



Analyse méthodologique de la modélisation numérique des équations de la physique des milieux continus à l'aide de la méthode des éléments finis : flux-expert, un système d'aide à la construction de logiciels

Philippe Massé

► To cite this version:

Philippe Massé. Analyse méthodologique de la modélisation numérique des équations de la physique des milieux continus à l'aide de la méthode des éléments finis : flux-expert, un système d'aide à la construction de logiciels. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1983. tel-00308718

HAL Id: tel-00308718

<https://theses.hal.science/tel-00308718>

Submitted on 1 Aug 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TH E S E

présentée à

- - -

L'UNIVERSITE L'INSTITUT
SCIENTIFIQUE & MEDICALE NATIONAL POLYTECHNIQUE

DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR D'ETAT ES-SCIENCES

par

Philippe MASSE

*

* *

Analyse méthodologique de la modélisation numérique des équations
de la physique des milieux continus à l'aide de la méthode des
éléments finis.

FLUX-EXPERT : un système d'aide à la construction de logiciels.

*

* *

Soutenue le

devant la Commission d'Examen

J U R Y

A

PREMIÈRE PARTIE

A.I	<u>METHODOLOGIE ET CHOIX DES EXEMPLES</u>	7
A.I.1	DOMAINE D'ANALYSE ET CHOIX MÉTHODOLOGIQUE	7
A.I.2	CHOIX DE LA MÉTHODE NUMÉRIQUE	7
A.II	<u>MODELISATION NUMERIQUE PAR ELEMENTS FINIS</u>	8
A.II.1	MINIMISATION DE FONCTIONNELLE ET MÉTHODES PROJECTIVES	9
A.II.2	TRAITEMENT DES OPÉRATEURS TEMPORELS	13
A.II.3	CAS DES PIÈCES EN MOUVEMENT	21
A.II.4	RÉSOLUTION DES SYSTÈMES NON LINÉAIRES	22
A.II.5	DISCRÉTISATION D'UNE ÉQUATION AUX DÉRIVÉES PARTIELLES PAR LA MÉTHODE DES ÉLÉMENTS FINIS	23
A.III	<u>MODELES DES PHENOMENES MAGNETIQUES</u>	26
A.III.1	LES MODÈLES SIMPLIFIÉS STATIQUES	26
A.III.2	LES MODÈLES SIMPLIFIÉS DYNAMIQUES BIDIMENTIONNELS	30
A.III.3	INFLUENCE DU SYSTÈME DE COORDONNÉES	34
A.IV	<u>MODELE DES PHENOMENES THERMODYNAMIQUES</u>	37
A.V	<u>MODELISATION NUMERIQUE DES EQUATIONS DE NAVIER-STOKES</u>	42
A.V.1	FLUIDES NON VISQUEUX EN RÉGIME STATIONNAIRE	43
A.V.2	ÉTUDE TRIDIMENSIONNELLE DES ÉQUATIONS DE NAVIER-STOKES DANS LA FORMULATION U,V,W,P POUR UN FLUIDE INCOMPRES- SIBLE	44

A.V.3 FORMULATION EN TERMES DE VORTICITÉ ET DE COURANT EN COORDONNÉES CARTÉSIENNES	52
A.V.4 FORMULATION EN TERMES DE VORTICITÉ ET DE COURANT EN COORDONNÉES AXISYMETRIQUES	58
A.VI <u>ETUDE DES EQUATIONS COUPLEES</u>	61
A.VI.1 DOMAINES D'ÉTUDE INCOMPATIBLES	61
A.VI.2 TEMPS DE RÉPONSE	63
A.VI.3 NATURE MATHÉMATIQUE DE LA VARIABLE	64
A.VI.4 TRAITEMENTS SIMULTANÉS ET TRAITEMENTS PARALLÈLES	64
A.VII CONCLUSION	66

B

DEUXIÈME PARTIE

B.I	<u>METHODOLOGIE</u>	70
B.I.1	INTRODUCTION	70
B.I.2	DIVERSES APPROCHES POSSIBLES	71
B.I.3	PRINCIPE D'UN SYSTÈME GÉNÉRATEUR	74
B.II	<u>LES CONCEPTS DE BASE</u>	78
B.II.1	STRATÉGIES, PROBLÈME POTENTIEL ÉLÉMENTAIRE	78
B.II.2	NOTION D'OPÉRATEURS ÉLÉMENTAIRES	81
B.III	<u>DESCRIPTION D'UN PROBLEME POTENTIEL ELEMENTAIRE</u>	89
B.III.1	TYPES DE VARIABLES	89
B.III.2	DÉCLARATION D'OPÉRATEURS	91
B.III.3	DESCRIPTION OPÉRATIONNELLE	92
B.IV	<u>TRAITEMENT D'UN PROBLEME REEL</u>	94
B.IV.1	STRUCTURE GÉNÉRALE DU GRAPHE DE DÉCISION	95
B.IV.2	CONSTRUCTION DU GRAPHE D'UN PROBLÈME RÉEL	98
B.IV.3	EXÉCUTION DU GRAPHE	99
B.IV.4	ENCHAÎNEMENT DYNAMIQUE DES OPÉRATEURS	100

B.V	<u>DETAIL DE CHAQUE CLASSE D'OPERATEURS</u>	100
B.V.1	FORMULE D'INTÉGRATIONS	101
B.V.2	CALCUL DES POLYNOMES	102
B.V.3	ÉVALUATION DES PROPRIÉTÉS	103
B.V.4	CALCUL DES INTÉGRANTS	106
B.V.5	CONSTITUTION DES SOUS-MATRICES	109
B.V.6	CONSTRUCTION DU SOUS-SYSTÈME	109
B.V.7	ASSEMBLAGE DU SYSTÈME LINÉARISÉ	111
B.V.8	RÉSOLUTION DU SYSTÈME LINÉAIRE	112
B.V.9	DEGRÉ DE LIBERTÉ D'UN PROBLÈME POTENTIEL ÉLÉMENTAIRE	112
B.VI	<u>STRATEGIES GENERALES ET MACHINE ABSTRAITE</u>	114
B.VI.1	TRAITEMENT OPÉRATIONNEL DES STRATÉGIES	114
B.VI.2	TRAITEMENT DES STRATÉGIES PAR MACHINE ABSTRAITE	116
B.VI.3	CHOIX DE LA MÉTHODE DU PROTOTYPE	117

TROISIÈME PARTIE

C.I	<u>INTRODUCTION</u>	121
C.II	<u>STRUCTURE GENERALE DU SYSTEME PHASES AUTONOMES</u>	123
C.II.1	ÉTAPES SUCCESSIVES DE LA CONSTRUCTION D'UN SYSTÈME	126
C.II.2	STRUCTURE DES INTERACTIONS UTILISATEUR-SYSTÈME	136
C.III	<u>EXEMPLES D'UTILISATIONS</u>	137
C.III.1	COMPARAISON ENTRE FLUX2D ET SA DESCRIPTION PAR FLUX EXPERT	138
C.III.2	UTILISATION DU FLUX EXPERT POUR RÉSOUDRE UN PROBLÈME THERMODYNAMIQUE TRANSITOIRE	144
C.III.3	UTILISATION DU FLUX EXPERT POUR DÉCRIRE L'ÉQUATION DE NAVIER-STOCKES BIDIMENSIONNELLE	152
C.IV	<u>SYNOPTIQUE DES COMMANDES DU PROTOTYPE</u>	159
C.IV.1	LES COMMANDES DE TÂCHES	159
C.IV.2	LES COMMANDES DE GESTION	160
C.IV.3	LES COMMANDES PROPRES À LA TÂCHE GÉNÉRATEUR	161
C.IV.4	LES COMMANDES DE LA TÂCHE MODÉLISATION DES PROPRIÉTÉS	171
C.IV.5	LES COMMANDES DE LA TÂCHE DE DESCRIPTION DU PROBLÈME RÉEL	172
C.IV.6	LES COMMANDES DE LA TÂCHE DE RÉOLUTION DE PROBLÈME	173
C.IV.7	LES COMMANDES DE LA TÂCHE D'EXPLOITATION DES RÉSULTATS	174

C.V	<u>FONDEMENTS INFORMATIQUES DU SYSTEME</u>	175
C.V.1	BASES DE DONNÉES STATIQUES ET DYNAMIQUES : PROGRAMMES TRANSFERT	175
C.V.2	BIBLIOTHÈQUES D'OPÉRATEURS ET CHARGEUR DYNAMIQUE	177
C.V.3	BIBLIOTHÈQUES FONCTIONNELLES	179
C.VI	<u>CONCLUSION</u>	183

I N T R O D U C T I O N

I N T R O D U C T I O N

La Conception Assistée par Ordinateur (C.A.O.) est une discipline dont le domaine d'application recouvre désormais toute la plage des sciences physiques de l'ingénieur. Mais ses liaisons avec la physique théorique sont souvent occultées par un carcan informatique qui rend difficilement accessibles modèles et hypothèses.

La méthodologie de la Conception Assistée par Ordinateur est immuable :

- le modèle d'une classe d'objets à concevoir a été pensé par le concepteur du système ;
- l'informatique lui a permis de matérialiser ce modèle sous forme de programmes ;
- l'utilisateur du système va pouvoir construire sa propre réalisation dans le cadre générique de ce modèle : en décrivant l'objet particulier qu'il veut étudier ;
- le système prend alors à sa charge les vérifications nécessaires pour déterminer si la description de l'objet à étudier est complète et cohérente vis-à-vis du modèle ; puis il effectue la simulation du fonctionnement de cet objet dans des conditions d'essais fixées par l'utilisateur.

La modélisation est le fondement de toute approche mettant en oeuvre la Conception Assistée par Ordinateur. Or, l'apanage du physicien est justement de modéliser cette réalité qu'il observe à travers le filtre de ses expériences. Nous devons conclure ce syllogisme par "donc le physicien doit être le véritable concepteur des modèles utilisés en CAO".

Ce qui tombe sous le coup de la logique peut être perturbé par des impératifs d'ordre bassement technique. En effet, la "matérialisation informatique d'un modèle" ne va pas sans contraintes dues aux limites imposées par les outils informatiques dont dispose le constructeur de systèmes de C.A.O.

Les modèles souples et évolutifs pensés par le physicien devront s'adapter aux techniques qui vont les matérialiser : de nombreux degrés de liberté vont être ainsi figés au point que, de toutes les variantes envisageables, une seule peut généralement être retenue. Au cours de cette phase, toujours plus ou moins douloureuse, il arrive quelquefois que le contact établi entre physicien et informaticien se rompe au détriment de la qualité du produit.

Car le physicien ne peut pas être le seul concepteur du modèle, il faut lui associer un mathématicien appliqué possédant la maîtrise des techniques informatiques, à moins que lui-même ne se forme à ces méthodes. Cette dernière solution le conduit hélas souvent à délaisser sa discipline d'origine pour voir peu à peu consommer son temps par la dévorante technique informatique.

Ce sont ces problèmes, quotidiennement vécus dans bien des laboratoires, qui nous ont conduit à mener le travail que nous présentons ci-dessous.

- *Le fond* : étudier le processus de matérialisation informatique d'un modèle physique, dans le cadre de la méthode des éléments finis.
- *Le but* : dégager une méthodologie qui assouplisse les contraintes apportées par l'informatique afin que les modèles utilisés dans les outils de C.A.O. puissent être évolutifs et conçus par les physiciens eux-mêmes.

- *La démarche* : proposer des exemples concrets de modèles physiques, généraliser à partir de ces cas afin de définir de nouveaux concepts, matérialiser ces concepts par des outils informatiques généraux.
- *Les résultats* : une méthodologie de construction de système de C.A.O.,
un système informatique mettant cette méthodologie en application dans le cadre de la méthode des éléments finis.
- *Un espoir enfin*, celui d'avoir peut être contribué à la création d'outils informatiques plus souples et mieux adaptés aux besoins des concepteurs de modèles.

Notre mémoire a été structuré en trois parties qui illustrent la démarche inductive que nous avons suivie.

La première partie présente les méthodes numériques classiques généralement utilisées pour traiter les équations aux dérivées partielles. Nous donnons également trois exemples concrets d'application dans les domaines magnétodynamique, thermodynamique et hydrodynamique.

La deuxième partie expose les outils conceptuels que nous avons élaborés afin de proposer une méthodologie générale de construction de système qui permette une description conversationnelle simple des équations aux dérivées partielles usuelles. Nous montrons comment un système générateur peut être mis à la disposition du physicien afin qu'il décrive aisément ses modèles d'équations aux dérivées partielles. Nous détaillons les méthodes de résolution de problèmes qui permettent de concevoir un programme général capable de résoudre tout problème concret posé dans le cadre générique d'un modèle décrit à l'aide du générateur.

La troisième partie expose enfin les caractéristiques du système FLUX-EXPERT que nous avons conçu selon ces principes. Nous décrivons en détail la structure informatique de ce système, et nous donnons en exemple trois utilisations de ce programme pour décrire des équations aux dérivées partielles choisies parmi les centres d'intérêt que nous avons présentés au cours de la première partie : magnétique, thermique et hydraulique.

Dans un document séparé, fourni en annexe, nous présentons la résolution de quelques problèmes concrets à l'aide de ces trois systèmes construits dynamiquement par notre générateur.

1ÈRE PARTIE

DES MODÈLES PHYSIQUES AUX MODÈLES NUMÉRIQUES :
ÉTUDE DE CAS

* * *

A.I METHODOLOGIE ET CHOIX DES EXEMPLES .-

A.I.1 DOMAINE D'ANALYSE ET CHOIX METHODOLOGIQUE

C'est une démarche inductive que nous avons suivie pour notre étude : nous avons analysé la modélisation d'un domaine de la physique, la magnéto-hydro-thermodynamique, qui nous a fourni les exemples à partir desquels nous avons élaboré une méthodologie générale. En effet, les équations de la MHD nous offrent un terrain d'étude suffisamment vaste et complexe présentant une grande variété de problèmes de modélisation : divers types de non linéarité~~s~~ dans les équations, des caractéristiques de matériaux variées, des conditions aux limites très complètes, et surtout la nécessité de coupler des équations mettant en jeu des phénomènes transitoires qui dépendent de constantes de temps très différentes. Nous présenterons tout d'abord chacun des problèmes magnétique~~s~~, thermique~~s~~, hydraulique~~s~~, de façon séparée en dégagant les caractéristiques qui leur sont propres, puis nous étudierons les difficultés posées par le couplage entre ces équations.

Nous avons tenté de ne conserver dans cette présentation que les points caractéristiques qui nous ont conduit à prendre des décisions.

A.I.2 CHOIX DE LA METHODE NUMERIQUE

Pour traiter les problèmes d'équations aux dérivées partielles dans des domaines aux formes compliquées, trois méthodes sont généralement utilisables : Différences finies, Eléments finis, Intégrales de frontières. Notre propos n'est pas de dresser ici une analyse comparative de ces trois méthodes, ceci à déjà été fait et nous conduirait hors du cadre de ce travail [14], [29].

La grande souplesse d'adaptation de la méthode des éléments finis à modéliser des phénomènes complexes dans des géométries élaborées, a conduit à la généralisation de son utilisation dans presque tous les problèmes aux dérivées partielles. C'est pourquoi notre étude sera menée dans ce cadre général, mais la méthode des différences finies sera également utilisée par notre système pour le traitement des phénomènes d'évolution temporelle [37]. Quant à la méthode intégrale, son domaine de prédilection demeure toujours le traitement des problèmes linéaires [26]. Bien que les phénomènes analysés soient pour la plupart non linéaires, de grandes parties des domaines étudiés ont souvent des caractéristiques linéaires. Nous n'avons donc pas exclu un couplage entre la méthode des éléments finis et la méthode intégrale, dans ces cas particuliers, à l'aide de macro-éléments ; mais nous n'avons pas approfondi ce problème difficile sur lequel beaucoup d'études sont en cours [40], [78], [86].

A.II MODELISATION NUMERIQUE PAR ELEMENTS FINIS .-

D'excellents ouvrages généraux ont été écrits sur la méthode des éléments finis, [18] [19] [24] [31] [39], c'est pourquoi, dans ce chapitre, nous ne présenterons brièvement que les notions dont nous aurons besoin dans la suite de ce mémoire. Les références bibliographiques permettront au lecteur de consulter le détail de chaque thème abordé.

Nous rappelons que notre propos n'est pas d'élaborer une nouvelle méthode de traitement des équations aux dérivées partielles mais d'analyser le processus de modélisation numérique de phénomènes physiques très généraux à l'aide de méthodes déjà classiques.

A.II.1 MINIMISATION DE FONCTIONNELLE ET MÉTHODES PROJECTIVES

La méthode des éléments finis repose sur la recherche d'une fonction globale représentant les phénomènes étudiés en tout point du domaine analysé. Ce domaine doit être décomposé au préalable en parties finies, adjacentes, appelées les éléments finis. La fonction globale recherchée sera construite par parties sur chacun de ces éléments.

L'inconnue ainsi élaborée devra vérifier globalement les équations aux dérivées partielles, les conditions aux limites associées et les propriétés de continuité requises par les phénomènes étudiés, cela sur chacun des éléments comme sur l'ensemble du domaine. Selon les équations traitées on peut généralement démontrer que si la taille de chaque élément tend vers zéro (le nombre d'éléments tendant vers l'infini), les équations aux dérivées partielles sont alors vérifiées en tout point du domaine.

Les éléments finis utilisés pour discrétiser le domaine étudié sont généralement regroupés par familles topologiques : segments, triangles, quadrilatères, tétraèdres, parallélépipèdes, prismes, etc... L'ensemble des éléments qui reconstituent le domaine est dénommé le découpage ou la discrétisation géométrique.

A l'intérieur d'un élément la fonction qui modélise le phénomène est définie par une interpolation polynomiale en fonction des valeurs de l'inconnue en chacun des sommets (ou noeuds) de cet élément.

Le principe de la méthode consiste à trouver la répartition de valeurs nodales qui entraîne la vérification de l'équation aux dérivées partielles du phénomène étudié et qui remplisse ses conditions aux limites.

Deux approches peuvent alors être suivies : la méthode

variationnelle ou une méthode projective.

METHODE VARIATIONNELLE DE RITZ RAYLEIGH.

Elle consiste à minimiser l'expression :

$$(1) \quad E = \iiint_{\Omega} F(x, y, z, I, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial z} \dots) dx dy dz + \iint_{\Gamma} \Phi(x, y, z, I) d\Gamma$$

où F et Φ sont des fonctionnelles supposées connues et I , $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial z}$... respectivement l'inconnue et ses dérivées.

L'équation d'Euler appliquée à ce problème fournit le problème différentiel équivalent :

$$(2) \quad \frac{\partial F}{\partial I} - \nabla \cdot V = 0 \text{ en tout point du domaine}$$

avec comme conditions aux limites :

$$(3) \quad \frac{\partial \Phi}{\partial I} + V \cdot n = 0 \text{ en tout point de la surface}$$

où V est le vecteur ayant pour composantes

$$\frac{\partial F}{\partial (\frac{\partial I}{\partial x})}, \quad \frac{\partial F}{\partial (\frac{\partial I}{\partial y})}, \quad \frac{\partial F}{\partial (\frac{\partial I}{\partial z})}$$

Si on approxime ^{che} la fonction inconnue I par une famille de fonctions :

$f_m(x, y, z, I_1, I_2, \dots, I_n)$ où les termes $I_1, I_2, \dots, I_n$ sont des paramètres (en fait les valeurs de l'inconnue aux noeuds du découpage ou éléments finis), les fonctionnelles F et Φ deviennent de simples fonctions de $x, y, z, I_1, I_2, \dots, I_n$; et l'intégrale E devient une fonction des seules valeurs $I_1, I_2, \dots, I_n$. La minimisation de cette quantité conduit alors au système d'équations :

$$(4) \quad \frac{\partial E}{\partial I_i} = 0$$

On choisit les fonctions f_m de telle sorte que
 $I_m(x,y,z) = f_m(x,y,z, \dot{i}_1, \dots, \dot{i}_n) = \sum_{j=1}^n \alpha_{mj}(x,y,z) I_j$ où les fonctions $\alpha_{mj}(x,y,z)$ vérifient les conditions aux limites et le degré de continuité nécessaire. Dans ce cas le système d'équations (4) fournit un système algébrique linéaire

$$(5) \quad [H] \dot{I} = R$$

Les termes de la matrice H et du vecteur R étant totalement déterminés par des intégrales de volume et de surface mettant en oeuvre les fonctions α_{mi} et leurs dérivées, qui ne dépendent que de la géométrie du domaine étudié.

Cette méthode n'est donc applicable que si on connaît une fonctionnelle équivalente au problème différentiel que l'on veut résoudre. Or, dans bien des cas on ne peut pas trouver de fonctionnelle [] notamment en ce qui concerne les phénomènes variant dans le temps et les équations associant des termes dérivés d'ordre pairs et impairs simultanément (cas des équations de Navier-Stokes avec $V \cdot \nabla V$ et $\nabla \cdot \eta \nabla V$).

METHODES PROJECTIVES, METHODE DE GALERKINE.

Elles traitent directement l'équation aux dérivées partielles que l'on veut résoudre.

$$(6) \quad D(x, y, z, I, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \dots) = 0$$

Comme dans la méthode de Ritz on va approximer ^{cher} les fonctions inconnues par des familles de fonctions de la forme $I_m = \sum_{j=1}^n \alpha_{mj}(x,y,z) I_j$. Si on choisit pour I_i une répartition quelconque on aura $D(x, \dots) \neq 0$

Cette dernière valeur, non nulle lorsque les valeurs I_i ne vérifient pas l'équation, est appelée le résidu.

La méthode consiste à projeter ce résidu sur une

famille de fonctions W_i indépendantes à l'aide d'un produit scalaire adéquat :

$$(7) \quad \iiint_{\Omega} \mathcal{D} w_i d\Omega$$

Les fonctions W_i engendrent un espace dans lequel pour que $\mathcal{D}$ soit nul il est nécessaire qu'il soit simultanément orthogonal à tous les vecteurs de base. De plus, comme la valeur de $\mathcal{D}$ ne dépend que des constantes $i_1, i_2, \dots, i_n$, nous disposons de n degrés de liberté, ce qui fixe donc le nombre des fonctions W_i et ~~par~~ conséquence le nombre d'équations.

La méthode de Galerkinge conduit souvent à prendre comme fonctions de projection les fonctions α_i elles-mêmes. Ceci conduit généralement alors au même système que celui que fournit la méthode de Ritz (quand une fonctionnelle existe) [39] mais d'autres fonctions peuvent être également choisies, comme nous le verrons pour les problèmes axisymétriques ou dans des cas d'inconnues vectorielles possédant des composantes interpolées par des polynômes de degrés différents. (On consultera l'ouvrage de Finlayson [20] pour approfondir la théorie de ces méthodes).

L'avantage de la méthode de Galerkinge réside donc dans le fait qu'elle ne nécessite pas la connaissance d'un principe variationnel associé à l'équation aux dérivées partielles traitée.

Dans les cas où un principe variationnel existe, on aura toujours intérêt à l'utiliser car d'une part il conduit directement à une forme symétrique des équations et d'autre part son interprétation énergétique est fort utile pour le calcul de grandeurs intégrales comme :

- les énergies stockées dans les dispositifs étudiés,
- les forces,
- les couples,

- et beaucoup de grandeurs globales que l'on peut souvent ainsi obtenir plus directement [74], [75].

Dans ce qui précède nous n'avons abordé que des problèmes d'équations aux dérivées partielles par rapport à l'espace, dans le cas où les équations dépendent du temps (équations de type parabolique ou hyperbolique) il faut envisager de nouvelles méthodes.

A.II.2 TRAITEMENT DES OPÉRATEURS TEMPORELS

Ce sont les techniques très classiques de résolution des systèmes d'équations différentielles qui sont utilisées pour traiter les phénomènes transitoires ou périodiques.

En effet, la méthode de Galerkin peut être appliquée à une équation dépendant du temps :

$$(8) \quad \mathcal{D} \left(\bar{I}, \frac{\partial \bar{I}}{\partial t}, \frac{\partial^2 \bar{I}}{\partial t^2}, \frac{\partial \bar{I}}{\partial x}, \frac{\partial \bar{I}}{\partial y}, \dots \right)$$

sans aucune difficulté particulière.

Elle nous conduit alors à deux types de systèmes différentiels suivant la nature de l'équation.

Pour les équations paraboliques on obtient un système de la forme :

$$(9) \quad [L] \frac{\partial \bar{I}}{\partial t} + [M] \bar{I} = R$$

Alors que pour les équations hyperboliques on arrive à :

$$(10) \quad [L] \frac{\partial^2 \bar{I}}{\partial t^2} + [M] \bar{I} = R$$

Les méthodes les plus couramment utilisées pour résoudre les équations différentielles de ce type sont :

- l'utilisation des nombres complexes,
- les méthodes de différences finies :
 - . les méthodes à pas séparés
 - . les méthodes à pas liés
 - . les méthodes de prédiction-corrrection.

Nous allons brièvement discuter l'adéquation de chacune aux problèmes éléments finis et nous ne détaillerons que les méthodes les plus adaptées à ces problèmes.

A.II.2.1 Les nombres complexes

Leur utilisation est limitée aux équations dont la matrice $[M]$ est obtenue par la discrétisation d'un problème elliptique linéaire (ΔI) et dont la matrice $[L]$ est constante. Alors, si les sources sont de formes sinusoïdales $\bar{R} = Re^{j(\omega t + \phi)}$, la solution peut être recherchée sous la forme d'un nombre complexe pour lequel

(11) $j\omega \bar{I}$ représente $\frac{\partial I}{\partial t}$ et $-\omega^2 \bar{I}$ représente $\frac{\partial^2 I}{\partial t^2}$ ce qui ramène le problème à :

$$(12) \quad (j\omega[L] + [M]) \bar{I} = \bar{R}$$

ou bien

$$(13) \quad ([M] - \omega^2[L]) \bar{I} = \bar{R}$$

Dans le cas où l'équation traitée permet leur utilisation, les nombres complexes fournissent donc une méthode extrêmement efficace qui sera préférée à toutes les autres.

Pour les problèmes linéaires périodiques non sinusoïdaux, la décomposition en séries de Fourier permet de ramener le problème général à la résolution des équations (12) ou (13) pour chaque harmonique.

A.II.2.2 Les différences finies

Elles consistent à décomposer en série de Taylor les phénomènes dépendant du temps et ~~à~~ ne conserver qu'un nombre restreint de termes.

$$I_n = I_{n-1} + \left(\frac{\partial I}{\partial t}\right)_{n-1} \Delta t + \left(\frac{\partial^2 I}{\partial t^2}\right)_{n-1} \Delta t^2 + \dots$$

Ces méthodes sont très utilisées car elles allient une grande simplicité ~~à~~ la possibilité d'obtenir plusieurs schémas de discrétisations ~~possibles~~ selon la précision ou la stabilité désirées.

. La méthode de Runge-Kutta

Elle met en oeuvre une discrétisation généralement d'ordre 4 qui implique de pouvoir expliciter les valeurs $\frac{\partial I}{\partial t}$, ce qui impose alors d'inverser la matrice $[L]$. Ceci est quelquefois possible directement, mais de nombreux phénomènes physiques conduisent à des problèmes où $[L]$ est une matrice singulière. Dans ces cas délicats il faut alors évaluer la restriction de $[L]$ qui est régulière et ne traiter que le sous-problème ainsi défini. D'autre part, cette méthode a un rayon de stabilité assez faible ($p=2,8$) ce qui risque d'imposer des pas de temps faibles. En conséquence de ces problèmes, cette méthode n'est généralement pas utilisée, bien que dans certains cas particuliers elle puisse fournir de très bons résultats et cela de façon très économique [121].

. Les méthodes d'ordre 1 ou 2 à pas séparés

On développe en série de Taylor l'inconnue I au voisinage de la solution :

$$(14) \quad I_n = I_{n-1} + \Delta t \left(\frac{\partial I}{\partial t}\right)_{n-1} + \frac{1}{2} \Delta t^2 \left(\frac{\partial^2 I}{\partial t^2}\right)_{n-1} + \frac{1}{3!} \Delta t^3 \left(\frac{\partial^3 I}{\partial t^3}\right)_{n-1} + \dots$$

ce qui fournit une approximation pour $\frac{\partial I}{\partial t}$.

. Pour obtenir une approximation de $\left(\frac{\partial^2 I}{\partial t^2}\right)$ il faut exécuter un développement supplémentaire qui lie les valeurs I_n et I_{n+1} .

Nous supposons que l'équation différentielle à résoudre est étudiée à l'instant $n\delta$ pris égal à $t_{n-1} + \delta\Delta t$ avec $0 \leq \delta \leq 1$ et nous définirons ces valeurs intermédiaires (15) comme la moyenne pondérée $I_{n\delta} = \delta I_{n-1} + (1-\delta) I_n$; ceci conduit aux approximations suivantes :

$$(16) \quad \left(\frac{\partial I}{\partial t}\right)_{n\delta} = \frac{I_n - I_{n-1}}{\Delta t}$$

et

$$(17) \quad \left(\frac{\partial^2 I}{\partial t^2}\right)_{n\delta} = \frac{I_{n+1} - 2I_n + I_{n-1}}{\Delta t^2}$$

ce qui, reporté dans les équations respectives, conduit aux systèmes :

Cas parabolique :

$$(18) \quad \left([L] + \Delta t (1-\delta) [M]\right) I_n = \Delta t \left(\delta R_{n-1} + (1-\delta) R_n\right) + \left([L] - \Delta t \delta [M]\right) I_{n-1}$$

Cas hyperbolique :

$$(19) \quad \left([L] + \Delta t^2 (1-\delta) [M]\right) I_{n+1} = \Delta t^2 \left(\delta R_{n-1} + (1-\delta) R_n\right) - \left[-2[L] + \Delta t^2 \delta [M]\right] I_n - [L] I_{n-1}$$

Remarques :

Dans ce qui précède nous avons supposé que les matrices $[L]$ et $[M]$ ne dépendaient pas de l'inconnue i .

. D'une manière générale, pour les équations paraboliques, les méthodes implicites : $\delta=0$, et Crank Nicholson : $\delta=\frac{1}{2}$, sont inconditionnellement stables alors que la méthode explicite $\delta=1$, fait intervenir une condition de stabilité en $\frac{\Delta t}{\Delta x^2}$ où x est le "pas" de la discrétisation spatiale [49]. Cette condition impose généralement des pas de temps beaucoup trop faibles, c'est pourquoi les méthodes $\delta=0$ et $\delta=1/2$, sont presque toujours préférées.

. Dans le cas d'une équation hyperbolique le schéma explicite $\delta=1$ est également conditionnellement stable mais fait intervenir un terme moins défavorable aux grands pas de temps car la stabilité dépend alors du rapport $\frac{\Delta t}{\Delta x}$ [38].

Dans le cas non linéaire où la matrice $[M]$ dépend de l'inconnue I les formules obtenues deviennent plus complexes.

Nous ne les détaillerons que pour le cas parabolique :

$$(20) \quad I_n = I_{n-1} + \Delta t \left(\delta \left(\frac{\partial I}{\partial t} \right)_{n-1} + (1-\delta) \left(\frac{\partial I}{\partial t} \right)_n \right)$$

qui multiplié par $[L]$ matrice indépendante de I conduit à :

$$(21) \quad [L] I_n = [L] I_{n-1} + \Delta t \left(\delta [L] \left(\frac{\partial I}{\partial t} \right)_{n-1} + (1-\delta) [L] \left(\frac{\partial I}{\partial t} \right)_n \right)$$

or $[L] \left(\frac{\partial I}{\partial t} \right)_{n-1} = R_{n-1} - [M]_{n-1} I_{n-1}$
 et $[L] \left(\frac{\partial I}{\partial t} \right)_n = R_n - [M]_n I_n$

ce qui fournit :

$$(22) \quad ([L] + \Delta t (1-\delta) [M]_n) I_n = ([L] - \Delta t \delta [M]_{n-1}) I_{n-1} + \Delta t (\delta R_{n-1} + (1-\delta) R_n)$$

. Enfin dans le cas où $[L]$ est une matrice également dépendante de I la méthode centrée ne permet plus de séparer les termes en I_n et I_{n-1} sans inverser la matrice. Nous revenons alors à un schéma purement implicite :

$$(23) \quad \left(\frac{\partial I}{\partial t} \right)_n = \frac{I_n - I_{n-1}}{\Delta t}$$

$$\text{et } [L_n] \left(\frac{\partial I}{\partial t} \right)_n = R_n - [H_n] I_n \text{ d'après (9)}$$

ce qui donne

$$(24) \quad [L_n] (I_n - I_{n-1}) + \Delta t [H_n] I_n = \Delta t R_n$$

. Les méthodes à pas liés

Elles permettent d'obtenir une très bonne précision mais imposent le stockage des valeurs sur un grand nombre de pas et conduisent à des systèmes à résoudre qui sont souvent prohibitifs vu le très grand nombre d'équations à traiter.

Le domaine d'utilisation de ces méthodes demeure les problèmes différentiels mettant en oeuvre quelques dizaines d'équations simultanées

. Méthodes prédiction correction

Devant le grand nombre d'itérations que peut demander la résolution d'un problème non linéaire par une méthode implicite, on doit chercher à minimiser les calculs à effectuer.

