1 = N1(I)
  NN2 = N2(I)
  NN3 = N3(I)
  X1 = X(NN1)
  X2 = X(NN2)
  X3 = X(NN3)
  Y1 = Y(NN1)
  Y2 = Y(NN2)
  Y3 = Y(NN3)
  A = Y2 - Y3
  B = X3 - X2
  C = Y3 - Y1
  D = X1 - X3
  E = Y1 - Y2
  F = X2 - X1
  U1 = V(2*NN1-1)
  U2 = V(2*NN2-1)
  U3 = V(2*NN3-1)
  V1 = V(2*NN1)
  V2 = V(2*NN2)
  V3 = V(2*NN3)
  U12 = 2.0*U1 + U2 + U3
  U22 = U1 + 2.0*U2 + U3
  U32 = U1 + U2 + 2.0*U3
  V12 = 2.0*V1 + V2 + V3
  V22 = V1 + 2.0*V2 + V3
  V32 = V1 + V2 + 2.0*V3
  K(NN1,4) = K(NN1,4) + S * (A*U12+E*V12)
  K(NN2,4) = K(NN2,4) + S * (C*U22+D*V22)
  K(NN3,4) = K(NN3,4) + S * (E*U32+F*V32)
  J = NN2 - NN1 + 4
  IF (J .GT. 7) J = J - IBAN
  IF (J .LT. 1) J = J + IBAN
  K(NN1,J) = K(NN1,J) + S * (C*U12+D*V12)
  J = NN3 - NN1 + 4
  IF (J .GT. 7) J = J - IBAN
  IF (J .LT. 1) J = J + IBAN
  K(NN1,J) = K(NN1,J) + S * (E*U12+F*V12)
  J = NN1 - NN2 + 4
  IF (J .GT. 7) J = J - IBAN
  IF (J .LT. 1) J = J + IBAN
  K(NN2,J) = K(NN2,J) + S * (A*U22+B*V22)
  J = NN3 - NN2 + 4
  IF (J .GT. 7) J = J - IBAN
  IF (J .LT. 1) J = J + IBAN
  K(NN2,J) = K(NN2,J) + S * (E*U22+F*V22)
  I = NN1 - NN3 + 4

```

```

      IF (J .GT. 7) J = J - IBAN
      IF (J .LT. 1) J = J + IBAN
      K(NN3,J) = K(NN3,J) + S * (A*U32+E*V32)
      J = NN2 - NN3 + 4
      IF (J .GT. 7) J = J - IBAN
      IF (J .LT. 1) J = J + IBAN
      K(NN3,J) = K(NN3,J) + S * (C*U32+D*V32)

```

```

END DO

```

```

C
C-----
C
1000  IF (OPTION .EQ. 'DL' .OR. OPTION .EQ. 'DC') THEN
      DO I=1,M
        DO J=1,7
          K(I,J) = K(I,J) * CINV(I)
        END DO
      END DO
END IF

```

```

C
C-----
C
      RETURN
      END

```

```

SUBROUTINE MATRG ( M , NELEM )

```

```

C
C=====
C
      CALCUL DE LA MATRICE (G)
C
C=====
C

```

```

      IMPLICIT REAL *4 (A-H,O-Z)
      IMPLICIT INTEGER *4 (I-N)
      REAL*4 G(740,7)
      CHARACTER*2 OPTION
      COMMON /CHDIX/ OPTION /MATCINV/ CINV(740)
      COMMON /MATG/ G /MATVIT/ V(2*740)
      COMMON /ENTHA/ H(740) /PDSI/ X(740),Y(740)
      COMMON /CONNEC/ IBAN,SURF,N1(1368),N2(1368),N3(1368)
      DATA ZERO/0.E0/,LN/1.E0/,DEUX/2.E0/,DOUZE/1.251/
      DATA QUATRE/4.0E0/,VINGTQUAT/24.0E0/

```

```

C
C-----
C
      DO I=1,M
        DO J=1,7
          G(I,J) = ZERO
        END DO
      END DO

```

```

C
C-----
C
      S = UN / ( QUATRE * SURF )
      DO I=1,NELEM
        NN1 = N1(I)
        NN2 = N2(I)
        NN3 = N3(I)

```

```

X1 = X(NN1)
X2 = X(NN2)
X3 = X(NN3)
Y1 = Y(NN1)
Y2 = Y(NN2)
Y3 = Y(NN3)
B1 = 3.768E-15
B2 = 3.768E-15
B3 = 3.768E-15
IF ( H(NN1) .GE. ZERO .AND. H(NN1) .LE. 9.53E8 ) B1 = ZERO
IF ( H(NN2) .GE. ZERO .AND. H(NN2) .LE. 9.53E8 ) B2 = ZERO
IF ( H(NN3) .GE. ZERO .AND. H(NN3) .LE. 9.53E8 ) B3 = ZERO
IF (H(NN1) .LT. ZERO) B1 = -8.938E-15
IF (H(NN2) .LT. ZERO) B2 = -8.938E-15
IF (H(NN3) .LT. ZERO) B3 = -8.938E-15
G(NN1,4) = G(NN1,4) + S * B1 *
+ ((Y2-Y3)*(Y2-Y3) + (X2-X3)*(X2-X3))
G(NN2,4) = G(NN2,4) + S * B2 *
+ ((Y3-Y1)*(Y3-Y1) + (X3-X1)*(X3-X1))
G(NN3,4) = G(NN3,4) + S * B3 *
+ ((Y1-Y2)*(Y1-Y2) + (X1-X2)*(X1-X2))
J = NN2 - NN1 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN1,J) = G(NN1,J) + S * B2 *
+ ((Y2-Y3)*(Y3-Y1) + (X2-X3)*(X3-X1))
J = NN3 - NN1 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN1,J) = G(NN1,J) + S * B3 *
+ ((Y2-Y3)*(Y1-Y2) + (X2-X3)*(X1-X2))
J = NN1 - NN2 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN2,J) = G(NN2,J) + S * B1 *
+ ((Y3-Y1)*(Y2-Y3) + (X3-X1)*(X2-X3))
J = NN3 - NN2 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN2,J) = G(NN2,J) + S * B2 *
+ ((Y3-Y1)*(Y1-Y2) + (X3-X1)*(X1-X2))
J = NN1 - NN3 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN3,J) = G(NN3,J) + S * B1 *
+ ((Y1-Y2)*(Y2-Y3) + (X1-X2)*(X2-X3))
J = NN2 - NN3 + 4
IF (J .GT. 7) J = J - IBAN
IF (J .LT. 1) J = J + IBAN
G(NN3,J) = G(NN3,J) + S * B2 *
+ ((Y1-Y2)*(Y3-Y1) + (X1-X2)*(X3-X1))
END DO

```

```

C
C-----
C

```

```

IF (OPTION .EQ. 'DQ') THEN
  DO I=1,M
    DO J=1,7
      G(I,J) = G(I,J) * CINV(I)
    END DO
  END DO
END DO
END IF

```

```

C

```

C-----  
C

RETURN  
END

107

SUBROUTINE MATRRF ( M , NELEM )

C  
C-----  
C  
C        CALCUL DE LA MATRICE ( R ).  
C    H  
C-----  
C

IMPLICIT REAL \*4 (A-H,D-Z)  
IMPLICIT INTEGER \*4 (I-N)  
CHARACTER\*2 OPTION  
COMMON /MATR/ R(740) /CHCIX/ OPTION  
COMMON /MATCINV/ CINV(740)  
COMMON /ENTHA/ H(740) /POSI/ X(740),Y(740)  
COMMON /CONNEC/ IBAN,SURF,N1(1368),N2(1368),N3(1368)  
DATA ZERO /0.0/ , UN /1.0/ , QUATRE /4.0/

C  
C-----  
C

DO I=1,M  
  R(I) = ZERO  
END DO

C  
C-----  
C

S = UN / ( QUATRE \* SURF )  
DO I=1,NELEM  
  NN1 = N1(I)  
  NN2 = N2(I)  
  NN3 = N3(I)  
  X1 = X(NN1)  
  X2 = X(NN2)  
  X3 = X(NN3)  
  Y1 = Y(NN1)  
  Y2 = Y(NN2)  
  Y3 = Y(NN3)  
  IF ( OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL' ) THEN  
    B1 = -37200.0  
    B2 = -37200.0  
    B3 = -37200.0  
  ELSE  
    B1 = -29780.0  
    B2 = -29730.0  
    B3 = -29780.0  
  END IF  
  IF ( H(NN1) .GE. ZERO .AND. H(NN1) .LE. 9.53E8 ) B1 = ZERO  
  IF ( H(NN2) .GE. ZERO .AND. H(NN2) .LE. 9.53E8 ) B2 = ZERO  
  IF ( H(NN3) .GE. ZERO .AND. H(NN3) .LE. 9.53E8 ) B3 = ZERO  
  IF ( H(NN1) .LT. ZERO ) THEN  
    IF ( OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL' ) THEN  
      B1 = 5400.0  
    ELSE  
      B1 = -152.0

```

      END IF
      END IF
      IF (H(NN2) .LT. ZERO) THEN
        IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
          B2 = 5400.0
        ELSE
          B2 = -152.0
        END IF
      END IF
      IF (H(NN3) .LT. ZERO) THEN
        IF (OPTION .EQ. 'BL' .OR. OPTION .EQ. 'DL') THEN
          B3 = 5400.0
        ELSE
          B3 = -152.0
        END IF
      END IF
      R(NN1) = R(NN1) + S * (
+      B1*((Y2-Y3)*(Y2-Y3)+(X2-X3)*(X2-X3)) +
+      B2*((Y2-Y3)*(Y3-Y1)+(X3-X2)*(X1-X3)) +
+      B3*((Y2-Y3)*(Y1-Y2)+(X2-X3)*(X1-X2)) )
      R(NN2) = R(NN2) + S * (
+      B1*((Y3-Y1)*(Y2-Y3)+(X3-X1)*(X2-X3)) +
+      B2*((Y3-Y1)*(Y3-Y1)+(X3-X1)*(X3-X1)) +
+      B3*((Y3-Y1)*(Y1-Y2)+(X3-X1)*(X1-X2)) )
      R(NN3) = R(NN3) + S * (
+      B1*((Y2-Y3)*(Y1-Y2)+(X2-X3)*(X1-X2)) +
+      B2*((Y1-Y2)*(Y3-Y1)+(X1-X2)*(X3-X1)) +
+      B3*((Y1-Y2)*(Y1-Y2)+(X1-X2)*(X1-X2)) )
      END DO
C
C-----
C
      IF (OPTION .EQ. 'DL' .OR. OPTION .EQ. 'DC') THEN
        DO I=1,M
          R(I) = R(I) * CINV(I)
        END DO
      END IF
C
C-----
C
      RETURN
      END

```

```

      SUBROUTINE RUNGEKUT ( N , DT )

```

```

C
C=====
C
C      INTEGRATION DU SYSTEME D'EQUATIONS :
C
C
C          .
C          -1
C          (H) = (C) (R) - (K)(H) - (G)(H )
C                      H
C
C=====
C
      IMPLICIT REAL *4 (A-H,D-Z)
      IMPLICIT INTEGER *4 (I-N)

```



```

PEAL *4 H1(740),H2(740),H3(740),H4(740),H5(740),H6(740)
REAL*4 K(740,7),G(740,7)
CHARACTER*2 OPTICN
COMMON /MATK/ K /MATG/ G
COMMON /MATR/ P(740) /CHOIX/ OPTICN
COMMON /POSI/ X(740),Y(740) /ENTHA/ H(740)
COMMON /COORD/ XL,YL,KY,KY
DATA C1/7.777778E-2/ , C3/3.555556E-1/ , C4/1.333333E-1/

```

109

```

C
C-----
C

```

```

IF (OPTION .EQ. 'DL') THEN
  DO I=1,N
    DO J=1,7
      G(I,J) = 0.0
    END DO
  END DO
END IF

```

```

C
C-----
C

```

```

I1 = KY + 1
I2 = KY + 2
A1 = H(1)
A2 = H(2)
A3 = H(I1)
A4 = H(I2)
H1(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+       G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4
+       + R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H(J)
  A2 = H(J1)
  A3 = H(J2)
  A4 = H(JI1)
  A5 = H(JI2)
  H1(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+         K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+         G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+         + R(I)
  J = J + 1
END DO
A1 = H(1)
A2 = H(KY)
A3 = H(I1)
A4 = H(I2)
A5 = H(2*KY+1)
A6 = H(2*KY+2)
H1(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+         K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+         G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+         G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+         + R(I1)
L = N - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY

```

```

      J3 = J2 + 1
      J4 = J3 + 1
      J5 = J4 + KY - 1
      J6 = J5 + 1
      A1 = H(J)
      A2 = H(J1)
      A3 = H(J2)
      A4 = H(J3)
      A5 = H(J4)
      A6 = H(J5)
      A7 = H(J6)
      H1(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+           G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+           G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+           G(I,7)*A7*A7
+           + P(I)
      J = J + 1
      END DO
      L = L + 1
      A1 = H(J)
      A2 = H(J+1)
      A3 = H(J+KY)
      A4 = H(J+KY+1)
      A5 = H(J+KY+2)
      A6 = H(N)
      H1(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+           K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 + G(L,2)*A2*A2 +
+           G(L,3)*A3*A3 + G(L,4)*A4*A4 + G(L,5)*A5*A5 +
+           G(L,6)*A6*A6
+           + R(L)
      L = L + 1
      N1 = N - 1
      I1 = N - KY - KY
      DO I=L,N1
        I2 = I1 + 1
        J1 = I1 + KY
        J2 = J1 + 1
        J3 = J2 + 1
        A1 = H(I1)
        A2 = H(I2)
        A3 = H(J1)
        A4 = H(J2)
        A5 = H(J3)
        H1(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+           G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+           + R(I)
        I1 = I1 + 1
      END DO
      A1 = H(N-KY-1)
      A2 = H(N-KY)
      A3 = H(N-1)
      A4 = H(N)
      H1(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+           G(N,1)*A1*A1 + G(N,2)*A2*A2 + G(N,3)*A3*A3 +
+           G(N,4)*A4*A4
+           + P(N)
      DO I=1,N
        IF (ABS(Y(I)-YL) .LT. 0.0005) H1(I) = 0.0
      END DO

```

C  
C-----

```

S = DT / 4.0
I1 = KY + 1
I2 = KY + 2
A1 = H(I1) + S*H1(I1)
A2 = H(I2) + S*H1(I2)
A3 = H(I1) + S*H1(I1)
A4 = H(I2) + S*H1(I2)
H2(I) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+       G(1,4)*A1*A1 + G(1,5)*A2*A2 +
+       G(1,6)*A3*A3 + G(1,7)*A4*A4
+       + R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H(J) + S*H1(J)
  A2 = H(J1) + S*H1(J1)
  A3 = H(J2) + S*H1(J2)
  A4 = H(JI1) + S*H1(JI1)
  A5 = H(JI2) + S*H1(JI2)
  H2(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+       K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+       G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+       + R(I)
  J = J + 1
END DO
A1 = H(1) + S*H1(1)
A2 = H(KY) + S*H1(KY)
A3 = H(I1) + S*H1(I1)
A4 = H(I2) + S*H1(I2)
A5 = H(2*KY+1) + S*H1(2*KY+1)
A6 = H(2*KY+2) + S*H1(2*KY+2)
H2(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+       K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+       G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+       G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+       + R(I1)
L = N - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  A1 = H(J) + S*H1(J)
  A2 = H(J1) + S*H1(J1)
  A3 = H(J2) + S*H1(J2)
  A4 = H(J3) + S*H1(J3)
  A5 = H(J4) + S*H1(J4)
  A6 = H(J5) + S*H1(J5)
  A7 = H(J6) + S*H1(J6)
  H2(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+       K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+       G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+       G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+       G(I,7)*A7*A7
+       + R(I)
  J = J + 1

```