A chaque pas de temps on doit évaluer l'inconnue au temps suivant à l'aide d'une méthode itérative. Les performances de ces méthodes, que nous verrons au prochain chapitre, dépendent bien sûr de la proximité de la valeur de départ

qu'on doit leur fournir, par rapport à la solution escomptée. On peut certes leur fournir comme valeur celle de l'inconnue au pas précédent mais on améliorera cela en approx^{chant}imant la valeur de départ à l'aide d'une méthode directe.

Ce principe est celui de la méthode prédiction correction qui se déroule donc en deux temps :

- évaluation d'une valeur de départ aussi proche que possible de la solution (par une méthode peu onéreuse si possible),
- calcul de la solution par une méthode implicite qui assure alors la stabilité du schéma.

Les techniques de prédiction sont généralement fondées sur des méthodes directes explicites.

Prenons le cas d'une méthode d'ordre deux : on développe en série la solution au voisinage du pas n .

$$(25) \quad I_{n+1} = I_n + \Delta t_n \left(\frac{\partial I}{\partial t} \right)_n + \frac{\Delta t_n^2}{2} \left(\frac{\partial^2 I}{\partial t^2} \right)_n + \frac{\Delta t_n^3}{3!} \left(\frac{\partial^3 I}{\partial t^3} \right)_n + \dots$$

de même au pas précédent

$$(26) \quad \left(\frac{\partial I}{\partial t} \right)_{n-1} = \left(\frac{\partial I}{\partial t} \right)_n - \Delta t_{n-1} \left(\frac{\partial^2 I}{\partial t^2} \right)_n + \frac{\Delta t_{n-1}^2}{2} \left(\frac{\partial^3 I}{\partial t^3} \right)_n + \dots$$

ce qui conduit à

$$(27) \quad I_{n+1} = I_n + \Delta t_n \left(\frac{\partial I}{\partial t} \right)_n + \frac{\Delta t_n^2}{2} \frac{\left(\frac{\partial I}{\partial t} \right)_n - \left(\frac{\partial I}{\partial t} \right)_{n-1}}{\Delta t_{n-1}} + \dots$$

avec une erreur de troncature :

$$(28) \quad \varepsilon_p = \left(\frac{\Delta t_n^2 \Delta t_{n-1}}{4} + \frac{\Delta t_n^3}{6} \right) \frac{\partial^3 I}{\partial t^3} \approx I_p^{n+1} - I_{n+1}^{\text{exact}}$$

Ceci fournit donc un moyen d'obtenir la valeur prédite I_{pn+1} .

Le schéma implicite déjà rencontré (24) fournira alors la valeur corrigée I_{cn+1} . Cela nécessite la mémorisation supplémentaire des dérivées $(\frac{\partial I}{\partial t})_n$ et $(\frac{\partial I}{\partial t})_{n-1}$, et le calcul de

$$(29) \quad \left(\frac{\partial I}{\partial t} \right)_{n+1} = \frac{2 (I_{n+1} - I_n)}{\Delta t_n} - \left(\frac{\partial I}{\partial t} \right)_n$$

mais les calculs de (27) et de (29) sont très peu coûteux par rapport au nombre d'itérations qu'ils font économiser dans la résolution de la méthode implicite directe. L'équation non linéaire (24) devient alors :

$$(30) \quad \left[\mathcal{L}_{n+1}(I_{pn+1}) \right] (I_{n+1} - I_n) + \Delta t \left[\mathcal{H}_{n+1}(I_{pn+1}) \right] I_{n+1} = \Delta t \mathcal{R}_{n+1}$$

Les matrices $\mathcal{L}$ et $\mathcal{H}$ qui dépendent de I_{n+1} sont alors calculées avec la valeur prédite I_{pn+1} .

Dans l'équation (30) on peut également évaluer l'erreur due à la troncature de la série ce qui donne (en tenant compte de (26))

$$(31) \quad \varepsilon_n = \frac{\Delta t_n^3}{12} \left(\frac{\partial^3 I}{\partial t^3} \right)_n \simeq I_{n+1 \text{ calculé}} - I_{\text{exact}}$$

Les deux évaluations séparées (28) et (31) permettent d'approximer directement ε_n car le calcul nous fournit I_{pn+1} et I_{n+1} ce qui donne :

$$(32) \quad \varepsilon_n = \frac{I_{n+1} - I_{pn+1}}{3 \left(1 + \frac{\Delta t_{n-1}}{\Delta t_n} \right)}$$

On voit donc que cette méthode permet très simplement d'évaluer l'erreur commise à chaque pas par (32), et donc en utilisant (31) d'en déduire une valeur optimale du pas de temps suivant :

Si ε^* représente l'erreur que l'on accepte pour l'étude en cours on obtient :

$$(33) \quad \Delta t_{n+1} = \Delta t_n \sqrt[3]{\frac{\varepsilon^*}{\varepsilon_n}}$$

Dans le cas où cette formule conduit à un pas Δt_{n+1} nettement plus petit que Δt_n il est alors prudent de reprendre les calculs à l'itération précédente en divisant alors Δt_n par 2.

A.II.3 CAS DES PIÈCES EN MOUVEMENT

L'étude de phénomènes physiques dans des ensembles comportant des pièces fixes et des pièces en mouvement conduit à considérer les dérivées temporelles comme $\frac{D}{Dt} = \frac{\partial}{\partial t} + V \cdot \nabla$, où V est la vitesse des solides ou des liquides en mouvement par rapport au repère fixe.

La méthode de Galerkin est encore valable pour l'opérateur $V \cdot \nabla$ mais les études en différences finies [33] [36] ont montré que les schémas de discrétisation symétrique (différences centrées) étaient instables. Dans la méthode des différences finies, le choix d'un schéma décentré amont permet de remédier à ce problème.

Dans la méthode des éléments finis, un phénomène similaire peut être mis en évidence : avec des fonctions de projections égales aux fonctions d'interpolation, on remarque une oscillation des résultats selon la discrétisation choisie. Le phénomène est moins sensible qu'avec les différences finies (où l'on rencontrerait une divergence importante) mais justifie

des études particulières qui ont conduit à l'utilisation de fonctions de projection mieux adaptées [105] [106].

A.II.4 RÉSOLUTION DES SYSTÈMES NON LINÉAIRES

Dans les paragraphes précédents nous avons vu que de nombreuses discrétisations conduisaient à des systèmes d'équations algébriques non linéaires.

Il semble que ce soit la méthode de Newton-Raphson qui conduise le plus efficacement aux résultats. Le principe de cette méthode consiste à développer en série la valeur du résidu au voisinage de la solution.

Si nous notons $\vec{I}$ la variable vectorielle représentant l'ensemble des valeurs nodales de l'inconnue, le système d'équations non linéaires à résoudre peut s'écrire $F(\vec{I}) = 0$. Soit $\vec{I}_0$ la solution recherchée. Il est possible de développer $F(\vec{I})$ au voisinage de $\vec{I}_0$ par :

$$(35) \quad F(\vec{I}) = F(\vec{I}_0) + \left[\frac{\partial F(\vec{I}_0)}{\partial \vec{I}} \right] \Delta \vec{I}_0 \approx 0$$

d'où $\vec{I}^{(k+1)} = \vec{I}^{(k)} + \Delta \vec{I}^{(k)}$

$$(36) \quad \left[\frac{\partial F(\vec{I}^{(k)})}{\partial \vec{I}^{(k)}} \right] \Delta \vec{I}^{(k)} = -F(\vec{I}^{(k)})$$

ou $\left[\frac{\partial F}{\partial \vec{I}} \right]$ représente la matrice Jacobienne du système $F(\vec{I})=0$.

Cet algorithme ne converge généralement que si la valeur de départ choisie n'est pas trop éloignée de la solution. C'est pourquoi une grande précaution doit être prise pour

élaborer ces valeurs de démarrage, ce qui justifie quelquefois d'utiliser des techniques de prédiction peu coûteuses mais très efficaces. Dans des cas délicats on peut avoir recours à des méthodes incrémentielles qui construisent artificiellement une suite de problèmes voisins ²portant d'un état connu uniforme pour parvenir aux conditions particulières de l'étude. Ainsi, chaque solution précédente fournit un "bon" point de départ pour l'approche suivante. C'est un véritable régime transitoire artificiel qui est ainsi reconstitué. (Dans les cas où plusieurs solutions peuvent exister c'est une technique qui est pratiquement indispensable : cas des équations de Navier-Stokes).

A.II.5 DISCRÉTISATION D'UNE ÉQUATION AUX DÉRIVÉES PARTIELLES PAR LA MÉTHODE DES ÉLÉMENTS FINIS

Il est toujours périlleux de vouloir proposer un guide méthodologique qui aspire à la généralité, mais l'analyse de nombreux cas particuliers nous a conduit à dégager un cadre très général au sein duquel la plupart des études de problèmes d'équations aux dérivées partielles pourront être menées à bien.

Quatre étapes sont généralement nécessaires pour transformer une équation fournie par la physique en un ensemble de systèmes algébriques linéaires traitables par un ordinateur.

a) Projection du résidu des équations à résoudre sur une famille de fonctions β_i par la méthode de Galerkin.

On dispose alors d'un système d'équations aux dérivées partielles de la forme :

$$(37) \quad \iiint \beta_i F\left(I, \frac{\partial I}{\partial x}, \dots, \frac{\partial I}{\partial t}, \dots\right) = 0$$

b) Prise en compte des conditions aux limites.

- les conditions de type Dirichlet (variable connue sur la limite) conduisent simplement à diminuer le nombre de degrés de liberté dans le choix des fonctions β_i et sont traitées par la suppression des équations correspondantes.

- les conditions naturelles sont obtenues par intégration par partie des termes dérivés d'ordre le plus élevé dans l'équation (37). Les termes obtenus sont alors transformés à l'aide du théorème d'Ostrogradski (en 3 dimensions) ou de Stokes (en 2 dimensions) pour conduire à des expressions dont la structure est généralement :

$$\iint_{\Gamma} \beta_i \mathcal{L} \cdot n \, d\Gamma \quad (\text{en 3D})$$

ou

$$\int_L \beta_i \mathcal{L} \cdot n \, dl \quad (\text{en 2D})$$

qui modélisent les échanges entre le domaine étudié et l'extérieur.

$\mathcal{L}$ étant un vecteur dépendant du problème à traiter (éventuellement fonction de l'inconnue)

n étant la normale extérieure à la surface du domaine.

c) Discrétisation des opérateurs de dérivation temporelle (s'ils existent dans l'équation) à l'aide d'un schéma aux différences finies.

Le choix de la méthode de discrétisation sera fortement dépendant de l'équation traitée mais on peut dégager les critères généraux suivants :

. Si les systèmes obtenus peuvent se mettre sous une forme matricielle à coefficients indépendants de l'in-

connue, (problème linéaire) les méthodes de type Crank-Nicholson assureront une précision correcte et une bonne stabilité.

. Si les coefficients dépendent de l'inconnue, les méthodes implicites seront alors préférées malgré leur précision plus faible et "l'effet de lissage" qu'elles présentent.

. Dans le cas de fortes non linéarités des méthodes de prédiction pourront être mises en oeuvre afin d'accélérer les convergences des processus de résolution des systèmes non-linéaires.

d) Discrétisation des inconnues à l'aide des polynômes d'interpolation et linéarisation.

Cette étape transforme alors les systèmes d'équations obtenus en systèmes algébriques. Si ces systèmes sont linéaires on peut alors considérer que l'étude mathématique est achevée. Dans le cas non linéaire, généralement, on linéarise le système obtenu à l'aide de la méthode de Newton-Raphson. Il faut donc évaluer l'expression littérale de la matrice Jacobienne du système à traiter, en dérivant chacune des équations par rapport à chacune des variables (ce qui conduit souvent à l'apparition de nombreux termes nouveaux, produits par les dérivations partielles nécessaires).

e) Enfin les conditions initiales devront être étudiées en détail.

Elle seront généralement fournies pour les problèmes d'évolution temporelle, mais dans les cas non linéaires statiques il faudra le plus souvent prévoir une analyse spéciale qui fournisse des valeurs initiales au procédé de résolution itératif.

Dans les chapitres qui suivent nous présentons de nombreux exemples concrets d'application de cette démarche dans les domaines magnétiques thermiques et hydrauliques.

A.III MODELES DES PHENOMENES MAGNETIQUES

Si nous négligeons les courants de déplacement ($\frac{\partial D}{\partial t}$) les équations de Maxwell s'écrivent :

$$(1) \quad \nabla \times E = - \frac{DB}{Dt} ; \quad \nabla \times H = I ; \quad \nabla \cdot B = 0 ; \quad \nabla \cdot D = \rho$$

avec $D = \epsilon E$ et $B = \mu H + B_r$

où ϵ et μ sont des tenseurs.

Si nous ajoutons à ces équations la loi d'Ohm,

$$(2) \quad I = \sigma E$$

tous les modèles classiques en découlent par le biais d'hypothèses simplificatrices successives.

Nous allons considérer quelques modèles parmi les plus utilisés. Nous ne détaillerons que les caractéristiques dont nous devrons tenir compte lors de la modélisation numérique.

A.III.1 LES MODÈLES SIMPLIFIÉS STATIQUES

- Le modèle de conduction électrique :

La loi de conservation du courant électrique dans un conducteur s'écrit $\nabla \cdot I = 0$. D'où, compte-tenu de $\nabla \times E = 0$, on obtient :

$$(3) \quad \nabla \cdot (\sigma \nabla \phi) = 0 \quad \text{en ayant posé} \quad E = -\nabla \phi.$$

La formule variationnelle équivalente s'écrit :

$$(4) \quad \min \left(\int_{\Omega} \frac{1}{2} I \cdot E + \int_{\Gamma} I \cdot n \phi \right)$$

où $I.n$ est la composante de I normale à la surface Γ .

Cette équation est la plus simple car elle est dépourvue de terme source et de plus, les conditions aux limites les plus couramment rencontrées dans les cas concrets sont très simples :

- . Φ potentiel électrique connu sur une section du conducteur d'arrivée de courant,
- . $I.n=0$ sur la limite extérieure d'un conducteur.
- . De plus σ est généralement indépendante de Φ , ce qui conduit au problème $\Delta\Phi=0$.

- Le modèle électrostatique :

L'absence de termes d'évolution ($:\frac{DB}{Dt}=0$) découple les équations de Maxwell et conduit à :

$$(5) \quad \nabla \times E = 0 \quad ; \quad \nabla \cdot D = \rho \quad \text{donc} \quad E = -\nabla \Phi$$

$$\text{d'où} \quad \nabla \cdot (\epsilon \nabla \Phi) = -\rho$$

avec la formulation variationnelle équivalente :

$$(6) \quad \min \left(\int_{\Omega} D \cdot dE - \rho \Phi + \int_{\Gamma} D_n \Phi \right) \quad [61]$$

les conditions aux limites naturelles s'expriment en fonction de grandeurs physiques facilement accessibles :

- . Potentiel électrique connu sur la limite Γ ;
- . Condition de symétrie où la composante normale de D est nulle : $\int_{\Gamma} D.n=0$

Cette équation (6) est la plus classique, elle est traitée par de nombreux codes [70] [73] [154] ... généralement en limitant les problèmes de conditions aux limites à $\Phi_{\Gamma}=f(x,y,z)$

ou $\int_{\Gamma} D \cdot n = 0$ et en supposant que la valeur de ϵ est indépendante de E (matériau linéaire).

- Les modèles magnétostatiques :

Là encore $\frac{DB}{Dt} = 0$ découple les équations de Maxwell qui se réduisent à :

$$(7) \quad \nabla \cdot B = 0 \quad ; \quad \nabla \times H = I$$

Dans les portions de matériaux dépourvues de sources, on obtient $\nabla \times H = 0$, d'où l'on déduit que le champ H dérive d'un potentiel scalaire ψ . Par ailleurs, le champ produit par les courants peut être calculé isolément par la loi de Biot et Savart, ce qui permet de réduire considérablement le nombre d'inconnues par rapport à l'utilisation d'un potentiel vecteur.

En présence de courant c'est alors l'équation :

$$(8) \quad \nabla \times \nabla \times A = I$$

avec $\nabla \cdot A = 0$ qu'il faut résoudre.

Dans ces deux cas une approche variationnelle peut être utilisée :

. Pour le potentiel scalaire la fonctionnelle est :

$$(9) \quad \iint [\int_0^H B \cdot dH] + \int_{\Gamma} B \cdot \frac{\partial \psi}{\partial n} \quad \text{ou} \quad H = H_c - \nabla \psi$$

. Alors que dans le cas du potentiel vecteur on obtient :

$$(10) \quad \iint [\int_0^B H dB - I A]$$

Tous ces modèles statiques simplifiés peuvent être traités de façon très classique et sont caractérisés par la présence d'un terme du 2ème ordre : soit $\nabla \cdot k(\nabla \dot{I})$ soit $\nabla \times (k \nabla \times I)$ dans lesquels k est une propriété dépendant éventuellement de l'inconnue $\dot{I}$. Cette non linéarité est généralement traitée par la méthode de Newton-Raphson dont la matrice Jacobienne comporte alors deux termes distincts :

$$\iint k \nabla \beta_i \cdot \nabla \alpha_j + \iint \frac{\partial k}{\partial \dot{I}_j} \nabla \beta_i \cdot \nabla \dot{I}$$

la grandeur $\frac{\partial k}{\partial \dot{I}_j}$ n'a généralement pas de sens physique direct et il est nécessaire de passer par l'intermédiaire des dérivations partielles auxiliaires. Prenons l'exemple du terme $\nabla \cdot \mu \nabla \Phi$: on remarque que μ est une fonction du gradient de l'inconnue principale : $\mu = f(H) = f(\nabla \Phi)$; les perméabilités étant généralement mesurées au laboratoire en fonction du module du champ H , ce sont les courbes $\mu = f(H.H)$ dont on peut disposer, ce qui conduit à :

$$(11) \quad \frac{\partial \mu}{\partial \Phi_j} = \frac{\partial \mu}{\partial H^2} \times 2.H \frac{\partial H}{\partial \Phi_j} .$$

Suivant la formule (11) les intégrants deviennent :

$$(12) \quad \iint \mu \nabla \beta_i \cdot \nabla \alpha_j + \iint 2 \frac{\partial \mu}{\partial H^2} \nabla \beta_i \cdot \nabla \Phi \nabla \alpha_j \cdot \nabla \Phi$$

Nous insistons sur ce type de problème car dans les équations thermiques ou hydrauliques des termes qui peuvent paraître similaires seront rencontrés ; mais nous montrerons que le traitement des non-linéarités des propriétés conduit à des expressions totalement différentes.

Remarque :

La prise en compte de l'anisotropie linéaire de certains matériaux ne pose pas de problème et nous avons facilement implanté leur traitement dans notre logiciel. Par contre, dans le cas de matériaux anisotropes non linéaires, l'élaboration des tenseurs μ ou ν est très délicate car la physique

ne nous fournit pas pour l'instant des modèles macroscopiques acceptables pour ces phénomènes. Une recherche devrait être développée dans ce sens.

A.III.2 LES MODÈLES SIMPLIFIÉS DYNAMIQUES BIDIMENSIONNELS

Les problèmes dynamiques imposent l'utilisation du potentiel vecteur.

Si on utilise la jauge de Coulomb $\nabla \cdot \mathbf{A} = 0$, on peut alors découpler les équations en $\mathbf{A}$ et ϕ et ne conserver que le système :

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times \nu \nabla \times \mathbf{A} = \mathbf{I}_{ex} \quad (13)$$

$$\nabla \cdot \mathbf{A} = 0$$

Dans une hypothèse bidimensionnelle le vecteur $\mathbf{A}$ n'a qu'une composante (perpendiculaire au plan d'étude) ce qui réduit donc considérablement le coût de résolution et assure $\nabla \cdot \mathbf{A} = 0$.

Cas linéaire :

Si la perméabilité magnétique ne dépend pas de B , l'utilisation des nombres complexes permet alors de traiter la dérivation temporelle en représentant $\sigma \frac{\partial \mathbf{A}}{\partial t}$ par $j\omega\sigma\bar{\mathbf{A}}$ où $\bar{\mathbf{A}}$ est un nombre complexe. L'équation (13) devient :

$$j\omega\sigma\bar{\mathbf{A}} + \nabla \times \nu \nabla \times \bar{\mathbf{A}} = \bar{\mathbf{I}}_{ex} \quad (14)$$

ω étant la pulsation des sources sinusoïdales.

Cette approche limite le domaine de l'étude aux phénomènes périodiques sinusoïdaux (ou décomposés en séries de Fourier).

L'effet de peau constitue une difficulté supplémentaire qui conduit souvent à des découpages très fins et donc très coûteux.

Cas non linéaire :

L'utilisation des nombres complexes n'est plus compatible avec le terme $\nabla x v(B) \nabla A$ ce qui impose alors l'utilisation de méthodes de traitement en pas à pas sur le temps.

Les méthodes de type Crank-Nicholson avec prédiction-correction conduisent à des formulations assez simples d'autant que le terme σ est généralement indépendant de A . Prenons pour exemple le traitement de l'équation :

$$(15) \quad \sigma \frac{DA_z}{Dt} + \nabla x v \nabla x A_z - I_{ex} = 0$$

par une méthode de Galerkinge.

Projetons le résidu sur la base des fonctions β_i :

$$(16) \quad \iint \sigma \beta_i \frac{DA_z}{Dt} + \iint \beta_i \cdot \nabla x (v \nabla x A_z) - \iint \beta_i I_{ex} = 0$$

$$\text{or} \quad \nabla \cdot (\beta_i x v \nabla x A_z) = (v \nabla x A_z) \cdot (\nabla x \beta_i) - \beta_i \cdot \nabla x (v \nabla x A_z)$$

donc (16) devient ;

$$(17) \quad \iint \sigma \beta_i \frac{DA_z}{Dt} + \iint (v \nabla x A_z) \cdot (\nabla x \beta_i) - \int \Gamma (\beta_i x (v \nabla x A_z)) \cdot n - \iint \beta_i I_{ex} = 0$$

dans lequel il faut détailler les formules des termes du 2ème ordre selon que v est un scalaire ou un tenseur :

$$(18) \quad \iint (v \nabla x A_z) \cdot (\nabla x \beta_i) =$$

$$\iint \left(v_{xx} x \frac{\partial A_z}{\partial y} - v_{xy} y \frac{\partial A_z}{\partial x} \right) \frac{\partial \beta_i}{\partial y} - \left(v_{yx} x \frac{\partial A_z}{\partial y} - v_{yy} y \frac{\partial A_z}{\partial x} \right) \frac{\partial \beta_i}{\partial x}$$

ou bien si v est un scalaire :

$$\iint v \left(\frac{\partial A_z}{\partial y} \frac{\partial \beta_i}{\partial y} + \frac{\partial A_z}{\partial x} \frac{\partial \beta_i}{\partial x} \right)$$

$$(19) \quad \int_{\Gamma} (\beta_i \mathbf{x} v \nabla \mathbf{x} A_z) \cdot n d\Gamma =$$

$$\int_{\Gamma} \beta_i \left(v_{yy} \frac{\partial A_z}{\partial x} - v_{yx} \frac{\partial A_z}{\partial y} \right) \cos(n, x) + \beta_i \left(v_{xx} \frac{\partial A_z}{\partial y} - v_{xy} \frac{\partial A_z}{\partial x} \right) \cos(n, y)$$

ou bien si v est un scalaire :

$$\int_{\Gamma} v \left(\beta_i \frac{\partial A_z}{\partial x} \cos(n, x) + \beta_i \frac{\partial A_z}{\partial y} \cos(n, y) \right)$$

Dans le cas où l'étude est conduite en présence de pièces en mouvement animées de la vitesse V nous devons détailler le terme $\frac{D}{Dt}$, ce qui conduit à :

$$(20) \quad \iint \sigma \beta_i \frac{DA_z}{Dt} = \iint \sigma \beta_i \frac{\partial A_z}{\partial t} + \iint \sigma \beta_i V \cdot \nabla A_z$$

Choisissons des fonctions α_j interpolant l'inconnue sur chaque élément :

$$(21) \quad A_z = \sum_k \alpha_k A_z^k$$

$$(22) \quad \frac{\partial A_z}{\partial t} = \sum_k \alpha_k \frac{\partial A_z^k}{\partial t}$$

l'équation (17) compte tenu de (18) (19) (20) (21) (22) peut être écrite :

$$(23) \quad [L] \frac{\partial A_z}{\partial t} + [M] A_z = K$$

avec

$$(24) \quad L_{ij} = \iint \sigma \beta_i \alpha_j$$

$$(25) \quad M_{ij} = \iint \sigma \beta_i V \cdot \nabla \alpha_j + \iint (v \nabla \mathbf{x} \alpha_j) \cdot (\nabla \mathbf{x} \beta_i)$$

$$(26) \quad K_i = \iint \beta_i I_{ex} + \int_{\Gamma} \beta_i x H \cdot n$$

On applique alors la méthode de Crank-Nicholson à l'équation (23) ce qui fournit :

$$(28) \quad \{[L] + \frac{\Delta t}{2} [M_n]\}_{A_{zn}} = \Delta t \left(\frac{K_{n-1} + K_n}{2} \right) + \{[L] - \frac{\Delta t}{2} [M_{n-1}]\}_{A_{zn-1}}$$

le deuxième membre de l'équation (28) n'est composé que de termes indépendants de l'inconnue A_{zn} , par contre les termes des matrices L et M en dépendent. Le système obtenu (28) est donc non linéaire. On applique alors la méthode ~~de~~ Newton-Raphson à ce système.

La matrice Jacobienne de (28) est :

$$(29) \quad [J] = \iint \alpha \beta_i \alpha_j + \frac{\Delta t}{2} \iint \alpha \beta_i \nabla \cdot \nabla \alpha_j + \frac{\Delta t}{2} \iint (\nabla \times \alpha_j) \cdot (\nabla \times \beta_i) + \frac{\Delta t}{2} \iint \frac{\partial v}{\partial A_j} (\nabla \times A_z) \cdot \nabla \times \beta_i$$

le dernier terme de (29) doit être développé en effet $\frac{\partial v}{\partial A_j}$ n'est pas directement ~~calculable~~ évaluable. On va donc procéder aux dérivations partielles intermédiaires classiques déjà présentées (on pose : $B = \nabla \times A$).

$$(30) \quad \iint \frac{\partial v}{\partial A_j} (\nabla \times A_z) \cdot \nabla \times \beta_i = \iint \frac{\partial v}{\partial (B \cdot B)} 2B \cdot \frac{\partial B}{\partial A_j} (\nabla \times A_z) \cdot (\nabla \times \beta_i) = 2 \iint \frac{\partial v}{\partial (B \cdot B)} \left(\frac{\partial A_z}{\partial x} \frac{\partial \alpha_j}{\partial x} + \frac{\partial A_z}{\partial y} \frac{\partial \alpha_j}{\partial y} \right) \left(\frac{\partial A_z}{\partial x} \frac{\partial \beta_i}{\partial x} + \frac{\partial A_z}{\partial y} \frac{\partial \beta_i}{\partial y} \right)$$

le terme $\frac{\partial v}{\partial (B \cdot B)}$ pouvant être évalué à partir de la courbe $v(B \cdot B)$ relevée au Laboratoire (dans le cas où v est un scalaire).

Il ne reste donc plus qu'à résoudre le système non linéaire (28) en appliquant la méthode de Newton-Raphson pour chaque pas de temps ΔT .

Dans le cas de fortes saturations, à chaque pas de

temps il faudra de nombreuses boucles pour résoudre le système non linéaire obtenu, ce qui peut justifier alors l'utilisation d'une prédétermination de la valeur de A_{nz} grâce à une méthode de prédiction. On peut notamment utiliser :

$$(31) \quad A_{nz}^{(P)} = A_{(n-1)z} + \Delta t_{(n-1)} \frac{\partial A_{(n-1)z}}{\partial t} + \frac{\Delta t_{(n-1)}^2}{2} \left[\frac{\frac{\partial A_{(n-1)z}}{\partial t} - \frac{\partial A_{(n-2)z}}{\partial t}}{\Delta t_{(n-2)}} \right]$$

qui nécessite de mémoriser $\frac{\partial A_{(n-1)z}}{\partial t}$, $\frac{\partial A_{(n-2)z}}{\partial t}$ et la valeur du pas $\Delta t_{(n-2)}$.

Dans le cas précédent l'effet de peau vient bien sûr toujours imposer un découpage très fin sur le bord des pièces conductrices.

A.III.3 INFLUENCE DU SYSTÈME DE COORDONNÉES

Jusqu'à présent nous avons présenté tous nos exemples dans le système Cartésien. Le système axisymétrique permet très souvent de simplifier considérablement des études :

- . soit en ramenant un problème tridimensionnel à un problème bidimensionnel (cas des systèmes où $A_r = A_z = 0$ et où seul A_θ est à prendre en compte,
- . soit en simplifiant des formulations notamment en tridimensionnel.

De très nombreux changements de variables ont été proposés, soit pour permettre l'utilisation d'intégrations analytiques, soit pour simplifier les formules obtenues [61], [65], [66]. Nous avons pu conduire diverses expériences de formulations avec le logiciel FLUX2D et nous avons montré que le changement de variable $A^* = rA_\theta$ conduisait d'une part à une grande simplification dans les équations et d'autre part à une très grande facilité d'écriture des conditions aux limites qui s'expriment en effet souvent en fonction du flux.

J.L. COULOMB a montré [69] que ce même changement de variable conduisait à une formulation beaucoup plus simple en tridimensionnel.

Dans tous les cas, quelles que soient les fonctions inconnues utilisées, les intégrants mis en oeuvre en coordonnées axisymétriques diffèrent toujours des termes obtenus en cartésien ce qui vient alourdir encore l'écriture d'un logiciel qui se voudrait général.

Exemple

Formulation d'un problème magnétodynamique linéaire en axe de symétrie mettant en oeuvre le changement de variable $\mathcal{A} = rA_\theta$.

L'application de la méthode de Galerkin conduit à l'équation :

$$(32) \quad \iint \sigma \beta_i \frac{\partial A}{\partial t} r dr dz + \iint \beta_i \nabla x \beta \nabla x A r dr dz = \iint \beta_i I r dr dz$$

dont la discrétisation directe conduit à l'expression :

$$(33) \quad \iint \sigma \beta_i \frac{\partial A}{\partial t} r dr dz + \iint v \left(\frac{\partial \beta_i}{\partial r} \frac{\partial A}{\partial r} \frac{\partial \beta_i}{\partial z} \frac{\partial A}{\partial z} \right) r dr dz + \iint v \frac{\beta_i A}{r} dr dz \\ + \iint v \left(\frac{\partial \beta_i}{\partial r} A + \beta_i \frac{\partial A}{\partial r} \right) dr dz = \iint \beta_i I r dr dz + \int_{\Gamma} v \beta_i \nabla(rA) \cdot n d\Gamma$$

dans laquelle on peut remarquer que les conditions aux limites ont été écrites en terme de $\nabla(rA)$ qui est colinéaire à B et non en terme de $\nabla(A)$ qui n'est pas colinéaire à B.

La formulation obtenue comporte malgré tout beaucoup de nouveaux termes par rapport à l'expression en coordonnées cartésiennes. Le changement de variable $\mathcal{A} = rA$ associé aux fonctions de projection $\beta = \frac{\alpha}{r}$ va considérablement simplifier l'expression qui devient :

$$(34) \quad \iint \sigma \alpha_i \frac{\partial \mathcal{A}}{\partial t} \frac{drdz}{r} + \iint v \left(\frac{\partial \alpha_i}{\partial r} \frac{\partial \mathcal{A}}{\partial r} + \frac{\partial \alpha_i}{\partial z} \frac{\partial \mathcal{A}}{\partial z} \right) \frac{drdz}{r} =$$

$$\iint \alpha_i I_r drdz + \int_{\Gamma} v \alpha_i \nabla \mathcal{A} \frac{drdz}{r}$$

dans laquelle on remarque que les conditions aux limites sont naturellement exprimées en fonction de $\mathcal{A} = rA$ donc colinéaires à B , car :

$$(35) \quad \int_{\Gamma} v \alpha_i \nabla \mathcal{A} \frac{drdz}{r} = \int_{\Gamma} \alpha_i H_T d\Gamma$$

En utilisant la notation des nombres complexes on obtient finalement :

$$(36) \quad \{ j\omega \iint \alpha_i \alpha_j^* \frac{drdz}{r} + \iint v \nabla \alpha_i \nabla \alpha_j^* \frac{drdz}{r} \} \bar{\mathcal{A}}$$

$$= \iint \alpha_i I_r drdz + \int_{\Gamma} \alpha_i H_{Tr} d\Gamma + j \{ \iint \alpha_i I_i drdz + \int_{\Gamma} \alpha_i H_{Ti} d\Gamma \}$$

où $\bar{I} = I_r + j I_i$

et $\bar{H}_T = H_{Tr} + j H_{Ti}$

L'équation (36) est un système linéaire symétrique que l'on peut alors résoudre par les méthodes classiques.

A.IV MODELE DES PHENOMENES THERMODYNAMIQUES .-

Nous supposerons que la capacité calorifique ρC est modélisée par un scalaire dépendant de la température, et que la conductibilité thermique est un tenseur symétrique non diagonal qui dépend également de la température.

Q est la densité de source de chaleur fonction du temps et de l'espace, supposée connue en tout point,
 V la vitesse des pièces ou des fluides en mouvement (fonction de l'espace supposée connue),
 ρ la masse volumique supposée constante,
 h le coefficient d'échange convectif caractérisant une paroi (fonction de la température),
 ϵ l'émissivité modélisant les propriétés de rayonnement des parois sur les limites.

Le bilan énergétique s'écrit (~~#~~produit tensoriel):

$$1. \quad \rho C \frac{DT}{Dt} + \nabla \cdot (-\bar{k} \nabla T) - q = 0$$

Une projection de Galerkin sur l'espace des fonctions β_i conduit au système d'équations :

$$2. \quad \iiint \beta_i \rho C \frac{DT}{Dt} - \iiint \beta_i \nabla \cdot \bar{k} \nabla T - \iiint \beta_i q = 0$$

On intègre par parties le terme de conduction qui, sur un volume Ω limité par une surface Γ , conduit à :

$$3. \quad \iiint_{\Omega} \beta_i \nabla \cdot (\bar{k} \nabla T) = \iiint_{\Omega} \nabla \cdot (\beta_i \bar{k} \nabla T) - \iiint_{\Omega} \nabla \beta_i (\bar{k} \nabla T)$$

qui par le théorème d'Ostrogradski nous donne :

$$4. \quad \iiint_{\Omega} \beta_i \nabla \cdot (\bar{k} \nabla T) = \iint_{\Gamma} \beta_i (\bar{k} \nabla T) \cdot n - \iiint_{\Omega} \nabla \beta_i (\bar{k} \nabla T)$$

où n est la normale extérieure de Γ .

Or les conditions aux limites sont données par ailleurs par divers types d'équations:

5. $(\bar{k} \nabla T) \cdot n = 0$ pas d'échange avec l'extérieur
6. $(\bar{k} \nabla T) \cdot n = -F_1(t)$ flux de chaleur échangé connu
7. $(\bar{k} \nabla T) \cdot n = -h(T - T_0)$ flux enlevé par convection sur une surface baignant dans un fluide de température ambiante connue T_0
8. $\bar{k} \nabla T \cdot n = -\epsilon \sigma (T^4 - T_0^4)$ flux émis par rayonnement (σ constante de Bolztzmann)

L'intégrale $\iint_{\Gamma} \beta_i (\bar{k} \nabla T) \cdot n$ peut donc s'exprimer à l'aide de l'une quelconque des formules 5,6,7,8 ou même leur combinaison (cas de 7 et 8). L'équation 2 devient alors :

$$+ \iiint_{\Omega} \beta_i \rho C \frac{DT}{Dt} + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k} \nabla T) - \iiint_{\Omega} \beta_i q$$

$$9. + \iint_{\Gamma} \beta_i h (T - T_0) + \iint_{\Gamma} \beta_i \epsilon \sigma (T^4 - T_0^4) = 0$$

que l'on peut réécrire en regroupant les valeurs constantes (à l'instant t) dans le deuxième membre, et en développant

$$\frac{D}{Dt} = \left(\frac{\partial}{\partial t} + v \cdot \nabla \right).$$

$$10. \iiint_{\Omega} \beta_i \rho C \frac{\partial T}{\partial t} + \iiint_{\Omega} \beta_i \rho C v \cdot \nabla T + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k} \nabla T) + \iint_{\Gamma} \beta_i h T + \iint_{\Gamma} \beta_i \epsilon \sigma T^4 = \iiint_{\Omega} \beta_i q + \iint_{\Gamma} \beta_i h T_0 + \iint_{\Gamma} \beta_i \epsilon \sigma T_0^4$$

En appliquant une interpolation polynomiale classique

$T = \sum \alpha_k T_k$, nous obtenons un système d'équations non linéaires en T^k dans lequel il faut de plus traiter le terme $\frac{\partial T}{\partial t}$.

Choisissons une méthode de différences finies implicite

du premier ordre

$$\left(\frac{\partial T}{\partial t}\right)_n = \frac{T_n - T_{n-1}}{\Delta t}$$

Ce choix nous fournit un système non linéaire dans lequel ρC , k , h , $\varepsilon \sigma$ sont des fonctions de la température.

$$\begin{aligned} 11. \quad & \iiint_{\Omega} \beta_i \rho C(T_n) T_n - \iiint_{\Omega} \beta_i \rho C(T_n) T_{n-1} + \iiint_{\Omega} \beta_i \rho C(T_n) V \cdot \nabla T_n \Delta t \\ & + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k}(T_n) \nabla T_n) \Delta t + \iint_{\Gamma} \beta_i h(T_n) T_n \Delta t + \iint_{\Gamma} \beta_i \varepsilon \sigma(T_n) T_n^4 \Delta t \\ & = \iiint_{\Omega} \beta_i q \Delta t + \iint_{\Gamma} \beta_i h(T_n) T_0 \Delta t + \iint_{\Gamma} \beta_i \varepsilon \sigma(T_n) T_0^4 \Delta t \end{aligned}$$

Appliquons la méthode de Newton Raphson à ce système et calculons pour cela sa matrice Jacobienne.

Les modèles des propriétés physiques des matériaux mis en oeuvre nous fournissent les valeurs :

$$\begin{aligned} & (\rho C)_n, \quad \left(\frac{\partial(\rho C)}{\partial T}\right)_n \\ & (\bar{k})_n, \quad \left(\frac{\partial \bar{k}}{\partial T}\right)_n \quad \text{tenseur} \\ & (h)_n, \quad \left(\frac{\partial h}{\partial T}\right)_n \\ & (\varepsilon \sigma)_n, \quad \left(\frac{\partial \varepsilon \sigma}{\partial T}\right)_n \end{aligned}$$

La dérivation partielle de l'équation 11 par rapport à T_1 nous fournit : (en remarquant que $\frac{\partial(x)}{\partial T_j} = \frac{\partial x}{\partial T} \frac{\partial T}{\partial T_j} = \frac{\partial x}{\partial T} \alpha_j$)

$$12. \quad \frac{\partial}{\partial T_{nj}} (\iiint_{\Omega} \beta_i \rho C(T_n) T_n) = \iiint_{\Omega} (\rho C)_n \beta_i \alpha_j + \iiint_{\Omega} T_n \left(\frac{\partial(\rho C)}{\partial T}\right)_n \beta_i \alpha_j$$

$$13. \quad \frac{\partial}{\partial T_{nj}} (\iiint_{\Omega} \beta_i \rho C(T_n) T_{n-1}) = \iiint_{\Omega} T_{n-1} \left(\frac{\partial(\rho C)}{\partial T}\right)_n \beta_i \alpha_j$$

$$14. \quad \frac{\partial}{\partial T_{nj}} (\iiint_{\Omega} \beta_i \rho C(T_n) V \cdot \nabla T_n) = \iiint_{\Omega} (\rho C)_n \beta_i V \cdot \nabla \alpha_j + \iiint_{\Omega} V \cdot \nabla T_n \left(\frac{\partial(\rho C)}{\partial T}\right)_n \beta_i$$

$$15. \quad \frac{\partial}{\partial T_{nj}} (\iiint_{\Omega} \nabla \beta_i \cdot (\bar{k}(T_n) \nabla T_n) = \iiint_{\Omega} \nabla \beta_i \cdot [(\bar{k})_n \nabla \alpha_j] \\ + \iiint_{\Omega} [(\frac{\partial \bar{k}}{\partial T})_n \nabla T_n] \cdot \nabla \beta_i \alpha_j$$

$$16. \quad \frac{\partial}{\partial T_{nj}} (\iint_{\Gamma} \beta_i h(T_n) T_n) = \iint_{\Gamma} (h)_n \beta_i \alpha_j + \iint_{\Gamma} T_n (\frac{\partial h}{\partial T})_n \beta_i \alpha_j$$

$$17. \quad \frac{\partial}{\partial T_{nj}} (\iint_{\Gamma} \beta_i \varepsilon \sigma(T_n) T_n^4) = \iint_{\Gamma} 4(\varepsilon \sigma)_n T_n^3 \beta_i \alpha_j + \iint_{\Gamma} T_n^4 (\frac{\partial \varepsilon \sigma}{\partial T})_n \beta_i \alpha_j$$

$$18. \quad \frac{\partial}{\partial T_{nj}} (\iint_{\Gamma} \beta_i h(T_n) T_o) = \iint_{\Gamma} T_o (\frac{\partial h}{\partial T})_n \beta_i \alpha_j$$

$$19. \quad \frac{\partial}{\partial T_{nj}} (\iint_{\Gamma} \beta_i \varepsilon \sigma(T_n) T_o^4) = \iint_{\Gamma} T_o^4 (\frac{\partial (\varepsilon \sigma)}{\partial T})_n \beta_i \alpha_j$$

La matrice Jacobienne est la somme de tous les termes matriciels ainsi calculés (12 à 19) dans lesquels toutes les valeurs de l'inconnue au pas n-1 sont connues et toutes les valeurs au pas de temps n sont ~~approximées~~ ^{choies} par les valeurs obtenues à l'itération (m-1) précédente du processus de Newton Raphson (nous ne les avons pas notées $T_{n,m-1}$ pour éviter de surcharger des expressions déjà bien fournies).

Pour toute autre méthode de traitement temporel (Crank Nicholson ou implicite ...) on pourrait de même obtenir les termes du système linéarisé correspondant.

Nous remarquerons seulement les termes caractéristiques des phénomènes thermiques :

a) un grand nombre de termes supplémentaires dans la matrice Jacobienne dus aux variations des termes de propriétés en fonction de la température.

b) Toutes les variations de propriétés s'expriment directement en fonction de l'inconnue principale (et non de son gradient ou de son rotationnel comme c'était le cas en magnétique).

c) La matrice Jacobienne ne peut jamais être symétrisée, en effet le terme de transport conduit à une valeur $\iiint \rho C \beta_i V \cdot \nabla \alpha_j$ intrinsèquement dissymétrique comme d'autre part le terme $\iiint \left(\left(\frac{\partial \bar{k}}{\partial T} \right)_n \nabla T_n \right) \cdot \nabla \beta_i \alpha_j$.

d) Une non linéarité très forte apparaît à cause du rayonnement ($\iiint 4 \epsilon \sigma T_n^3 \beta_i \alpha_j$) qui risque de perturber le conditionnement de la matrice Jacobienne si ce terme est localement très important.

La modélisation que nous venons de proposer pour le rayonnement n'est valable que pour une étude aux limites pour lesquelles l'on suppose aucun effet de réflexions. Dans le cas d'un rayonnement mutuel entre des parois internes au dispositif étudié il faudrait prendre en compte les effets de réflexions à l'aide de macro-éléments appropriés.

e) Si on ne met pas en oeuvre des techniques particulières (décentrage géométrique, intégration décalée ou éléments sur-paramétriques [105], [106], [107], le terme de transport ∇T traité par des éléments isoparamétriques imposera un découpage en accord avec le nombre de Peclet caractérisant l'étude $h < \frac{2k}{\rho C |U|}$.

En conclusion de cette brève étude nous pouvons remarquer que l'approche mathématique n'est pas très complexe mais conduit à un grand nombre de termes intégrants qu'il est assez fastidieux de manipuler. Nous montrerons que les concepts définis dans la deuxième partie viennent non seulement simplifier considérablement l'implantation informatique d'un tel problème mais fournissent également une notation plus claire beaucoup plus simple à manipuler.

A.V MODELISATION NUMERIQUE DES EQUATIONS DE NAVIER-STOKES

Pour un fluide compressible l'équation de Navier-Stokes s'écrit :

$$\rho \frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = \rho \mathbf{F} - \nabla P - \frac{2}{3} \nabla (\eta \nabla \cdot \mathbf{U}) + \nabla \cdot (\eta \nabla \mathbf{U}) + \nabla \cdot (\eta \dot{\mathbf{U}})$$

dans laquelle nous avons noté :

$\mathbf{U}$ la vitesse

	u
	v
	w

$\nabla \mathbf{U}$ un vecteur ayant pour composante

	∇u
	∇v
	∇w

$\mathbf{F}$ les densités de forces de volumes appliquées au fluide

$\dot{\mathbf{U}}$ le tenseur des dérivées premières de $\mathbf{U}$

$$\frac{\partial u}{\partial x} \quad \frac{\partial v}{\partial x} \quad \frac{\partial w}{\partial x}$$

$$\frac{\partial u}{\partial y} \quad \frac{\partial v}{\partial y} \quad \frac{\partial w}{\partial y}$$

$$\frac{\partial u}{\partial z} \quad \frac{\partial v}{\partial z} \quad \frac{\partial w}{\partial z}$$

Si le fluide est supposé incompressible la loi de conservation de masse, nous donne $\nabla \cdot \mathbf{U} = 0$ ce qui simplifie l'équation qui devient alors :

$$\rho \frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = \rho \mathbf{F} + \zeta$$

dans laquelle ζ est le tenseur $\nabla \cdot (\eta \nabla \mathbf{U}) + \nabla \cdot (\eta \dot{\mathbf{U}}) - \nabla P$

Pour conserver plus de généralité à notre étude nous

supposerons que la viscosité η n'est pas constante (ce qui peut être important dans la modélisation de procédés mettant en oeuvre des métaux en cours de solidification).

De nombreuses formulations de ces équations peuvent être obtenues. Il n'est pas possible de conduire ici une étude exhaustive des diverses approches qui s'offrent au chercheur, nous n'avons retenu que les plus souvent utilisées en modélisation numérique.

Nous présenterons tout d'abord brièvement les modèles statiques mettant en oeuvre des fonctions de potentiel ou de courant.

Puis nous conduirons l'étude complète du modèle direct en (u, v, w, p) . Ce travail théorique sera conduit dans l'espace tridimensionnel, le cas bidimensionnel (beaucoup plus réaliste sur le plan pratique) pourra en être déduit très simplement comme cas particulier.

Enfin, nous présenterons les méthodes bidimensionnelles mettant en oeuvre des fonctions de courant ψ et de vorticité ω .

A.V.1 FLUIDES NON VISQUEUX EN RÉGIME STATIONNAIRE

L'hypothèse $\eta=0$ simplifie considérablement l'équation qui se réduit alors à l'équation d'incompressibilité $\nabla \cdot \mathbf{U}=0$.

En choisissant une fonction de potentiel Φ , telle que $u = \frac{\partial \Phi}{\partial x}$ et $v = \frac{\partial \Phi}{\partial y}$, cette équation devient :

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0$$

De même pour les flots irrotationnels, $\nabla \times \mathbf{U}=0$ peut s'exprimer :

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \quad \text{en posant } U = \begin{vmatrix} \frac{\partial \psi}{\partial y} \\ -\frac{\partial \psi}{\partial x} \end{vmatrix}$$

ψ étant alors la fonction de courant.

Dans les systèmes de coordonnées à axe de symétrie, on obtient :

$$\frac{\partial^2 \Phi}{\partial z^2} + \frac{\partial^2 \Phi}{\partial r^2} + \frac{1}{r} \frac{\partial \Phi}{\partial r} = 0$$

pour la fonction de potentiel et pour la fonction de courant :

$$\nabla \times \mathbf{V} = -\frac{1}{r} \frac{\partial^2 \psi}{\partial z^2} - \frac{1}{r} \frac{\partial^2 \psi}{\partial r^2} + \frac{1}{r^2} \frac{\partial \psi}{\partial r} = 0$$

avec

$$\begin{vmatrix} V_r = -\frac{1}{r} \frac{\partial \psi}{\partial z} \\ V_z = \frac{1}{r} \frac{\partial \psi}{\partial r} \end{vmatrix}$$

Une projection de type Galerkin sur ces équations conduit classiquement aux expressions :

$$\iint \left(\frac{\partial N_i}{\partial r} \frac{\partial \psi}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \psi}{\partial z} \right) r dr dz + 2 \iint N_i \frac{\partial \psi}{\partial r} dr dz = \int_{\Gamma} N_i \nabla \psi \cdot \mathbf{n}$$

$$\iint \left(\frac{\partial N_i}{\partial r} \frac{\partial \Phi}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \Phi}{\partial z} \right) r dr dz = \iint N_i \nabla \Phi \cdot \mathbf{n}$$

La forme de ces équations étant extrêmement classique nous ne détaillerons pas leur traitement qui est donc rigoureusement similaire aux cas déjà rencontrés en magnétique et en thermique [19].

A.V.2 ÉTUDE TRIDIMENSIONNELLE DES ÉQUATIONS DE NAVIER-STOKES DANS LA FORMULATION U,V,W,P POUR UN FLUIDE INCOMPRESSIBLE

L'équation à résoudre s'écrit :

$$(1) \quad \rho \frac{DU}{Dt} = \rho F - \nabla P + \nabla \cdot (\eta \nabla U) + \nabla \cdot (\eta \dot{U})$$

Pour la composante x l'équation (1) devient ;

$$(2) \quad \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = \rho F_x$$

$$+ \frac{\partial}{\partial x} (2\eta \frac{\partial u}{\partial x} - P) + \frac{\partial}{\partial y} (\eta \frac{\partial u}{\partial y} + \eta \frac{\partial v}{\partial x}) + \frac{\partial}{\partial z} (\eta \frac{\partial u}{\partial z} + \eta \frac{\partial w}{\partial x})$$

Pour la composante y :

$$(3) \quad \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = \rho F_y$$

$$+ \frac{\partial}{\partial x} (\eta \frac{\partial v}{\partial x} + \eta \frac{\partial u}{\partial y}) + \frac{\partial}{\partial y} (2\eta \frac{\partial v}{\partial y} - P) + \frac{\partial}{\partial z} (\eta \frac{\partial v}{\partial z} + \eta \frac{\partial w}{\partial y})$$

Et pour la composante en z :

$$(4) \quad \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = \rho F_z$$

$$+ \frac{\partial}{\partial x} (\eta (\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z})) + \frac{\partial}{\partial y} (\eta (\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z})) + \frac{\partial}{\partial z} (2\eta \frac{\partial w}{\partial z} - P)$$

à ces équations il faut ajouter la condition d'incompressibilité :

$$(5) \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

Il faut donc résoudre simultanément les équations (2), (3), (4), (5). Soient N_i des fonctions de projections pour la méthode de Galerkin.

Après projection et intégration par partie des termes du 2ème ordre on obtient :

. Pour la 1ère composante :

$$(6) \quad \iiint \rho N_i \frac{\partial u}{\partial t} + \iiint \rho N_i \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right)$$

$$+ \iiint 2\eta \frac{\partial N_i}{\partial x} \frac{\partial u}{\partial x} - \iiint \frac{\partial N_i}{\partial x} P + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial u}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial v}{\partial x} +$$

$$+ \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial u}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial w}{\partial x} = \iiint N_i F_x + \iint_{\Gamma} N_i \tau_{x.n}$$

. Pour la 2ème composante :

$$(7) \quad \iiint \rho N_i \frac{\partial v}{\partial t} + \iiint \rho N_i \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) \\ + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial v}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial u}{\partial y} + \iiint 2\eta \frac{\partial N_i}{\partial y} \frac{\partial v}{\partial y} - \iiint \frac{\partial N_i}{\partial y} P \\ + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial v}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial w}{\partial y} = \iiint N_i F_y + \iint_{\Gamma} N_i \tau_y \cdot n$$

. Et pour la 3ème composante :

$$(8) \quad \iiint \rho N_i \frac{\partial w}{\partial t} + \iiint \rho N_i \left(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) \\ + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial w}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial u}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial w}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial v}{\partial z} \\ + \iiint 2\eta \frac{\partial N_i}{\partial z} \frac{\partial w}{\partial z} - \iiint \frac{\partial N_i}{\partial z} P = \iiint N_i F_z + \iint_{\Gamma} N_i \tau_z \cdot n$$

et l'équation d'incompressibilité devient avec les fonctions de projection M_i :

$$(9) \quad \iiint M_i \frac{\partial u}{\partial x} + \iiint M_i \frac{\partial v}{\partial y} + \iiint M_i \frac{\partial w}{\partial z} = 0$$

Dans les équations (6), (7), (8), τ_x, τ_y, τ_z représentent les lignes x, y, z du tenseur :

$$\begin{array}{ccc} 2\eta \frac{\partial u}{\partial x} - P & \left| \eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right| & \left| \eta \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right| \\ \hline \eta \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & \left| 2\eta \frac{\partial v}{\partial y} - P \right| & \left| \eta \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right| \\ \hline \eta \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) & \left| \eta \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \right| & \left| 2\eta \frac{\partial w}{\partial z} - P \right| \end{array}$$

Nous reviendrons en fin de chapitre sur la signification de ces conditions aux limites.

Nous avons donc le système (6), (7), (8), (9) à résoudre. Dans ce cas encore on peut écrire le système sous la forme générale déjà rencontrée.

$$(11) \quad [L] \frac{\partial \dot{\mathbf{I}}}{\partial t} + [M] \dot{\mathbf{I}} = \mathcal{R}$$

les termes de la matrice $[L]$ étant indépendants de $\dot{\mathbf{I}}$, on peut envisager toutes les formes de discrétisation du terme $[L] \frac{\partial \mathbf{I}}{\partial t}$ présentées au chapitre A.II.

On peut remarquer que la matrice $[L]$ est dépourvue de couplage entre inconnues et s'écrit :

$$\begin{bmatrix} \iiint \rho N_i \alpha_j & 0 & 0 & 0 \\ 0 & \iiint \rho N_i \alpha_j & 0 & 0 \\ 0 & 0 & \iiint \rho N_i \alpha_j & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \\ \frac{\partial w}{\partial t} \\ (P) \end{matrix}$$

Par contre, la matrice $[M]$ présente de nombreux termes de couplages qui vont conduire à une matrice Jacobienne presque pleine. Si nous choisissons un schéma de discrétisation temporelle de type Crank-Nicholson on doit résoudre :

$$(12) \quad \left[L + \frac{\Delta t}{2} M_n \right] \mathbf{I}_n = \Delta t \frac{K_n + K_{n-1}}{2} + \left[L - \frac{\Delta t}{2} M_{n-1} \right] \mathbf{I}_{n-1}$$

Nous considérons la matrice Jacobienne construite par blocs sous la forme :

$$\begin{matrix} J_{uu} & J_{uv} & J_{uw} & J_{up} \\ J_{vu} & J_{vv} & J_{vw} & J_{vp} \\ J_{wu} & J_{wv} & J_{ww} & J_{wp} \\ J_{pu} & J_{pv} & J_{pw} & J_{pp} \end{matrix}$$

On utilisera les notations suivantes :

$$(13) \quad [I] = [L] + \frac{\Delta t}{2} \{[M] + [DM]\}$$

$$L_{uu} = L_{vv} = L_{ww} = \iiint \rho N_i \alpha_j$$

$$L_{lm} = 0 \quad \text{si } l = m \text{ et } L_{pp} = 0$$

$$M_{uu} = \iiint \rho N_i \alpha_j \frac{\partial u}{\partial x} + \iiint 2\eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial \alpha_j}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial \alpha_j}{\partial z}$$

$$M_{uv} = \iiint \rho N_i \alpha_j \frac{\partial u}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial \alpha_j}{\partial x}$$

$$M_{uw} = \iiint \rho N_i \alpha_j \frac{\partial u}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial \alpha_j}{\partial x}$$

$$M_{up} = \iiint \frac{\partial N_i}{\partial x} \beta_j$$

$$M_{vu} = \iiint \rho N_i \alpha_j \frac{\partial v}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial y}$$

$$M_{vv} = \iiint \rho N_i \alpha_j \frac{\partial v}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial x} + \iiint 2\eta \frac{\partial N_i}{\partial y} \frac{\partial \alpha_j}{\partial y} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial \alpha_j}{\partial z}$$

$$M_{vw} = \iiint \rho N_i \alpha_j \frac{\partial v}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial z} \frac{\partial \alpha_j}{\partial y}$$

$$M_{vp} = \iiint \frac{\partial N_i}{\partial y} \beta_j$$

$$M_{wu} = \iiint \rho N_i \alpha_j \frac{\partial w}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial z}$$

$$M_{wv} = \iiint \rho N_i \alpha_j \frac{\partial w}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial z}$$

$$M_{ww} = \iiint \rho N_i \alpha_j \frac{\partial w}{\partial z} + \iiint \eta \frac{\partial N_i}{\partial x} \frac{\partial \alpha_j}{\partial x} + \iiint \eta \frac{\partial N_i}{\partial y} \frac{\partial \alpha_j}{\partial y} + \iiint 2\eta \frac{\partial N_i}{\partial z} \frac{\partial \alpha_j}{\partial z}$$

$$M_{wp} = \iiint \frac{\partial N_i}{\partial z} \beta_j \quad M_{pu} = \iiint M_i \frac{\partial \alpha_j}{\partial x}$$

$$M_{pv} = \iiint M_i \frac{\partial \alpha_j}{\partial y} \quad M_{pw} = \iiint M_i \frac{\partial \alpha_j}{\partial z} \quad M_{pp} = 0$$

et pour la matrice DM :

$$DM_{uu} = DM_{vv} = DM_{ww} = \iiint \rho N_i \frac{\partial \alpha_j}{\partial x} u + \iiint \rho N_i \frac{\partial \alpha_j}{\partial y} v + \iiint \rho N_i \frac{\partial \alpha_j}{\partial z} w$$

et

$$DM_{lm} = 0 \quad \text{si } l = m$$

et $DM_{pp} = 0$

La matrice Jacobienne J est donc ainsi définie totalement.

On peut remarquer qu'elle n'est pas symétrique et qu'elle comporte un très grand nombre d'intégrants.

L'intégration d'un problème hydraulique tridimensionnel sera donc très coûteuse, mais il nous a paru important de conduire l'analyse de ce problème complet, bien que nous n'envisagions pas pour l'instant de traiter des problèmes concrets avec cette méthode. En effet, cette étude nous fournit un exemple idéal de test pour évaluer les outils de description d'équations que nous avons conçus. Au chapitre suivant nous montrerons qu'il est tout-à-fait possible de décrire cette équation avec la méthode opérationnelle proposée.

Le traitement de l'équation (12) par la méthode de Newton-Raphson impose donc pour chaque pas de temps, la résolution d'un système non linéaire. Afin de réduire le coût de résolution la prédétermination d'une valeur U_n^p proche de la solution peut être effectuée à l'aide de l'algorithme de prédiction déjà rencontré.

- Etude des conditions aux limites dans cette formulation :

Nous avons montré que les conditions aux limites naturelles de ces équations s'expriment en fonction du tenseur τ .

$$\begin{array}{c|c|c}
 P - 2\eta \frac{\partial u}{\partial x} & -\eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & -\eta \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\
 \hline
 -\eta \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & P - 2\eta \frac{\partial v}{\partial y} & -\eta \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\
 \hline
 -\eta \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) & \eta \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) & P - 2\eta \frac{\partial w}{\partial z}
 \end{array}$$

On ne peut donc en aucun cas fixer la valeur de la pression comme condition aux limites. En effet, la pression est une variable implicite qui est assujettie à la contrainte $\nabla \cdot \mathbf{U} = 0$; si on fixe la pression en un point, $\nabla \cdot \mathbf{U}$ n'est plus annulée par l'effet de la pression, il faut alors ne plus tenir compte de cette équation dans le système (qui serait alors surabondant) ; mais si on fait cela l'équation d'incompressibilité n'est plus vérifiée et la conservation de la masse ne sera vérifiée ni localement ni globalement [109].

a) Cas des enceintes totalement closes

Les conditions sont fournies sous forme de dirichlet exclusivement. La pression n'intervenant que par son gradient il faut fixer sa valeur en un seul point. Il faut alors supprimer l'équation correspondante dans $\nabla \cdot \mathbf{U} = 0$.

Remarque ;

Les valeurs des vitesses fournies aux parois doivent bien entendu toujours vérifier $\iiint \nabla \cdot \mathbf{U} = 0$ c'est-à-dire $\iint_{\Gamma} \vec{\mathbf{U}} \cdot \vec{\mathbf{n}} \, d\Gamma = 0$.

b) Cas des sorties de chenal

Si le terme de transport domine ($Re \gg 1$) la condition $\tau = 0$ convient.

(*Re: nombre de Reynolds*)

Si on veut imposer un mouvement unidirectionnel au fluide il faut alors imposer $\tau_n=0$, $U_t=0$. Si on veut imposer la pression il faut alors l'approximer par la condition sur le tenseur τ_n car si $Re \gg 1$, $2\eta \frac{\partial u}{\partial n}$ est faible devant $-P$ mais ceci demeure une approximation [10].

c) Cas des lignes de symétrie

La condition $U_n=0$ s'impose et également $\tau_t=0$ sur la ligne de symétrie.

En conclusion de cette étude on peut remarquer que les difficultés rencontrées sont d'ordre très divers :

- C'est le terme $U \cdot \nabla U$ qui pose le problème le plus crucial en effet il est à la fois fortement non linéaire et non symétrique et provoque de plus des instabilités numériques.
- La non linéarité de η entraîne deux types de problèmes . Si η ne dépend que des coordonnées (par exemple par l'effet d'une dépendance avec la température) le problème n'est guère plus complexe que dans le cas où η est supposée constante .
Mais si η dépend de la vitesse directement (en fonction du module de celle-ci) alors le nombre de termes à calculer devient très grand.
- Les conditions aux limites influent au plus haut degré sur le choix d'une formulation. Nous montrerons les grandes difficultés que l'on peut rencontrer dans certains cas.
- Le terme de pression ∇P couplé à l'équation d'incompressibilité $\nabla \cdot U=0$ contribue à mal conditionner la matrice générale. Cela doit être atténué par le choix de fonctions de projection M de degré inférieur aux

fonctions d'interpolations.

- Enfin compte-tenu de tous les problèmes précédemment cités, des études tridimensionnelles sont très délicates et chaque fois que cela est possible une analyse bidimensionnelle permet de simplifier non seulement le nombre de termes à calculer mais également de choisir des formulations souvent plus adaptées.

A.V.3 FORMULATION EN TERMES DE VORTICITÉ ET DE COURANT EN COORDONNÉES CARTÉSIENNES

L'équation générale s'écrit :

$$(14) \quad \rho \frac{DU}{dt} = \rho F - \nabla P + \nabla \cdot (\eta \nabla U) + \nabla \cdot \eta \dot{U}$$

on développe :

$$(15) \quad \frac{DU}{dt} = \frac{\partial U}{\partial t} + U \cdot \nabla U = \frac{\partial U}{\partial t} + \frac{1}{2} \nabla (U \cdot U) - U \times (\nabla \times U)$$

d'où :

$$(16) \quad \frac{\partial U}{\partial t} + \frac{1}{2} \nabla (U \cdot U) - U \times (\nabla \times U) = \rho F - \nabla P + \nabla \cdot (\eta \nabla U) + \nabla \cdot \eta \dot{U}$$

Si on prend le rotationnel des deux membres tous les gradients vont disparaître, c'est-à-dire $\nabla \times (\nabla (U \cdot U)) = 0$ et $\nabla \times (\nabla P) = 0$; il reste alors, en notant $\Omega = \nabla \times U$:

$$(17) \quad \frac{\partial \Omega}{\partial t} - \nabla \times (U \times \Omega) = \rho \nabla \times F + \nabla \times (\nabla \cdot \eta \nabla U + \nabla \cdot \eta \dot{U})$$

On développe le terme $\nabla \times (U \times \Omega)$:

$$(18) \quad \nabla \times (U \times \Omega) = U (\nabla \cdot \Omega) - \Omega (\nabla \cdot U) - (U \cdot \nabla) \Omega + (\Omega \cdot \nabla) U$$

or $\nabla \cdot \Omega = 0$ car $\Omega = \nabla \times U$
 et $\nabla \cdot U = 0$ incompressibilité

d'où :

$$\nabla \mathbf{x} (\mathbf{U} \times \Omega) = -(\mathbf{U} \cdot \nabla) \Omega + (\Omega \cdot \nabla) \mathbf{U}$$

d'où l'équation globale :

$$(19) \quad \frac{\partial \Omega}{\partial t} + (\mathbf{U} \cdot \nabla) \Omega = (\Omega \cdot \nabla) \mathbf{U} + \rho \nabla \mathbf{x} \mathbf{F} + \nabla \mathbf{x} (\nabla \cdot \eta \nabla \mathbf{U} + \nabla \cdot \eta \dot{\mathbf{U}})$$

Si l'étude est conduite en deux dimensions ou en axe de symétrie Ω est perpendiculaire au plan de l'étude et $\Omega \cdot \nabla \mathbf{U} = 0$.

Première remarque

L'utilisation de la vorticité n'a d'intérêt que dans une étude en deux dimensions ou en axe de symétrie. Dans ce cas on obtient un système dont l'inconnue n'a qu'une composante (Ω_z) ou ($\Omega\theta$). En 3 dimensions cette représentation offre beaucoup moins d'intérêt par rapport à la représentation en ($\mathbf{U}, \mathbf{P}$).