```

END DO
L = L + 1
A1 = H(J) + S*H1(J)
A2 = H(J+1) + S*H1(J+1)
A3 = H(J+KY) + S*H1(J+KY)
A4 = H(J+KY+1) + S*H1(J+KY+1)
A5 = H(J+KY+2) + S*H1(J+KY+2)
A6 = H(N) + S*H1(N)
H2(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+       K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+       G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+       G(L,5)*A5*A5 + G(L,6)*A6*A6
+       R(L)
L = L + 1
N1 = N - 1
I1 = N - KY - KY
DO I=L,N1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H(I1) + S*H1(I1)
  A2 = H(I2) + S*H1(I2)
  A3 = H(J1) + S*H1(J1)
  A4 = H(J2) + S*H1(J2)
  A5 = H(J3) + S*H1(J3)
  H2(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+       K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+       G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+       R(I)
  I1 = I1 + 1
END DO
A1 = H(N-KY-1) + S*H1(N-KY-1)
A2 = H(N-KY) + S*H1(N-KY)
A3 = H(N-1) + S*H1(N-1)
A4 = H(N) + S*H1(N)
H2(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+       G(N,1)*A1*A1 + G(N,2)*A2*A2 + G(N,3)*A3*A3 + G(N,4)*A4*A4
+       R(N)
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.0005) H2(I) = 0.0
END DO

```

```

C
C-----
C

```

```

S1 = DT / 8.0
S2 = S1
I1 = KY + 1
I2 = KY + 2
A1 = H(1) + S1*H1(1) + S2*H2(1)
A2 = H(2) + S1*H1(2) + S2*H2(2)
A3 = H(I1) + S1*H1(I1) + S2*H2(I1)
A4 = H(I2) + S1*H1(I2) + S2*H2(I2)
H3(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+       G(1,4)*A1*A1 + G(1,5)*A2*A2 +
+       G(1,6)*A3*A3 + G(1,7)*A4*A4
+       R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2

```

```

A1 = H(J) + S1*H1(J) + S2*H2(J)
A2 = H(J1) + S1*H1(J1) + S2*H2(J1)
A3 = H(J2) + S1*H1(J2) + S2*H2(J2)
A4 = H(JI1) + S1*H1(JI1) + S2*H2(JI1)
A5 = H(JI2) + S1*H1(JI2) + S2*H2(JI2)
H3(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+      K(I,7)*A5      + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+      G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+      + R(I)
  J = J + 1
END DO
J5 = 2*KY + 1
J6 = J5 + 1
A1 = H(1) + S1*H1(1) + S2*H2(1)
A2 = H(KY) + S1*H1(KY) + S2*H2(KY)
A3 = H(I1) + S1*H1(I1) + S2*H2(I1)
A4 = H(I2) + S1*H1(I2) + S2*H2(I2)
A5 = H(J5) + S1*H1(J5) + S2*H2(J5)
A6 = H(J6) + S1*H1(J6) + S2*H2(J6)
H3(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+      K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+      G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+      G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+      + R(I1)
L = N - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  A1 = H(J) + S1*H1(J) + S2*H2(J)
  A2 = H(J1) + S1*H1(J1) + S2*H2(J1)
  A3 = H(J2) + S1*H1(J2) + S2*H2(J2)
  A4 = H(J3) + S1*H1(J3) + S2*H2(J3)
  A5 = H(J4) + S1*H1(J4) + S2*H2(J4)
  A6 = H(J5) + S1*H1(J5) + S2*H2(J5)
  A7 = H(J6) + S1*H1(J6) + S2*H2(J6)
  H3(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+      K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+      G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+      G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+      G(I,7)*A7*A7
+      + R(I)
  J = J + 1
END DO
L = L + 1
J2 = J + 1
J3 = J + KY
J4 = J3 + 1
J5 = J4 + 1
A1 = H(J) + S1*H1(J) + S2*H2(J)
A2 = H(J2) + S1*H1(J2) + S2*H2(J2)
A3 = H(J3) + S1*H1(J3) + S2*H2(J3)
A4 = H(J4) + S1*H1(J4) + S2*H2(J4)
A5 = H(J5) + S1*H1(J5) + S2*H2(J5)
A6 = H(N) + S1*H1(N) + S2*H2(N)
H3(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+      K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+      G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+      G(L,5)*A5*A5 + G(L,6)*A6*A6

```

```

L = L + 1
N1 = N - 1
I1 = N - KY - KY
DO I=L,N1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H(I1) + S1*H1(I1) + S2*H2(I1)
  A2 = H(I2) + S1*H1(I2) + S2*H2(I2)
  A3 = H(J1) + S1*H1(J1) + S2*H2(J1)
  A4 = H(J2) + S1*H1(J2) + S2*H2(J2)
  A5 = H(J3) + S1*H1(J3) + S2*H2(J3)
  H3(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+         K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+         G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+         + R(I)
  I1 = I1 + 1
END DO
J1 = N - KY - 1
J2 = J1 + 1
J3 = N - 1
A1 = H(J1) + S1*H1(J1) + S2*H2(J1)
A2 = H(J2) + S1*H1(J2) + S2*H2(J2)
A3 = H(J3) + S1*H1(J3) + S2*H2(J3)
A4 = H(N) + S1*H1(N) + S2*H2(N)
H3(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+         G(N,1)*A1*A1 + G(N,2)*A2*A2 + G(N,3)*A3*A3 + G(N,4)*A4*A4
+         + R(N)
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.0005) H3(I) = 0.0
END DO

```

C  
C-----  
C

```

S2 = - DT / 2.0
S3 = DT
I1 = KY + 1
I2 = KY + 2
A1 = H(1) + S2*H2(1) + S3*H3(1)
A2 = H(2) + S2*H2(2) + S3*H3(2)
A3 = H(I1) + S2*H2(I1) + S3*H3(I1)
A4 = H(I2) + S2*H2(I2) + S3*H3(I2)
H4(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+         G(1,4)*A1*A1 + G(1,5)*A2*A2 +
+         G(1,6)*A3*A3 + G(1,7)*A4*A4
+         + R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H(J) + S2*H2(J) + S3*H3(J)
  A2 = H(J1) + S2*H2(J1) + S3*H3(J1)
  A3 = H(J2) + S2*H2(J2) + S3*H3(J2)
  A4 = H(JI1) + S2*H2(JI1) + S3*H3(JI1)
  A5 = H(JI2) + S2*H2(JI2) + S3*H3(JI2)
  H4(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+         K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+         G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+         + R(I)

```

```

      J = J + 1
    END DO
    J5 = 2*KY + 1
    J6 = J5 + 1
    A1 = H(1) + S2*H2(1) + S3*H3(1)
    A2 = H(KY) + S2*H2(KY) + S3*H3(KY)
    A3 = H(I1) + S2*H2(I1) + S3*H3(I1)
    A4 = H(I2) + S2*H2(I2) + S3*H3(I2)
    A5 = H(J5) + S2*H2(J5) + S3*H3(J5)
    A6 = H(J6) + S2*H2(J6) + S3*H3(J6)
    H4(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+           K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+           G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+           G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+           + R(I1)
    L = N - I1
    J = 1
    DO I=I2,L
      J1 = J + 1
      J2 = J + KY
      J3 = J2 + 1
      J4 = J3 + 1
      J5 = J4 + KY - 1
      J6 = J5 + 1
      A1 = H(J) + S2*H2(J) + S3*H3(J)
      A2 = H(J1) + S2*H2(J1) + S3*H3(J1)
      A3 = H(J2) + S2*H2(J2) + S3*H3(J2)
      A4 = H(J3) + S2*H2(J3) + S3*H3(J3)
      A5 = H(J4) + S2*H2(J4) + S3*H3(J4)
      A6 = H(J5) + S2*H2(J5) + S3*H3(J5)
      A7 = H(J6) + S2*H2(J6) + S3*H3(J6)
      H4(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+           G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+           G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+           G(I,7)*A7*A7
+           + R(I)
      J = J + 1
    END DO
    L = L + 1
    J2 = J + 1
    J3 = J + KY
    J4 = J3 + 1
    J5 = J4 + 1
    A1 = H(J) + S2*H2(J) + S3*H3(J)
    A2 = H(J2) + S2*H2(J2) + S3*H3(J2)
    A3 = H(J3) + S2*H2(J3) + S3*H3(J3)
    A4 = H(J4) + S2*H2(J4) + S3*H3(J4)
    A5 = H(J5) + S2*H2(J5) + S3*H3(J5)
    A6 = H(N) + S2*H2(N) + S3*H3(N)
    H4(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+           K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+           G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+           G(L,5)*A5*A5 + G(L,6)*A6*A6
+           + R(L)
    L = L + 1
    N1 = N - 1
    I1 = N - KY - KY
    DO I=L,N1
      I2 = I1 + 1
      J1 = I1 + KY
      J2 = J1 + 1
      J3 = J2 + 1

```

```

A1 = H(I1) + S2*H2(I1) + S3*H3(I1)
A2 = H(I2) + S2*H2(I2) + S3*H3(I2)
A3 = H(J1) + S2*H2(J1) + S3*H3(J1)
A4 = H(J2) + S2*H2(J2) + S3*H3(J2)
A5 = H(J3) + S2*H2(J3) + S3*H3(J3)
H4(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+      K(I,5)*A5      + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+      G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+      + R(I)
I1 = I1 + 1
END DO
J1 = N - KY - 1
J2 = J1 + 1
J3 = N - 1
A1 = H(J1) + S2*H2(J1) + S3*H3(J1)
A2 = H(J2) + S2*H2(J2) + S3*H3(J2)
A3 = H(J3) + S2*H2(J3) + S3*H3(J3)
A4 = H(N) + S2*H2(N) + S3*H3(N)
H4(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+      G(N,1)*A1*A1+G(N,2)*A2*A2+G(N,3)*A3*A3+G(N,4)*A4*A4
+      + R(N)
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.0005) H4(I) = 0.0
END DO

```

```

C
C-----
C

```

```

S1 = 3.0/16.0 * ET
S4 = 9.0/16.0 * ET
I1 = KY + 1
I2 = KY + 2
A1 = H(1) + S1*H1(1) + S4*H4(1)
A2 = H(2) + S1*H1(2) + S4*H4(2)
A3 = H(I1) + S1*H1(I1) + S4*H4(I1)
A4 = H(I2) + S1*H1(I2) + S4*H4(I2)
H5(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+      G(1,4)*A1*A1 + G(1,5)*A2*A2 +
+      G(1,6)*A3*A3 + G(1,7)*A4*A4
+      + R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H(J) + S1*H1(J) + S4*H4(J)
  A2 = H(J1) + S1*H1(J1) + S4*H4(J1)
  A3 = H(J2) + S1*H1(J2) + S4*H4(J2)
  A4 = H(JI1) + S1*H1(JI1) + S4*H4(JI1)
  A5 = H(JI2) + S1*H1(JI2) + S4*H4(JI2)
  H5(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+      K(I,7)*A5      + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+      G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A4*A4
+      + R(I)
  J = J + 1
END DO
J5 = 2*KY + 1
J6 = J5 + 1
A1 = H(1) + S1*H1(1) + S4*H4(1)
A2 = H(KY) + S1*H1(KY) + S4*H4(KY)
A3 = H(I1) + S1*H1(I1) + S4*H4(I1)
A4 = H(I2) + S1*H1(I2) + S4*H4(I2)
A5 = H(J5) + S1*H1(J5) + S4*H4(J5)

```



```

H5(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+      K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+      G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A2*A3 +
+      G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+      + R(I1)
L = N - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  A1 = H(J) + S1*H1(J) + S4*H4(J)
  A2 = H(J1) + S1*H1(J1) + S4*H4(J1)
  A3 = H(J2) + S1*H1(J2) + S4*H4(J2)
  A4 = H(J3) + S1*H1(J3) + S4*H4(J3)
  A5 = H(J4) + S1*H1(J4) + S4*H4(J4)
  A6 = H(J5) + S1*H1(J5) + S4*H4(J5)
  A7 = H(J6) + S1*H1(J6) + S4*H4(J6)
  H5(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+      K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+      G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+      G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+      G(I,7)*A7*A7
+      + R(I)
  J = J + 1
END DO
L = L + 1
J2 = J + 1
J3 = J + KY
J4 = J3 + 1
J5 = J4 + 1
A1 = H(J) + S1*H1(J) + S4*H4(J)
A2 = H(J2) + S1*H1(J2) + S4*H4(J2)
A3 = H(J3) + S1*H1(J3) + S4*H4(J3)
A4 = H(J4) + S1*H1(J4) + S4*H4(J4)
A5 = H(J5) + S1*H1(J5) + S4*H4(J5)
A6 = H(N) + S1*H1(N) + S4*H4(N)
H5(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+      K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+      G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+      G(L,5)*A5*A5 + G(L,6)*A6*A6
+      + R(L)
L = L + 1
N1 = N - 1
I1 = N - KY - KY
DO I=L,N1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H(I1) + S1*H1(I1) + S4*H4(I1)
  A2 = H(I2) + S1*H1(I2) + S4*H4(I2)
  A3 = H(J1) + S1*H1(J1) + S4*H4(J1)
  A4 = H(J2) + S1*H1(J2) + S4*H4(J2)
  A5 = H(J3) + S1*H1(J3) + S4*H4(J3)
  H5(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+      K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+      G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+      + R(I)

```

```

END DO
J1 = N - KY - 1
J2 = J1 + 1
J3 = N - 1
A1 = H(J1) + S1*H1(J1) + S4*H4(J1)
A2 = H(J2) + S1*H1(J2) + S4*H4(J2)
A3 = H(J3) + S1*H1(J3) + S4*H4(J3)
A4 = H(N) + S1*H1(N) + S4*H4(N)
H5(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+ G(N,1)*A1*A1 + G(N,2)*A2*A2 + G(N,3)*A3*A3 + G(N,4)*A4*A4
+ R(N)
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.0005) H5(I) = 0.0
END DO

```