L'équation se réduit donc à :

$$(20) \quad \rho \frac{\partial \Omega}{\partial t} + (\mathbf{U} \cdot \nabla) \Omega = \rho \nabla \mathbf{x} \mathbf{F} + \nabla \mathbf{x} (\nabla \cdot \eta \nabla \mathbf{U} + \nabla \cdot \eta \dot{\mathbf{U}})$$

nous sommes donc amenés à prendre le rotationnel du tenseur de contraintes. Si nous développons ces dérivations en tenant compte de la variation de η on peut montrer qu'il est impossible d'éliminer totalement la vitesse $\mathbf{U}$ en fonction de Ω dans le terme $\nabla \mathbf{x} (\nabla \cdot \eta \nabla \mathbf{U} + \nabla \cdot \eta \dot{\mathbf{U}})$.

Deuxième remarque

Le changement de variable $\Omega = \nabla \mathbf{x} \mathbf{U}$ n'est utilisable efficacement que dans le cas où $\eta = \text{constante}$. Car alors $\nabla \mathbf{x} (\nabla \cdot \eta \dot{\mathbf{U}}) = 0$ et $\nabla \cdot \eta \nabla \Omega = \eta \Delta \Omega$.

L'équation à résoudre s'écrit donc :

$$(21) \quad \rho \frac{\partial \Omega}{\partial t} + (U \cdot \nabla) \Omega - \eta \Delta \Omega = \rho \nabla \times F$$

il reste à déterminer la pression et la vitesse.

Pour déterminer la pression, prenons la divergence de l'équation de Navier-Stokes :

$$(22) \quad \nabla \cdot \rho \left(\frac{\partial U}{\partial t} + U \cdot \nabla U \right) + \nabla \cdot (\nabla P) - \nabla \cdot \eta \Delta U = \nabla \cdot (\rho F)$$

ce qui donne :

$$(23) \quad \rho \frac{\partial}{\partial t} (\nabla \cdot U) + \rho \nabla \cdot (U \cdot \nabla) U + \Delta P - \nabla \cdot \eta \Delta U = \nabla \cdot (\rho F)$$

d'où l'équation donnant la pression :

$$(24) \quad \Delta P + \rho \nabla \cdot (U \cdot \nabla) U = \nabla \cdot (\rho F)$$

On retrouve une équation de Poisson classique.

Il nous reste à établir l'expression de la vitesse U. On introduit la fonction de courant ψ telle que :

$$U = \begin{vmatrix} \frac{\partial \psi}{\partial y} \\ - \frac{\partial \psi}{\partial x} \end{vmatrix}$$

dans ce cas $\nabla \times U$ devient $-\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2}$ qui est égal à Ω .

Dans le cas où $\eta = \text{constante}$ on obtient les trois équations :

$$(25) \quad \rho \frac{\partial \Omega}{\partial t} + U \cdot \nabla \Omega - \eta \Delta \Omega = \rho \nabla \times F$$

$$(26) \quad \Delta P + \rho \nabla \cdot (U \cdot \nabla U) = \nabla \cdot (\rho F)$$

$$(27) \quad \Delta \psi = -\Omega$$

Dans ce système seules les équations (25) et (27) sont couplées alors que l'équation (26) peut être résolue séparément.

Les équations (25) et (27) une fois développées deviennent :

$$(28) \quad \rho \left(\frac{\partial \Omega}{\partial t} + \frac{\partial \psi}{\partial y} \frac{\partial \Omega}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial \Omega}{\partial y} \right) - \eta \left(\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \Omega}{\partial y^2} \right) = \rho \nabla \times F$$

$$(29) \quad \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\Omega$$

En utilisant une projection de type Galerkin sur l'ensemble des fonctions N_i et M_i on obtient :

$$(30) \quad \iint \rho \left(\frac{\partial \Omega}{\partial t} N_i + \frac{\partial \psi}{\partial y} \frac{\partial \Omega}{\partial x} N_i - \frac{\partial \psi}{\partial x} \frac{\partial \Omega}{\partial y} N_i \right) - \eta \frac{\partial^2 \Omega}{\partial x^2} N_i - \eta \frac{\partial^2 \Omega}{\partial y^2} N_i = \iint \rho \nabla \times F$$

$$(31) \quad \iint \frac{\partial^2 \psi}{\partial x^2} M_i + \frac{\partial^2 \psi}{\partial y^2} M_i = - \iint \Omega M_i$$

Considérons la première équation que l'on intègre par partie. Le terme Laplacien donne classiquement :

$$-\eta \iint \frac{\partial}{\partial x} \left(\frac{\partial \Omega}{\partial x} \right) N_i = +\eta \iint \frac{\partial \Omega}{\partial x} \frac{\partial N_i}{\partial x} - \eta \iint \frac{\partial}{\partial x} \left(\frac{\partial \Omega}{\partial x} N_i \right)$$

$$-\eta \iint \frac{\partial}{\partial y} \left(\frac{\partial \Omega}{\partial y} \right) N_i = +\eta \iint \frac{\partial \Omega}{\partial y} \frac{\partial N_i}{\partial y} - \eta \iint \frac{\partial}{\partial y} \left(\frac{\partial \Omega}{\partial y} N_i \right)$$

et

$$-\eta \iint \nabla \cdot \left(\frac{\partial \Omega}{\partial x} N_i \right) = - \int_{\Gamma} N_i \frac{\partial \Omega}{\partial n} d\Gamma$$

le terme de transport peut être également traité par partie en :

$$\iint \frac{\partial \psi}{\partial y} \frac{\partial \Omega}{\partial x} N_i = - \iint \frac{\partial \psi}{\partial y} \Omega \frac{\partial N_i}{\partial x} + \iint \frac{\partial \psi}{\partial y} \frac{\partial}{\partial x} (\Omega N_i)$$

et

$$\iint \frac{\partial \psi}{\partial x} \frac{\partial \Omega}{\partial y} N_i = + \iint \frac{\partial \psi}{\partial x} \Omega \frac{\partial N_i}{\partial y} - \iint \frac{\partial \psi}{\partial x} \frac{\partial}{\partial y} (\Omega N_i)$$

et

$$\iint \frac{\partial \psi}{\partial y} \frac{\partial}{\partial x} (\Omega N_i) - \iint \frac{\partial \psi}{\partial x} \frac{\partial}{\partial y} (\Omega N_i) = \int_{\Gamma} \frac{\partial \psi}{\partial y} \Omega N_i \cdot l_x - \int_{\Gamma} \frac{\partial \psi}{\partial x} \Omega N_i \cdot l_y$$

Considérons l'équation (31) que l'on peut également traiter par partie, on obtient :

$$\begin{aligned} (32) \quad \iint \frac{\partial^2 \psi}{\partial x^2} M_i + \frac{\partial^2 \psi}{\partial y^2} M_i &= - \iint \frac{\partial \psi}{\partial x} \frac{\partial M_i}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial M_i}{\partial y} + \iint \frac{\partial}{\partial x} (M_i \frac{\partial \psi}{\partial x}) + \frac{\partial}{\partial y} (M_i \frac{\partial \psi}{\partial y}) \\ &= - \iint \frac{\partial \psi}{\partial x} \frac{\partial M_i}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial M_i}{\partial y} + \int_{\Gamma} M_i \frac{\partial \psi}{\partial n} \cdot n = - \iint M_i \Omega \end{aligned}$$

On obtient alors le système en $\Omega \psi$:

$$\begin{aligned} (33) \quad \iint \rho N_i \frac{\partial \Omega}{\partial t} - \iint \Omega \frac{\partial \psi}{\partial y} \frac{\partial N_i}{\partial x} + \iint \Omega \frac{\partial \psi}{\partial x} \frac{\partial N_i}{\partial y} \\ + \iint \eta \frac{\partial \Omega}{\partial x} \frac{\partial N_i}{\partial x} + \iint \eta \frac{\partial \Omega}{\partial y} \frac{\partial N_i}{\partial y} = \iint \rho \nabla_x F N_i \\ + \int_{\Gamma} N_i \frac{\partial \Omega}{\partial n} d\Gamma - \int_{\Gamma} \frac{\partial \psi}{\partial y} \Omega N_i \cdot n_x + \int_{\Gamma} \frac{\partial \psi}{\partial x} \Omega N_i \cdot n_y \end{aligned}$$

n_x et n_y étant les cosinus directeurs de la normale

$$(34) \quad \iint \frac{\partial \psi}{\partial x} \frac{\partial M_i}{\partial x} + \iint \frac{\partial \psi}{\partial y} \frac{\partial M_i}{\partial y} = \iint N_i \Omega + \int_{\Gamma} M_i \frac{\partial \psi}{\partial n} \cdot n$$

que l'on peut mettre sous la forme :

$$(35) \quad [L] \frac{\partial \chi}{\partial t} + [M] \chi = R$$

avec $\chi = \begin{vmatrix} \Omega \\ \psi \end{vmatrix}$

Si on envisage un traitement par une méthode de type Crank-Nicholson, on retrouve l'équation classique :

$$(36) \quad [L' + \frac{\Delta t}{2} M_n] \chi_n = \Delta t \frac{K_n + K_{n-1}}{2} + [L - \frac{\Delta t}{2} M_{n-1}] \chi_{n-1}$$

dont on peut évaluer facilement la matrice Jacobienne :

$$J = L + \frac{\Delta t}{2} (M + DM)$$

Des difficultés supplémentaires proviennent des conditions aux limites. En effet :

. Les conditions aux limites de l'équation (34) ne posent aucun problème, car $\frac{\partial \psi}{\partial n}$ s'exprime directement en fonction de U donc les conditions naturelles sont généralement connues physiquement pour cette équation ($\psi = \text{constante}$ ou $\frac{\partial \psi}{\partial n} = 0$).

. Ce n'est pas le cas de l'équation (33) dont les conditions aux limites s'écrivent en fonction de $\frac{\partial \Omega}{\partial n}$ et Ω qui ne s'expriment pas simplement en fonction de U.

. Sur une limite dépourvue de paroi (entrée ou sortie de chenal) on peut fixer $\frac{\partial \psi}{\partial n} = 0$ et $\frac{\partial \Omega}{\partial n} = 0$.

Il en est autrement sur les parois fixes qui sont sources de vorticit  . Il faut alors calculer la valeur des int  grales $\int_{\Gamma} N_i \frac{\partial \Omega}{\partial n}$ et $\int_{\Gamma} \frac{\partial \psi}{\partial v} \Omega n_x - \int_{\Gamma} \frac{\partial \psi}{\partial x} \Omega n_y$ qui ne sont pas nulles (n_x et n_y   tant les cosinus directeurs de la normale).

- On peut utiliser la m  thode classique suivante [109] :    la paroi $\frac{\partial \psi}{\partial t}$ est nul donc l'  quation $\frac{\partial^2 \psi}{\partial t^2} + \frac{\partial^2 \psi}{\partial n^2} = -\Omega$ que l'on peut int  grer (dans lesquelles t et n repr  sentent la tangente et la normale).

- On peut   galement ^{cher} approximer la variation de Ω en fonction de la g  om  trie sur la paroi (variation lin  aire par exemple) et int  grer cette   quation analytiquement si la limite est simple (num  riquement si non). Cela conduit alors    une relation lin  aire suppl  mentaire entre les vorticit  s aux noeuds de la paroi et les fonctions de courant suppos  es connues.

- Une autre méthode consiste à discrétiser localement l'élément en un maillage plus fin pour évaluer plus exactement la valeur de $\frac{\partial \Omega}{\partial n}$ et calculer la contribution de l'intégrale correspondante. Cette technique bien adaptée aux différences finies est délicate à mettre en oeuvre avec les éléments finis.

Dans tous les cas on peut remarquer que, si la paroi présente un angle, la normale aura en ce point une double définition ce qui conduit à de nouvelles difficultés (généralement résolues en prenant la moyenne).

A.V.4 FORMULATION EN TERMES DE VORTICITÉ ET DE COURANT EN COORDONNÉES AXISYMÉTRIQUES

Nous ne détaillerons pas le traitement de cette formulation qui suit la même démarche que l'analyse en coordonnées cartésiennes. Nous ne fournissons ici que les résultats de cette discrétisation.

Rappel

Définition de la fonction de courant en axisymétrique

$$(37) \quad \left| \begin{array}{l} V_r = - \frac{1}{r} \frac{\partial \psi}{\partial z} \\ V_\theta = 0 \\ V_z = \frac{1}{r} \frac{\partial \psi}{\partial r} \end{array} \right.$$

$$(38) \quad \rho \frac{\partial \Omega}{\partial t} + \rho \left[-\frac{1}{r} \frac{\partial \psi}{\partial z} \frac{\partial \Omega}{\partial r} + \frac{1}{r} \frac{\partial \psi}{\partial r} \frac{\partial \Omega}{\partial z} \right] - \eta \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \Omega}{\partial r} \right) + \frac{\partial^2 \Omega}{\partial z^2} - \frac{\Omega}{r^2} \right] = \rho \nabla \times F$$

On projète sur le champ de polynomes N_i , on obtient alors :

$$\begin{aligned}
 (39) \quad & -\rho \iint N_i \frac{\partial \beta_j}{\partial z} \psi_j \frac{\partial \alpha_k}{\partial r} dr dz + \rho \iint N_i \frac{\partial \beta_j}{\partial r} \psi_j \frac{\partial \alpha_k}{\partial z} dr dz \\
 & + \eta \iint \left(\frac{\partial N_i}{\partial r} \frac{\partial \alpha_k}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \alpha_k}{\partial z} \right) r dr dz = \rho \iint N_i \nabla x F r dr dz \\
 & + \int_{\Gamma} N_i \frac{\partial \Omega}{\partial r} \cos(n, r) r dr dz + \int_{\Gamma} N_i \frac{\partial \Omega}{\partial z} \cos(n, z) r dr dz
 \end{aligned}$$

Ce dernier terme de conditions aux limites peut directement s'exprimer en terme de Ω à l'aide des diverses techniques, présentées dans l'étude en coordonnées cartésiennes, qui peuvent être facilement transposées.

D'autre part le terme :

$$(40) \quad \rho \iint \left(-N_i \frac{\partial \psi}{\partial z} \frac{\partial \alpha_j}{\partial r} + N_i \frac{\partial \psi}{\partial r} \frac{\partial \alpha_j}{\partial z} \right) dr dz$$

peut être intégré par partie et fournit alors les termes :

$$\begin{aligned}
 (41) \quad & \iint \frac{\partial}{\partial r} \left(-N_i \frac{\partial \psi \Omega}{\partial z} \right) + \frac{\partial \psi}{\partial z} \left(N_i \frac{\partial \psi \Omega}{\partial r} \right) dr dz \\
 & + \iint \left(\frac{\partial N_i}{\partial r} \frac{\partial \psi}{\partial z} - \frac{\partial N_i}{\partial z} \frac{\partial \psi}{\partial r} \right) \Omega dr dz
 \end{aligned}$$

et donc à un nouveau terme de conditions aux limites :

$$(42) \quad \int_{\Gamma} N_i \frac{\partial \psi}{\partial z} \Omega \cos(n, r) - \int_{\Gamma} N_i \frac{\partial \psi}{\partial r} \Omega \cos(n, z)$$

Enfin, l'équation de définition de Ω s'écrit $\nabla x V = \Omega$ ce qui conduit à :

$$\begin{aligned}
 (43) \quad & \iint \frac{\partial N_i}{\partial r} \frac{\partial \psi}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \psi}{\partial z} dr dz + \iint \frac{\partial N_i}{\partial r} \frac{\partial \psi}{\partial r} dr dz \\
 & = \iint N_i \Omega r dr dz + \int_{\Gamma} N_i \frac{\partial \psi}{\partial r} \cos(n, r) d\Gamma + \int_{\Gamma} N_i \frac{\partial \psi}{\partial z} \cos(n, z) d\Gamma
 \end{aligned}$$

dont les conditions aux limites peuvent s'écrire en fonction de V .

Quant à l'expression de l'équation donnant la pression elle devient alors :

$$\begin{aligned}
 (44) \quad & \iint N_i \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial P}{\partial r} \right) + \frac{\partial}{\partial z} \left(\frac{\partial P}{\partial z} \right) \right] r dr dz \\
 & + \iint N_i \left[\frac{1}{r} \frac{\partial}{\partial r} (r (V \cdot \nabla) V_r) + \frac{\partial}{\partial z} ((V \cdot \nabla) V_z) \right] r dr dz \\
 & = \iint \rho N_i \nabla \cdot F r dr dz
 \end{aligned}$$

ce qui fournit après intégration par partie :

$$\begin{aligned}
 (45) \quad & + \iint \left(\frac{\partial N_i}{\partial r} \frac{\partial P}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial P}{\partial z} \right) r dr dz + \iint \left(\frac{\partial N_i}{\partial r} (V \cdot \nabla) V_r + \frac{\partial N_i}{\partial z} (V \cdot \nabla) V_z \right) r dr dz \\
 & = \iint \rho N_i \nabla \cdot F r dr dz + \int_{\Gamma} N_i \left(\frac{\partial P}{\partial r} + (V \cdot \nabla) V_r \right) \cos(n, r) r d\Gamma \\
 & + \int_{\Gamma} N_i \left(\frac{\partial P}{\partial z} + (V \cdot \nabla) V_z \right) \cos(n, z) r d\Gamma
 \end{aligned}$$

CONCLUSION

Quelle que soit la formulation que l'on choisisse les équations de Navier-Stokes conduisent à un système non linéaire dont les inconnues ont plusieurs composantes (ω, ψ, P) ou bien (u, v, P en 2 dimensions) ou bien (u, v, w, p dans le cas tridimensionnel). Le couplage entre ces composantes est toujours très fort ce qui conduit à un système matriciel assez volumineux.

La non linéarité due essentiellement au terme de transport $(V \cdot \nabla) V$ est très importante et conduit à des problèmes numériques délicats dès que le nombre de Reynolds devient trop grand.

Les problèmes posés par la turbulence ne semblent pas pouvoir être pris en compte dans le détail avec les modèles que nous avons présentés.

Enfin, nous avons rappelé que les formulations en terme

de vorticité peuvent conduire à des problèmes délicats pour l'expression des conditions aux limites, c'est pourquoi il semble que les formulations en terme de u, v, w, p leur soient généralement préférées.

A.VI ETUDE DES EQUATIONS COUPLEES

Jusqu'à présent nous avons considéré isolément chacun des phénomènes que nous avons étudié. Or, dans de nombreux dispositifs, les grandeurs magnétiques, thermiques et hydrauliques interagissent étroitement.

Il est tout-à-fait envisageable de pouvoir étudier simultanément ces phénomènes en considérant une inconnue généralisée dont les composantes seront dans le cas bidimensionnel par exemple :

$\begin{array}{l|l} u & \text{vitesse des fluides} \\ v & \end{array}$

$\begin{array}{ll} p & \text{pression des fluides} \\ t & \text{température des corps} \\ A & \text{champ magnétique} \end{array}$

La méthode de Galerkin peut alors être appliquée au système obtenu et les opérateurs temporels peuvent également être discrétisés par des méthodes de type Crank-Nicholson. Les difficultés rencontrées sembleraient donc, au premier abord, n'être dues qu'au grand nombre de composantes de l'inconnue généralisée. En fait, le problème est beaucoup plus complexe et 3 écueils majeurs, doivent être évités.

A.VI.1 DOMAINES D'ÉTUDE INCOMPATIBLES

Si on veut pouvoir résoudre simultanément les équations des phénomènes couplés, il faut que chacune des équations prise

séparément soit définie sur le domaine complet de l'étude.

Cela n'est quelquefois pas le cas. Prenons pour exemple l'étude d'un problème de brassage électromagnétique de métaux fondus [99], [113], [135].

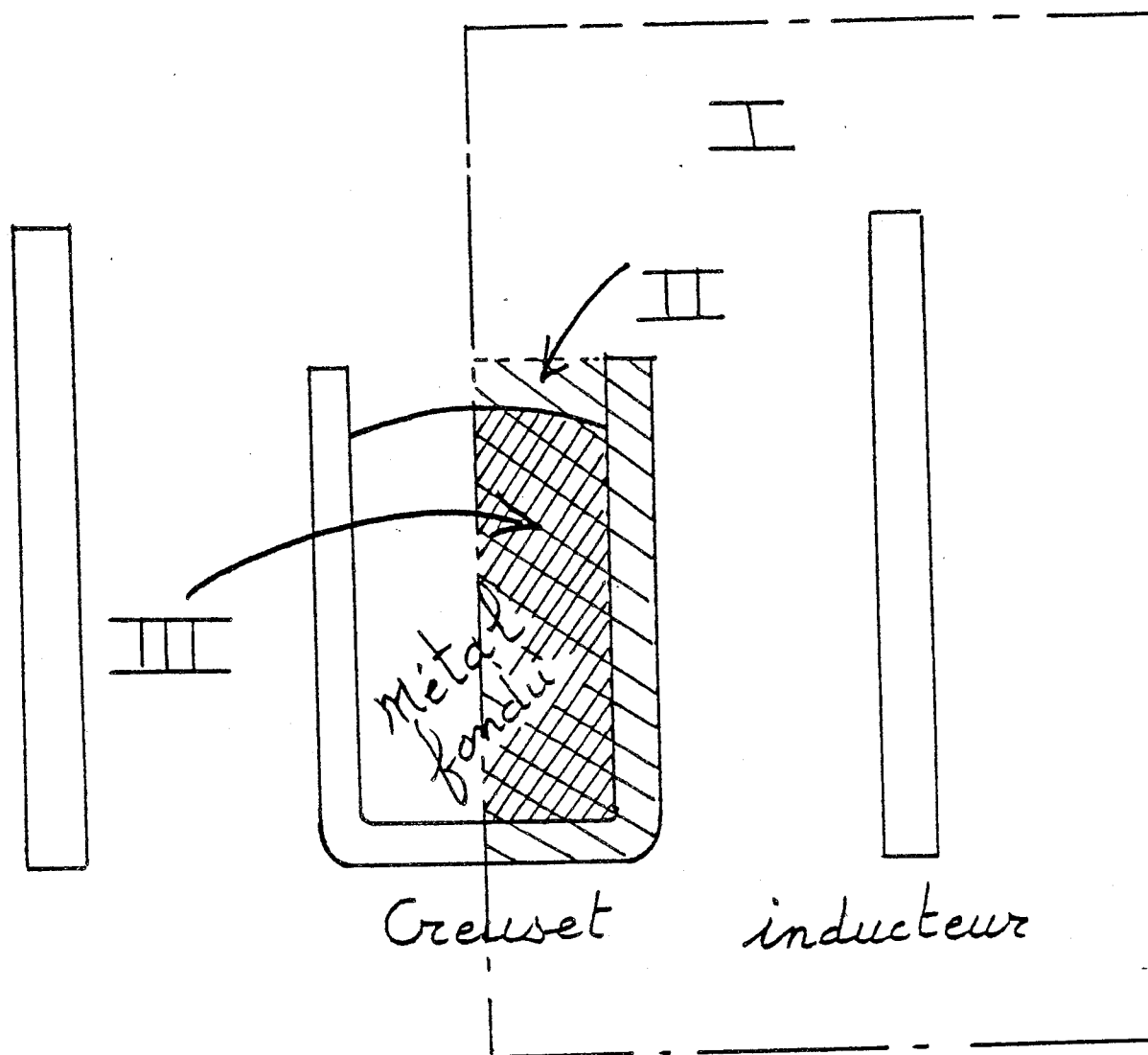


Figure 1 Schéma de principe d'un creuset

L'étude des phénomènes magnétiques doit être conduite dans le domaine I alors que les problèmes thermiques peuvent être réduits à la zone II et pour les problèmes hydrauliques, c'est la zone III seule qui doit être étudiée.

Il est donc impossible de résoudre simultanément les

problèmes couplés complets dans le domaine I puisque notamment l'équation hydraulique n'y est pas partout définie.

Le domaine d'étude implique donc quelquefois de découpler les équations en plusieurs systèmes séparés qui seront résolus tout à tour.

A.VI.2 TEMPS DE RÉPONSE

Dans l'étude des problèmes de propagation ou de diffusion les temps caractéristiques de chacun des phénomènes sont généralement très divers.

Si nous reprenons l'exemple concret déjà abordé au paragraphe précédent nous pouvons remarquer que :

- . les phénomènes magnétiques sont caractérisés par un temps de réponse très bref de l'ordre de quelques milli-secondes,
- . les phénomènes hydrauliques sont beaucoup plus lents et atteignent la seconde,
- . et les phénomènes thermiques sont pour leur part encore plus lents.

Si on envisage une résolution simultanée des trois équations magnéto-hydro-thermodynamiques, la discrétisation des termes en $\frac{\partial}{\partial t}$ va conduire à choisir un pas de temps nécessairement compatible avec le phénomène le plus rapide (le problème magnétique), ce qui va conduire à réaliser un très grand nombre de calculs inutiles pour les problèmes hydrothermiques et donc augmentera inutilement le budget de l'étude.

Cette remarque conduit donc également à aborder les problèmes de couplage à l'aide de systèmes distincts qui ne se

"synchronisent" que lorsque le temps de réponse caractéristique de chacun le nécessite.

A.VI.3 NATURE MATHÉMATIQUE DE LA VARIABLE

Enfin, on doit considérer également des cas où la nature de chacune des équations conduit à choisir des variables de type mathématique différent pour les composantes de l'inconnue.

Dans l'exemple que nous avons déjà cité, l'équation des phénomènes magnétiques est une équation linéaire (car les propriétés des matériaux mis en oeuvre v et σ sont des constantes). Les sources de courants d'excitation étant souvent sinusoïdales, l'utilisation des nombres complexes s'impose pour modéliser la variable A potentiel vecteur. Or, les équations thermiques et hydrauliques demeurent intrinsèquement non linéaires et imposent donc l'utilisation de nombres réels.

Malgré de nombreuses tentatives, il nous a semblé impossible de concevoir une méthode qui permette de traiter simultanément une inconnue globale dont certaines composantes seraient réelles et d'autres seraient de type complexe.

Là encore il faut envisager le traitement de chacune des équations séparément.

A.VI.4 TRAITEMENTS SIMULTANÉS ET TRAITEMENTS PARALLÈLES

Nous avons donc envisagé le traitement des couplages suivant deux méthodes très distinctes.

METHODE DU TRAITEMENT SIMULTANE

Dans cette méthode l'inconnue est alors considérée comme une variable vectorielle dont toutes les composantes doivent être de même type : REELLES ou COMPLEXES.

Les algorithmes de traitement des dérivées temporelles sont appliqués simultanément à toutes les composantes de l'inconnue, à savoir :

- . même pas de temps dans le cas des composantes de type réel,
- . même pulsation dans le cas des composantes de type complexe.

Chacune des composantes doit être définie sur tout le domaine de l'étude.

METHODE DE TRAITEMENTS PARALLELES

Dans cette approche, chaque phénomène est traité séparément, et conduit à un système autonome (qui peut bien entendu concerner une variable vectorielle).

Dans un tel cas, chaque système peut mettre en oeuvre une variable de type mathématique différent (Réel, Complexe) des constantes de temps ou des pulsations distinctes et divers domaines d'étude.

On peut alors envisager un traitement multiprocesseur ou monoprocesseur pour lequel il devra y avoir d'une part un partage des résultats de tous les systèmes (afin de pouvoir mettre à jour les termes de couplage) et d'autre part un programme superviseur assurant la synchronisation de l'ensemble des résolutions suivant une stratégie compatible avec les divers temps de réponse de chacun des systèmes physiques.

A.VII CONCLUSION

Tout au long de cette première partie nous avons montré sur de nombreux exemples comment la méthode des éléments finis pouvait modéliser divers phénomènes physiques représentés par leurs équations aux dérivées partielles.

Au terme de chacune des études que nous avons détaillées nous avons obtenu un système d'équations algébriques linéaire ou non linéaire à résoudre. Il serait idéal que l'essentiel du travail du chercheur consiste à établir les formules théoriques qui constituent ces équations discrétisées.

En fait d'une manière générale, une fois que l'on a obtenu les formules des équations discrétisées à résoudre, il faut se plonger dans une phase de programmation extrêmement longue et fastidieuse et qui demande de très bonnes connaissances en informatique.

Nous nous sommes donné comme but de concevoir un système informatique qui permette au chercheur de décrire conversationnellement les équations discrétisées à résoudre. Toute la technique informatique sera prise en charge par le système ainsi conçu.

Nous allons montrer dans la deuxième partie de ce mémoire comment une telle description est possible et surtout comment un programme général peut résoudre n'importe quel problème concret selon n'importe quels jeux d'équations descriptibles par notre système.

Afin que le fruit de notre travail ne soit pas seulement d'ordre théorique nous avons tenté de concevoir un système suffisamment élaboré pour pouvoir prendre en compte chacun des modèles numériques que nous avons donnés en exemple dans cette première partie.

Nous tiendrons compte donc de la grande variété de modèles de propriétés physiques de conditions aux limites et de sources pouvant intervenir dans des études concrètes. Les phénomènes transitoires devront pouvoir être traités par des modèles divers (implicites, explicites, Crank-Nicholson, prédiction-correction, nombres complexes). Les non-linéarités seront prises en compte avec une utilisation généralisée de la méthode Newton-Raphson. Les équations vectorielles pourront être décrites dans le cas de systèmes simultanément couplés.

Plusieurs systèmes d'équations seront descriptibles isolément et pourront donner lieu à des stratégies de résolutions complexes tenant compte des couplages que l'on pourra réaliser entre ces systèmes.

Enfin, nous essaierons de tenir compte de la grande diversité que peut présenter la méthode des éléments finis en laissant une grande souplesse dans le choix des polynômes (degré variable et simplexes quelconques) ou dans les méthodes de résolution de systèmes linéaires.

L'étude nous avons conduite est indépendante de la dimension de l'espace d'étude choisi. Nos programmes seront donc utilisables pour des problèmes en 1, 2 ou 3 dimensions, en coordonnées cartésiennes ou axisymétriques.

2ÈME PARTIE

MÉTHODOLOGIE GÉNÉRALE DE TRAITEMENT
D'UNE ÉQUATION PAR LA MÉTHODE DES ÉLÉMENTS FINIS :

FONDEMENTS CONCEPTUELS D'UN SYSTÈME GÉNÉRATEUR

* * *

INTRODUCTION .-

Beaucoup de travail a été fait pour fournir aux ingénieurs des outils de Conception Assistée par Ordinateur qui les libèrent presque totalement de l'emprise de la technique. Aucune connaissance poussée en informatique n'est désormais nécessaire pour utiliser les grands logiciels interactifs.

Par contre, le physicien concepteur d'un logiciel aux éléments finis ~~et~~ l'ingénieur de laboratoire qui veut adapter un programme à ses propres besoins, doivent se "plonger" dans cette informatique dévoreuse de temps: 10 % de physique, 90 % de programmation.

L'adaptation d'un logiciel à un nouveau domaine d'hypothèses peut paraître plus aisé. Mais bien souvent la documentation fait cruellement défaut, et l'expérience montre qu'il est parfois plus rapide de construire un nouvel outil plutôt que de tenter de modifier un logiciel mal documenté.

Notre but est d'alléger au maximum le travail de programmation d'un nouveau système et également de rendre ce dernier le plus évolutif possible afin de pouvoir l'adapter aux besoins de chaque ~~utilisateur~~, *concepteur*

Les techniques de résolution de problèmes et de construction assistée de système permettent à l'heure actuelle de surmonter ces difficultés. Nous allons étudier en détail les diverses étapes d'un tel processus en nous appuyant sur les expériences que nous avons pu réaliser.

Enfin, lorsque l'on peut disposer d'outils d'aide à la construction de système, il faut se poser la question: qu'elle peut être la structure globale la ~~plus~~ *mieux* adaptée aux besoins actuels de la CAO utilisant les éléments finis ? C'est sur cette

structure que nous concluons en montrant quelques exemples de réalisation.

Avant de détailler la méthodologie que nous préconisons il nous a semblé indispensable de situer notre travail parmi les diverses approches qui s'offraient à nous et de justifier les options que nous avons prises.

BI

DIVERSES APPROCHES POSSIBLES

Différentes écoles se sont employées à résoudre le problème posé par l'adaptation de la méthode des éléments finis au traitement d'équations multiples.

LES GRANDS LOGICIELS : tels NASTRAN, ASKA ces grands codes traitent de façon relativement monolithique diverses équations. De par leur taille, ils offrent un choix qui peut permettre au physicien de trouver parmi l'ensemble des possibilités de ces systèmes un modèle numérique qui corresponde à ses besoins. Mais dans le cas où aucun des modèles ne peut convenir, il est pratiquement impossible à l'utilisateur de pouvoir intervenir pour adapter ces grands codes à son problème. Seule l'équipe qui a conçu le logiciel peut entreprendre des modifications, et même dans ce cas, la complexité de ces programmes réduit énormément les possibilités d'extensions. La taille critique une fois atteinte, un tel système ne peut plus continuer à se développer sans poser de gros problèmes de fiabilité. Ces systèmes sont donc pratiquement figés mais ils ont une efficacité optimale dans leur domaine d'utilisation : les très gros problèmes.