```

C
C-----
C

```

```

S1 = -3.0/7.0 * ET
S2 = 2.0/7.0 * ET
S3 = 12.0/7.0 * ET
S4 = -12.0/7.0 * ET
S5 = 8.0/7.0 * ET
I1 = KY + 1
I2 = KY + 2
A1 = H(1) + S1*H1(1) + S2*H2(1) + S3*H3(1) + S4*H4(1) +
+ S5*H5(1)
A2 = H(2) + S1*H1(2) + S2*H2(2) + S3*H3(2) + S4*H4(2) +
+ S5*H5(2)
A3 = H(I1) + S1*H1(I1) + S2*H2(I1) + S3*H3(I1) +
+ S4*H4(I1) + S5*H5(I1)
A4 = H(I2) + S1*H1(I2) + S2*H2(I2) + S3*H3(I2) +
+ S4*H4(I2) + S5*H5(I2)
H6(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+ G(1,4)*A1*A1 + G(1,5)*A2*A2 +
+ G(1,6)*A3*A3 + G(1,7)*A4*A4
+ R(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H(J) + S1*H1(J) + S2*H2(J) + S3*H3(J) + S4*H4(J) +
+ S5*H5(J)
  A2 = H(J1) + S1*H1(J1) + S2*H2(J1) + S3*H3(J1) +
+ S4*H4(J1) + S5*H5(J1)
  A3 = H(J2) + S1*H1(J2) + S2*H2(J2) + S3*H3(J2) +
+ S4*H4(J2) + S5*H5(J2)
  A4 = H(JI1) + S1*H1(JI1) + S2*H2(JI1) + S3*H3(JI1) +
+ S4*H4(JI1) + S5*H5(JI1)
  A5 = H(JI2) + S1*H1(JI2) + S2*H2(JI2) + S3*H3(JI2) +
+ S4*H4(JI2) + S5*H5(JI2)
  H6(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+ K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+ G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+ R(I)
  J = J + 1
END DO
J5 = 2*KY + 1
J6 = J5 + 1
A1 = H(1) + S1*H1(1) + S2*H2(1) + S3*H3(1) + S4*H4(1) +
+ S5*H5(1)

```



```

+       R(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+       K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+       G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+       G(L,5)*A5*A5 + G(L,6)*A6*A6
+       + R(L)
L = L + 1
N1 = N - 1
I1 = N - KY - KY
DO I=L,N1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H(I1) + S1*H1(I1) + S2*H2(I1) + S3*H3(I1) +
+       S4*H4(I1) + S5*H5(I1)
  A2 = H(I2) + S1*H1(I2) + S2*H2(I2) + S3*H3(I2) +
+       S4*H4(I2) + S5*H5(I2)
  A3 = H(J1) + S1*H1(J1) + S2*H2(J1) + S3*H3(J1) +
+       S4*H4(J1) + S5*H5(J1)
  A4 = H(J2) + S1*H1(J2) + S2*H2(J2) + S3*H3(J2) +
+       S4*H4(J2) + S5*H5(J2)
  A5 = H(J3) + S1*H1(J3) + S2*H2(J3) + S3*H3(J3) +
+       S4*H4(J3) + S5*H5(J3)
  H6(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+       K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+       G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+       + R(I)
  I1 = I1 + 1
END DO
J1 = N - KY - 1
J2 = J1 + 1
J3 = N - 1
A1 = H(J1) + S1*H1(J1) + S2*H2(J1) + S3*H3(J1) +
+       S4*H4(J1) + S5*H5(J1)
A2 = H(J2) + S1*H1(J2) + S2*H2(J2) + S3*H3(J2) +
+       S4*H4(J2) + S5*H5(J2)
A3 = H(J3) + S1*H1(J3) + S2*H2(J3) + S3*H3(J3) +
+       S4*H4(J3) + S5*H5(J3)
A4 = H(N) + S1*H1(N) + S2*H2(N) + S3*H3(N) + S4*H4(N) +
+       S5*H5(N)
H6(N) = K(N,1)*A1 + K(N,2)*A2 + K(N,3)*A3 + K(N,4)*A4 +
+       G(N,1)*A1*A1+G(N,2)*A2*A2+G(N,3)*A3*A3+G(N,4)*A4*A4
+       + R(N)
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.0005) H6(I) = 0.0
END DO
C
C-----
C
DO I=1,N
  H1(I) = -1.0 * (C1*(H1(I)+H6(I))+C3*(H3(I)+H5(I))+C4*H4(I))
END DO
C
C-----
C
DO I=1,N
  H(I) = H(I) + DT * H1(I)
END DO
C
C-----
C
RETURN
END

```

SUBROUTINE CRANKNICHDL ( N , DT )

```

C
C=====
C
C      INTEGRATION DU SYSTEME :
C
C
C      (C)(H) + (K)(H) = (R)
C
C=====
C

```

```

      IMPLICIT INTEGER*4 (I-N)
      IMPLICIT REAL*4 (A-H,O-Z)
      REAL*4 B(740),K(740,7),A(740,7)
      COMMON /MATC/ C(740,7) /MATK/ K
      COMMON /MATR/ R(740)
      COMMON /COORD/ XL,YL,KX,KY
      COMMON /PDSI/ X(740),Y(740)
      COMMON /ENTHA/ H(740)
      DATA DOMIN /1.0E7/

```

```

C
C-----
C
C      L' ORDRE DE LA MATRICE = N
C      LA LARGEUR DE LA BANDE = M
C      LA DEMIE LARGEUR DE BANDE = MS
C
C-----
C

```

```

      MS = KY + 2
      M = 2 * MS - 1
      IF (M .GT. 43) THEN
        WRITE (6,10)
        STOP
      END IF

```

```

C
C-----
C
      S = DT / 2.0
      DO I=1,N
        IF (ABS(Y(I)-YL) .LT. 0.002) R(I) = 0.0
        A(I,1) = C(I,1) - S * K(I,1)
        A(I,2) = C(I,2) - S * K(I,2)
        A(I,3) = C(I,3) - S * K(I,3)
        A(I,4) = C(I,4) - S * K(I,4)
        A(I,5) = C(I,5) - S * K(I,5)
        A(I,6) = C(I,6) - S * K(I,6)
        A(I,7) = C(I,7) - S * K(I,7)
      END DO

```

```

C
C-----
C
      DO I=1,N
        DO J=1,7
          K(I,J) = C(I,J) + S * K(I,J)
        END DO
      END DO

```

C  
C-----  
C

```
I1 = KY + 1
I2 = KY + 2
B(1) = A(1,4)*H(1) + A(1,5)*H(2) + A(1,6)*H(I1) + A(1,7)*H(I2)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  B(I) = A(I,3)*H(J) + A(I,4)*H(J1) + A(I,5)*H(J2) +
+       A(I,6)*H(JI1) + A(I,7)*H(JI2)
  J = J + 1
END DO
B(I1) = A(I1,2)*H(1) + A(I1,3)*H(KY) + A(I1,4)*H(I1) +
+       A(I1,5)*H(I2) + A(I1,6)*H(2*KY+1) + A(I1,7)*H(2*KY+2)
L = N - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  B(I) = A(I,1)*H(J) + A(I,2)*H(J1) + A(I,3)*H(J2) + A(I,4)*
+       H(J3) + A(I,5)*H(J4) + A(I,6)*H(J5) + A(I,7)*H(J6)
  J = J + 1
END DO
L = L + 1
E(L) = A(L,1)*H(J) + A(L,2)*H(J+1) + A(L,3)*H(J+KY) +
+       A(L,4)*H(J+KY+1) + A(L,5)*H(J+KY+2) + A(L,6)*H(N)
L = L + 1
N1 = N - 1
I1 = N - KY - KY
DO I=L,N1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  B(I) = A(I,1)*H(I1) + A(I,2)*H(I2) + A(I,3)*H(J1) +
+       A(I,4)*H(J2) + A(I,5)*H(J3)
  I1 = I1 + 1
END DO
B(N) = A(N,1)*H(N-KY-1) + A(N,2)*H(N-KY) + A(N,3)*H(N-1) +
+       A(N,4)*H(N)
```

C  
C-----  
C

```
DO I=1,N
  IF (ABS(Y(I)-YL) .LT. 0.002) THEN
    B(I) = H(I) * DDMIN
  ELSE
    B(I) = B(I) - DT * R(I)
  END IF
END DO
```

C  
C-----  
C

CALL SLBSN ( N , E , 2 )

C

```

C
DO I=1,N
  IF (ABS(Y(I)-YL) .GT. 0.002) H(I) = B(I)
END DO

```

```

C
C-----
C
10  FORMAT(3(/), ' *** LA BANDE DE LA MATRICE EST TROP GRANDE *** '
+    ,/, ' ARRET DE L'EXECUTION DANS CRANK_NICHOLSON',3(/))

```

```

C
C-----
C
RETURN
END

```

SUBROUTINE SCL1 ( M , DT )

```

C
C=====
C
C    CALCUL DE H1 POUR LA METHODE DE PREDICTION-CORR.
C
C=====
C

```

```

IMPLICIT REAL*4 (A-H,D-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*4 K(740,7),CH0(740)
COMMON /ENTHA/ H(740),H0(740),H1(740)
COMMON /DERIV/ F0(740),F1(740),F2(740),F3(740)
COMMON /PDSI/X(740),Y(740) /CGERR/ XL,YL,KX,KY
COMMON /MATK/ K /MATG/ G(740,7)
COMMON /MATR/ R(740)
COMMON /MATC/ C(740,7)
DATA DOMIN /1.0E7/

```

```

C
C-----
C
DO I=1,M
  H0(I) = H(I)
END DO

```

```

C
C-----
C
I1 = KY + 1
I2 = KY + 2
A1 = H0(I1)
A2 = H0(I2)
A3 = H0(I1)
A4 = H0(I2)
F0(I1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+       G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4
+       + F(1)
J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1

```

```

      JI2 = J + I2
      A1 = H0(J)
      A2 = H0(J1)
      A3 = H0(J2)
      A4 = H0(JI1)
      A5 = H0(JI2)
      F0(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+           K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+           G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+           + R(I)
      J = J + 1
      END DO
      A1 = H0(1)
      A2 = H0(KY)
      A3 = H0(I1)
      A4 = H0(I2)
      A5 = H0(2*KY+1)
      A6 = H0(2*KY+2)
      F0(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+           K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+           G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+           G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+           + R(I1)
      L = M - I1
      J = 1
      DO I=I2,L
        J1 = J + 1
        J2 = J + KY
        J3 = J2 + 1
        J4 = J3 + 1
        J5 = J4 + KY - 1
        J6 = J5 + 1
        A1 = H0(J)
        A2 = H0(J1)
        A3 = H0(J2)
        A4 = H0(J3)
        A5 = H0(J4)
        A6 = H0(J5)
        A7 = H0(J6)
        F0(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+           G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+           G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+           G(I,7)*A7*A7
+           + R(I)
        J = J + 1
      END DO
      L = L + 1
      A1 = H0(J)
      A2 = H0(J+1)
      A3 = H0(J+KY)
      A4 = H0(J+KY+1)
      A5 = H0(J+KY+2)
      A6 = H0(M)
      F0(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+           K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+           G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+           G(L,5)*A5*A5 + G(L,6)*A6*A6
+           + R(L)
      L = L + 1
      M1 = M - 1
      I1 = M - KY - KY
      DO I=L,M1

```



```

--      --
J1 = I1 + KY
J2 = J1 + 1
J3 = J2 + 1
A1 = H0(I1)
A2 = H0(I2)
A3 = H0(J1)
A4 = H0(J2)
A5 = H0(J3)
F0(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+       K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+       G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+       + R(I)
      I1 = I1 + 1
      END DO
      A1 = H0(M-KY-1)
      A2 = H0(M-KY)
      A3 = H0(M-1)
      A4 = H0(M)
      F0(M) = K(M,1)*A1 + K(M,2)*A2 + K(M,3)*A3 + K(M,4)*A4 +
+           G(M,1)*A1*A1 + G(M,2)*A2*A2 + G(M,3)*A3*A3 +
+           G(M,4)*A4*A4
+           + R(M)
C
C-----
C
      DO I=1,M
      F0(I) = -F0(I)
      END DO
C
C-----
C
      I1 = KY + 1
      I2 = KY + 2
      A1 = H0(1)
      A2 = H0(2)
      A3 = H0(I1)
      A4 = H0(I2)
      CH0(1) = C(1,4)*A1 + C(1,5)*A2 + C(1,6)*A3 + C(1,7)*A4
      J = 1
      DO I=2,KY
      J1 = J + 1
      J2 = J + 2
      JI1 = J + I1
      JI2 = J + I2
      A1 = H0(J)
      A2 = H0(J1)
      A3 = H0(J2)
      A4 = H0(JI1)
      A5 = H0(JI2)
      CH0(I) = C(I,3)*A1 + C(I,4)*A2 + C(I,5)*A3 + C(I,6)*A4 +
+           C(I,7)*A5
      J = J + 1
      END DO
      KY1 = 2*KY + 1
      KY2 = 2*KY + 2
      A1 = H0(1)
      A2 = H0(KY)
      A3 = H0(I1)
      A4 = H0(I2)
      A5 = H0(KY1)
      A6 = H0(KY2)
      CH0(I1) = C(I1,2)*A1 + C(I1,3)*A2 + C(I1,4)*A3 +

```

```

+          C(I1,5)*A4 + C(I1,6)*A5 + C(I1,7)*A6
L = M - I1
J = 1
DO I=I2,L
    J1 = J + 1
    J2 = J + KY
    J3 = J2 + 1
    J4 = J3 + 1
    J5 = J4 + KY - 1
    J6 = J5 + 1
    A1 = H0(J)
    A2 = H0(J1)
    A3 = H0(J2)
    A4 = H0(J3)
    A5 = H0(J4)
    A6 = H0(J5)
    A7 = H0(J6)
    CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+          C(I,5)*A5 + C(I,6)*A6 + C(I,7)*A7
    J = J + 1
END DO
L = L + 1
A1 = H0(J)
A2 = H0(J+1)
A3 = H0(J+KY)
A4 = H0(J+KY+1)
A5 = H0(J+KY+2)
A6 = H0(M)
CH0(L) = C(L,1)*A1 + C(L,2)*A2 + C(L,3)*A3 + C(L,4)*A4 +
+          C(L,5)*A5 + C(L,6)*A6
L = L + 1
M1 = M - 1
I1 = M - KY - KY
DO I=L,M1
    I2 = I1 + 1
    J1 = I1 + KY
    J2 = J1 + 1
    J3 = J2 + 1
    A1 = H0(I1)
    A2 = H0(I2)
    A3 = H0(J1)
    A4 = H0(J2)
    A5 = H0(J3)
    CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+          C(I,5)*A5
    I1 = I1 + 1
END DO
A1 = H0(M-KY-1)
A2 = H0(M-KY)
A3 = H0(M-1)
A4 = H0(M)
CH0(M) = C(M,1)*A1 + C(M,2)*A2 + C(M,3)*A3 + C(M,4)*A4
C
C-----
C
DO I=1,M
    H1(I) = CH0(I) + DT * F0(I)
END DO
C
C-----
C
DO I=1,M
    IF (ABS(Y(I)-YL) .LT. 0.002) H1(I) = H(I) * DOMIN

```

```
END DO  
CALL SLASN ( M , H1 , 1 )
```

127

```
C  
C-----  
C
```

```
DO I=1,M  
  IF (ABS(Y(I)-YL) .GT. 0.002) H(I) = H1(I)  
END DO
```

```
C  
C-----  
C
```

```
I1 = KY + 1  
I2 = KY + 2  
A1 = H(I1)  
A2 = H(I2)  
A3 = H(I1)  
A4 = H(I2)  
F1(I) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +  
+ G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4  
+ R(1)  
J = 1  
DO I=2,KY  
  J1 = J + 1  
  J2 = J + 2  
  JI1 = J + I1  
  JI2 = J + I2  
  A1 = H(J)  
  A2 = H(J1)  
  A3 = H(J2)  
  A4 = H(JI1)  
  A5 = H(JI2)  
  F1(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +  
+ K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +  
+ G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5  
+ R(I)  
  J = J + 1  
END DO  
A1 = H(I1)  
A2 = H(KY)  
A3 = H(I1)  
A4 = H(I2)  
A5 = H(2*KY+1)  
A6 = H(2*KY+2)  
F1(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +  
+ K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +  
+ G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +  
+ G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6  
+ R(I1)  
L = M - I1  
J = 1  
DO I=I2,L  
  J1 = J + 1  
  J2 = J + KY  
  J3 = J2 + 1  
  J4 = J3 + 1  
  J5 = J4 + KY - 1  
  J6 = J5 + 1  
  A1 = H(J)  
  A2 = H(J1)  
  A3 = H(J2)  
  A4 = H(J3)  
  A5 = H(J4)  
  A6 = H(J5)
```