LES SYSTEMES BIBLIOTHEQUES : le plus caractéristique de ces systèmes est certainement MODULEF. Par essence, ces systèmes se veulent évolutifs, car composés de modules que l'on peut programmer séparément et que l'on peut assembler en respectant des règles strictes. L'aspect documentaire devient dans ce cas primordial et la gestion d'un tel système est assez délicate. Mais les progrès sont assez considérables ; en effet, le concepteur

d'un nouveau système va trouver tout programmés un grand nombre des algorithmes dont il a besoin. Les éléments manquants devront être réalisés au format standard. L'aide est donc appréciable au point de vue de la programmation mais elle est beaucoup moins efficace au niveau de l'architecture de système. Enfin, il peut être particulièrement difficile de se plonger dans toute la documentation nécessaire afin de pouvoir dégager les pièces utiles à une nouvelle création. L'utilisation de tels systèmes n'est donc pas à la portée des chercheurs qui ne possèdent pas d'une part une solide formation informatique et d'autre part une très bonne connaissance du contenu des bibliothèques.

LES SYSTEMES UTILISANT DES MACHINES ABSTRAITES

Le principe de tels systèmes réside dans la création d'une machine virtuelle adaptée au traitement des informations très particularisées qui sont mises en oeuvre dans la méthode des éléments finis. Des macro-instructions permettent de traiter l'information, ~~des~~ ^{programmes} ~~utilitaires~~ système gèrent automatiquement les données et les entrées-sorties.

La programmation d'un programme d'éléments finis doit alors être effectuée dans un langage de haut niveau géré par la machine virtuelle. Un interpréteur traite alors l'exécution du système ainsi construit. Cette approche, caractéristique des travaux de Schrem [175] [176], est dans la droite ligne des systèmes de CAO bâtis sur le modèle de ICES dans le domaine [173] - [174] du génie civil. Dans une recherche antérieure nous avons été amenés à travailler sur de tels systèmes (ESPACE, CASCADE) qui ne mettaient pas en oeuvre la technique des éléments finis mais dont la base reposait sur le principe d'une machine abstraite [158], [164], [167]. Le travail nous a permis de mettre en évidence les avantages mais également les inconvénients que peut présenter ce type d'approche.

L'aide méthodologique est considérable car la structure du système devient beaucoup plus simple grâce à la machine

abstraite qui accepte un langage de très haut niveau. Toute la partie algorithmique s'en trouve donc simplifiée. Sur le plan de l'écriture de système cette méthode est certainement la plus performante.

Les problèmes concrets se rencontrent généralement en trois points cruciaux.

- Si les langages mis en oeuvre sont idéaux pour l'algorithme, ils sont par contre généralement assez lourds et surtout assez peu performants pour le calcul numérique (compilations puis interprétation sur la machine abstraite). Or malgré les progrès technologiques la méthode des éléments finis met encore en oeuvre des problèmes dont la taille peut entraîner des coûts prohibitifs. On est alors tenté de faire réaliser les calculs par des modules FORTRAN qui posent à leur tour de délicats problèmes lorsque l'on manipule des données de taille variable .
- La construction d'un système aux éléments finis avec ce type d'approche demande à l'auteur un degré d'abstraction (et de formation) très poussé ce qui l'éloigne encore plus de la physique et des mathématiques appliquées.
- Enfin contrairement aux systèmes bibliothèques, la programmation demeure totalement à la charge du concepteur et cela peut être fastidieux. En effet, la méthode des éléments finis met en oeuvre des traitements immuables inhérents à cette technique, dont les degrés de liberté sont bien délimités et pour lesquels une aide à la programmation pourrait être apportée au concepteur.

En conclusion des ces diverses approches, nous pouvons constater que la méthode idéale n'existe pas et il n'est pas de notre prétention de vouloir la créer. Ce que nous allons proposer

se veut pragmatique, gardant présent à l'esprit le but que nous nous sommes fixés au départ de cette recherche : faciliter la tâche du concepteur d'un système d'éléments finis afin qu'il puisse consacrer la plus grande partie de son énergie à la modélisation physico-mathématique de son problème.

Nous avons donc dès le début de notre travail pris des options bien déterminées (que nous avons justifiées dans des publications prospectives [158] [168]).

PRINCIPE D'UN SYSTEME GENERATEUR

L'aide que nous avons choisie d'apporter au concepteur d'un système va être concrétisée tout d'abord par un programme général dénommé "générateur" qui va permettre la description conversationnelle assistée des équations à résoudre et des méthodes utilisées pour les résoudre.

Nous appellerons système objet tout système ainsi décrit à l'aide de notre générateur. Chaque système objet sera autonome et pourra être mis en oeuvre comme un système aux éléments finis classique sans qu'aucune connaissance du générateur ne soit demandée à son utilisateur.

On pourra donc sur chaque système objet :

1. découper un sujet en éléments finis
2. affecter propriétés et conditions aux limites
3. faire résoudre le problème réel ainsi décrit
4. présenter les résultats.

La partie la plus complexe à décrire est bien sûr la résolution de problème qui dépend étroitement des équations et des méthodes choisies. Le découpage en éléments finis et la définition des propriétés et conditions aux limites peuvent être généralisées à partir des informations fournies pour décrire équations et méthodes. Les étapes 1,2 et 4 pourront être

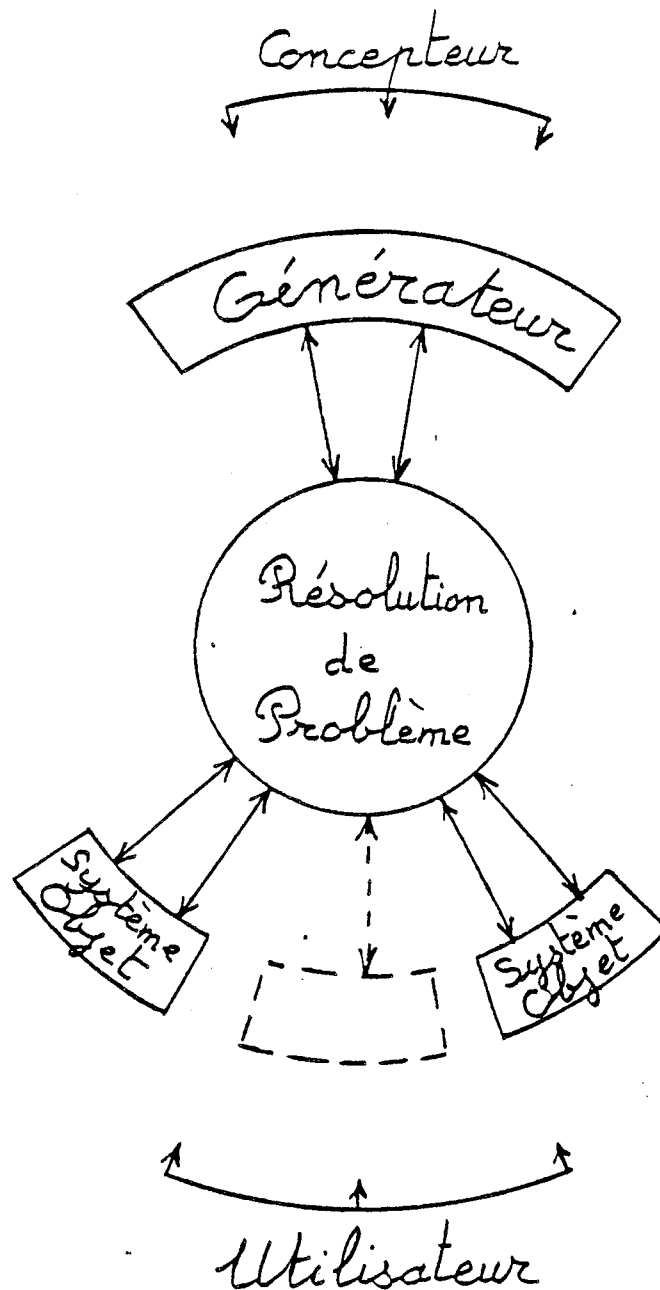


Figure 2 Générateur et système objet.

traitées par des programmes généraux programmés par nos soins et s'adaptant automatiquement à la description fournie pour l'étape 3 qui constituera donc l'essentiel du travail d'un concepteur de système.

Nous n'avons pas retenu le principe de la machine abstraite pour l'ensemble de notre approche, en effet nous montrons que le traitement de l'informations/ éléments finis peut être dissocié en deux niveaux hiérarchiques bien distincts grâce

aux concepts de stratégie de résolution et problèmes potentiels élémentaires que nous définirons au chapitre suivant. L'idée de base de cette approche est de séparer l'algorithmique classique que l'on rencontre dans tout problème d'une part, de la technique très particulière aux éléments finis d'autre part.

Nous exposerons ensuite comment la description de problème à l'aide d'un graphe d'enchaînement d'opérateurs permet de définir à la fois les équations à résoudre et la structure du système éléments finis qui les traitera. Grâce à cette approche l'utilisation de bibliothèque d'opérateurs assouplit considérablement la tâche du concepteur d'un système objet.

L'aide apportée sur le plan de la programmation est donc presque totale puisque le concepteur n'aura généralement qu'à choisir ses opérateurs dans les bibliothèques conçues à cet effet. Sur le plan de l'algorithmique, il recevra une aide permanente puisque l'outil de description de problème le guidera pas à pas suivant une méthodologie rigoureuse contrôlée par le générateur.

On peut donc considérer que notre système repose essentiellement sur une méthode de description/résolution de problème qui met en oeuvre des bibliothèques d'opérateurs.

Néanmoins, nous montrerons que le traitement de sujets complexes (non linéaires, dépendants du temps, systèmes couplés, etc...) ne peut pas être ramené à la résolution d'un seul type de problème mais à l'enchaînement de plusieurs traitements successifs. C'est dans ce domaine (la stratégie de résolution) que l'approche mettant en oeuvre une machine abstraite peut porter des fruits de façon performante.

Le langage de programmation choisi, le FORTRAN, permet de traiter avec une grande efficacité d'exécution tous les calculs nécessaires ; ce qui garantit des temps de réponse acceptables pour les systèmes objets générés. Seuls des modules de traitement numériques (les opérateurs) devront être programmés dans ce langage. En dehors de cette dernière tâche tout le

travail d'un concepteur de système objet sera géré par le générateur de façon totalement conversationnelle et assistée.

Le point de vue de l'utilisateur d'un système objet ne doit pas non plus être laissé de côté car il ne faut pas oublier que le but final n'est pas la construction d'un système mais son utilisation ; et cela bien sûr par une autre personne que le concepteur lui-même. Dans ce domaine nous avons bénéficié de l'expérience que nous a apportée l'utilisation industrielle du logiciel FLUX [58], [49], [45]. Les outils que nous avons implantés ont été réalisés sur ces bases (en apportant quelques améliorations par rapport à la version diffusée aujourd'hui). Cela garantit une qualité d'interactivité, que nous espérons suffisante, pour chacun des systèmes objets construits par notre générateur.

Enfin, nous montrerons comment la documentation interne et externe d'un tel système a été réalisée afin que la plupart des documents nécessaires puissent être élaborés automatiquement, facilitant ainsi la mise au point et la maintenance selon des normes industrielles.

B.II LES CONCEPTS DE BASE .-

Si nous considérons les diverses discrétisations que nous avons présentées dans la première partie, nous pouvons mettre en évidence deux niveaux bien distincts dans le traitement d'une équation par la méthode des éléments finis.

B.II.1 STRATÉGIES ET PROBLÈMES POTENTIELS

La stratégie de résolution : elle peut être concrétisée :

- . par l'algorithme de Newton-Raphson, de Newton-Kantorovitch, etc..., pour un système non linéaire ;
- . par les méthodes classiques : implicite, Crank-Nicholson, Prédiction correction, etc..., pour le traitement des variables temporelles ;
- . par des bouclages entre équations couplées, traitées par des systèmes d'équations distinctes ou résolues par des traitements simultanés.

Les problèmes élémentaires : ils doivent être résolus successivement au cours de la mise en oeuvre d'une stratégie.

Par exemple la méthode Newton Raphson va demander le traitement de deux types de problèmes d'éléments finis.

- résolution de l'équation linéarisée par une méthode directe pour obtenir les valeurs initiales approchées des inconnues,

- résolutions itérées de l'équation Jacobienne du système non linéaire.

La méthode prédiction correction nécessite le traitement de 3 types de problèmes :

- utilisation d'une discrétisation implicite pour démarrer le processus (elle-même composée des deux types de problèmes utilisés par la méthode de Newton-Raphson).
- résolution itérée de la discrétisation de Crank-Nicholson non linéaire.

Nous constatons que le traitement d'une même équation peut donc être effectuée par diverses stratégies et qu'au sein d'une même stratégie plusieurs types de problèmes élémentaires doivent être successivement résolus.

D'autre part, la stratégie de résolution met en oeuvre des algorithmes généraux qui sont totalement indépendants de la méthode des éléments finis, alors que c'est la construction de chaque problème élémentaire qui utilise cette technique.

C'est pourquoi nous traiterons de façon totalement différente ces deux niveaux. Nous allons tout d'abord analyser les méthodes de résolution des divers problèmes élémentaires qui peuvent être obtenus à partir d'équations quelconques puis nous reviendrons sur le traitement des stratégies à la fin de ce volume.

PROBLEME REEL ET PROBLEME POTENTIEL ELEMENTAIRE

Un problème élémentaire à résoudre dépend d'une part, de l'équation caractérisant le phénomène physique étudié mais bien entendu aussi de l'objet étudié. Comme nous l'avons vu dans le principe d'un générateur permettant la construction d'un "système objet", ce dernier pourra servir à la résolution d'une infinité de problèmes concrets (dans lesquels les formes et les matériaux pourront varier).

Au sein d'un même système objet, l'ensemble de tous les problèmes réels élémentaires, que l'on peut construire en décrivant divers sujets d'étude, forme une classe d'équivalence.

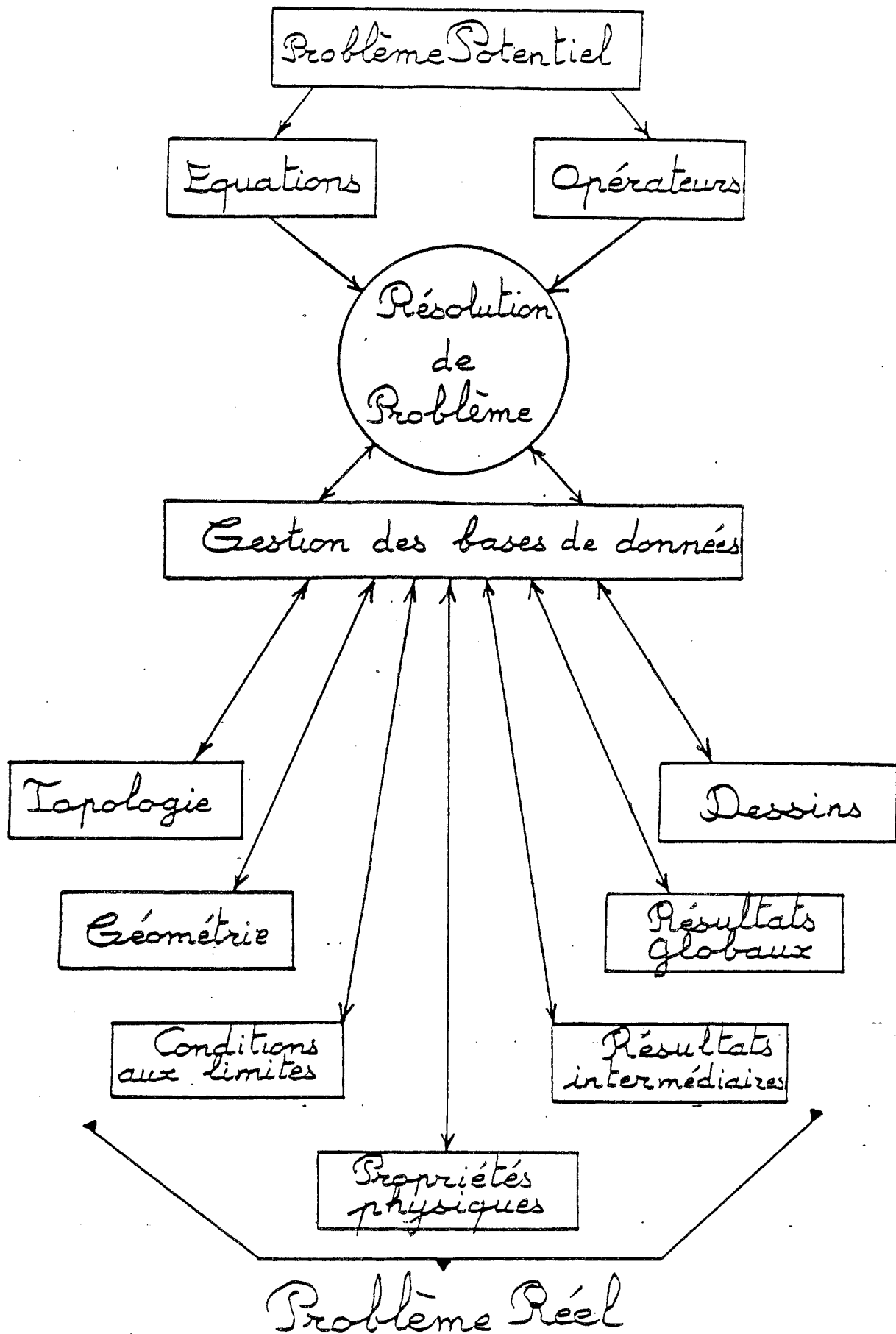


Figure 3 Relation entre problème potentiel et problèmes réels.

C'est cette classe d'équivalence que nous dénommons PROBLEME POTENTIEL et qui nous permet de faire abstraction des découpages concrets traités et des matériaux réels les constituant, pour ne prendre en compte que les aspects de méthode de résolution de problème ~~et~~ de description d'équation ~~\$~~

B.II.2 NOTION D'OPÉRATEUR ÉLÉMENTAIRE

Détaillons les diverses étapes qui ~~constituent~~ ^{composent} la résolution d'un problème élémentaire : il est toujours constitué par la construction et la résolution d'un système linéaire (principe intrinsèque de la discrétisation). Le système linéaire est obtenu par assemblage de la sous-matrice et du sous-vecteur second ~~member~~ correspondant à chacun des éléments.

Chaque sous-matrice ~~et~~ ^{ou} sous-vecteur est donné par la formule de l'équation discrétisée, à laquelle on a appliqué une méthode de linéarisation.

Citons quelques exemples que nous avons déjà détaillés dans la première partie :

Cas du problème de diffusion thermique non linéaire (cf 1ère partie ch. III) ; après une discrétisation temporelle implicite l'équation à résoudre s'écrit :

$$\begin{aligned} & \iiint_{\Omega} \beta_i \rho C_n T_n - \iiint_{\Omega} \beta_i \rho C_n T_{n-1} + \iiint_{\Omega} \beta_i \rho C_n V \cdot \nabla T_n \Delta t + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{K}_n \nabla T_n) \Delta t \\ & + \iint_{\Gamma} \beta_i h_n T_n \Delta t + \iint_{\Gamma} \beta_i \varepsilon \sigma_n T_n^3 T_n \Delta t = \iiint_{\Omega} \beta_i q_n \Delta t + \iint_{\Gamma} \beta_i h_n T_{an} \Delta t + \iint_{\Gamma} \beta_i \varepsilon \sigma_n T_{an}^4 \Delta t \end{aligned}$$

dont nous avons montré que la matrice Jacobienne était composée des termes matriciels suivants :

$$\begin{aligned}
 & \iiint_{\Omega} \rho C_n \beta_i \alpha_j + \iiint_{\Omega} T_n \left(\frac{\partial(\rho C)}{\partial T} \right)_n \beta_i \alpha_j - \iiint_{\Omega} T_{n-1} \left(\frac{\partial(\rho C)}{\partial T} \right)_n \beta_i \alpha_j + \iiint_{\Omega} (\rho C)_n \beta_i V \cdot \nabla \alpha_j \\
 & + \iiint_{\Omega} V \cdot \nabla T_n \left(\frac{\partial(\rho C)}{\partial T} \right)_n \beta_i \alpha_j + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k} \nabla \alpha_j) + \iiint_{\Omega} \left(\frac{\partial k}{\partial T} \right)_n \nabla T_n \cdot \nabla \beta_i \alpha_j + \iint_{\Gamma} h \beta_i \alpha_j \\
 & + \iint_{\Gamma} T_n \frac{\partial h}{\partial T} \beta_i \alpha_j + 4 \iint_{\Gamma} \varepsilon \sigma_n T_n^3 \beta_i \alpha_j + \iint_{\Gamma} T_n^4 \left(\frac{\partial \varepsilon \sigma}{\partial T} \right)_n \beta_i \alpha_j - \iint_{\Gamma} T_{n-1}^4 \left(\frac{\partial h}{\partial T} \right)_n \beta_i \alpha_j - \iint_{\Gamma} T_{n-1}^4 \left(\frac{\partial \varepsilon \sigma}{\partial T} \right)_n \beta_i \alpha_j
 \end{aligned}$$

Cas de l'équation de Naviers-Stokes en 2 dimensions à viscosité constante (simplification de l'étude générale conduite au chapitre AV).

$$\begin{aligned}
 & \cdot \iint \rho N_i u_n - \iint \rho N_i u_{n-1} + \Delta t \iint N_i \rho U_n \cdot \nabla u_n + \Delta t \eta \iint \frac{\partial N_i}{\partial x} \frac{\partial u_n}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial u_n}{\partial y} \\
 & \quad - \Delta t \iint \frac{\partial N_i}{\partial x} P = \Delta t \iint N_i F_x + \Delta t \int_{\Gamma} N_i \tau_x \cdot n \\
 & \cdot \iint \rho N_i v_n - \iint \rho N_i v_{n-1} + \Delta t \iint N_i \rho U_n \cdot \nabla v_n + \Delta t \eta \iint \frac{\partial N_i}{\partial x} \frac{\partial v_n}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial v_n}{\partial y} \\
 & \quad - \Delta t \iint \frac{\partial N_i}{\partial y} P = \Delta t \iint N_i F_y + \Delta t \int_{\Gamma} N_i \tau_y \cdot n \\
 & \cdot \iint M_i \frac{\partial u_n}{\partial x} + \iint M_i \frac{\partial v_n}{\partial y} = 0
 \end{aligned}$$

Dans ce cas nous sommes en présence d'un système d'équations intrinsèquement non linéaires de par la présence des termes en $U \cdot \nabla u$ et $U \cdot \nabla v$.

Parmi les équations, caractéristiques des phénomènes classiques de conduction convection diffusion rayonnement, on peut remarquer que les formules obtenues dépendent d'une part de l'équation de départ à résoudre, d'autre part de la méthode utilisée pour séparer les variables d'espace et de temps et pour linéariser les problèmes obtenus.

Or ces méthodes mettent en oeuvre des opérateurs très simples :

- . combinaisons linéaires de matrices pour la discrétisation des dérivées temporelles par la méthode des différences finies,
- . dérivation des systèmes matriciels non linéaires pour les méthodes de traitement des non-linéarités.

Si nous généralisons les équations précédemment étudiées nous pouvons obtenir la formulation suivante :

$$[L] \left\{ \frac{\partial I}{\partial t} \right\} + [M] \{I\} = \{KS\} + \{KV\}$$

dans laquelle :

- . $[L]$ et $[M]$ sont des matrices dont les termes peuvent éventuellement dépendre des inconnues,
- . $\{KS\}$ le vecteur des sources (ou de leurs équivalents) indépendantes du temps,
- . $\{KV\}$ le vecteur des sources évolutives.

L'ensemble des méthodes de construction de systèmes linéarisés que nous avons étudié peut être obtenu à l'aide de simples combinaisons d'expressions matricielles, mettant en oeuvre les matrices déjà rencontrées (en divers instants de discrétisation temporelle) et les dérivées de ces expressions par rapport à l'inconnue I , lorsque ces équations sont non linéaires.

Pour toutes les équations que nous avons traitées, on peut remarquer qu'elles mettent en oeuvre les matrices suivantes :

L_n		$[L_n] \left\{ \frac{\partial I}{\partial t} \right\}_n + [M_n] \{I\}_n = KS + KV_n$
M_n		
DL_n		
DM_n		$[L_{n-1}] \left\{ \frac{\partial I}{\partial t} \right\}_{n-1} + [M_{n-1}] \{I\}_{n-1} = KS + KV_{n-1}$
L_{n-1}		
M_{n-1}		$\left[\frac{\partial}{\partial I} ([M] \{I\}) \right] = [M] + [DM]$
DL_{n-1}		
DM_{n-1}		
KS		$\left[\frac{\partial}{\partial I} ([L] \{I\}) \right] = [L] + [DL]$
KV_n		
KV_{n-1}		

On peut certainement envisager des méthodes du traitement qui fassent intervenir d'autres types de matrices mais, dans tous les cas, les matrices mises en jeu seront toujours peu nombreuses et directement définies à partir des matrices constituant l'équation de départ.

L'ensemble des matrices qui permettent de mettre en oeuvre les diverses méthodes de résolution sont dénommées "MATRICES CONSTITUTIVES".

La formule qui permet de construire un système discrétisé linéarisé à l'aide de ces matrices est appelée "OPERATEUR DE CONSTRUCTION".

Nous pouvons reprendre les exemples d'équations déjà citées en leur appliquant les concepts que nous venons de définir.

L'équation de la diffusion thermique non linéaire s'écrit alors :

$$[L]_n T_n - [L]_{n-1} T_{n-1} + [M]_n T_n \Delta t = (KS + KV_n) \Delta t$$

dont la matrice Jacobienne s'écrit :

$$[L_n] + [DL_n] - [DL_{n-1}] + \Delta t ([M_n] + [DM_n])$$

où les matrices constitutives ont pour formule :

$$\begin{aligned} [L]_n &= [L]_{n-1} = \left[\iiint_{\Omega} (\rho C)_n \beta_i \alpha_j \right] \\ [M]_n &= \left[\iiint_{\Omega} \rho C_n \beta_i V \cdot \nabla \alpha_j + \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k}_n \nabla \alpha_j) + \iint_{\Gamma} h_n \beta_i \alpha_j + \iint_{\Gamma} \varepsilon \sigma_n T_n^3 \beta_i \alpha_j \right] \\ [DL]_n &= \left[\iiint_{\Omega} T_n \left(\frac{\partial \rho C}{\partial T} \right)_n \beta_i \alpha_j \right] \\ [DL]_{n-1} &= \left[\iiint_{\Omega} T_{n-1} \left(\frac{\partial \rho C}{\partial T} \right)_n \beta_i \alpha_j \right] \\ [DM]_n &= \left[\iiint_{\Omega} V \cdot \nabla T_n \left(\frac{\partial \rho C}{\partial T} \right)_n \beta_i \alpha_j + \iiint_{\Omega} \left(\frac{\partial \bar{k}}{\partial T} \right)_n \nabla T_n \cdot \nabla \beta_i \alpha_j + \iint_{\Gamma} T_n \left(\frac{\partial h}{\partial T} \right)_n \beta_i \alpha_j \right. \\ &\quad \left. + 3 \iint_{\Gamma} (\varepsilon \sigma)_n T_n^3 \beta_i \alpha_j + \iint_{\Gamma} T_n^4 \left(\frac{\partial \varepsilon \sigma}{\partial T} \right)_n \beta_i \alpha_j - \iint_{\Gamma} T_{an} \left(\frac{\partial h}{\partial T} \right)_n \beta_i \alpha_j \right] \\ [KS] &= \left[\iiint_{\Omega} \beta_i q_s \right] \\ [KV]_n &= \left[\iiint_{\Omega} \beta_i q_{vn} + \iint_{\Gamma} \beta_i h_n T_{an} + \iint_{\Gamma} \beta_i \varepsilon \sigma_n T_{an}^4 \right] \end{aligned}$$

Dans le cas de l'équation de Navier-Stokes déjà présentée la formulation de l'équation globale demeure identique :

$$[L_n]I_n - [L_{n-1}]I_{n-1} + [M]I_n \Delta t = (KS + KV_n) \Delta t$$

mais l'inconnue est alors vectorielle et les matrices sont définies par bloc.

Soit : $\vec{I} \begin{vmatrix} u \\ v \\ p \end{vmatrix}$ on détermine alors pour la matrice M par exemple :

$$M_{uu} = \left[\iint \rho N_i N_j \frac{\partial u}{\partial x} + \eta \iint \nabla N_i \cdot \nabla N_j \right]$$

$$M_{uv} = \left[\iint \rho N_i N_j \frac{\partial u}{\partial y} \right]$$

$$M_{up} = \left[\iint \frac{\partial N_i}{\partial x} M_j \right]$$

$$M_{vu} = \left[\iint \rho N_i N_j \frac{\partial v}{\partial x} \right]$$

$$M_{vv} = \left[\iint \rho N_i N_j \frac{\partial v}{\partial y} + \eta \iint \nabla N_i \cdot \nabla N_j \right]$$

$$M_{vp} = \left[\iint \frac{\partial N_i}{\partial y} M_j \right]$$

$$M_{pu} = \left[\iint M_i \frac{\partial N_j}{\partial x} \right]$$

$$M_{pv} = \left[\iint M_i \frac{\partial N_j}{\partial y} \right]$$

$$M_{pp} = 0$$

Dans les exemples que nous venons de citer nous pouvons remarquer que chaque matrice constitutive est la somme de plusieurs sous matrices. Dans le cas où l'inconnue I est vectorielle (cf équation de Navier-Stokes par exemple) les matrices constitutives doivent être définies par "bloc".

Les matrices qui composent (par bloc) chaque matrice constitutive sont appelées les "matrices intégrantes".

L'action de combinaison des matrices intégrantes pour obtenir les matrices constitutives est dénommée "opérateur de constitution".

Donnons à nouveau quelques exemples de matrices intégrantes rencontrées.

$$\begin{aligned}
 TE_{ij} &= \iiint_{\Omega} \rho C \beta_i \alpha_j && \text{terme principal de stockage} \\
 TEI_{ij} &= \iiint_{\Omega} T_n \left(\frac{\partial \rho C}{\partial T} \right)_n \beta_i \alpha_j && \text{terme jacobien de stockage} \\
 SEC_{ij} &= \iiint_{\Omega} \nabla \beta_i \cdot (\bar{k})_n \nabla \alpha_j && \text{terme principal de conduction anisotrope} \\
 SECI_{ij} &= \iiint_{\Omega} \left(\frac{\partial \bar{k}}{\partial T} \right)_n \nabla T_n \cdot \nabla \beta_i \alpha_j && \text{terme jacobien de conduction anisotrope} \\
 SEFi_j &= \iiint_{\Omega} (\rho C)_n \beta_i V \cdot \nabla \alpha_j && \text{terme principal de convection forcée} \\
 SEFC_{ij} &= \iiint_{\Omega} V \cdot \nabla T_n \left(\frac{\partial (\rho C)}{\partial T} \right)_n \beta_i \alpha_j && \text{terme jacobien de convection forcée} \\
 VES_i &= \iiint_{\Omega} \beta_i q_s && \text{sources statiques} \\
 VED_i &= \iiint_{\Omega} \beta_i (qv)_n && \text{sources dynamiques} \\
 VEC_i &= \iint_{\Gamma} \beta_i (h)_n T_{an} && \text{flux convectif entrant} \\
 NEC_{ij} &= \iint_{\Gamma} (h_n) \beta_i \alpha_j && \text{terme principal de convection libre} \\
 NECI_{ij} &= \iint_{\Gamma} T_n \left(\frac{\partial h}{\partial T} \right)_n \beta_i \alpha_j && \text{terme jacobien de convection libre} \\
 NER_{ij} &= \iint_{\Gamma} (\varepsilon \sigma)_n T_n^3 \beta_i \alpha_j && \text{terme principal de rayonnement} \\
 &\text{etc} \dots
 \end{aligned}$$

On peut remarquer que tous les intégrants sont composés d'expressions polynomiales (projection et interpolation) des dérivées de ces expressions, et des propriétés physiques provenant soit des matériaux qui constituent l'élément ($\rho, v, k, \dots$) soit des propriétés qui caractérisent les conditions aux limites de Neumann non homogène ($v \frac{\partial A}{\partial n}, h, hT_a, \dots$). Pour la plupart, ces valeurs ne sont pas constantes sur un élément. Ceci justifie à priori une intégration par méthode numérique qui s'impose également si on utilise des éléments curvilignes.

Nous dénommerons "calcul d'intégrant" l'opérateur

permettant d'évaluer une matrice intégrante.

Pour pouvoir effectuer son travail, cet opérateur impose le calcul préliminaire des propriétés intervenant dans la formule de la matrice intégrante, ce qui doit être réalisé à l'aide de l'opérateur "calcul de propriété". Il en est de même pour les polynômes évalués par l'opérateur "calcul de polynome".

Enfin, le choix des formules d'intégration au sein de l'élément("abscisses et poids")compte tenu des géométries curvilignes ,constitue le dernier degré de liberté concrétisé par l'opérateur "formule d'intégration".