```

      A7 = H(J6)
      F1(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+           G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+           G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+           G(I,7)*A7*A7
+           + R(I)

      J = J + 1
    END DO
    L = L + 1
    A1 = H(J)
    A2 = H(J+1)
    A3 = H(J+KY)
    A4 = H(J+KY+1)
    A5 = H(J+KY+2)
    A6 = H(M)
    F1(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+           K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+           G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+           G(L,5)*A5*A5 + G(L,6)*A6*A6
+           + R(L)

    L = L + 1
    M1 = M - 1
    I1 = M - KY - KY
    DO I=L,M1
      I2 = I1 + 1
      J1 = I1 + KY
      J2 = J1 + 1
      J3 = J2 + 1
      A1 = H(I1)
      A2 = H(I2)
      A3 = H(J1)
      A4 = H(J2)
      A5 = H(J3)
      F1(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+           G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+           + R(I)

      I1 = I1 + 1
    END DO
    A1 = H(M-KY-1)
    A2 = H(M-KY)
    A3 = H(M-1)
    A4 = H(M)
    F1(M) = K(M,1)*A1 + K(M,2)*A2 + K(M,3)*A3 + K(M,4)*A4 +
+           G(M,1)*A1*A1 + G(M,2)*A2*A2 + G(M,3)*A3*A3 +
+           G(M,4)*A4*A4
+           + R(M)

C
C-----
C
      DO I=1,M
        F1(I) = -F1(I)
      END DO

C
C-----
C
      RETURN
      END

```

SUBROUTINE SOL2 ( M , DT )

129

```

C
C=====
C
C      CALCUL DE H2 POUR LA METHODE DE PREDICTION-CORP.
C
C=====
C

```

```

      IMPLICIT REAL*4 (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
      REAL*4 K(740,7),CH1(740)
      REAL*4 H2P(740)
      COMMON /ENTHA/ H(740),H0(740),H1(740),H2(740)
      COMMON /DERIV/ F0(740),F1(740),F2(740)
      COMMON /PESI/X(740),Y(740) /COORD/ XL,YL,KX,KY
      COMMON /MATK/ K /MATG/ G(740,7)
      COMMON /MATR/ R(740)
      COMMON /MATC/ C(740,7)
      DATA DOMIN /1.0E7/

```

```

C
C-----
C
C      PREDICTION DE (H ) = (H )
C                    2          2P
C
C-----
C

```

```

      I1 = KY + 1
      I2 = KY + 2
      A1 = H(1)
      A2 = H(2)
      A3 = H(I1)
      A4 = H(I2)
      CH1(1) = C(1,4)*A1 + C(1,5)*A2 + C(1,6)*A3 + C(1,7)*A4
      J = 1
      DO I=2,KY
         J1 = J + 1
         J2 = J + 2
         JI1 = J + I1
         JI2 = J + I2
         A1 = H(J)
         A2 = H(J1)
         A3 = H(J2)
         A4 = H(JI1)
         A5 = H(JI2)
         CH1(I) = C(I,3)*A1 + C(I,4)*A2 + C(I,5)*A3 + C(I,6)*A4 +
+              C(I,7)*A5
         J = J + 1
      END DO
      A1 = H(1)
      A2 = H(KY)
      A3 = H(I1)
      A4 = H(I2)
      A5 = H(2*KY+1)
      A6 = H(2*KY+2)
      CH1(I1) = C(I1,2)*A1 + C(I1,3)*A2 + C(I1,4)*A3 +
+              C(I1,5)*A4 + C(I1,6)*A5 + C(I1,7)*A6
      L = M - I1
      J = 1
      DO I=I2,L
         J1 = J + 1
         J2 = J + KY

```

```

      J3 = J2 + 1
      J4 = J3 + 1
      J5 = J4 + KY - 1
      J6 = J5 + 1
      A1 = H(J)
      A2 = H(J1)
      A3 = H(J2)
      A4 = H(J3)
      A5 = H(J4)
      A6 = H(J5)
      A7 = H(J6)
      CH1(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+           C(I,5)*A5 + C(I,6)*A6 + C(I,7)*A7
      J = J + 1
      END DO
      L = L + 1
      A1 = H(J)
      A2 = H(J+1)
      A3 = H(J+KY)
      A4 = H(J+KY+1)
      A5 = H(J+KY+2)
      A6 = H(M)
      CH1(L) = C(L,1)*A1 + C(L,2)*A2 + C(L,3)*A3 + C(L,4)*A4 +
+           C(L,5)*A5 + C(L,6)*A6
      L = L + 1
      M1 = M - 1
      I1 = M - KY - KY
      DO I=L,M1
        I2 = I1 + 1
        J1 = I1 + KY
        J2 = J1 + 1
        J3 = J2 + 1
        A1 = H(I1)
        A2 = H(I2)
        A3 = H(J1)
        A4 = H(J2)
        A5 = H(J3)
        CH1(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+           C(I,5)*A5
        I1 = I1 + 1
      END DO
      A1 = H(M-KY-1)
      A2 = H(M-KY)
      A3 = H(M-1)
      A4 = H(M)
      CH1(M) = C(M,1)*A1 + C(M,2)*A2 + C(M,3)*A3 + C(M,4)*A4
C
C-----
C
      DO I=1,M
        H2P(I) = CH1(I) + 1.5 * DT * F1(I) - 0.5 * DT * F0(I)
      END DO
C
C-----
C
      DO I=1,M
        IF (ABS(Y(I)-YL) .LT. 0.002) H2P(I) = H(I) * DDMIN
      END DO
      CALL SLBSN ( M , H2P , 1 )
C
C-----
C
      CORRECTION DE (H ) = (H )

```

```

DO I=1,5
  I1 = KY + 1
  I2 = KY + 2
  A1 = H2P(1)
  A2 = H2P(2)
  A3 = H2P(I1)
  A4 = H2P(I2)
  F2(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+         G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4
+         + R(1)
  J = 1
DO I=2,KY
  J1 = J + 1
  J2 = J + 2
  JI1 = J + I1
  JI2 = J + I2
  A1 = H2P(J)
  A2 = H2P(J1)
  A3 = H2P(J2)
  A4 = H2P(JI1)
  A5 = H2P(JI2)
  F2(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+         K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+         G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+         + R(I)
  J = J + 1
END DO
A1 = H2P(1)
A2 = H2P(KY)
A3 = H2P(I1)
A4 = H2P(I2)
A5 = H2P(2*KY+1)
A6 = H2P(2*KY+2)
F2(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+         K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+         G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+         G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+         + R(I1)
L = M - I1
J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  A1 = H2P(J)
  A2 = H2P(J1)
  A3 = H2P(J2)
  A4 = H2P(J3)
  A5 = H2P(J4)
  A6 = H2P(J5)
  A7 = H2P(J6)
  F2(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+         K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+         G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+         G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+         G(I,7)*A7*A7
+         + R(I)
  J = J + 1
END DO

```

```

L = L + 1
A1 = H2P(J)
A2 = H2P(J+1)
A3 = H2P(J+KY)
A4 = H2P(J+KY+1)
A5 = H2P(J+KY+2)
A6 = H2P(M)
F2(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+       K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+       G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+       G(L,5)*A5*A5 + G(L,6)*A6*A6
+       + R(L)
L = L + 1
M1 = M - 1
I1 = M - KY - KY
DO I=L,M1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H2P(I1)
  A2 = H2P(I2)
  A3 = H2P(J1)
  A4 = H2P(J2)
  A5 = H2P(J3)
  F2(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+       K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+       G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+       + R(I)
  I1 = I1 + 1
END DO
A1 = H2P(M-KY-1)
A2 = H2P(M-KY)
A3 = H2P(M-1)
A4 = H2P(M)
F2(M) = K(M,1)*A1 + K(M,2)*A2 + K(M,3)*A3 + K(M,4)*A4 +
+       G(M,1)*A1*A1 + G(M,2)*A2*A2 + G(M,3)*A3*A3 +
+       G(M,4)*A4*A4
+       + R(M)
C
DO I=1,M
  F2(I) = -F2(I)
END DO
C
DO I=1,M
  H2(I) = CH1(I) + 0.5 * DT * ( F1(I) + F2(I) )
END DO
C
DO I=1,M
  IF(ABS(Y(I))-YL) .LT. 0.002) H2(I) = H(I) * DDIMIN
END DO
CALL SLBSM ( M , H2 , 1 )
C
ECART = 0.0
DO I=1,M
  DELTA = ABS( (F2(I)-H2P(I))/H2(I) )
  IF (DELTA .GE. ECART) ECART = DELTA
END DO
IF (ECART .LE. 0.02) GOTO 200
C
DO I=1,M
  H2P(I) = H2(I)
END DO

```



```

      END DO
C
C-----
C
      WRITE (6,10) ECART
10    FORMAT (1X,'CRITERE DE CONVERGENCE NON SATISFAIT AU PAS
      + *2 AVEC UN ECART RELATIF=',F5.2)
C
C-----
C
200   DO I=1,M
      IF (ABS(Y(I)-YL) .GT. 0.002) H(I) = H2(I)
      END DO
C
C-----
C
      RETURN
      END

```

```

      SUBROUTINE SOL3 ( M , DT )
C
C=====
C
C      CALCUL DE H3 POUR LA METHODE DE PREDICTION-CORR.
C
C=====
C
      IMPLICIT REAL*4 (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
      REAL*4 K(740,7),CH2(740)
      REAL*4 H3P(740)
      COMMON /ENTHA/ H(740),H0(740),H1(740),H2(740),H3(740)
      COMMON /DERIV/ F0(740),F1(740),F2(740),F3(740)
      COMMON /PDSI/X(740),Y(740) /COORD/ XL,YL,KX,KY
      COMMON /MATK/ K /MATG/ G(740,7)
      COMMON /MATR/ R(740)
      COMMON /MATC/ C(740,7)
      DATA DOMIN /1.0E7/
C
C-----
C
C      PREDICTION DE ( H ) = ( H )
C
C
C-----
C
      I1 = KY + 1
      I2 = KY + 2
      A1 = H(I1)
      A2 = H(I2)
      A3 = H(I1)
      A4 = H(I2)
      CH2(1) = C(1,4)*A1 + C(1,5)*A2 + C(1,6)*A3 + C(1,7)*A4
      J = 1
      DO I=2,KY
        J1 = J + 1
        J2 = J + 2
        JI1 = J + I1
        JI2 = J + I2

```

```

      A1 = H(J)
      A2 = H(J1)
      A3 = H(J2)
      A4 = H(JI1)
      A5 = H(JI2)
      CH2(I) = C(I,3)*A1 + C(I,4)*A2 + C(I,5)*A3 + C(I,6)*A4 +
+          C(I,7)*A5
      J = J + 1
      END DO
      A1 = H(1)
      A2 = H(KY)
      A3 = H(I1)
      A4 = H(I2)
      A5 = H(2*KY+1)
      A6 = H(2*KY+2)
      CH2(I1) = C(I1,2)*A1 + C(I1,3)*A2 + C(I1,4)*A3 +
+          C(I1,5)*A4 + C(I1,6)*A5 + C(I1,7)*A6
      L = M - I1
      J = 1
      DO I=I2,L
          J1 = J + 1
          J2 = J + KY
          J3 = J2 + 1
          J4 = J3 + 1
          J5 = J4 + KY - 1
          J6 = J5 + 1
          A1 = H(J)
          A2 = H(J1)
          A3 = H(J2)
          A4 = H(J3)
          A5 = H(J4)
          A6 = H(J5)
          A7 = H(J6)
          CH2(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+          C(I,5)*A5 + C(I,6)*A6 + C(I,7)*A7
          J = J + 1
      END DO
      L = L + 1
      A1 = H(J)
      A2 = H(J+1)
      A3 = H(J+KY)
      A4 = H(J+KY+1)
      A5 = H(J+KY+2)
      A6 = H(M)
      CH2(L) = C(L,1)*A1 + C(L,2)*A2 + C(L,3)*A3 + C(L,4)*A4 +
+          C(L,5)*A5 + C(L,6)*A6
      L = L + 1
      M1 = M - 1
      I1 = M - KY - KY
      DO I=L,M1
          I2 = I1 + 1
          J1 = I1 + KY
          J2 = J1 + 1
          J3 = J2 + 1
          A1 = H(I1)
          A2 = H(I2)
          A3 = H(J1)
          A4 = H(J2)
          A5 = H(J3)
          CH2(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+          C(I,5)*A5
          I1 = I1 + 1
      END DO

```

```

A1 = H(M-KY-1)
A2 = H(M-KY)
A3 = H(M-1)
A4 = H(M)
CH2(M) = C(M,1)*A1 + C(M,2)*A2 + C(M,3)*A3 + C(M,4)*A4

```

```

C
C-----
C

```

```

DO I=1,M
  H3P(I) = CH2(I) + 23.0/12.0 * DT * F2(I) -
+         16.0/12.0 * DT * F1(I) + 5.0/12.0 * DT * F0(I)
END DO

```

```

C
C-----
C

```

```

DO I=1,M
  IF (ABS(Y(I)-YL) .LT. 0.002) H3P(I) = H(I) * DOMIN
END DO
CALL SLBSN ( M , H3P , 1 )

```

```

C
C-----
C
C
C

```

```

COPRECTION DE ( H ) = ( H )
                2P      3

```

```

DO III=1,5
  I1 = KY + 1
  I2 = KY + 2
  A1 = H3P(I1)
  A2 = H3P(I2)
  A3 = H3P(I1)
  A4 = H3P(I2)
  F3(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+         G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4
+         + R(1)
  J = 1
  DO I=2,KY
    J1 = J + 1
    J2 = J + 2
    JI1 = J + I1
    JI2 = J + I2
    A1 = H3P(J)
    A2 = H3P(J1)
    A3 = H3P(J2)
    A4 = H3P(JI1)
    A5 = H3P(JI2)
    F3(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+         K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+         G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+         + R(I)
    J = J + 1
  END DO
  A1 = H3P(1)
  A2 = H3P(KY)
  A3 = H3P(I1)
  A4 = H3P(I2)
  A5 = H3P(2*KY+1)
  A6 = H3P(2*KY+2)
  F3(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+         K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+         G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+         G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+         + R(I1)
  L = M - I1

```

```

J = 1
DO I=I2,L
  J1 = J + 1
  J2 = J + KY
  J3 = J2 + 1
  J4 = J3 + 1
  J5 = J4 + KY - 1
  J6 = J5 + 1
  A1 = H3P(J)
  A2 = H3P(J1)
  A3 = H3P(J2)
  A4 = H3P(J3)
  A5 = H3P(J4)
  A6 = H3P(J5)
  A7 = H3P(J6)
  F3(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+         K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+         G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+         G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+         G(I,7)*A7*A7
+         + R(I)
  J = J + 1
END DO
L = L + 1
A1 = H3P(J)
A2 = H3P(J+1)
A3 = H3P(J+KY)
A4 = H3P(J+KY+1)
A5 = H3P(J+KY+2)
A6 = H3P(M)
F3(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+         K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+         G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+         G(L,5)*A5*A5 + G(L,6)*A6*A6
+         + R(L)
L = L + 1
M1 = M - 1
I1 = M - KY - KY
DO I=L,M1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H3P(I1)
  A2 = H3P(I2)
  A3 = H3P(J1)
  A4 = H3P(J2)
  A5 = H3P(J3)
  F3(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+         K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+         G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+         + R(I)
  I1 = I1 + 1
END DO
A1 = H3P(M-KY-1)
A2 = H3P(M-KY)
A3 = H3P(M-1)
A4 = H3P(M)
F3(M) = K(M,1)*A1 + K(M,2)*A2 + K(M,3)*A3 + K(M,4)*A4 +
+         G(M,1)*A1*A1 + G(M,2)*A2*A2 + G(M,3)*A3*A3 +
+         G(M,4)*A4*A4
+         + R(M)

```

```

DO I=1,M
  F3(I) = -F3(I)
END DO

C
DO I=1,M
  H3(I) = C42(I) + 5.0/12.0 * DT * F3(I) +
+       8.0/12.0 * DT * F2(I) - 1.0/12.0 * DT * F1(I)
END DO

C
DO I=1,M
  IF(ABS(Y(I)-YL) .LT. 0.002) H3(I) = H(I) * DOMIN
END DO
CALL SLBSN ( M , H3 , 1 )

C
ECART = 0.0
DO I=1,M
  DELTA = ABS( (H3(I)-H3P(I))/H3(I) )
  IF (DELTA .GE. ECART) ECART = DELTA
END DO
IF (ECART .LE. 0.02) GOTO 200

C
DO I=1,M
  H3P(I) = H3(I)
END DO
END DO

C
C-----
C
WRITE (6,10) ECART
10  FORMAT (1X,"CRITERE DE CONVERGENCE NON SATISFAIT AU PAS
+ #3 AVEC UN ECART RELATIF=",F5.2)

C
C-----
C
200  DO I=1,M
      IF (ABS(Y(I)-YL) .GT. 0.002) H(I) = H3(I)
    END DO

C
C-----
C
RETURN
END

```

```

SUBROUTINE PREDICOR ( M , DT , LM )

```

```

C
C=====
C
C  CALCUL DE H  AVEC LA METHODE DE PREDICTION-CORR.
C              LM
C              (METHODE DE HAMMING)
C
C=====
C
IMPLICIT REAL*4 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*4 K(740,7),CH0(740)
REAL*4 H4(740),H4P(740)
COMMON /ENTHA/ H(740),H0(740),H1(740),H2(740),H3(740)

```

```

COMMON /DERIV/ F0(740),F1(740),F2(740),F3(740)
COMMON /POSI/X(740),Y(740) /COORD/ XL,YL,KX,KY
COMMON /MATK/ K /MATG/ G(740,7)
COMMON /MATP/ P(740)
COMMON /MATC/ C(740,7)
DATA DDMIN /1.0E7/

```

138

```

C
C-----
C
C      PREDICTION
C
C-----
C
      I1 = KY + 1
      I2 = KY + 2
      A1 = H0(1)
      A2 = H0(2)
      A3 = H0(I1)
      A4 = H0(I2)
      CH0(1) = C(1,4)*A1 + C(1,5)*A2 + C(1,6)*A3 + C(1,7)*A4
      J = 1
      DO I=2,KY
          J1 = J + 1
          J2 = J + 2
          JI1 = J + I1
          JI2 = J + I2
          A1 = H0(J)
          A2 = H0(J1)
          A3 = H0(J2)
          A4 = H0(JI1)
          A5 = H0(JI2)
          CH0(I) = C(I,3)*A1 + C(I,4)*A2 + C(I,5)*A3 + C(I,6)*A4 +
+              C(I,7)*A5
          J = J + 1
      END DO
      A1 = H0(1)
      A2 = H0(KY)
      A3 = H0(I1)
      A4 = H0(I2)
      A5 = H0(2*KY+1)
      A6 = H0(2*KY+2)
      CH0(I1) = C(I1,2)*A1 + C(I1,3)*A2 + C(I1,4)*A3 +
+              C(I1,5)*A4 + C(I1,6)*A5 + C(I1,7)*A6
      L = M - I1
      J = 1
      DO I=I2,L
          J1 = J + 1
          J2 = J + KY
          J3 = J2 + 1
          J4 = J3 + 1
          J5 = J4 + KY - 1
          J6 = J5 + 1
          A1 = H0(J)
          A2 = H0(J1)
          A3 = H0(J2)
          A4 = H0(J3)
          A5 = H0(J4)
          A6 = H0(J5)
          A7 = H0(J6)
          CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+              C(I,5)*A5 + C(I,6)*A6 + C(I,7)*A7
          J = J + 1
      END DO

```

```

L = L + 1
A1 = H0(J)
A2 = H0(J+1)
A3 = H0(J+KY)
A4 = H0(J+KY+1)
A5 = H0(J+KY+2)
A6 = H0(M)
CH0(L) = C(L,1)*A1 + C(L,2)*A2 + C(L,3)*A3 + C(L,4)*A4 +
+ C(L,5)*A5 + C(L,6)*A6
L = L + 1
M1 = M - 1
I1 = M - KY - KY
DO I=L,M1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = H0(I1)
  A2 = H0(I2)
  A3 = H0(J1)
  A4 = H0(J2)
  A5 = H0(J3)
  CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+ C(I,5)*A5
  I1 = I1 + 1
END DO
A1 = H0(M-KY-1)
A2 = H0(M-KY)
A3 = H0(M-1)
A4 = H0(M)
CH0(M) = C(M,1)*A1 + C(M,2)*A2 + C(M,3)*A3 + C(M,4)*A4
C
C-----
C
DO I=1,M
  H4P(I) = CH0(I) + 4.0/3.0 * DT * ( 2.0*F3(I) - F2(I)
+ 2.0*F1(I) )
END DO
C
C-----
C
DO I=1,M
  IF (ABS(Y(I)-YL) .LT. 0.002) H4P(I) = H(I) * DOMIN
END DO
CALL SLBSN ( M , H4P , 1 )
C
C-----
C
C      MODIFICATION DE LA PREDICTION
C-----
C
IF (LM .GE. 5) THEN
  FRAC = 112.0 / 121.0
  DO I=1,M
    H0(I) = H4P(I) + FRAC * F0(I)
  END DO
ELSE
  DO I=1,M
    H0(I) = H4P(I)
  END DO
END IF
C

```

```

C-----
C
C      CORRECTION
C
C-----
C
      I1 = KY + 1
      I2 = KY + 2
      S1 = 1.0 / 3.0
      S2 = 9.0
      A1 = S1*(S2*H3(1)-H1(1))
      A2 = S1*(S2*H3(2)-H1(2))
      A3 = S1*(S2*H3(I1)-H1(I1))
      A4 = S1*(S2*H3(I2)-H1(I2))
      CH0(1) = C(1,4)*A1 + C(1,5)*A2 + C(1,6)*A3 + C(1,7)*A4
      J = 1
      DO I=2,KY
        J1 = J + 1
        J2 = J + 2
        JI1 = J + I1
        JI2 = J + I2
        A1 = S1*(S2*H3(J)-H1(J))
        A2 = S1*(S2*H3(J1)-H1(J1))
        A3 = S1*(S2*H3(J2)-H1(J2))
        A4 = S1*(S2*H3(JI1)-H1(JI1))
        A5 = S1*(S2*H3(JI2)-H1(JI2))
        CH0(I) = C(I,3)*A1 + C(I,4)*A2 + C(I,5)*A3 + C(I,6)*A4 +
+          C(I,7)*A5
        J = J + 1
      END DO
      A1 = S1*(S2*H3(1)-H1(1))
      A2 = S1*(S2*H3(KY)-H1(KY))
      A3 = S1*(S2*H3(I1)-H1(I1))
      A4 = S1*(S2*H3(I2)-H1(I2))
      J5 = 2*KY + 1
      J6 = J5 + 1
      A5 = S1*(S2*H3(J5)-H1(J5))
      A6 = S1*(S2*H3(J6)-H1(J6))
      CH0(I1) = C(I1,2)*A1 + C(I1,3)*A2 + C(I1,4)*A3 +
+          C(I1,5)*A4 + C(I1,6)*A5 + C(I1,7)*A6
      L = M - I1
      J = 1
      DO I=I2,L
        J1 = J + 1
        J2 = J + KY
        J3 = J2 + 1
        J4 = J3 + 1
        J5 = J4 + KY - 1
        J6 = J5 + 1
        A1 = S1*(S2*H3(J)-H1(J))
        A2 = S1*(S2*H3(J1)-H1(J1))
        A3 = S1*(S2*H3(J2)-H1(J2))
        A4 = S1*(S2*H3(J3)-H1(J3))
        A5 = S1*(S2*H3(J4)-H1(J4))
        A6 = S1*(S2*H3(J5)-H1(J5))
        A7 = S1*(S2*H3(J6)-H1(J6))
        CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+          C(I,5)*A5 + C(I,6)*A6 + C(I,7)*A7
        J = J + 1
      END DO
      L = L + 1
      J1 = J + 1
      J3 = J + KY

```



```

J4 = J3 + 1
J5 = J4 + 1
A1 = S1*(S2*H3(J)-H1(J))
A2 = S1*(S2*H3(J1)-H1(J1))
A3 = S1*(S2*H3(J2)-H1(J2))
A4 = S1*(S2*H3(J3)-H1(J3))
A5 = S1*(S2*H3(J4)-H1(J4))
A6 = S1*(S2*H3(M)-H1(M))
CH0(L) = C(L,1)*A1 + C(L,2)*A2 + C(L,3)*A3 + C(L,4)*A4 +
+ C(L,5)*A5 + C(L,6)*A6
L = L + 1
M1 = M - 1
I1 = M - KY - KY
DO I=L,M1
  I2 = I1 + 1
  J1 = I1 + KY
  J2 = J1 + 1
  J3 = J2 + 1
  A1 = S1*(S2*H3(I1)-H1(I1))
  A2 = S1*(S2*H3(I2)-H1(I2))
  A3 = S1*(S2*H3(J1)-H1(J1))
  A4 = S1*(S2*H3(J2)-H1(J2))
  A5 = S1*(S2*H3(J3)-H1(J3))
  CH0(I) = C(I,1)*A1 + C(I,2)*A2 + C(I,3)*A3 + C(I,4)*A4 +
+ C(I,5)*A5
  I1 = I1 + 1
END DO
J1 = M - KY - 1
J2 = J1 + 1
J3 = M - 1
A1 = S1*(S2*H3(J1)-H1(J1))
A2 = S1*(S2*H3(J2)-H1(J2))
A3 = S1*(S2*H3(J3)-H1(J3))
A4 = S1*(S2*H3(M)-H1(M))
CH0(M) = C(M,1)*A1 + C(M,2)*A2 + C(M,3)*A3 + C(M,4)*A4
C
C-----
C
DO III=1,5
  I1 = KY + 1
  I2 = KY + 2
  A1 = H0(1)
  A2 = H0(2)
  A3 = H0(I1)
  A4 = H0(I2)
  F0(1) = K(1,4)*A1 + K(1,5)*A2 + K(1,6)*A3 + K(1,7)*A4 +
+ G(1,4)*A1*A1 + G(1,5)*A2*A2 + G(1,6)*A3*A3 + G(1,7)*A4*A4
+ R(1)
  J = 1
  DO I=2,KY
    J1 = J + 1
    J2 = J + 2
    JI1 = J + I1
    JI2 = J + I2
    A1 = H0(J)
    A2 = H0(J1)
    A3 = H0(J2)
    A4 = H0(JI1)
    A5 = H0(JI2)
    F0(I) = K(I,3)*A1 + K(I,4)*A2 + K(I,5)*A3 + K(I,6)*A4 +
+ K(I,7)*A5 + G(I,3)*A1*A1 + G(I,4)*A2*A2 +
+ G(I,5)*A3*A3 + G(I,6)*A4*A4 + G(I,7)*A5*A5
+ R(I)

```

```

      J = J + 1
    END DO
    A1 = H0(1)
    A2 = H0(KY)
    A3 = H0(I1)
    A4 = H0(I2)
    A5 = H0(2*KY+1)
    A6 = H0(2*KY+2)
    F0(I1) = K(I1,2)*A1 + K(I1,3)*A2 + K(I1,4)*A3 +
+           K(I1,5)*A4 + K(I1,6)*A5 + K(I1,7)*A6 +
+           G(I1,2)*A1*A1 + G(I1,3)*A2*A2 + G(I1,4)*A3*A3 +
+           G(I1,5)*A4*A4 + G(I1,6)*A5*A5 + G(I1,7)*A6*A6
+           + R(I1)
    L = M - I1
    J = 1
    DO I=I2,L
      J1 = J + 1
      J2 = J + KY
      J3 = J2 + 1
      J4 = J3 + 1
      J5 = J4 + KY - 1
      J6 = J5 + 1
      A1 = H0(J)
      A2 = H0(J1)
      A3 = H0(J2)
      A4 = H0(J3)
      A5 = H0(J4)
      A6 = H0(J5)
      A7 = H0(J6)
      F0(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +
+           K(I,5)*A5 + K(I,6)*A6 + K(I,7)*A7 +
+           G(I,1)*A1*A1 + G(I,2)*A2*A2 + G(I,3)*A3*A3 +
+           G(I,4)*A4*A4 + G(I,5)*A5*A5 + G(I,6)*A6*A6 +
+           G(I,7)*A7*A7
+           + R(I)
      J = J + 1
    END DO
    L = L + 1
    A1 = H0(J)
    A2 = H0(J+1)
    A3 = H0(J+KY)
    A4 = H0(J+KY+1)
    A5 = H0(J+KY+2)
    A6 = H0(M)
    F0(L) = K(L,1)*A1 + K(L,2)*A2 + K(L,3)*A3 + K(L,4)*A4 +
+           K(L,5)*A5 + K(L,6)*A6 + G(L,1)*A1*A1 +
+           G(L,2)*A2*A2 + G(L,3)*A3*A3 + G(L,4)*A4*A4 +
+           G(L,5)*A5*A5 + G(L,6)*A6*A6
+           + R(L)
    L = L + 1
    M1 = M - 1
    I1 = M - KY - KY
    DO I=L,M1
      I2 = I1 + 1
      J1 = I1 + KY
      J2 = J1 + 1
      J3 = J2 + 1
      A1 = H0(I1)
      A2 = H0(I2)
      A3 = H0(J1)
      A4 = H0(J2)
      A5 = H0(J3)
      F0(I) = K(I,1)*A1 + K(I,2)*A2 + K(I,3)*A3 + K(I,4)*A4 +

```