Nous venons de montrer que la résolution de tout problème élémentaire peut être effectuée par l'enchaînement des huit classes d'opérateurs suivants :

1 . Formules d'intégration

Coordonnées & poids d'intégrations

2 . Calcul des polynomes

Valeurs des polynomes et dérivées aux points d'intégration

3 . Calcul des propriétés

Valeurs des propriétés

4 . Calcul des intégrants

Sous-matrices et sous-vecteurs intégrants

5 . Constitution

Sous-matrices et sous-vecteurs constitutifs

6 . Contruction du système linéarisé

Sous-matrice et sous-vecteur d'assemblage

7 . Assemblage

Matrice et deuxième membre du système

8 . Résolution du système linéaire algébrique

B.III DESCRIPTION D'UN PROBLEME POTENTIEL .-

Pour décrire un problème potentiel nous avons à notre disposition les opérateurs qui ont été définis au paragraphe précédent. Il faut ajouter la notion de variable (traitée par ces opérateurs) pour que le langage de description que nous proposons soit complet.

B.III.1 TYPES DES VARIABLES

Deux types de variables doivent être considérées :

- . celles qui dépendent de l'équation ou des méthodes de traitement ;
- . celles qui sont immuables et qui ne dépendent que de la technique des éléments finis.

Ces dernières constitueront les variables prédéfinies, implicitement connues par le système et dont la définition ne sera pas à la charge du concepteur d'un système objet. L'analyse de l'ensemble des opérateurs utilisés par notre méthode conduit à prédéfinir les variables suivantes :

t : temps
 Δt : pas de temps
 ω : pulsation des sources
MA : matrice assemblée
KA : deuxième membre assemblé
 SMA_i : sous-matrices d'assemblage
 SKA_i : sous-vecteurs d'assemblage

auxquelles nous devons ajouter les sous matrices constitutives déjà rencontrées :

L : l_{ni} termes temporels à l'instant n
LO : $l_{n-1,i}$ termes temporels à l'instant n-1

DL : $DL_{n,i}$ ^{des} dérivée partielle termes temporels partie non linéaire à l'instant n

DLO : $DL_{n-1,i}$ ^{des} dérivée partielle termes temporels partie non linéaire à l'instant n-1

M : $M_{n,i}$ termes d'espace à l'instant n

MO : $M_{n-1,i}$ termes d'espace à l'instant n-1

DM : $DM_{n,i}$ dérivée partielle des termes d'espace partie non linéaire à l'instant n

DMO : $DM_{n-1,i}$ dérivée partielle des termes d'espace partie non linéaire à l'instant n-1

KS : KS_i termes sources statiques

KV : $KV_{n,i}$ termes sources évolutifs à l'instant n

KVO : $KV_{n-1,i}$ termes sources évolutifs à l'instant n-1

Remarque : Vu la grande généralité des matrices constitutives, nous n'avons pas choisi de laisser au concepteur la possibilité de les définir lui-même.

Dans le cas où une méthode de traitement mettrait en oeuvre d'autres matrices constitutives, c'est au gestionnaire du système générateur qu'il revient de modifier ces variables pré-définies. (Cette action est tout-à-fait compatible avec les techniques de traitement mises en oeuvre qui sont toutes indépendantes du nombre et du type des matrices constitutives choisies).

Les variables qui doivent être définies par le concepteur d'un système objet sont les suivantes :

- . L'inconnue principale (nom de chacune des composantes) type réel ou complexe
- . Chacune des matrices intégrantes (on ne prend en compte que des variables de type réel, mais qui peuvent être des scalaires, des vecteurs ou des tableaux quelconques (tenseurs), les propriétés de type complexe doivent être décrites sous forme d'un couple de deux réels a, b ou r, θ).

- . Chacune des matrices intégrantes, qui sont toutes du type réel et peuvent être constituées soit d'un vecteur (homogène à une source) soit d'une matrice carrée dont la symétrie ou la dissymétrie dépend de l'opérateur qui l'évalue.

A ces déclarations préliminaires il convient d'ajouter le choix des options globales qui définissent :

*La dimension de l'espace d'étude 1D, 2D ou 3D

*Le degré des polynômes d'interpolations et de projection (on a prévu de pouvoir traiter les degrés 1,2,4).

B.III.2 DÉCLARATION DES OPÉRATEURS

Tous les opérateurs que nous avons présentés précédemment sont gérés par le système générateur.

Les opérateurs prédéclarés

Dans chacune des classes d'opérateurs que nous avons présentées, l'opérateur "vide" peut être utile. Il est donc prédéclaré par le système générateur.

Tous les opérateurs déclarés lors d'utilisations précédentes au cours de la construction de systèmes objets peuvent être mémorisés. Cette tâche assez fastidieuse ne sera donc pas à la charge du descripteur d'un nouveau système.

C'est donc seulement les nouveaux opérateurs que le concepteur d'un système objet devra déclarer de la même manière que les variables.

Le langage de description est alors complet et permet la définition de chaque problème potentiel.

à l'aide des intégrants qui les composent.

Par exemple la matrice M sera composée de :

$$\begin{aligned}
 & \iiint_{\Omega} C_n \beta_i V \cdot \nabla \alpha_j && \text{intégrant nommé SEF} \\
 & + \\
 & \iiint_{\Omega} \nabla \beta_i \cdot (k \nabla \alpha_j) && \text{intégrant nommé SEC} \\
 & + \\
 & \iint_{\Gamma} (h)_n \beta_i \alpha_j && \text{intégrant nommé NEC} \\
 & + \\
 & \iint_{\Gamma} \epsilon \sigma T_n^3 \beta_i \alpha_j && \text{intégrant nommé NER}
 \end{aligned}$$

En pratique la description d'un problème potentiel élémentaire peut être facilement réalisée de façon interactive, une fois que les opérateurs mis en oeuvre ont été décrits et lorsque les variables concernées ont été définies.

. Le concepteur d'un système objet doit tout d'abord choisir l'opérateur de construction de système correspondant, (parmi l'ensemble des opérateurs de ce type présents en bibliothèque).

. Le programme générateur a alors accès à la liste des matrices constitutives utilisées par l'opérateur choisi; il peut alors demander au concepteur, pour chacune des matrices constitutives nécessaires, la liste des matrices intégrantes qui les composent.

Le générateur dispose alors de toutes les informations nécessaires à la construction d'un problème objet potentiel.

Lorsque l'utilisateur d'un système objet va demander la résolution d'un problème réel, l'outil général de résolution va consulter d'une part le problème potentiel préalablement décrit et d'autre part les données concrètes qui définissent le problème réel (topologie du découpage, coordonnées des points, valeur des propriétés physiques). Nous allons décrire le principe de

fonctionnement de ce programme.

B.IV TRAITEMENT D'UN PROBLEME REEL .-

Le mode de traitement d'un problème réel repose sur le principe suivant : pour chaque problème l'algorithme construit un graphe de décision qui va définir les opérateurs et les variables nécessaires à sa résolution.

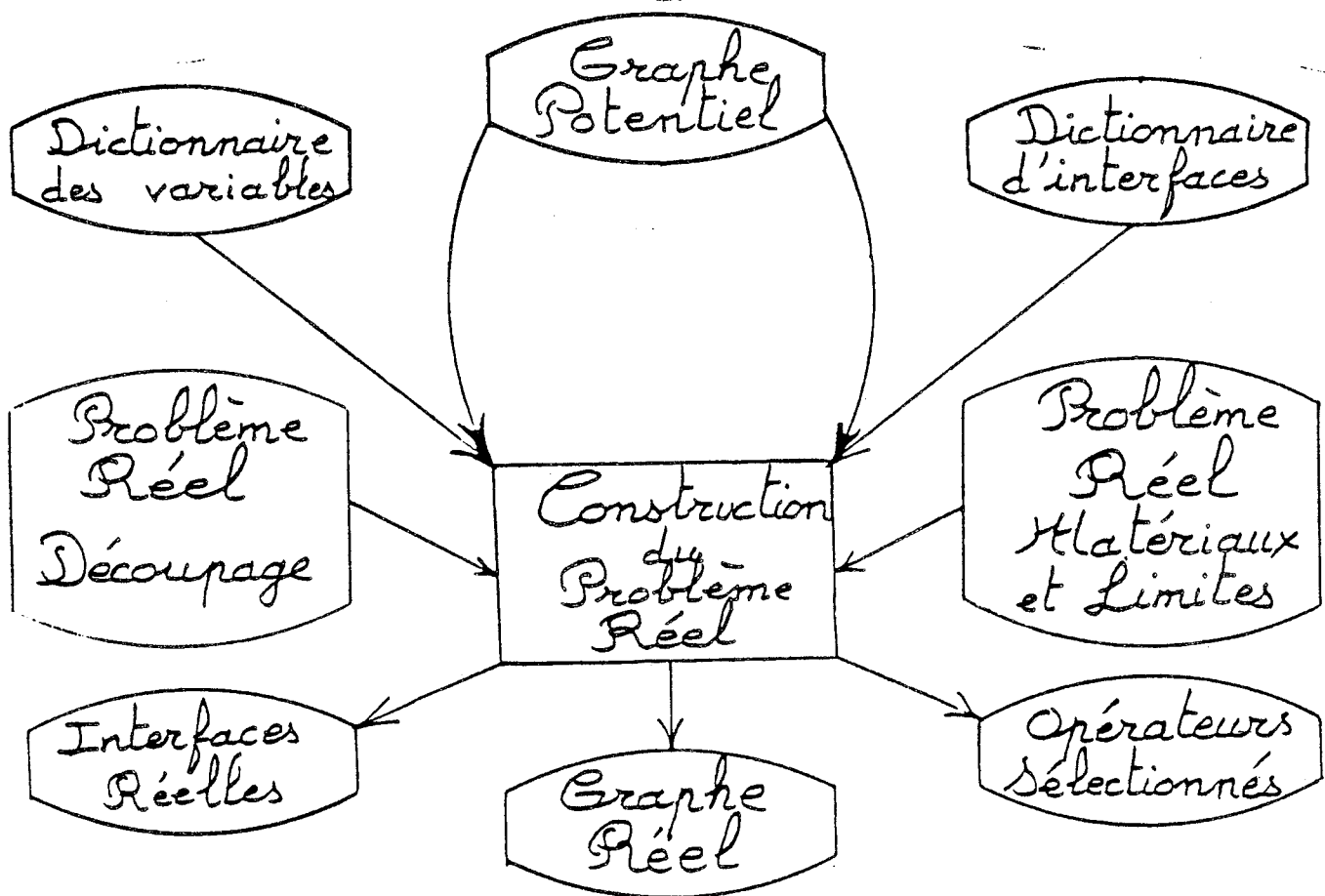


Figure 4 La construction du graphe de résolution.

B.IV.1 STRUCTURE GÉNÉRALE DU GRAPHE

Rappelons brièvement l'enchaînement des opérateurs qui composent un problème potentiel élémentaire.

intégration	{	1. calcul des formules d'intégration
		2. calcul des polynômes
		3. calcul des propriétés
		4. calcul des sous-matrices intégrantes
		5. calcul des sous-matrices constitutives
		6. construction du sous-système linéaire
résolution	{	7. assemblage
		8. résolution du système linéaire

L'étape de résolution de système est autonome et succède à l'assemblage qui est constitué d'une boucle classique sur l'ensemble des éléments. En outre, on suppose que les sous-matrices d'assemblage ont été stockées au préalable sur disque ; les phases 7 et 8 sont dénommées phase de résolution. Les phases 1 à 6 composent ce que nous dénommerons la phase globale d'intégration. Nous avons pris l'option de traiter la succession de ces étapes pour chaque élément de façon totalement indépendante (ce qui est particulièrement bien adapté à des structures de calculateurs multiprocesseurs parallèles). Nous allons détailler, pour un élément quelconque, comment le programme de résolution de problèmes peut calculer les matrices d'assemblages correspondant à l'équation décrite dans le problème potentiel.

Pour chaque élément l'algorithme procède en deux temps :

. Analyse du graphe de décision qui permet de définir les variables et les opérateurs nécessaires au calcul de la sous-matrice d'assemblage et le sous-vecteur.

. Calcul de ces valeurs à l'aide des opérateurs définis par le graphe suivant l'ordre donné par ses niveaux successifs.

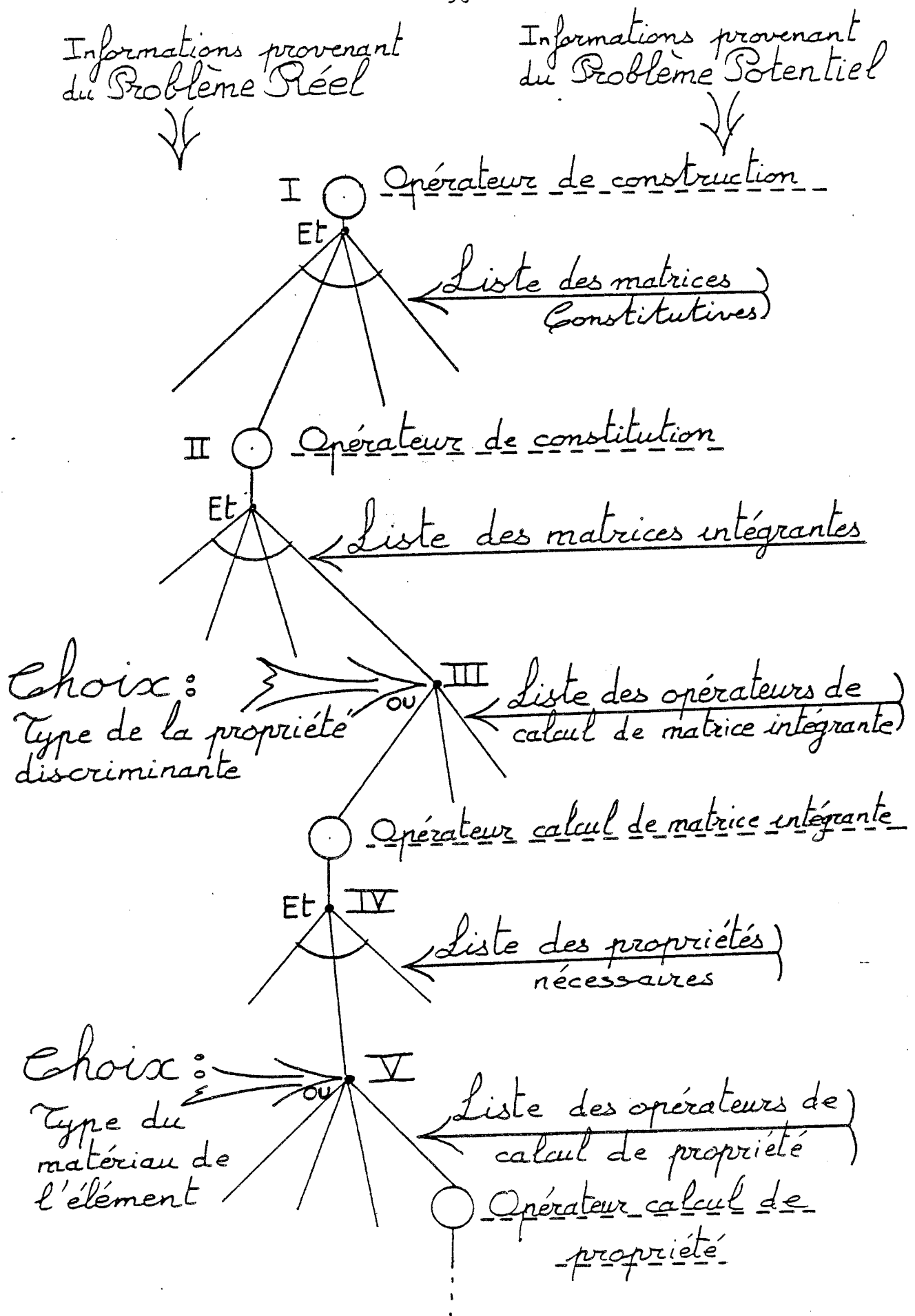


Figure 4 Début : graphe réel.

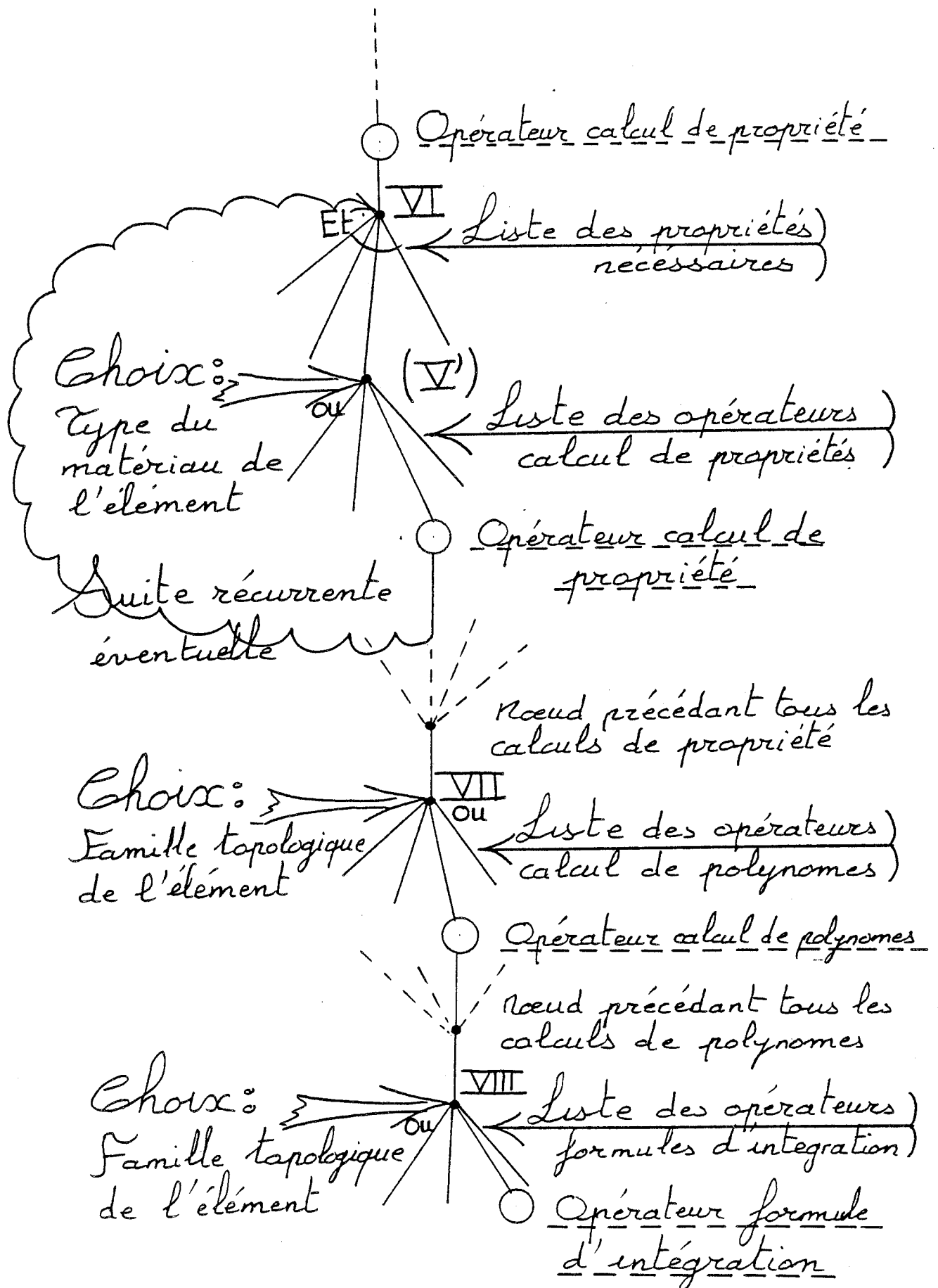


Figure 4 Fin : graphe réel.

B.IV.2 CONSTRUCTION DU GRAPHE DE DÉCISION D'UN PROBLÈME RÉEL

Deux sources d'informations sont utilisées par l'algorithme.

- . La description du problème potentiel : (fournie par le concepteur d'un système objet à l'aide du générateur).

- . Le problème réel : le découpage du domaine et la définition de ses propriétés (fournis par l'utilisateur du système objet).

Noeud_décisionnel_I

A chaque problème potentiel est associé un seul opérateur de construction. L'algorithme en déduit la liste des matrices constitutives qu'il faudra calculer.

Noeud_décisionnel_II

Pour chaque matrice constitutive sélectionnée par le niveau précédent l'algorithme trouve dans le problème potentiel la liste des matrices intégrantes qui les composent.

Noeud_décisionnel_III

Pour chaque matrice intégrante plusieurs opérateurs de calculs peuvent être mis en oeuvre pour les évaluer (selon le type de la propriété discriminante, cf ~~SBV~~ 4.4) ; l'algorithme consulte alors la description de la matrice intégrante dans le problème potentiel : il y trouve le nom de la propriété discriminante. Il scrute alors les données du problème réel pour déterminer le type de la propriété discriminante de l'élément en cours de traitement. Cela permet donc de sélectionner l'opérateur de calcul de matrice intégrante nécessaire.

Noeud_décisionnel_IV

Le problème potentiel fournit pour chaque opérateur de

matrice intégrante la liste des propriétés nécessaires à son calcul.

Noeud_décisionnel_V

Le problème réel (notamment les propriétés de l'élément en cours de calcul) permet de déterminer l'opérateur que l'on doit utiliser pour calculer chaque propriété.

Noeud_décisionnel_VI

Quand un opérateur de calcul de propriété a besoin à son tour d'une autre propriété le problème potentiel décrit la liste des propriétés nécessaires.

Ces deux dernières phases sont donc récurrentes jusqu'à ce que toutes les propriétés nécessaires soient fournies par des modèles ne nécessitant pas le calcul de nouvelles propriétés (aucune récursivité n'est bien sûr permise à ce niveau).

Noeud_décisionnel_VII

Il s'agit alors de déterminer selon le type d'élément les opérateurs de calcul de polynômes nécessaires. C'est le problème réel qui fournit cette information.

Noeud_décisionnel_VIII

Le type d'élément permet également le choix de l'opérateur formule d'intégration.

B.IV.3 EXÉCUTION DU GRAPHE

Lorsque l'algorithme d'analyse de problème réel a construit le graphe, il dispose de la liste de toutes les informations à calculer et de tous les opérateurs nécessaires. Il lui suffit alors d'enchaîner le calcul de ces opérateurs. Deux piles peuvent servir à cet enchaînement, d'une part la liste des opérateurs et d'autre part celle de leurs interfaces d'entrée. Grâce au mode d'analyse du graphe un opérateur trouvera toujours dans la pile

d'interface les informations qui lui sont nécessaires et qui auront été automatiquement calculées au préalable.

Nous ne détaillerons pas plus le traitement technique de ces informations mais il est évident que l'on peut l'adapter aisément à une machine multiprocesseur (en construisant plusieurs piles) et que l'on peut optimiser le nombre d'opérateurs utiles en détectant d'éventuelles répétitions.

B.IV.4 ENCHAÎNEMENT DYNAMIQUE DES OPÉRATEURS

Afin de pouvoir concevoir un programme général capable de traiter tous les problèmes potentiels descriptibles par le générateur et tous les problèmes réels posés par les utilisateurs de chaque système objet, il faut disposer :

- d'une gestion dynamique de mémoire au sein de bases de données qui concrétisent problème potentiel et problème réel.
- de la possibilité d'appeler dynamiquement l'un quelconque des opérateurs d'une même classe.

Cette dernière particularité, beaucoup moins classique, impose la création d'outils informatiques spécialement conçus pour notre système. (Appel dynamique de sous-programmes à l'exécution sans que ceux-ci n'aient été définis lors de la compilation). En effet, nous voulons que l'algorithme proposé puisse être programmé et compilé une fois pour toutes sans que le concepteur d'un système objet n'ait à le modifier. Nous exposerons dans la 3ème partie la réalisation technique d'un tel outil.

B.V DESCRIPTION DES OPERATEURS ELEMENTAIRES .-

Nous venons de montrer comment il est possible de décrire la résolution de tout problème potentiel à l'aide d'un enchaînement

d'opérateurs.

Dans ce chapitre nous allons détailler chacun des opérateurs en caractérisant les applications qu'ils concrétisent ainsi que leurs ensembles de départ et d'arrivée.

De plus pour chaque cas nous délimiterons ce qui est immuable (et donc peut être programmé une fois pour toutes) de ce qui est évolutif et doit être laissé au choix du concepteur d'un système objet.

B.V.1 OPÉRATEUR FORMULES D'INTÉGRATIONS

Toutes les formules d'intégration numériques envisagées ici sont de la forme :

$$\int f d\Omega = \sum_i \rho_i f(V_i)$$

où V_i est le vecteur $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ des coordonnées des points

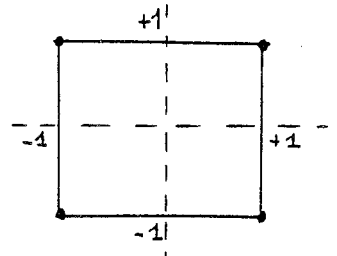
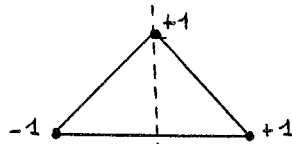
(dits points d'intégrations) et ρ_i les poids associés à chacun de ces points.

Généralement ces valeurs sont calculables une fois pour toutes sur des éléments réguliers normés.

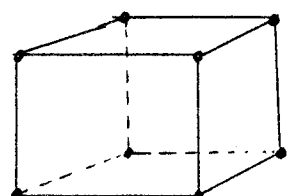
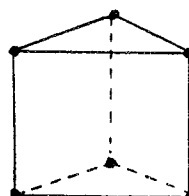
. linéiques



. surfaciques



. volumiques

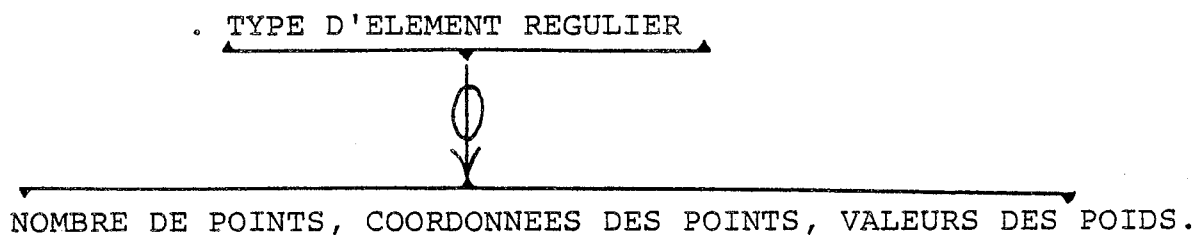


. etc.....

Les méthodes de Gauss sont les plus employées ; elles utilisent en général peu de points dans chaque direction (maximum 10), mais d'autres formules peuvent être employées (notamment pour les triangles, prismes et tétraèdres). Pour des raisons de précision il faut éviter d'utiliser des points d'intégration dont les poids associés seraient de signes différents ou de valeurs trop disparates.

Dans le cas d'éléments non réguliers ou curvilignes on peut toujours trouver un changement de coordonnées qui ramène l'intégration dans les bornes $-1 ; +1$ [25]. Il est alors nécessaire de corriger les poids d'intégration P_i , en les multipliant par le déterminant de la transformation utilisée. Ceci peut être effectué par un programme standard indépendant des formules d'intégration choisies.

Le concepteur d'un système doit donc uniquement choisir pour chaque type d'élément mis en oeuvre dans son projet, une ou plusieurs formules d'intégrations. L'opérateur "formule d'intégration" réalisera donc l'application suivante :



B.V.2 OPÉRATEURS CALCULS DE POLYNOMES

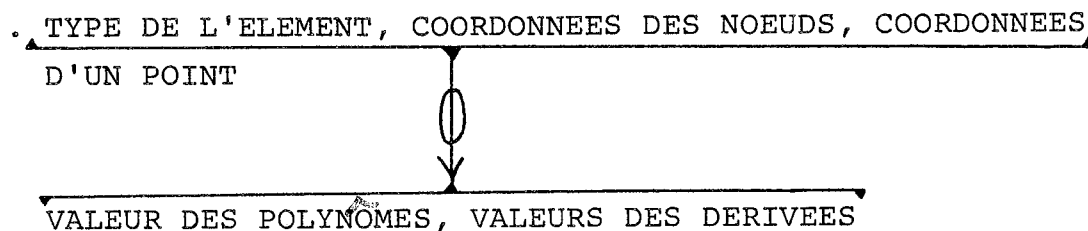
Le choix des polynômes d'interpolation dépend de trois facteurs :

- le type de l'équation traitée (doit-on assurer uniquement la continuité de la variable ou également celle des dérivées ?). Ceci va déterminer la formule mathématique des polynômes nécessaires (pour les équations

données en exemples les polynômes de Lagrange conviennent).

- Le degré de l'interpolation choisie. On rappelle (cf Navier-Stokes) que divers degrés d'interpolation peuvent coexister dans un même système.

- La forme de l'élément interpolé ^{chaque} tant opérateur "calcul de polynome" doit donc concrétiser l'application suivante :



Remarque : Là encore les polynômes peuvent être définis directement sur des éléments réguliers normés, le passage à l'élément fini curviligne étant effectué par un opérateur programmé une fois pour toutes applicable à tous les opérateurs calcul de polynome (calcul des dérivées en coordonnées curvilignes).

B.V.3 OPÉRATEURS CALCULS DE PROPRIÉTÉS

C'est dans ce domaine que la physique nous apporte la plus grande variété de possibilités. Il est donc absolument impossible de concevoir un seul opérateur qui puisse modéliser toutes les propriétés. Une très grande souplesse doit donc être préservée.

Avant de passer à des opérateurs élaborés nous pouvons déjà traiter quelques cas triviaux

- . Propriété nulle
- . Propriété constante

Ces opérateurs devront donc faire partie du système et ne seront pas à la charge d'un concepteur.

Pour les problèmes traités par couplage il est fort utile de pouvoir utiliser des propriétés qui ont été calculées au préalable par un système aux éléments finis. Les polynômes (que l'on a déjà présentés) peuvent donc fournir un moyen efficace d'interpoler ces valeurs nodales. Ce type d'opérateur peut également être pré-programmé pour ne pas être à la charge du concepteur.

Il reste donc à définir la structure générale des opérateurs que le concepteur devra créer pour modéliser ses propriétés. Nous avons prévu de fournir à ces opérateurs les informations suivantes :

- les coordonnées des points où l'on doit calculer la propriété XYZ
- la valeur du temps t
- la valeur de la variable principale aux points de calcul (chaque composante dans le cas vectoriel) I
- la valeur du gradient ou du rotationnel de la variable principale (ou de ses composantes) $\nabla I_i, \nabla X I_i$
- la valeur d'une première propriété auxiliaire (ou de son gradient ou de son rotationnel)
- la valeur d'une seconde propriété auxiliaire (ou de son gradient ou de son rotationnel)
- une table quelconque définie par l'utilisateur qui contiendra une liste de nombres réels (ce qui peut servir à définir des coefficients variables, pour des formules analytiques, des coordonnées de points lorsque les opérateurs modélisent des relevés expérimentaux, etc...)
- le nombre de valeurs de la table précédente.

L'opérateur le plus général que nous proposons peut être noté :

$$f(x, y, z, t, \begin{Bmatrix} I \\ \nabla I \\ \nabla \times I \end{Bmatrix}, \begin{Bmatrix} V1 \\ \nabla V1 \\ \nabla \times V1 \end{Bmatrix}, \begin{Bmatrix} V2 \\ \nabla V2 \\ \nabla \times V2 \end{Bmatrix}, \text{Table, nb d'éléments})$$

parmi toutes les expériences que nous avons réalisées ou seulement envisagées, cet opérateur nous a paru largement suffisant. Il permet de modéliser des castrès complexes comme un cycle d'hystérésis qui dépend du temps de la température, de la pression, du rotationnel de l'inconnue principale, et que l'on définirait par des réseaux de courbes interpolées par des fonctions splines.

Remarque : L'opérateur calcul de propriété sera mis en oeuvre dans notre système pour calculer quatre types distincts de valeurs :

- . Les propriétés classiques de matériaux ($\mu, \nu, \eta, \sigma, \dots$) valeurs de type volumique.
- . Des variables auxiliaires (souvent elles-mêmes calculées au préalable par éléments finis dans les systèmes couplés) (Température, Champ de Vitesse ...).
- . Des conditions aux limites de Neumann non homogène : valeur d'un flux connu sur une paroi ($k \frac{\partial T}{\partial n}$), ($\nu \frac{\partial A}{\partial n}$) ou bien des propriétés caractérisant l'échange des parois (h), (hT_0), (ces valeurs étant alors de type surfacique).
- . Des conditions aux limites de type Dirichlet lorsque celles-ci dépendent du temps (valeur de type surfacique).

B.V.4 OPÉRATEUR DE CALCUL D'INTÉGRANT

Les exemples que nous avons déjà cités montrent la grande diversité des formules que l'on doit prendre en compte. Une nouvelle notion va nous permettre de déterminer le type d'un intégrant d'après la nature de l'une de ses propriétés : la propriété discriminante.