```

+           K(I,5)*A5 + G(I,1)*A1*A1 + G(I,2)*A2*A2 +
+           G(I,3)*A3*A3 + G(I,4)*A4*A4 + G(I,5)*A5*A5
+           + R(I)
      I1 = I1 + 1
END DO
A1 = H0(M-KY-1)
A2 = H0(M-KY)
A3 = H0(M-1)
A4 = H0(M)
F0(M) = K(M,1)*A1 + K(M,2)*A2 + K(M,3)*A3 + K(M,4)*A4 +
+       G(M,1)*A1*A1 + G(M,2)*A2*A2 + G(M,3)*A3*A3 +
+       G(M,4)*A4*A4
+       + R(M)

```

```

C
DO I=1,M
  F0(I) = -F0(I)
END DO

```

```

C
FRAC = 3.0 / 8.0 * DT
DO I=1,M
  H4(I) = CH0(I) + FRAC * ( F0(I) + 2.0*F3(I) - F2(I) )
END DO

```

```

C
DO I=1,M
  IF (ABS(Y(I)-YL) .LT. 0.002) H4(I) = H(I) * DDIMN
END DO
CALL SLBSN ( M , H4 , 1 )

```

```

C
ECART = 0.0
DO I=1,M
  IF (ABS(H4(I)) .GE. 0.000001) THEN
    DELTA = ABS( (H4(I)-H0(I))/H4(I) )
    IF (DELTA .GE. ECART) ECART = DELTA
  END IF
END DO
IF (ECART .LE. 0.02) GO TO 200

```

```

C
DO I=1,M
  H0(I) = H4(I)
END DO
END DO

```

```

C
C-----
C
WRITE(6,10) LM,ECART
10  FORMAT (1X,'CRITERE DE CONVERGENCE NON SATISFAIT AU PAS
+ #',I4,/,1X,'AVEC LM ECART RELATIF =',F5.2)

```

```

C
C-----
C
200  DO I=1,M
      IF (ABS(Y(I)-YL) .GT. 0.002 .AND. LM .EQ. 4) H(I)=H4(I)
      IF (ABS(Y(I)-YL) .GT. 0.002 .AND. LM .GE. 5) THEN
        H(I) = 112.0/121.0 * H4(I) + 9.0/121.0 * H4P(I)
      END IF
      H0(I) = H1(I)
      H1(I) = H2(I)
      H2(I) = H3(I)
      H3(I) = H(I)
      F1(I) = F2(I)
      F2(I) = F3(I)
      F3(I) = F0(I)
      F0(I) = H4(I) - H4P(I)

```

END DO

C  
C-----  
C

RETURN  
END

144

SUBROUTINE MOUVEMENT ( M , DT )

C  
C=====

C  
C RESOLUTION DES EQUATIONS DE NAVIER-STOKES EN  
C UTILISANT LE LOGICIEL -- SOLA -- .

C  
C=====

C  
C IMPLICIT REAL\*4 (A-H,C-Z)  
C IMPLICIT INTEGER\*4 (I-N)  
C REAL\*4 HU(38,21),HV(38,21),GY(38,21),NU,DT  
C COMMON /NAV/ XPUT(6)

C  
C-----  
C

C  
C CALL MATHU ( HU )  
C CALL MATHV ( HV )  
C CALL TEMPLIQ ( HU , HV )  
C CALL MATGY ( GY , HV )

C  
C-- VISCOSITE CINEMATIQUE DE L'ALUMINIUM A 700°C  
C

C  
C NU = 1.22E-06

C  
C-----  
C

C  
C XPUT(1) = DT  
C XPUT(2) = NU

C  
C-----  
C

C  
C CALL SOLA ( GY , HU , HV )  
C CALL TRANSF ( M )

C  
C-----  
C

C  
C RETURN  
C END

SUBROUTINE MATHU ( HU )

C  
C=====

C  
C INTERPOLATION DE L'ENTHALPIE CORRESPONDANT AUX  
C COMPOSANTES "U" .

C  
C=====

```

C
  IMPLICIT REAL*4 (A-H,O-Z)
  IMPLICIT INTEGER*4 (I-N)
  REAL*4 HU(36,21)
  COMMON /ENTHA/ H(740) /COORD/ XL,YL,KX,KY

```

```

C
C-----
C

```

```

  DO I=1,KX
    DO J=2,KY
      L = (I-1)*KY + J - 1
      HU(I,J) = (H(L) + H(L+1)) / 2.0
    END DO
  END DO

```

```

C
C-----
C

```

```

  RETURN
  END

```

SUBROUTINE MATHV ( HV )

```

C
C=====
C
C   INTERPOLATION DE L'ENTHALPIE CORRESPONDANT AUX
C   COMPOSANTES "V" .
C
C=====
C

```

```

  IMPLICIT REAL*4 (A-H,O-Z)
  IMPLICIT INTEGER*4 (I-N)
  REAL*4 HV(36,21)
  COMMON /ENTHA/ H(740) /COORD/ XL,YL,KX,KY

```

```

C
C-----
C

```

```

  DO I=2,KX
    DO J=1,KY
      L = (I-2)*KY + J
      HV(I,J) = (H(L) + H(L+KY)) / 2.0
    END DO
  END DO

```

```

C
C-----
C

```

```

  RETURN
  END

```

SUBROUTINE TEMPLIC ( HU , HV )

```

C
C=====
C
C   CALCUL DE LA TEMPERATURE MOYENNE DU LIQUIDE.
C
C=====
C

```

```

IMPLICIT REAL*4 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*4 HUC(35,21),HVC(35,21)
COMMON /COORD/ XL,YL,KX,KY /TLIC/ TMOY

```

146

```

C
C-----
C

```

```

      TMOY = 0.0
      N = 0

```

```

C
C-----
C

```

```

      DO I=1,KX
        DO J=2,KY
          IF (HUC(I,J) .GE. 9.53E8) THEN
            CALL TL ( T , HUC(I,J) )
            TMOY = TMOY + T
            N = N + 1
          END IF
        END DO
      END DO
      DO I=2,KX
        DO J=1,KY
          IF (HVC(I,J) .GE. 9.53E8) THEN
            CALL TL ( T , HVC(I,J) )
            TMOY = TMOY + T
            N = N + 1
          END IF
        END DO
      END DO

```

```

C
C-----
C

```

```

      TMOY = TMOY / FLGAT(N)

```

```

C
C-----
C

```

```

      RETURN
      END

```

```

      SUBROUTINE MATGY ( GY , HV )

```

```

C
C=====

```

```

      CALCUL DE LA FORCE DE COPPS EXERCEE SUR LES NOEUDS
      "V" .

```

```

C
C=====

```

```

      IMPLICIT REAL*4 (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
      REAL*4 GY(35,21),HV(35,21)
      COMMON /COORD/ XL,YL,KX,KY /TLIQ/ TMOY
      DATA BETAG / 1.619E-3 /

```

```

C
C-----
C

```

```

      DO I=2,KX
        DO J=1,KY

```

```

      IF (HV(I,J) .LT. 9.54E8) THEN
        GY(I,J) = 0.0
      ELSE
        CALL TL ( T , HV(I,J) )
        GY(I,J) = BETAG * ( T - TMOY )
      END IF
    END DO
  END DO

```

```

C
C-----
C
      RETURN
      END

```

SUBROUTINE TL ( T , H )

```

C
C-----
C
      CALCUL DE LA TEMPERATURE T CORRESPONDANT A
      L'ENTHALPIE H .
C
C-----
C

```

```

      IMPLICIT REAL*4 (A-H,D-Z)
      IMPLICIT INTEGER*4 (I-N)

```

```

C
C-----
C

```

$$T = 2.345E-27 * F * H * H + 2.563E-18 * H * H + 3.744E-7 * H + 571.7$$

```

C
C-----
C

```

```

      RETURN
      END

```

SUBROUTINE SOLA ( GY , HU , HV )

```

C
C-----
C
      CALCUL DU CHAMP DE VITESSE A L'AIDE DU LOGICIEL
      -- SOLA --
C
C-----
C

```

```

      IMPLICIT REAL*4 (A-H,D-Z)
      IMPLICIT INTEGER*4 (I-N)
      REAL*4 NU,HU(38,21),HV(38,21),GY(38,21)
      REAL*4 UN(38,21),VN(38,21)
      INTEGER*4 FLG
      COMMON /NAV/ XPUT(6) /COORD/ XL,YL,KX,KY
      COMMON /UVP/ U(38,21),V(38,21),P(38,21)

```

```

C
C-----

```

```

RHOL = 9.53E+9
ISAR = KX - 1
JBAR = KY - 1
DELX = XL / FLOAT(KX-1)
DELY = YL / FLOAT(KY-1)
DELT = XPUT(1)
NU = XPUT(2)
OMG = XPUT(3)
ALPHA = XPUT(4)
WL = XPUT(5)
WR = XPUT(6)

```

```

C
C *** COMPUTE CONSTANT TERMS AND INITIALIZE NECESSARY VARIABLES ****
C

```

```

IMAX=IBAR+2
JMAX=JBAR+2
IM1=IMAX-1
JM1=JMAX-1
RDX=1.0/DELX
RDY=1.0/DELY
BETA=OMG/(2.*DELT*(RDX**2+RDY**2))
ITER = 0
FLG = 1
DO I=1,IMAX
  DO J=1,JMAX
    UN(I,J) = U(I,J)
    VN(I,J) = V(I,J)
  END DO
END DO

```

```

C
C *** COMPUTE TEMPORARY L AND V ****
C

```

```

DO 1100 I=2,IM1
DO 1100 J=2,JM1
FUX=((UN(I,J)+UN(I+1,J))*UN(I,J)+UN(I+1,J))
1   +ALPHA*ABS(UN(I,J)+UN(I+1,J))*(UN(I,J)-UN(I+1,J))
2   -(UN(I-1,J)+UN(I,J))*UN(I-1,J)+UN(I,J))
3   -ALPHA*ABS(UN(I-1,J)+UN(I,J))*(UN(I-1,J)-UN(I,J))/
4   (4.0*DELX)
FUY=((VN(I,J)+VN(I+1,J))*UN(I,J)+UN(I,J+1))
1   +ALPHA*ABS(VN(I,J)+VN(I+1,J))*(UN(I,J)-UN(I,J+1))
2   -(VN(I,J-1)+VN(I+1,J-1))*UN(I,J-1)+UN(I,J))
3   -ALPHA*ABS(VN(I,J-1)+VN(I+1,J-1))*(UN(I,J-1)-UN(I,J))/
4   (4.0*DELY)
FVX=((UN(I,J)+UN(I,J+1))*VN(I,J)+VN(I+1,J))
1   +ALPHA*ABS(UN(I,J)+UN(I,J+1))*(VN(I,J)-VN(I+1,J))
2   -(UN(I-1,J)+UN(I-1,J+1))*VN(I-1,J)+VN(I,J))
3   -ALPHA*ABS(UN(I-1,J)+UN(I-1,J+1))*(VN(I-1,J)-VN(I,J))/
4   (4.0*DELX)
FVY=((VN(I,J)+VN(I,J+1))*VN(I,J)+VN(I,J+1))
1   +ALPHA*ABS(VN(I,J)+VN(I,J+1))*(VN(I,J)-VN(I,J+1))
2   -(VN(I,J-1)+VN(I,J))*VN(I,J-1)+VN(I,J))
3   -ALPHA*ABS(VN(I,J-1)+VN(I,J))*(VN(I,J-1)-VN(I,J))/
4   (4.0*DELY)
VISX=NU*((UN(I+1,J)-2.0*UN(I,J)+UN(I-1,J))/(DELX**2)
1   +(UN(I,J+1)-2.0*(UN(I,J))+UN(I,J-1))/(DELY**2))
VISY=NU*((VN(I+1,J)-2.0*VN(I,J)+VN(I-1,J))/(DELX**2)
1   +(VN(I,J+1)-2.0*VN(I,J)+VN(I,J-1))/(DELY**2))
U(I,J)=U(I,J)+DELT*((P(I,J)-P(I+1,J))*RDX-FUX-FUY+VISX)
V(I,J)=V(I,J)+DELT*((P(I,J)-P(I,J+1))*RDY+GY(I,J)-FVX-FVY+VISY)
1100 CONTINUE
C

```



```

C *** SET BOUNDARY CONDITIONS *****
C
2000   DO 2200 J=1,JMAX
        IWL = JNINT(WL)
        GOTO(2020,2040),IWL
2020   U(1,J)=0.0
        V(1,J)=V(2,J)
        GOTO 2100
2040   U(1,J)=0.0
        V(1,J)=-V(2,J)
2100   IWR = JNINT(WR)
        GOTO(2120,2140),IWR
2120   U(IM1,J)=0.0
        V(IMAX,J)=V(IM1,J)
        GOTO 2200
2140   U(IM1,J)=0.0
        V(IMAX,J)=-V(IM1,J)
2200   CONTINUE
        DO 2500 I=1,IMAX
        V(I,JM1)=0.0
        U(I,JMAX)=U(I,JM1)
        V(I,1)=0.0
        U(I,1)=-U(I,2)
2500   CONTINUE
C
C *** SPECIAL BOUNDARY CONDITIONS *****
C
        DO I=2,IM1
        DO J=1,JM1
            IF (HV(I,J) .LT. RHQL) V(I,J) = 0.0
        END DO
    END DO
        DO I=1,IM1
        DO J=2,JM1
            IF (HU(I,J) .LT. RHQL) U(I,J) = 0.0
        END DO
    END DO
C
C *** HAS CONVERGENCE BEEN REACHED? *****
C
        IF (FLG .EQ. 0) RETURN
        ITER = ITER+1
        IF (ITER .GE. 500) RETURN
        FLG=0
C
C *** COMPUTE UPDATED CELL PRESSURE AND VELOCITIES *****
C
        DO 3500 J=2,JM1
        DO 3500 I=2,IM1
        D = RDX*(U(I,J)-U(I-1,J))+RDY*(V(I,J)-V(I,J-1))
        IF (ABS(D) .GE. 1.0E-3) FLG = 1
        DELP = -BETA*D
        P(I,J)=P(I,J)+DELP
        U(I,J)=U(I,J)+DELT*RDX*DELP
        U(I-1,J)=U(I-1,J)-DELT*RDX*DELP
        V(I,J)=V(I,J)+DELT*RDY*DELP
        V(I,J-1)=V(I,J-1)-DELT*RDY*DELP
3500   CONTINUE
        GOTO 2000
C
C-----
C
        END

```

## SUBROUTINE TRANSF ( M )

```

C
C-----
C
C   FAIT CORRESPONDRE LES COMPOSANTES DES VITESSES
C   CALCULEES PAR SGLA AUX COMPOSANTES NODALES DU
C   MAILLAGE EN ELEMENTS FINIS.
C
C-----
C
C   IMPLICIT REAL*4 (A-H,D-Z)
C   IMPLICIT INTEGER*4 (I-N)
C   COMMON /UVP/ U(38,21),V(38,21) /COORD/ XL,YL,KX,KY
C   COMMON /MATVIT/ VIT(2*740) /ENTHA/ H(740)
C
C-----
C
C   DO I=1,KX
C     DO J=1,KY
C       K = (I-1)*KY + J
C       LU = 2*K - 1
C       LV = LU + 1
C       VIT(LU) = (U(I,J) + U(I,J+1)) / 2.0
C       VIT(LV) = (V(I,J) + V(I+1,J)) / 2.0
C     END DO
C   END DO
C
C-----
C
C   DO I=1,M
C     IF (H(I) .LT. 9.53E2) THEN
C       VIT(2*I-1) = 0.0
C       VIT(2*I)   = 0.0
C     END IF
C   END DO
C
C-----
C
C   RETURN
C   END

```

## SUBROUTINE SLBSN ( M , H , ICHOIX )

```

C
C-----
C
C
C
C   CALCUL DE (H) A PARTIR DE (A)(H) = (B) POUR
C   UN SYSTEME BANDE.
C   REF. : C.A. BREBBIA , A.J. FERRANTE
C           COMPUTATIONAL METHODS FOR THE SOLUTION OF
C           ENGINEERING PROBLEMS.
C
C   ICHOIX = 1 ==> EQ. D'ENERGIE (PREDICTION-CORR.)

```

```

C
C
C-----
C
    IMPLICIT REAL*4 (A-H,O-Z)
    IMPLICIT INTEGER*4 (I-N)
    REAL*4 H(740),A(740,43),MK(740,7)
    COMMON /MATC/ C(740,7) /MATK/ MK
    COMMON /CGORD/ XL,YL,KX,KY
    COMMON /POSI/ X(740),Y(740)
    DATA DOMIN /1.0E7/

```

```

C
C-----
C
    MS = KY + 2
    KBAN = 2*MS - 1
    IF (KBAN .GT. 43) THEN
        WRITE (6,10)
        STOP
    END IF

```

```

C
C-----
C
    DO I=1,M
        DO J=1,KBAN
            A(I,J) = 0.0
        END DO
    END DO

```

```

C
C-----
C
    IF (ICHDIX .EQ. 1) THEN
        J4 = MS
        J5 = J4 + 1
        J6 = MS + KY
        J7 = J6 + 1
        J3 = J4 - 1
        J2 = MS - KY
        J1 = J2 - 1
        DO I=1,M
            A(I,J1) = C(I,1)
            A(I,J2) = C(I,2)
            A(I,J3) = C(I,3)
            A(I,J4) = C(I,4)
            A(I,J5) = C(I,5)
            A(I,J6) = C(I,6)
            A(I,J7) = C(I,7)
            IF (ABS(Y(I)-YL) .LT. 0.002) THEN
                A(I,J4) = A(I,J4) + DOMIN
            END IF
        END DO
    END IF

```

1000

```

C
C-----
C
    IF (ICHDIX .EQ. 2) THEN
        J1 = MS - KY - 1
        J2 = J1 + 1
        J3 = MS - 1
        J4 = MS
        J5 = MS + 1
        J6 = MS + KY
        J7 = J6 + 1
    END IF

```

```

      DO I=1,M
        A(I,J1) = MK(I,1)
        A(I,J2) = MK(I,2)
        A(I,J3) = MK(I,3)
        A(I,J4) = MK(I,4)
        A(I,J5) = MK(I,5)
        A(I,J6) = MK(I,6)
        A(I,J7) = MK(I,7)
        IF (ABS(Y(I)-YL) .LT. 0.002) THEN
          A(I,J4) = A(I,J4) + DDMIN
        END IF
1005      END DO
      END IF
C
C-----
C
      MNOMB = M
C
C-----
C
      N1 = MNOMB - 1
      DO K = 1,N1
        CC = A(K,MS)
        K1 = K + 1
        IF (ABS(CC)-0.000001)1,1,3
1        WRITE (6,2) K
        STOP
3        NI = K1 + MS - 2
        L = JMINO (NI , MNOMB)
        DO J = K1,L
          K2 = J - K + MS
          A(K,K2) = A(K,K2) / CC
        END DO
        H(K) = H(K) / CC
        DO I=K1,L
          K2 = MS + K - I
          CC = A(I,K2)
          DO J = K1,L
            K2 = J - I + MS
            K3 = J - K + MS
            A(I,K2) = A(I,K2) - CC * A(K,K3)
          END DO
          H(I) = H(I) - CC * H(K)
        END DO
      END DO
      IF (ABS(A(MNOMB,MS))-0.000001) 4,4,111
4      WRITE(6,2) K
      STOP
111     H(MNOMB) = H(MNOMB) / A(MNOMB,MS)
      DO I=1,N1
        K = MNOMB - I
        K1 = K + 1
        NI = K1 + MS - 2
        L = JMINO (NI,MNOMB)
        DO J = K1,L
          K2 = J - K + MS
          H(K) = H(K) - A(K,K2) * H(J)
        END DO
      END DO
      CONTINUE
C
C-----
C

```

```

2      FORMAT ( '***** SINGULIERITE = LE CASN', 1000
10     FORMAT (3(/), ' *** LA BANDE DE LA MATRICE EST TROP GRANDE ***'
      +      ,/, ' ARRET DE L'EXECUTION DANS SLBSN', 3(/))
C
C-----
C
      RETURN
      END

```

```

      SUBROUTINE CPU
C
C=====
C
C      ECRITURE DU TEMPS CPU DANS LE FICHIER TEMPERATURES.DAT
C
C=====
C
      INTEGER*4 TIMER_DATA, TIMER_ADDR, TIMER_ROUTINE, STATUS
      EXTERNAL TIMER_ROUTINE
      INTEGER*4 LIB$INIT_TIMER, LIB$SHOW_TIMER, LIB$FREE_TIMER
      EXTERNAL LIB$INIT_TIMER, LIB$SHOW_TIMER, LIB$FREE_TIMER
      LOGICAL FIRST
      DATA FIRST/.TRUE./
C
      TIMER_DATA=2
      IF(FIRST)THEN
C
      STATUS = LIB$INIT_TIMER (TIMER_ADDR)
      IF(.NOT.STATUS)CALL LIB$SIGNAL (%VAL(STATUS))
      FIRST=.FALSE.
C
      RETURN
C
      ENDIF
C
      STATUS = LIB$SHOW_TIMER (TIMER_ADDR,,TIMER_ROUTINE,TIMER_DATA)
      IF(.NOT.STATUS)CALL LIB$SIGNAL (%VAL(STATUS))
      STATUS = LIB$FREE_TIMER (TIMER_ADDR)
      IF(.NOT.STATUS)CALL LIB$SIGNAL (%VAL(STATUS))
C
      FIRST=.TRUE.
C
      RETURN
      END
C
C
      INTEGER FUNCTION TIMER_ROUTINE (STATS, TIMER_DATA)
C
      CHARACTER*(*) STATS
      INTEGER TIMER_DATA
C
      OPEN (1, FILE='TEMPERATURES.DAT', STATUS='OLD', ACCESS='APPEND')
      WRITE(1,10)
10     FORMAT(//, ' STATISTIQUES DE L'EXECUTION:',/, 80('-'))
      WRITE(1,*)STATS
      WRITE(1,20)
20     FORMAT(80('-'),/)
C
      TIMER_ROUTINE = 1

```

C

CLOSE(1)  
END

```

PROGRAM PREP
C-----C
C
C      CE PROGRAMME EST UTILISEE POUR CREER LES
C      FICHIER DONNEES.DAT ET SOLIDE.DAT S'ILS
C      NE LE SONT PAS. S'ILS SONT DEJA CREEES, CE
C      PROGRAMME FAIT UNE MISE A JOUR DE CES
C      FICHIERS UTILISES PAR LE PROGRAMME FUSION.
C-----C
C
C      Ecrit par: Jocelyn Trambly, etudiant
C                  a l'Universite du Quebec a
C                  Chicoutimi (juin 1986).
C-----C
C
      CHARACTER*3 CONV(2)
      CHARACTER*2 OPTION
      INTEGER*4 CONVECTION
C-----C
C
      OPEN (1,FILE='DONNEES.DAT',
+          STATUS='UNKNOWN')
C-----C
C
      CONV(1) = 'oui'
      CONV(2) = 'non'
C-----C
C
      READ (1,*,END=1000) XL,YL,KX,KY
      READ (1,*) TEMPS
      READ (1,*) DT
      READ (1,*) K,TSOL,TLIQ,TSURF
      READ (1,*) CONVECTION
      READ (1,*) OMEGA,ALPHA
      READ (1,*) WL,WR
      READ (1,10) OPTION
      READ (1,*) SXG,SXD,SY
C-----C
C
1      PRINT *,CHAR(12)
      PRINT *,CHAR(12)
      WRITE (6,15)
      WRITE (6,20)
      WRITE (6,30) XL
      WRITE (6,40) YL
      WRITE (6,50) KX
      WRITE (6,60) KY
      WRITE (6,70) OPTION
      IF (CONVECTION .EQ. 1) THEN
          L = 1
      ELSE

```

```

L = 2
WL = 2.0
WR = 2.0
OMEGA = 1.8
ALPHA = 0.5
END IF
WRITE (6,80) CONV(L)
IF (L .EQ. 1) THEN
  WRITE (6,90) OMEGA
  WRITE (6,100) ALPHA
  IF (ABS(WL-2.0) .LT. 0.01) THEN
    L1 = 2
  ELSE
    L1 = 1
  END IF
  WRITE (6,110) CONV(L1)
  IF (ABS(WR-2.0) .LT. 0.01) THEN
    L1 = 2
  ELSE
    L1 = 1
  END IF
  WRITE (6,120) CONV(L1)
END IF
IF (K .EQ. 1) THEN
  WRITE (6,130) CONV(1)
ELSE
  WRITE (6,130) CONV(2)
  WRITE (6,140) TSCL
  WRITE (6,150) TLIQ
  WRITE (6,160) TSURF
END IF
WRITE (6,170) TEMPS
WRITE (6,180) DT
WRITE (6,21)
WRITE (6,190)

```

C  
C-----  
C

```

WRITE (6,200)
READ (5,*) L
IF (L .EQ. 0) GOTO 2000
PRINT #,CHAR(12)
GOTO (501,502,503,504,505,506,507,508,509,
+      510,511,512,513,514,515,516) ,L

```

C  
C-----  
C

```

501  WRITE (6,601)
601  FORMAT(1X,'LA LARGEUR DU BAIN (m) ? ',F)
    READ (5,*) XL
    GOTO 1
502  WRITE (6,602)
602  FORMAT(1X,'LA HAUTEUR DU BAIN (m) ? ',F)
    READ (5,*) YL
    GOTO 1
503  WRITE (6,603)
603  FORMAT(1X,'LE NOMBRE DE NOEUDS EN x ? ',F)
    READ (5,*) KX
    IF (KX .GT. 37) THEN
      PRINT #,' LE NOMBRE DE NOEUDS EN x DOIT ETRE < 38'
      GOTO 503
    END IF
    GOTO 1

```



```

604  FORMAT(1X, 'LE NOMBRE DE NOEUDS EN y ? ', $)
      READ (5,*) KY
      IF (KY .GT. 20) THEN
          PRINT *, ' LE NOMBRE DE NOEUDS EN y DOIT ETRE < 21'
          GOTO 504
      END IF
      GOTO 1
505  PRINT *, ' L'OPTION ?'
      PRINT *, ' 1=> DL'
      PRINT *, ' 2=> FL'
      PRINT *, ' 3=> DQ'
      WRITE (6,605)
605  FORMAT(1X, ' 4=> BQ ', $)
      READ (5,*) J
      IF (J .EQ. 1) OPTION = 'DL'
      IF (J .EQ. 2) OPTION = 'FL'
      IF (J .EQ. 3) OPTION = 'DQ'
      IF (J .EQ. 4) OPTION = 'BQ'
      GOTO 1
506  IF (CONVECTION .EQ. 0) THEN
          CONVECTION = 1
      ELSE
          CONVECTION = 0
      END IF
      GOTO 1
507  PRINT *, ' LE PARAMETRE DE RELAXATION (OMEGA) ?'
      WRITE (6,607)
607  FORMAT(1X, '          EX: 1.8          ', $)
      READ (5,*) OMEGA
      IF (OMEGA .GE. 2.0) THEN
          PRINT *, ' LE PARAMETRE DE RELAXATION DOIT ETRE < 2'
          GOTO 507
      END IF
      GOTO 1
508  PRINT *, ' LE COEFFICIENT ALPHA ( OKALPHA1 ) ?'
      WRITE (6,608)
608  FORMAT(1X, '          ex: 0.7          ', $)
      READ (5,*) ALPHA
      GOTO 1
509  IF (WL .GT. 1.0) THEN
          WL = 1.0
      ELSE
          WL = 2.0
      END IF
      GOTO 1
510  IF (WR .GT. 1.0) THEN
          WR = 1.0
      ELSE
          WR = 2.0
      END IF
      GOTO 1
511  IF (K .EQ. 1) THEN
          K = 2
      ELSE
          K = 1
      END IF
      GOTO 1
512  WRITE (6,612)
612  FORMAT(1X, 'LA TEMPERATURE DU SOLIDE (°C) ? ', $)
      READ (5,*) TSOL
      GOTO 1
513  WRITE (6,613)

```

```

513  FORMAT(1X, 'LA TEMPERATURE DU LIQUIDE ("C) ? ', $)
      READ (5,*) TLIQ
      GOTO 1
514  WRITE (6,614)
514  FORMAT(1X, 'LA TEMPERATURE IMPOSEE ("C) ? ', $)
      READ (5,*) TSURF
      GOTO 1
515  WRITE (6,615)
515  FORMAT(1X, 'LA DUREE DE LA SIMULATION (sec) ? ', $)
      READ (5,*) TMPS
      GOTO 1
516  WRITE (6,616)
516  FORMAT(1X, 'LE PAS DE TEMPS (sec) ? ', $)
      READ (5,*) DT
      IF (DT .GT. 0.4) THEN
        PRINT *, ' ATTENTION, LA CONVERGENCE EST MEILLEURE '
        PRINT *, ' LORSQUE LE PAS EST INFERIEUR OU EGAL A 0.4 sec.'
      END IF
      GOTO 1

C
C-----
C
1000  PRINT *,CHAR(12)
      WRITE (6,1100)
1100  FORMAT(1X, 'LA LARGEUR DU BAIN (m) ? ', $)
      ACCEPT *,XL
      WRITE (6,1200)
1200  FORMAT(1X, 'LA HAUTEUR DU BAIN (m) ? ', $)
      ACCEPT *,YL
1001  WRITE (6,1101)
1101  FORMAT(1X, 'LE NOMBRE DE NOEUDS EN x ? ', $)
      ACCEPT *,KX
      IF (KX .GT. 37) THEN
        PRINT *, ' LE NOMBRE DE NOEUDS EN x DOIT ETRE < 38 '
        GOTO 1001
      END IF
1002  WRITE (6,1102)
1102  FORMAT(1X, 'LE NOMBRE DE NOEUDS EN y ? ', $)
      ACCEPT *,KY
      IF (KY .GT. 20) THEN
        PRINT *, ' LE NOMBRE DE NOEUDS EN y DOIT ETRE < 21 '
        GOTO 1002
      END IF
      PRINT *, 'L' OPTION DESIREE ? '
      PRINT *, '      1=> DL '
      PRINT *, '      2=> PL '
      PRINT *, '      3=> DQ '
      WRITE (6,1201)
1201  FORMAT(1X, '      4=> BQ      ', $)
      ACCEPT *,L
      IF (L .EQ. 1) OPTION = 'DL'
      IF (L .EQ. 2) OPTION = 'PL'
      IF (L .EQ. 3) OPTION = 'DQ'
      IF (L .EQ. 4) OPTION = 'BQ'
      PRINT *, ' AVEC OU SANS CONVECTION NATURELLE ? '
      PRINT *, '      1=> AVEC '
      WRITE (6,1202)
1202  FORMAT(1X, '      2=> SANS      ', $)
      ACCEPT *,L
      IF (L .EQ. 1) CONVECTION = 1
      IF (L .EQ. 2) CONVECTION = 0
      IF (CONVECTION .EQ. 1) THEN
1003  WRITE (6,1103)

```