Nous avons vu que suivant les phénomènes physiques modélisés un même intégrant pouvait être évalué par divers types de formules qui tiennent compte des différents modèles des propriétés. Un exemple caractéristique de ce problème est donné par le terme $\nabla \cdot k \nabla T$ dans lequel k est soit un tenseur (conductibilité anisotrope) soit un scalaire (conductibilité isotrope) il conduit comme nous l'avons montré dans la première partie à deux manières distinctes de calculer l'intégrant correspondant.

$$SE = \iiint k \vec{\nabla} \alpha_i \cdot \vec{\nabla} \alpha_j \quad \text{produits scalaires}$$

ou bien

$$SE = \iiint \nabla \alpha_i \cdot k \nabla \alpha_j \quad \text{produits tensoriels}$$

Certes il est mathématiquement possible de considérer un scalaire comme un tenseur diagonal équivalent $\begin{pmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{pmatrix}$ mais le coût d'une telle généralisation est prohibitif sur le plan informatique. Le mode de chaque intégrant sera donc associé à une propriété (appelée propriété discriminante : conductibilité k dans l'exemple proposé) dont le type (réel, vecteur, tenseur) permettra de sélectionner l'opérateur de calcul de matrice intégrante correspondant respectivement à chacun de ces cas.

D'autre part, si on considère la même équation mais dans des systèmes de coordonnées distincts (cartésien ou axisymétrique) et avec des inconnues pouvant avoir subi des changements de variables (changement de variable $\mathcal{A} = \mathcal{R}$ très utile en axisymétrique), le nombre de formules de calcul d'intégrant

devient extrêmement élevé. Si on prévoit un seul type d'opérateur pour traiter le calcul d'un intégrant on voit qu'il faudra que le concepteur d'un système programme chacun des cas nécessaires ce qui peut conduire à un travail fastidieux.

Mais beaucoup de ces intégrants ont des structures similaires :

$$\iint \rho \nabla \alpha_i \cdot \nabla \alpha_j d\Omega, \iint \rho \nabla \alpha_i \nabla \alpha_j r d\Omega, \iint v \nabla \alpha_i \nabla \alpha_j \frac{\partial \Omega}{r} \dots$$

C'est pourquoi on propose de décomposer le calcul en deux étapes :

- évaluation d'un élément différentiel généralisé du type :

$$\mu d\Omega, \mu r d\Omega, v \frac{d\Omega}{r}, \dots = dG$$

par un opérateur noté Pds

- et calcul de l'expression proprement dite :

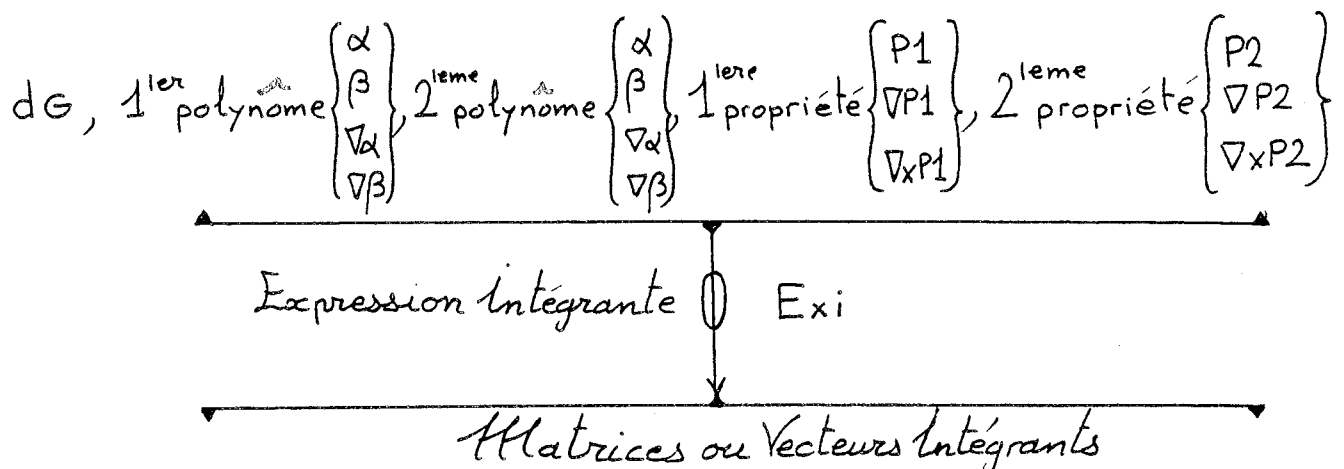
$$\iint \nabla \beta_i \cdot \nabla \alpha_j dG, \iint \beta_i \alpha_j dG, \iint \beta_i dG \dots$$

par un opérateur noté Exi

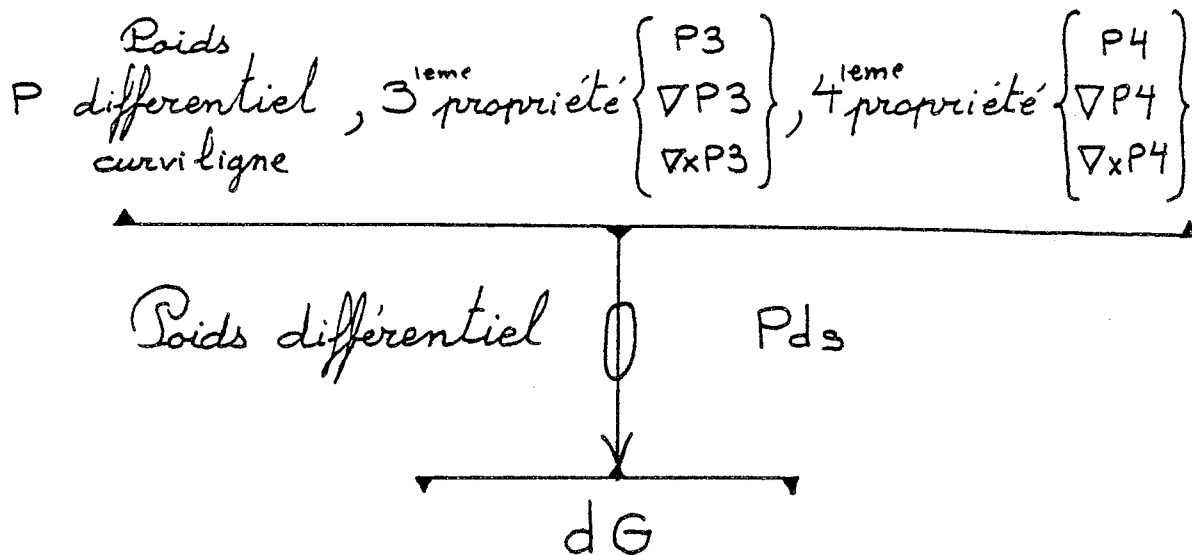
On obtient grâce à la combinaison de ces deux opérateurs un nombre de programmes beaucoup plus restreint.

L'opérateur de calcul d'intégrant sera donc noté Pds ⊗ Exi

L'opérateur Exi : il effectue une application dont la structure la plus générale que nous proposons est :



dans laquelle dG a été évalué par un opérateur quelconque réalisant une application du type Pds représentable par :



Dans laquelle P le poids différentiel géométrique est la valeur évaluée par l'opérateur calculant les formules d'intégrations (corrigée par le changement de coordonnées curvilignes déjà expliqué).

Donnons quelques exemples de décomposition de calcul de matrice intégrante à l'aide de ces opérateurs.

Supposons de les opérateurs suivants soient disponibles.

$$Exi = \iint P_i \nabla \alpha_i \cdot \nabla \alpha_j dG$$

$$Pds1 = dx dy ; Pds2 = r dr dz ; Pds3 = \frac{dr dz}{r}$$

alors dans un tel cas:

$$Pds1 \otimes Exi(u) \text{ calcule : } \iint u \nabla \alpha_i \cdot \nabla \alpha_j dx dy /$$

$$Pds2 \otimes Exi(k) \text{ calcule : } \iint k \nabla \alpha_i \cdot \nabla \alpha_j r dr dz /$$

$$Pds3 \otimes Exi(v) \text{ calcule : } \iint v \nabla \alpha_i \cdot \nabla \alpha_j \frac{dr dz}{r} /$$

etc...

B.V.5 CONSTITUTION DES SOUS-MATRICES

Nous avons vu que cet opérateur est unique et effectue la sommation des matrices intégrantes pour reconstituer les matrices constitutives. Il peut donc être programmé au niveau du système une fois pour toutes. Il ne sera donc pas à la charge d'un concepteur.

Compte-tenu de la liste des matrices constitutives préalablement déclarée et de la liste des matrices intégrantes nécessaires à la reconstitution de chaque matrice cet opérateur très simple réalise les sommations nécessaires.

Même lorsque l'inconnue est un nombre complexe (c'est le cas d'une discrétisation du type $[j\omega L + M]\{\bar{I}\} = \{K_r + jK_i\}$) les matrices constitutives sont toujours de type réel.

B.V.6 CONSTRUCTION DES ^{CA} SOUS-MATRICES D'ASSEMBLAGE ET DU SECON MEMBRE

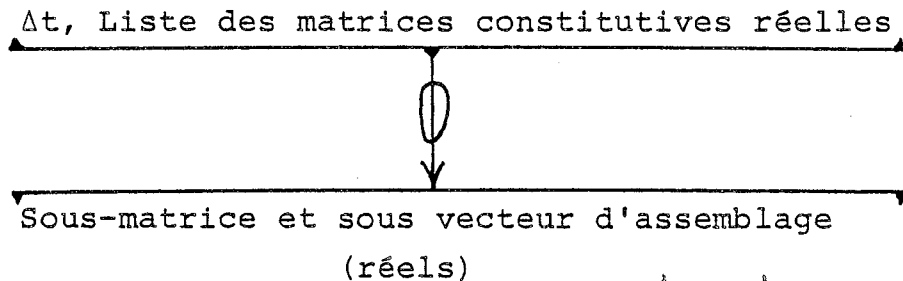
C'est à ce niveau qu'intervient la méthode choisie pour, d'une part séparer temps et espace, et d'autre part linéariser les éventuels systèmes non linéaires obtenus.

Nous pouvons citer quelques méthodes déjà rencontrées dans la 1ère partie et que nous avons reformulées à l'aide des notations déjà introduites.

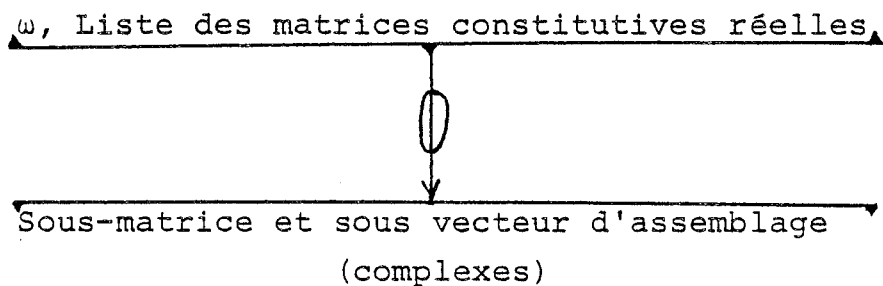
- . Cas linéaire (nombres réels)
 $[M]I = KS$
- . Cas non linéaire Newton-Raphson (nombres réels)
 $R = [M]I - KS, [J] = [M] + [DM]$
- . Cas linéaire (nombres complexes)
 $\{[M] + j\omega[L]\}I = KS + j.KV$
- . Cas linéaire dynamique implicite
 $\{[L] + dt[M]\}I = [L]I_0 + dt \{KS + KV\}$

- . Cas non linéaire dynamique implicite Newton-Raphson
 $R = [L](I - I_0) + dt \{ [M]I - KS - KV \}$
 $[J] = [L] + [DL] - [DLO] + dt \{ [M] + [DM] \}$
- . Crank Nicholson linéaire
 $\{ [L] + dt.r.[M] \} I = [L]I_0 - dt \{ s[M]I_0 - KS - sKVO - r.KV \}$
- . Crank Nicholson non linéaire méthode de Newton Raphson
 $R = [L](I - I_0) + dt \{ r[M]I + s[MO]I_0 - rKV \}$
 $[J] = [L] + dt.r. \{ [M] + [DM] \}$

L'opérateur général correspondant réalise une application dont la structure est :



Remarquons que dans le cas où ^{les éléments} la nature mathématique des matrices obtenues ~~doit être~~ des nombres complexes l'opérateur à la structure : ^{son}



Nous distinguerons donc entre deux types d'opérateurs de construction (réel et complexe).

B.V.7 ASSEMBLAGE DU SYSTÈME LINÉARISÉ

Après l'application à tous les éléments de l'opérateur de construction on dispose de la sous-matrice et du sous vecteur deuxième membre pour chacun des éléments.

La phase d'assemblage est très technique et il nous a semblé important de ne pas la laisser à la charge du concepteur d'un système. Nous avons donc construit un opérateur d'assemblage apte à traiter une grande variété de problèmes.

- . Parmi les problèmes à assembler on distingue tout d'abord deux classes :
 - les problèmes où l'on doit assembler seulement le deuxième membre (problèmes linéaires multisources, méthode de Newton-Kantorovitch ...)
 - les problèmes où la matrice et le deuxième membre doivent être traités.
- . On doit également aborder différemment :
 - les cas traitant des nombres réels
 - les cas mettant en oeuvre des nombres complexes.
- . La structure de la matrice à construire (symétrie éventuelle, mode de stockage) influe également sur l'algorithme.
 - Deux cas sont donc envisagés : symétrique et non symétrique
 - L'algorithme peut s'adapter à divers modes de stockage (parmi ceux qui seront choisis pour l'opérateur de résolution de système)
- . Le traitement des conditions aux limites constitue le dernier degré de liberté de l'opérateur d'assemblage.

Remarquons tout d'abord que les conditions de type

Neumann non homogène sont traduites au sein des équations par des matrices intégrantes ($\iint_{\Gamma} v \frac{\partial A}{\partial n} \alpha_i$, $\iint_{\Gamma} h \alpha_i \alpha_i$ ) et n'interviennent donc pas explicitement au niveau de l'assemblage puisqu'elles participent à l'élaboration des matrices constitutives. Il suffit donc de traiter à ce niveau les conditions de type Dirichlet et les conditions topologiques. Nous avons prévu à cet effet de prendre en compte des conditions de types :

Cycliques	$I_i = I_s$
Anticycliques	$I_i = -I_s$
Translation	$I_i = I_j + \phi$

qui ont été largement utilisées dans le système FLUX et permettent très souvent de diminuer considérablement les domaines à étudier. D'autres types de conditions pourraient être ajoutés aisément.

Nous pensons que l'opérateur général ainsi conçu permet de traiter un large éventail de problèmes qui ne devraient pas restreindre les degrés de liberté des systèmes objets conçus à l'aide de notre générateur. ^{Le concepteur} L'utilisateur sera ainsi déchargé de ce problème souvent délicat : concevoir un programme d'assemblage et réaliser ce dernier.

B.V.8 RÉSOLUTION DES SYSTÈMES LINÉAIRES

Notre recherche n'a pas porté sur ce domaine (au demeurant très important), nous avons simplement prévu que le concepteur nous fournirait un opérateur de résolution de systèmes linéaires dont les degrés de liberté sont :

- Type de la variable Réelle.Complexe
- Structure de la matrice Symétrique.Disymétrique
- Mode de stockage

Des recherches conduites par ailleurs ont pu tester

l'efficacité de certains opérateurs [61]

- . ICCG pour les matrices symétriques définies positives
- . gradient biconjugué pour les matrices dissymétriques

et cela dans un mode de stockage adapté à la structure très creuse des matrices (stockage par matrice symbolique). Les opérateurs correspondants sont bien sûr programmés et à la disposition des utilisateurs qui ne voudraient pas mettre en oeuvre leur propre méthode.

B.V.9 DEGRÉ DE LIBERTÉ D'UN PROBLÈME POTENTIEL ÉLÉMENTAIRE

Dans les paragraphes précédents, nous avons mis en évidence les divers opérateurs de traitement de l'information caractéristiques de la résolution d'un problème aux éléments finis. Nous avons montré qu'un certain nombre de tâches étaient immuables et pouvaient être prises en compte par le système ou par des opérateurs généraux programmés une fois pour toutes. Le concepteur doit donc déterminer les opérateurs qui lui sont nécessaires parmi seulement chacune des 6 classes suivantes :

- . formules d'intégration
- . calcul des polynômes
- . évaluation des propriétés
- . calcul des intégrants
- . construction d'un sous-système
- . résolution du système linéaire

Bien entendu, dans la partie "bibliothèque" de notre système ~~l'utilisateur~~ ^{le concepteur} pourra trouver un grand nombre de ces opérateurs déjà disponibles et il n'aura à sa charge que la réalisation de quelques opérateurs supplémentaires qui lui ~~feront~~ ^{permettent} défaut pour la construction d'un système objet, et qui viendront grossir le nombre des possibilités offertes dans la bibliothèque.

La plus grande part du travail du concepteur d'un système objet ne sera donc pas constituée par la programmation des opérateurs que l'on vient de présenter mais elle se limitera

essentiellement à la description des problèmes potentiels à l'aide des opérateurs préalablement définis.

B.VI STRATEGIES GENERALES ET MACHINE ABSTRAITE

Dans le cas d'une équation statique linéaire un seul problème potentiel permet de traiter le sujet qui se compose donc d'un seul problème réel. Mais dans le cas d'une équation plus complexe (problèmes évolutifs ou problèmes non linéaires) il faut utiliser des méthodes itératives (Newton-Raphson, Newton-Kantorovitch, Crank-Nicholson, prédiction correction, etc...) qui mettent en oeuvre plusieurs types de problèmes (plusieurs problèmes potentiels) et pour chacun nécessitent la résolution de plusieurs problèmes réels successifs.

Nous avons dénommé "Stratégie générale de résolution" l'algorithme qui traduit ce type de traitement.

Deux méthodes informatiques peuvent être envisagées pour matérialiser les diverses stratégies utilisables : la méthode opérationnelle ou la mise en oeuvre d'une machine abstraite.

B.VI.1 TRAITEMENT OPERATIONNEL DES STRATEGIES

Cette approche peut être étayée par la remarque suivante : le nombre de stratégies que l'on peut envisager est relativement faible. Il est donc possible de préprogrammer les stratégies les plus classiques et ainsi de concevoir une bibliothèque d'opérateurs très généraux capables de traiter n'importe quel sujet sur n'importe quel système objet. Cela revient à utiliser pour les stratégies la même méthode de traitement que pour les problèmes objets.

Avantages de cette solution

- Les stratégies préprogrammées ne sont pas à la charge

du concepteur d'un système objet, ce qui allège considérablement son travail qui se limitera alors à la description de chacun des problèmes potentiels mis en oeuvre par les stratégies existantes.

- L'utilisateur d'un système objet pourra choisir une stratégie adaptée à son sujet d'étude. Il lui suffira de spécifier au programme général de résolution le nom de l'opérateur de stratégie choisie.

- Chaque opérateur de stratégie peut être programmé en Fortran ce qui garantit une bonne rapidité d'exécution.

Inconvénients

- Si le concepteur d'un système objet ne trouve pas parmi les opérateurs de stratégie un cas qui conviennent à ses équations, il lui faudra écrire un nouvel opérateur. Mais la conception d'un tel programme demande des connaissances en informatique qui ne sont généralement pas du ressort du physicien concepteur d'un système objet.

- Le langage choisi pour cette programmation n'est pas très bien adapté au traitement algorithmique dynamique. Une bonne connaissance des outils du système est nécessaire pour assurer la compatibilité avec l'utilisation de l'outil de résolution de problèmes.

Nous concluons de cette analyse, que le traitement opérationnel des stratégies doit être limité au cas où l'introduction d'une nouvelle stratégie dans le système, reste l'apanage du gestionnaire du générateur et n'est pas à la charge du concepteur d'un système objet.

B.VI.2 TRAITEMENT DES STRATÉGIES PAR MACHINE ABSTRAITE

Une telle méthode est utilisable si on peut dégager au sein de l'ensemble des stratégies envisageables des phases de traitement de l'information suffisamment générales pour pouvoir être concrétisées par des macro-instructions de la machine abstraite.

On peut vérifier aisément que les stratégies déjà citées sont des algorithmes très classiques mettant en oeuvre :

- des opérations vectorielles classiques
- la résolution d'un système linéaire (prise en compte par l'outil de résolution du problème qui constitue par lui-même l'une des macro-instructions envisagées)
- les instructions classiques du traitement algorithmique (tant que, jusqu'à ce que, choix multiple, ...)
- des entrées-sorties conversationnelles qui peuvent être gérées par des macro-instructions (choix dans un menu, lecture de données)
- des déclarations de variables compatibles avec l'outil de résolution du problème et la gestion des bases de données (mais indépendantes de la logique interne de ces programmes).

Les avantages de cette approche :

- la stratégie peut être décrite dans un langage algorithmique de haut niveau beaucoup plus agréable à manipuler que le fortran par exemple.
- La programmation d'une stratégie peut être réalisée

par une personne qui n'est pas au courant des outils informatiques mis en oeuvre dans le reste du système (contrairement à la solution précédente).

- La possibilité de conserver les stratégies déjà programmées demeure.

L'inconvénient principal de cette méthode est le temps de réponse qui est nécessairement grevé par les techniques d'interprétation utilisées par la machine abstraite.

D'autre part les outils informatiques mis en oeuvre par une machine abstraite sont relativement lourds à réaliser et viennent encore augmenter la taille et la complexité d'un système nécessairement déjà bien volumineux.

B.VI.3 CHOIX DE LA MÉTHODE IMPLANTÉE DANS NOTRE PROTOTYPE

Dans le prototype que nous avons construit c'est le traitement opérationnel des stratégies que nous avons mis en oeuvre, pour des raisons d'efficacité d'exécution mais surtout parce que c'était la solution la plus simple et donc la plus facile à mettre au point. On pourra envisager d'implanter un traitement par machine abstraite ultérieurement lorsque les concepts mis en oeuvre dans notre système auront été validés par une utilisation plus dense que les tests que nous avons pu réaliser.

Nous présentons en conclusion de ce chapitre la structure générale de notre système à l'aide d'un schéma global dans lequel on retrouvera l'ensemble des éléments que nous avons présentés :

- le générateur
permettant la description de la stratégie et de chaque problème potentiel
- l'outil d'analyse de problème qui parcourt le graphe

des décisions selon les indications combinées du problème potentiel et du problème réel

- l'outil de résolution qui résout le problème précédemment construit.

Nous avons omis sur ce schéma les outils de description du problème réel et les outils de visualisation des résultats qui peuvent être totalement autonomes et n'interviennent donc pas directement au cours de la résolution d'un problème réel.

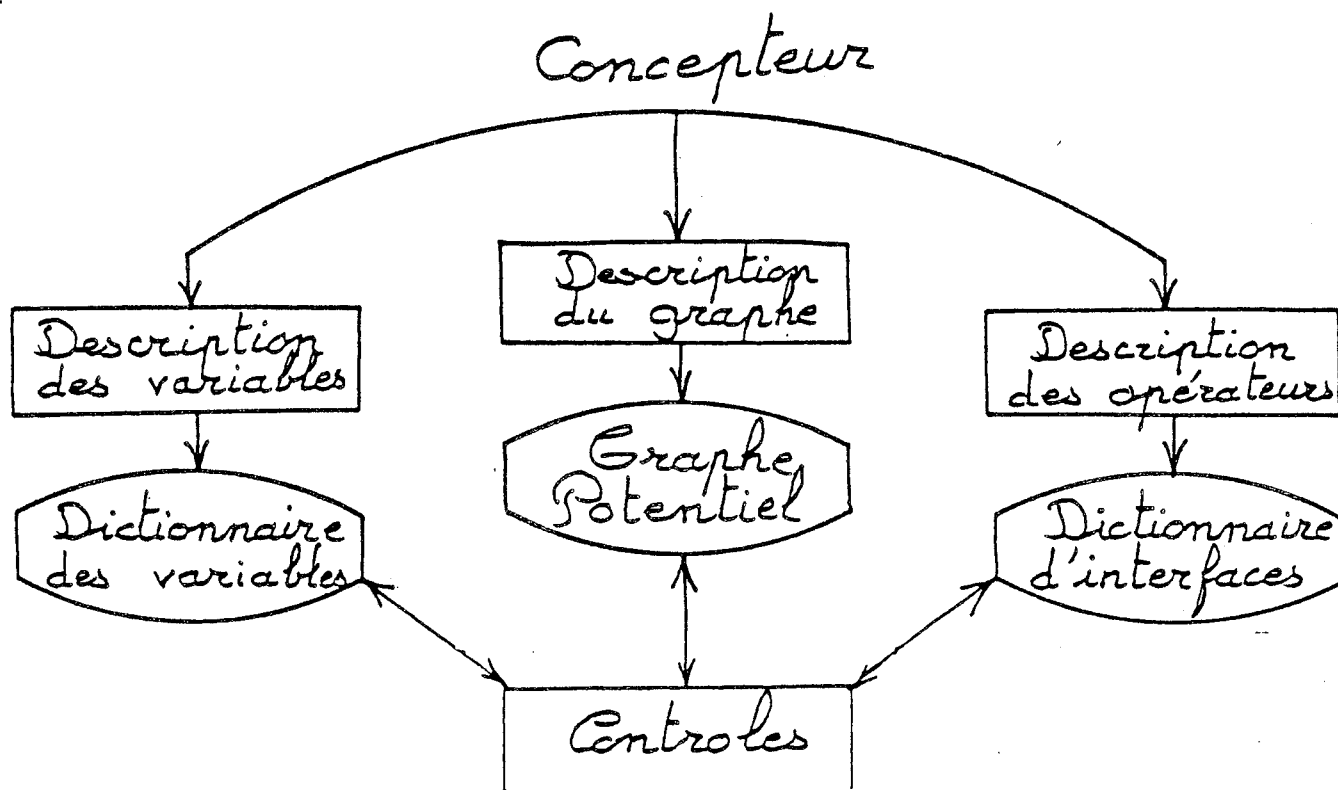


Figure 6 Le système générateur.

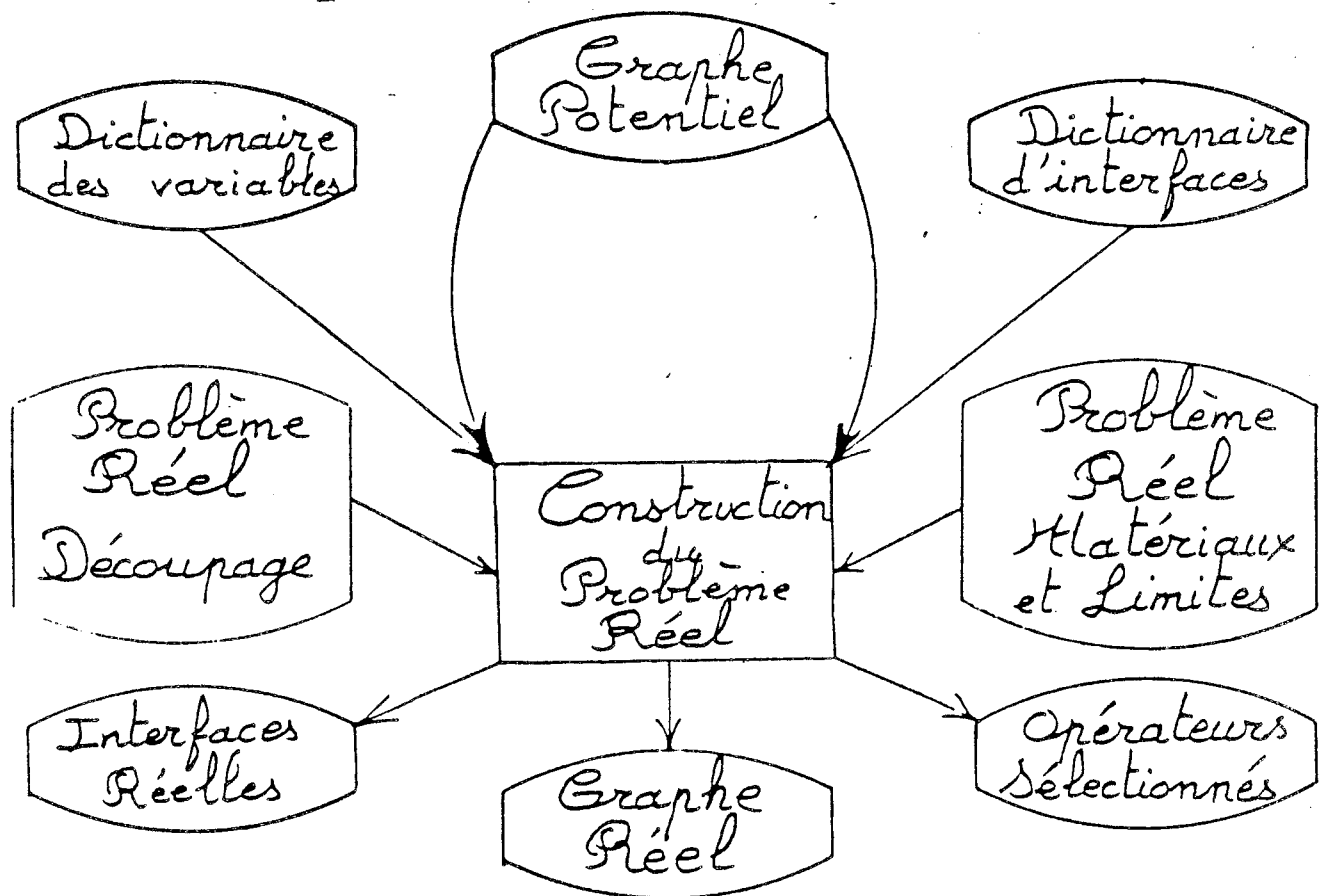


Figure 7 Construction du problème réel.

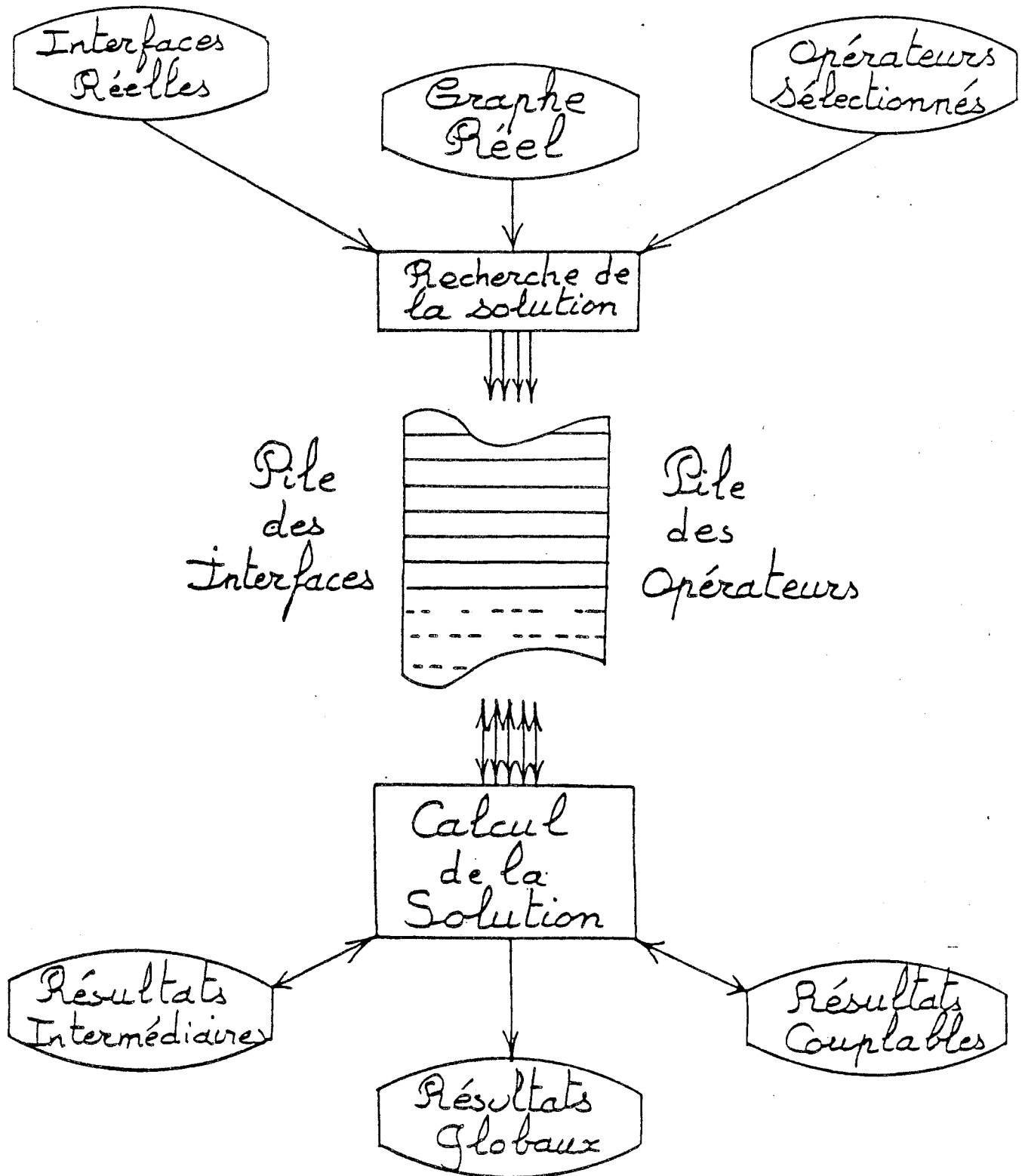


Figure 8 Résolution du problème réel.