```

1103     FORMAT(1X, ' LE PARAMETRE DE RELAXATION (ex 1.8) ? ', $)
        ACCEPT #, OMEGA
        IF (OMEGA .GE. 2.0) THEN
            PRINT #, ' LE PARAMETRE DE RELAXATION DOIT ETRE < 2'
            GOTO 1003
        END IF
        PRINT #, ' LE COEFFICIENT ALPHA ( 0 a 1 ) ?'
        WRITE (6,1203)
1203     FORMAT(1X, '          ex: 0.7          ', $)
        ACCEPT #, ALPHA
        PRINT #, ' PLAN DE SYMETRIE A LA GAUCHE DU BAIN ?'
        PRINT #, '          1=> OUI'
        WRITE (6,1204)
1204     FORMAT(1X, '          2=> NON          ', $)
        ACCEPT #, L
        IF (L .EQ. 1) WL = 1.0
        IF (L .EQ. 2) WL = 2.0
        PRINT #, ' PLAN DE SYMETRIE A LA DROITE DU BAIN ?'
        PRINT #, '          1=> OUI'
        WRITE (6,1205)
1205     FORMAT(1X, '          2=> NON          ', $)
        ACCEPT #, L
        IF (L .EQ. 1) WR = 1.0
        IF (L .EQ. 2) WR = 2.0
        END IF
        PRINT #, ' LES CONDITIONS INITIALES SONT DES ANCIENS CALCULS ?'
        PRINT #, '          1=> OUI'
        WRITE (6,1206)
1206     FORMAT(1X, '          2=> NON          ', $)
        ACCEPT #, K
        IF (K .EQ. 1) PRINT #, ' VOUS DEVEZ AVOIR LES FICHIER
+ H.DAT ET UV.DAT'
        IF (K .EQ. 2) THEN
            WRITE (6,1207)
1207     FORMAT(1X, 'LA TEMPERATURE DU SOLIDE (°C) ? ', $)
            ACCEPT #, TSOL
            WRITE (6,1208)
1208     FORMAT(1X, 'LA TEMPERATURE DU LIQUIDE (°C) ? ', $)
            ACCEPT #, TLIQ
            WRITE (6,1209)
1209     FORMAT(1X, 'LA TEMPERATURE IMPOSEE (°C) ? ', $)
            ACCEPT #, Tsurf
        END IF
        WRITE (6,1211)
1211     FORMAT(1X, 'LE PAS DE TEMPS (sec) ? ', $)
        ACCEPT #, DT
        IF (DT .GT. 0.4) THEN
            PRINT #, ' ATTENTION, LA CONVERGENCE EST MEILLEURE'
            PRINT #, ' LORSQUE LE PAS EST INFÉRIEUR OU ÉGAL A 0.4 sec.'
        END IF
        WRITE (6,1210)
1210     FORMAT(1X, 'LA DURÉE DE LA SIMULATION (sec) ? ', $)
        ACCEPT #, TEMPS
        IF (K .NE. 1) THEN
            WRITE (6,1212)
1212     FORMAT(1X, 'LA COORDONNÉE GAUCHE DU SOLIDE (m) ? ', $)
            ACCEPT #, SXG
            WRITE (6,1213)
1213     FORMAT(1X, 'LA COORDONNÉE DROITE DU SOLIDE (m) ? ', $)
            ACCEPT #, SXD
            WRITE (6,1214)
1214     FORMAT(1X, 'LA HAUTEUR DU SOLIDE (m) ? ', $)
            ACCEPT #, SY

```

```
END IF
GOTO 1
```

160

```
C
C-----
C
2000 PRINT *,CHAR(12)
      IF (K .NE. 2) GOTO 4000
      WRITE (6,15)
      WRITE (6,20)
      WRITE (6,3000) SXG
      WRITE (6,3010) SXD
      WRITE (6,3020) SY
      WRITE (6,21)
      WRITE (6,190)
```

```
C
C-----
C
      WRITE (6,200)
      READ (5,*) L
      IF (L .EQ. 0) GOTO 4000
      PRINT *,CHAR(12)
      GOTO (3501,3502,3503) ,L
```

```
C
C-----
C
3501 WRITE (6,3601)
3601 FORMAT(1X,'LA COORDONNEE GAUCHE DU SOLIDE ? ',%)
      ACCEPT *,SXG
      GOTO 2000
3502 WRITE (6,3602)
3602 FORMAT(1X,'LA COORDONNEE DROITE DU SOLIDE ? ',%)
      ACCEPT *,SXD
      GOTO 2000
3503 WRITE (6,3603)
3603 FORMAT(1X,'LA HALTEUR DU SOLIDE ? ',%)
      ACCEPT *,SY
      GOTO 2000
```

```
C
C-----
C
4000 PRINT *,CHAR(12)
      REWIND (1)
      WRITE (1,2010) XL,YL,KX,KY
      WRITE (1,2020) TEMPS
      WRITE (1,2030) DT
      WRITE (1,2040) K,TSOL,TLIQ,TSURF
      WRITE (1,2045) CONVECTION
      WRITE (1,2050) OMEGA,ALPHA
      WRITE (1,2060) WL,WR
      WRITE (1,2070) CPTIDN
      WRITE (1,2080) SXG,SXD,SY
      CLOSE (1)
```

```
C
C-----
C
      DO I=1,3
        PRINT *
      END DO
      PRINT *,' ==> FICHER DONNEES.DAT CREE <== '
      IF (K .EQ. 1) THEN
        DO I=1,3
          PRINT *
        END DO
```

END IF

C  
C-----  
C

```

IF (K .EQ. 2) THEN
  OPEN (2, FILE='SOLIDE.DAT', STATUS='UNKNOWN')
  DX = XL / FLOAT(KX-1)
  DY = YL / FLOAT(KY-1)
  N1 = JINT( SXG / DX + 0.0001) + 1
  N2 = JNINT( SXD / DX ) + 1
  N3 = JINT( SY / DY + 0.0001) + 1
  DO I=N1,N2
    DO J=1,N3
      L = (I-1) * KY + J
      WRITE (2,3E00) L
    END DO
  END DO
  CLOSE (2)
  PRINT *, ' ==> FICHER SOLIDE.DAT CREE <== '
  DO I=1,3
    PRINT *
  END DO
END IF

```

C  
C-----  
C

```

10  FORMAT (A3)
15  FORMAT (/)
20  FORMAT (1X, '.....')
+ .....")
21  FORMAT (1X, ':.....:.....')
+ .....:")
30  FORMAT (1X, ': LARGEUR DU BAIN           : ', F7.3,
+      ' (m)           : #1"')
40  FORMAT (1X, ': HAUTEUR DU BAIN           : ', F7.3,
+      ' (m)           : #2"')
50  FORMAT (1X, ': NOMBRE DE NOEUDS EN x       : ', I7,
+      ' : #3"')
60  FORMAT (1X, ': NOMBRE DE NOEUDS EN y       : ', I7,
+      ' : #4"')
70  FORMAT (1X, ': OPTION CHOISIE             : ', A7,
+      ' : #5"')
80  FORMAT (1X, ': CONVECTION NATURELLE       : ', A8,
+      ' : #6"')
90  FORMAT (1X, ': PARAMETRE DE RELAXATION     : ', F8.2,
+      ' : #7"')
100 FORMAT (1X, ': COEFFICIENT ALPHA          : ', F8.2,
+      ' : #8"')
110 FORMAT (1X, ': PLAN DE SYMETRIE A GAUCHE  : ', A8,
+      ' : #9"')
120 FORMAT (1X, ': PLAN DE SYMETRIE A DROITE  : ', A8,
+      ' : #10"')
130 FORMAT (1X, ': COND. INIT. PREDEFINITES   : ', A8,
+      ' : #11"')
140 FORMAT (1X, ': TEMPERATURE DU SOLIDE      : ', F7.1,
+      ' ("C) : #12"')
150 FORMAT (1X, ': TEMPERATURE DU LIQUIDE     : ', F7.1,
+      ' ("C) : #13"')
160 FORMAT (1X, ': TEMPERATURE IMPOSEE       : ', F7.1,
+      ' ("C) : #14"')
170 FORMAT (1X, ': DUREE DE LA SIMULATION    : ', F7.1,
+      ' (sec) : #15"')
180 FORMAT (1X, ': PAS DE TEMPS              : ', F7.2,

```

```

+          " (sec) :      #16")
190  FORMAT (/,T30,"AUCUN CHANGEMENT",T54,"*0")
200  FORMAT (/,T20,"QUE DESIREZ-VOUS CHANGER ? ",3)
2010 FORMAT (2(1X,F5.3),2(1X,I2)," 1 1")
2020  FORMAT (1X,F8.3)
2030  FORMAT (1X,F7.3)
2040  FORMAT (1X,I1,3(1X,F7.2))
2045  FORMAT (1X,I1)
2050  FORMAT (2(1X,F4.2))
2060  FORMAT (2(1X,F3.1))
2070  FORMAT (A3)
2080  FORMAT (3(1X,F7.3))
3000  FORMAT (1X,": COORD. GAUCHE DU SOLIDE : ",F7.3,
+      " (m) :      #1")
3010  FORMAT (1X,": COORD. DROITE DU SOLIDE : ",F7.3,
+      " (m) :      #2")
2020  FORMAT (1X,": HAUTEUR DU SOLIDE : ",F7.3,
+      " (m) :      #3")
3800  FORMAT (1X,I3)
C
C-----
C
      END

```

ANNEXE 2

DÉVELOPPEMENT DES TERMES FVX, FVY ET VISY  
APPARAISSANT DANS SOLA

## ANNEXE 2

## DÉVELOPPEMENT DES TERMES FVX, FVY ET VISY APPARAISSANT DANS SOLA

L'algorithme utilisé dans SOLA est une simplification de la méthode "Marker-and-Cell" (MAC)<sup>1</sup> formulée en coordonnées Eulériennes et utilisant les variables primitives. La notation sous forme de différences finies utilisée dans le rapport de Hirt et all est:

$p_{i,j}^n$  = la pression au centre de la cellule (i,j) au temps  $n \Delta t$  divisée par la densité du fluide (voir figure 10).

$u_{i,j}^n$  = la composante selon x de la vitesse prise au centre du côté droit de la cellule (i,j) au temps  $n \Delta t$  (voir figure 10).

$v_{i,j}^n$  = la composante selon y de la vitesse prise au centre du dessus de la cellule (i,j) au temps  $n \Delta t$  (voir figure 10).

Le schéma aux différences représentant l'équation de continuité (équation 8) pour une cellule typique (i,j) s'écrit en coordonnées cartésiennes:

$$\frac{1}{\Delta x} (u_{i,j}^{n+1} - u_{i-1,j}^{n+1}) + \frac{1}{\Delta y} (v_{i,j}^{n+1} - v_{i,j-1}^{n+1}) = 0$$

<sup>1</sup> Harlow, F.H., Welch, J.E., Shannon, J.P., Daly, B.J., "The MAC Method: A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces", Los Alamos Scientific Laboratory, report LA-3425, mars 1966.



avec  $\Delta x$  = la largeur de la cellule (selon x).

$\Delta y$  = la hauteur de la cellule (selon y).

Dans la pratique, cette équation donne comme divergence une valeur  $D \approx 0$ .

Les équations de Navier-Stokes (équations 9) sont approximées par les schémas aux différences

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \left( \frac{1}{\Delta x} (p_{i,j}^n - p_{i+1,j}^n) - FUX - FUY + VISX \right)$$

et

$$v_{i,j}^{n+1} = v_{i,j}^n + \Delta t \left( \frac{1}{\Delta y} (p_{i,j}^n - p_{i,j+1}^n) + g\beta(T - T_0) - FVX - FVY + VISY \right)$$

où:

$$FUX = \frac{1}{4\Delta x} \left( (u_{i,j} + u_{i+1,j})^2 + \alpha |u_{i,j} + u_{i+1,j}| (u_{i,j} - u_{i+1,j}) - \right. \\ \left. (u_{i-1,j} + u_{i,j})^2 - \alpha |u_{i-1,j} + u_{i,j}| (u_{i-1,j} - u_{i,j}) \right)$$

$$FUY = \frac{1}{4\Delta y} \left( (v_{i,j} + v_{i+1,j})(u_{i,j} + u_{i,j+1}) + \right. \\ \left. \alpha |v_{i,j} + v_{i+1,j}| (u_{i,j} - u_{i,j+1}) - (v_{i,j-1} + v_{i+1,j-1})(u_{i,j-1} + u_{i,j}) - \right. \\ \left. \alpha |v_{i,j-1} + v_{i+1,j-1}| (u_{i,j-1} - u_{i,j}) \right)$$

$$FVX = \frac{1}{4\Delta x} \left( (u_{i,j} + u_{i,j+1})(v_{i,j} + v_{i+1,j}) + \right. \\ \left. \alpha |u_{i,j} + u_{i,j+1}| (v_{i,j} - v_{i+1,j}) - (u_{i-1,j} + u_{i-1,j+1})(v_{i-1,j} + v_{i,j}) - \right. \\ \left. \alpha |u_{i-1,j} + u_{i-1,j+1}| (v_{i-1,j} - v_{i,j}) \right)$$

$$FVY = \frac{1}{4\Delta y} \left( (v_{i,j} + v_{i,j+1})^2 + \alpha |v_{i,j} + v_{i,j+1}| (v_{i,j} - v_{i,j+1}) - \right. \\ \left. (v_{i,j-1} + v_{i,j})^2 - \alpha |v_{i,j-1} + v_{i,j}| (v_{i,j-1} - v_{i,j}) \right)$$

$$\text{VISX} = \Delta \left( \frac{1}{4x^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{4y^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \right)$$

$$\text{VISY} = \Delta \left( \frac{1}{4x^2} (v_{i+1,j} - 2v_{i,j} + v_{i-1,j}) + \frac{1}{4y^2} (v_{i,j+1} - 2v_{i,j} + v_{i,j-1}) \right) .$$

Toutes les composantes des vitesses se trouvant dans FUX, FUY, FVX, FVY, VISX, VISY sont évaluées au temps  $n\Delta t$ . Le coefficient de différentiation en amont  $\alpha$  aide à diminuer le risque d'oscillations numériques pouvant survenir lorsque les équations aux différences sont centrées ( $\alpha=0$ ). Si  $\alpha=1$ , alors les équations aux différences sont dites différences avant (forward-difference) reconnues comme étant moins précises que les différences centrées (central-difference). L'optimum de  $\alpha$  se situe entre 0 et 1.

Les itérations sur chaque pas de temps  $\Delta t$  se font selon la technique de Gauss-Seidel point par point. À chaque itération, une correction est appliquée aux termes de pression et de vitesse. Un facteur de relaxation  $\omega$  est utilisé pour accélérer la convergence. L'arrêt des itérations se produit lorsque la valeur de la divergence de la vitesse,  $D$ , est inférieure à un critère de convergence.