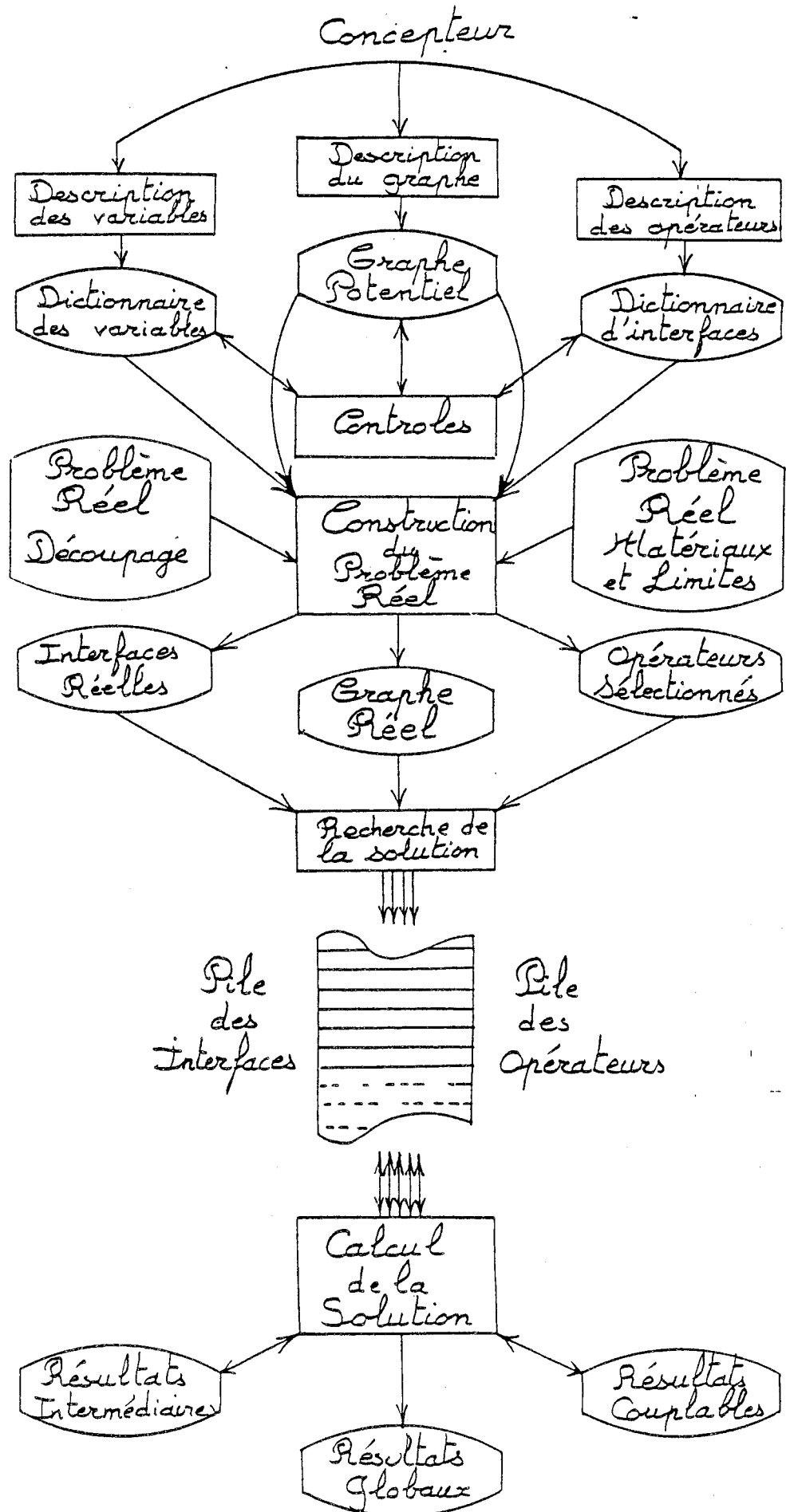


Figure 9 Structure globale du système.

3ÈME PARTIE

FLUX EXPERT UN PROTOTYPE DE SYSTÈME EXPERT FONDÉ SUR LA MÉTHODE DES ÉLÉMENTS FINIS

* * *

INSTITUT IMAG
Informatique, Mécanique Appliquée et Recherche
CNRS-INPG-UMG
MÉDIATHÈQUE
B.P. 53 X
38041 GRENOBLE CEDEX
FRANCE
Tél. 76.51.46,36

C.I INTRODUCTION .-

L'écriture d'un système expert est une vaste entreprise. Avant de démarrer cette tâche nous en avons évalué l'ampleur, et, devant la difficulté, nous avons décidé de forger une méthode de travail et des outils généraux qui permettent la construction d'un système fiable, facile à mettre au point puis à maintenir.

Le découpage logique en programmes indépendants dont chacun correspond à une phase du travail de l'utilisateur est apparu comme une nécessité essentielle. En effet, les phases ainsi bien délimitées ont des caractéristiques informatiques (temps d'unité centrale et taille de la mémoire utilisée) très disparates, de même les interactions homme-machine sont très particularisées : certaines tâches utilisent avant tout le graphique (découpage) d'autres exclusivement l'alphanumérique (générateur), d'autres enfin peu, voire pas, d'interaction (résolution). Toutes les parties de notre système sont donc autonomes au point de pouvoir être éventuellement traitées par des machines totalement différentes : a titre d'exemple les programmes générateurs ont été écrits et testés sur miniordinateur, alors que les autres parties l'ont été sur grosse machine.

Cette totale indépendance nous a conduit à définir des structures de bases de données assez particulières sur lesquelles nous reviendrons en détail. De même, les outils généraux que tous ces programmes utilisent ont été conçus en fonction de la portabilité imposée par ~~cette~~ ^{leur} implantation sur plusieurs machines.

Malgré la multiplicité des phases de travail, une unité de mode d'interaction était nécessaire afin que

l'utilisateur ait besoin d'un minimum de connaissance du système pour l'utiliser et pour que l'apprentissage réalisé au cours d'une phase soit utile lors des suivantes. C'est pourquoi les structures des dialogues et des langages sont communes à toutes les phases et toute interaction est auto-documentée.

Les programmes qui ont été écrits ont une structure très modulaire. L'outil de base du fonctionnement de ce système est la notion de bibliothèque d'algorithmes opérateurs ; c'est pourquoi beaucoup de travail a été réalisé dans la gestion des bibliothèques. Ceci nous a imposé une programmation très structurée avec un système de documentation et d'aide à la mise au point automatique qui s'est avéré bien utile.

Le système a été programmé en FORTRAN à cause de la grande efficacité de ce langage à l'exécution et également à cause de sa très large diffusion qui en fait un langage presque universellement supporté par les machines. Certes de nombreux outils ont été nécessaires pour augmenter la souplesse de programmation (réservation dynamique de variables, appels dynamiques de programmes, gestion de textes, ...) nous reviendrons sur ces aspects techniques dans les prochains chapitres.

Bien sûr un langage tel que ADA aurait résolu par lui-même de nombreux problèmes de réalisation auxquels nous nous sommes heurtés, mais d'une part ce langage n'existait pas lors du démarrage de ce projet et, d'autre part, à l'heure actuelle, les compilateurs ADA sont si peu nombreux qu'il nous aurait été impossible de mener à bien la construction de ce prototype.

C.II STRUCTURE GENERALE DU SYSTEME .-

Nous allons décrire le système suivant deux éclairages distincts.

Tout d'abord celui de l'utilisateur ce qui nous permettra de présenter brièvement le mode d'emploi de notre système.

Dans un deuxième temps, nous analyserons la structure des programmes qui le constituent.

C.II.1 LES PHASES AUTONOMES DU SYSTEME

phases destinées aux concepteurs des systèmes objets

Le système expert est composé de six tâches correspondant chacune à un programme indépendant. La tâche génération qui permet la description des équations à résoudre et le choix des techniques de traitement de ces équations.

phases destinées aux utilisateurs de ces systèmes

La tâche de discrétisation géométrique qui traite la description de l'objet à étudier et permet son découpage en éléments finis.

La tâche de modélisation des propriétés qui prend à sa charge la description des modèles des propriétés physiques des matériaux qui peuvent constituer tous les objets à étudier.

La tâche de définition d'un problème qui affecte à un découpage les conditions aux limites et les modèles des propriétés des matériaux composant l'objet à étudier.

La tâche de résolution qui calcule les valeurs nodales des inconnues.

La tâche d'exploitation qui évalue les grandeurs ponctuelles ou intégrales à partir des valeurs nodales et présente les résultats à l'utilisateur.

Chacune de ces phases est autonome et peut être implantée sur une machine qui lui est propre. Une base de données mémorise le travail achevé dans chaque phase ce qui permet de constituer ainsi des bases de savoir faire utilisables par toutes les autres parties du système. Il est possible d'interrompre une tâche à tout instant et de la poursuivre ultérieurement, mais seul un travail achevé par une phase peut être traité par les suivantes.

Dans l'utilisation de chaque phase on peut mettre en évidence des étapes logiques de travail dont nous allons présenter le détail :

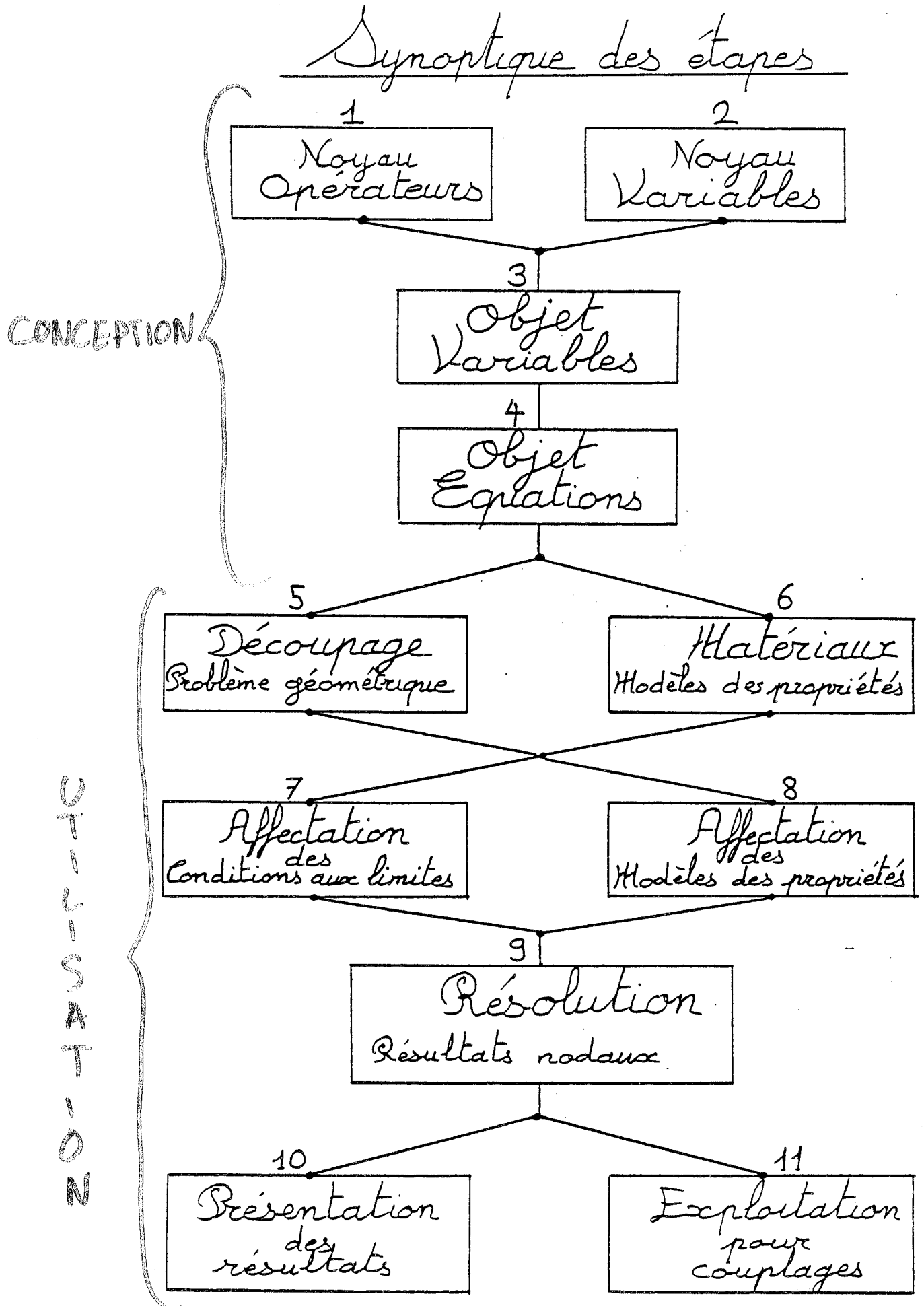


Figure 10 Les étapes du système.

C.II.2 LES ÉTAPES SUCCESSIVES DE LA CONSTRUCTION D'UN SYSTÈME

C.II.2.1 Les étapes de la phase générateur

- Etape 1

Elle consiste en la description des algorithmes dynamiques du système. Chacun de ces algorithmes constitue la matérialisation des opérateurs que nous avons présentés dans la deuxième partie :

- . Calcul des abscisses et poids d'intégration
- . Calcul des interpolations polynomiales
- . Calcul des propriétés
- . Calcul des intégrants
- . Constitution des matrices
- . Construction du système
- . Résolution de système linéaire

Réunis dans une même bibliothèque, tous les programmes travaillant sur une même classe d'opérateur possèdent la même structure d'interface d'entrée-sortie. Le nombre de programmes de chaque groupe est indéterminé : le concepteur d'un système peut programmer autant de nouveaux opérateurs qu'il le désire. Il lui suffit de se conformer à la structure de l'interface d'entrée associée. Le programme générateur permet la gestion conversationnelle de toutes les classes d'opérateurs. Chaque programme d'une classe doit être écrit en fortran absolument classique sans aucune contrainte ; il doit être compilé séparément du reste du système qui demeure totalement inchangé. C'est le superviseur de résolution qui se chargera de l'appel dynamique d'un opérateur particulier qu'il aura sélectionné.

Au cours de cette phase de description, l'utilisateur doit répondre aux questions posées par le système afin de définir chacun des opérateurs utilisés. Ce mode de travail limite le risque d'erreur lors de la déclaration d'une interface.

Le générateur analyse autant que possible la cohérence des paramètres qui lui sont fournis et indique le résultat de ses tests. Le travail réalisé peut être visualisé, modifié, sauvegardé à tout instant.

Les résultats de cette description constituent une base de données dynamiques appelée "NOYAU-OPERATEURS".

Ce travail est effectué une fois pour toutes et lors de la construction d'un système objet, il n'est généralement pas à reprendre. Seuls des compléments peuvent être ajoutés si un opérateur nécessaire à un problème n'a pas encore été créé dans la base Noyau-Opérateurs.

- Etape 2

Elle consiste en la description de toutes les variables physiques utilisables par tous les systèmes que l'on envisage d'écrire à l'aide du générateur.

Cette phase globale de déclaration de variables permet d'éviter des homonymies lors de couplages entre systèmes.

L'ensemble de ces variables constitue la base de données "NOYAU-VARIABLES". Comme pour la base Noyau-Opérateurs seuls quelques compléments seront à lui apporter lors de la construction d'un système mettant en oeuvre des variables encore non utilisées par les systèmes précédents.

Remarques

C'est à l'aide des variables et des opérateurs définis dans le noyau que l'utilisateur va pouvoir décrire le système qu'il veut construire.

Le concepteur Les étapes 1 et 2 sont totalement disjointes. L'utilisateur peut les aborder dans l'ordre de son choix ou même

les imbriquer. Lors des phases suivantes, il ne pourra reprendre les étapes 1 ou 2 que pour ajouter des déclarations qu'il aurait oubliées, car toute modification ou suppression dans le noyau annule les travaux réalisés dans les phases suivantes.

- Etape 3

Elle consiste en la déclaration des variables utilisées par le système objet en cours de construction.

Ces variables doivent déjà appartenir à la base de données "Noyau-variables".

De plus l'utilisateur va devoir définir :

- . les inconnues de son système objet,
- . la dimension de l'espace étudié (1, 2 ou 3),
- . le type de coordonnées utilisées cartésiennes ou axisymétriques
- . le type de polynômes d'interpolation (degré des approximations pour les variables),
- . enfin, pour chacune des propriétés dont il aura besoin pour décrire ses équations, l'utilisateur doit définir les types de modèles qu'il sera possible d'utiliser pour résoudre les divers problèmes :
 - valeurs constantes
 - valeurs tabulées globales
 - valeurs tabulées par élément
 - valeurs tabulées par région
 - valeurs calculées par un opérateur de la classe des calculs de propriétés.

Cette dernière possibilité permet de fixer diverses façons de calculer une même propriété : en effet, plusieurs types de courbes peuvent modéliser un même phénomène. Par exemple :

- . une formule analytique
- . une interpolation par splines
- . une approximation par moindres carrés
- . etc...

L'ensemble de ces déclarations forme une base de données appelée : "OBJET-VARIABLES".

- Etape 4

Après les étapes précédemment décrites toutes les variables nécessaires pour définir les équations sont disponibles. ~~L'utilisateur~~ ^{Le concepteur} devra tout d'abord déclarer chacun des intégrants élémentaires en lui donnant un nom et en choisissant l'opérateur associé à chacun des cas de calcul. Nous rappelons qu'un même intégrant peut être calculé par des opérateurs différents selon le type de la propriété discriminante qui la caractérise (cf § ~~2ème~~ ^{2ème} partie).

Enfin, il lui faudra fournir la composition de chacune des matrices constitutives nécessaires aux méthodes de construction qu'il aura choisies. En effet, la connaissance d'un opérateur de construction et la constitution de chacune des matrices de cet opérateur définissent ~~entièrement~~ ^{entièrement} l'équation à résoudre.

Cette phase est donc particulièrement simple, mais elle est très éloignée de l'écriture usuelle d'une équation discrétisée sous forme d'une formule mettant en oeuvre la notation classique : $\iiint A_i . A_j \sigma dx dy dz + \dots + \dots = \iiint \dots$. C'est pourquoi une fois la description formelle achevée, l'utilisateur peut faire visualiser par le système l'équation qu'il a décrite : un programme spécial de mise au format reconstitue la formule de l'équation dans la notation classique.

Remarque

par le concepteur

Une fois les étapes 1,2,3,4 achevées, l'utilisateur dispose d'un "système objet" complet qu'il va pouvoir utiliser. La base de données "OBJET-SYSTEME" est la réunion des 4 bases : NOYAU-OPERATEURS, NOYAU-VARIABLES, OBJET-VARIABLES, OBJET-EQUATION.

C.II.2.2 La tâche de discrétisation géométrique

Nous ne détaillerons pas ici les phases internes d'un découpage en éléments finis. En effet, notre travail de recherche n'a pas du tout porté sur ce domaine qui est presque totalement indépendant des équations qui vont traiter la discrétisation géométrique et topologique obtenue. Seul le degré des polynômes et le type des éléments fournis par la discrétisation doivent être compatibles avec les options choisies pour le système objet.

Afin de pouvoir connecter divers programmes de découpage à notre système, nous avons défini avec précision la structure de la base de données dont notre programme a besoin pour pouvoir travailler : ("PROBLEME-GEOMETRIE").

Remarque

Outre les informations géométriques (coordonnées des noeuds) et topologiques classiques (numéro des noeuds de chaque élément) notre système doit pouvoir avoir accès à la notion de région.

Une région est un ensemble d'éléments (non nécessairement connexe) qui possèdent les mêmes propriétés physiques. Cette définition est valable pour des éléments finis linéiques surfaciques et volumiques.

Cette notion de région est utilisée à la fois dans le

calcul des intégrants et dans l'élaboration des conditions aux limites.

Nous disposons d'un découpeur 2 Dimensions déjà opérationnel (le système ENTREE de FLUX2D) c'est pourquoi nous avons écrit les programmes assurant la compatibilité avec notre système et ce découpeur. Dès que nous pourrons disposer d'un découpeur 3 Dimensions, nous serons en mesure de tester notre système sur cette nouvelle classe de problèmes avec plus de souplesse.

C.II.2.3 La tâche de modélisation des propriétés

Une seule étape constitue cette phase. Elle correspond au besoin suivant : lorsqu'un système objet vient d'être construit il peut être utilisé pour étudier une très grande diversité de problèmes ; mais assez souvent les problèmes traités peuvent être regroupés par classes de similitude (par exemple lors de l'analyse d'un nouveau prototype industriel).

Plusieurs problèmes concrets similaires vont donc souvent être traités successivement. Il serait fastidieux de devoir redéfinir pour chaque problème des données immuables telles les propriétés physiques des matériaux disponibles pour la construction de ces prototypes.

~~Programme~~
Notre ~~prototype~~ permet de définir tous les modèles dont on peut avoir besoin au cours des problèmes d'une même classe, exemples :

- Courbe de saturation magnétique des aciers classiques
- Variation de la capacité calorifique du cuivre en fonction de la température
- Valeur de la perméabilité magnétique du vide
- etc ...

L'utilisateur devra déclarer tous les modèles dont il pourra avoir besoin, en leur donnant un nom et en précisant la façon dont ils seront calculés.

- Le modèle "valeur nulle" est prédéfini.

- Dans le cas de valeurs tabulées (globales élémentaires ou régionales) l'utilisateur doit fournir le nom du fichier contenant ces valeurs. Lorsque plusieurs systèmes objets sont couplés c'est le traitement d'un problème précédent qui doit avoir construit ces fichiers.

- Dans le cas de valeurs calculées par les opérateurs de "calcul de propriétés", l'utilisateur doit fournir pour chacun des modèles, le nom de l'opérateur qui calcule la propriété et l'interface nécessaire à cette fonction (paramètre des formules analytiques, coordonnées des points d'interpolation pour les courbes, etc ...). On peut remarquer qu'une même fonction peut servir pour des modèles différents.

L'ensemble des modèles définis constitue une base de données appelée "MODELES-MATERIAUX" qui peut être sauvegardée, restaurée, augmentée, visualisée à volonté.

C.II.2.4 La tâche de définition d'un problème réel

Les étapes accomplies précédemment permettent de disposer d'un système objet complet, d'une banque de modèles de propriétés de divers matériaux, et d'un sujet d'étude découpé en éléments finis. Tout est prêt pour décrire le problème à résoudre.

- Etape 7

Elle permet d'affecter les conditions aux limites.

Seules les conditions de type Dirichlet doivent être définies à ce niveau ; en effet, les conditions de Neumann non homogènes sont prises en compte dans l'équation elle-même (intégrales linéïques en 2D ou de surface en 3D).

Sur chacune des régions de la frontière (surfactive ou linéïque) l'utilisateur doit choisir le type de conditions aux limites parmi les possibilités suivantes :

- | | |
|-----------------|--------------------------------|
| - Cycliques | conditions topologiques |
| - Anticycliques | |
| - Translation | (topologie + valeur numérique) |
| - Dirichlet | (valeurs numériques) |

Pour les conditions qui nécessitent une valeur numérique (translation et Dirichlet) si cette grandeur est constante (indépendante du temps), l'utilisateur doit fournir la valeur correspondante, mais dans le cas où les conditions aux limites dépendent du temps c'est le nom du modèle qui les calcule qui doit être fourni. Ce modèle doit bien sûr avoir été défini au préalable dans la banque "Modèle-Matériaux".

- Etape 8

Pour chacune des régions, l'utilisateur doit fournir le modèle décrivant chacune des propriétés du matériau qui la constitue. Ceci doit être réalisé aussi bien pour les régions internes que pour les régions aux limites comportant des conditions de Neumann non homogènes.

Une fois les étapes 7 et 8 achevées, l'utilisateur dispose de deux bases de données globales qui résument son travail :

- . la base OBJET-SYSTEME déjà décrite
- . la base PROBLEME qui est composée

de la réunion des informations suivantes :

- : PROBLEME-GEOMETRIE
- : MODELE-MATERIAUX
- : CONDITIONS AUX LIMITES
- : NATURE DES REGIONS

C.II.2.5 Etape de la tâche de résolution

Une seule étape constitue cette phase à laquelle il suffit de transmettre les deux bases OBJET-SYSTEME et PROBLEME pour qu'elle dispose de toutes les informations nécessaires à la résolution du problème réel.

L'utilisateur de ce niveau a seulement la charge de contrôler la bonne marche de la résolution :

- choix de la stratégie
- choix des pas nécessaires
- choix des précisions souhaitées

Les programmes de ce niveau rajoutent aux deux bases de données précédentes la base "RESULTAT-NODAUX" ce qui permet ainsi un traitement différé par le dernier niveau.

C.II.2.6 Etapes de la phase d'exploitation

Disposant de la solution et de toutes les informations précédentes, les programmes de ce niveau ont pour charge de présenter les résultats à l'utilisateur.

On doit distinguer deux classes d'informations :

- . les grandeurs directes qui sont constituées par les valeurs nodales des inconnues et leurs interpolations polynomiales (de même pour les dérivées de ces valeurs)

- . Les grandeurs nécessitant un calcul supplémentaire (valeurs intégrales notamment). Cette phase de calcul supplémentaire devrait être configurable selon l'équation traitée donc devrait pouvoir être décrite au sein du "système OBJET". Nous n'avons pas eu le temps nécessaire pour aborder ce problème fort intéressant qui constitue un sujet de recherche dans le prolongement logique de notre travail. Dans notre prototype, seuls les programmes de visualisation de résultats bruts ont été réalisés (pour des géométries bidimensionnelles).

La phase de traitement comporte deux étapes logiques.

- Etape 10

Elle consiste en la visualisation des résultats ; pour l'instant les valeurs de chaque composante de l'inconnue peuvent être présentées en tout point du sujet étudié ; les lignes iso-valeurs peuvent être tracées, les valeurs maximales localisées.

Si le problème dépend du temps, l'utilisateur doit sélectionner au préalable l'échantillon de temps où il veut visualiser les résultats.

- Etape 11

Cette dernière phase est optionnelle et ne sert qu'aux problèmes couplés résolus par des systèmes objets distincts.

Les valeurs nodales calculées sont dans ce cas nécessaires à un autre système objet, soit sous forme de valeurs tabulées globales, soit régionales, soit élémentaires.

L'utilisateur doit alors extraire les valeurs nécessaires et constituer les fichiers correspondants.

Ces fichiers seront lus lors de l'étape (6) de définition

des modèles du problème couplé correspondant (qui portera donc nécessairement sur le même découpage à moins que l'utilisateur ait lui-même réalisé les interpolations multi-grilles correspondantes. Il sera possible d'ajouter à notre système les outils nécessaires au traitement automatique des interpolations multi-grilles).

C.II.3 STRUCTURE DES INTERACTIONS UTILISATEUR-SYSTÈME

Afin de simplifier l'utilisation de nos programmes, la structure des interactions est homogène tout au long des étapes que nous avons présentées.

Seule l'étape de découpage échappe à cette règle car elle est totalement indépendante de notre système.

Pour chacune des phases la structure du dialogue est arborescente : à chaque branche de l'arbre le menu des branches inférieures est proposé.

Une clef particulière permet de remonter au niveau de conversation de son choix ceci qu'elle que soit l'interaction en cours.

La structure des dialogues n'est pas évolutive mais tous les mots clefs sont dynamiques et sont définis implicitement par l'utilisateur. Ceci est réalisé grâce à un principe fort simple : les termes utilisés dans chacune des phases sont ceux qui ont été définis par l'utilisateur dans les déclarations qu'il a réalisées au cours des étapes précédentes. Il est donc possible d'adapter très facilement le contenu et la forme des dialogues aux besoins spécifiques de chaque système objet.

A chaque interaction un contrôle de la correction des paramètres est effectué ; si une erreur est détectée, l'utilisateur en est informé par un message. Dans la grande majorité

des cas la question est reposée et l'utilisateur peut faire afficher l'ensemble des réponses correctes autorisées, si l'erreur commise est plus grave l'action en cours est abandonnée et l'utilisateur doit reformuler sa commande.

La description d'un système objet et son utilisation pour résoudre des problèmes est essentiellement interactive mais la complexité des travaux à réaliser fait qu'il est très délicat de décrire un système sans une préparation préalable. Pour faciliter ce travail, nous avons écrit des programmes de visualisation très détaillés pour les travaux effectués au cours de chaque étape. L'utilisateur peut faire imprimer sur liste l'état des bases de données pour préparer le travail à effectuer dans les phases ultérieures.

Nous donnons au chapitre suivant plusieurs exemples de définition de systèmes objets. On pourra y consulter des exemples d'interactions entre notre système et l'utilisateur.

C.III EXEMPLES D'UTILISATION ET ANALYSE DES PERFORMANCES DU GENERATEUR .-

Nous allons donner trois exemples d'utilisation de notre système expert afin d'illustrer concrètement la méthode que nous avons présentée dans la deuxième partie de notre mémoire.

Le premier exemple concerne la description du système déjà existant FLUX2D à l'aide notre système. Cette étude nous a permis, d'une part, de tester notre prototype et d'autre part d'évaluer ses performances par rapport à un système industriel déjà optimisé.

Le deuxième exemple choisi : le traitement des phénomènes de diffusion thermique en géométrie bidimensionnelle et

axisymétrique, permet d'illustrer la description d'un problème mettant en oeuvre des propriétés non linéaires, des conditions aux limites assez variées et des études de régimes transitoires.

Le troisième exemple traite l'équation de Navier-Stokes en formulation u, v, p dans un repère cartésien bidimensionnel et met en évidence l'utilisation d'une variable vectorielle qui conduit à un système couplé dissymétrique.

C.III.1 COMPARAISON AVEC LE SYSTÈME FLUX2D

Le système FLUX2D [58] [56] offre à ses utilisateurs le traitement de plusieurs types d'équations : les problèmes magnéto-statiques à matériaux non linéaires, magnétodynamiques à matériaux linéaires, en coordonnées bidimensionnelles cartésiennes ou axisymétriques.

A l'aide de notre système, nous avons pu reconstituer chacun des cas traités par le système FLUX2D, mais, nous ne présenterons ici que le cas magnétodynamique en coordonnées axisymétriques, afin de ne pas alourdir cette illustration.

Nous avons déjà présenté l'analyse théorique de l'équation magnétodynamique au chapitre (III) (1ère partie). Nous reprenons ici directement l'équation discrétisée que nous avons obtenue en coordonnées axisymétriques.

$$\left\{ \int \omega \iint_{\Omega} \sigma \alpha_i \alpha_j \frac{dr dz}{r} + \iint_{\Omega} \nu \nabla \alpha_i \cdot \nabla \alpha_j \frac{dr dz}{r} \right\} \bar{A} = \iint_{\Omega} \alpha_i J R dr dz + \int_{\Gamma} \alpha_i H T R d\Gamma + \int \left(\iint_{\Omega} \alpha_i J I dr dz + \int_{\Gamma} \alpha_i H T I d\Gamma \right)$$

Nous devons tout d'abord choisir les identificateurs des diverses variables que nous allons utiliser.

l'inconnue : AV	: A_0 composante du potentiel vecteur
	NU : inverse de la perméabilité
	SIG : conductibilité
les	DJR : partie réelle des sources
propriétés	DJI : partie imaginaire des sources
	HTR : champ extérieur partie réelle
	HTI : champ extérieur partie imaginaire

Pour chaque propriété, il faut définir les types de modèles qui pourront être utilisés pour décrire des matériaux composant des problèmes réels. On peut choisir les cas retenus parmi les possibilités ci-dessous :

- . Valeur constante
- . Valeur tabulée en tous les noeuds du domaine
- . Valeur tabulée sur chaque élément du domaine
- . Valeur tabulée par région
- . Valeur calculée par un opérateur propriété

Dans ce dernier cas on doit alors spécifier le nombre de fonctions distinctes que l'on pourra utiliser et pour chacune il faut fournir le nom de l'opérateur de calcul de propriété qui effectuera le calcul (cet opérateur doit bien sûr exister dans la base NOYAU-OPERATEUR).

Dans l'exemple décrit, nous avons retenu les types constants ou tabulés pour toutes les propriétés et nous n'avons pas utilisé les valeurs calculées par un opérateur propriété.

Il faut ensuite spécifier :

- . le type de l'inconnue : COMPLEXE
- . la dimension de l'espace d'étude : 2
- . le type de coordonnées utilisées : AXISYMETRIQUE
- . les polynomes utilisés : ISOPARAMETRIQUE de degré 2

En effet puisque nous avons choisi des fonctions de projection égales aux fonctions d'interpolation, un seul type de polynôme est nécessaire (nous l'avons pris de degré 2 pour être conforme au système FLUX2D).

Nous abordons alors la phase de description des intégrants. Nous rappelons que nous avons six intégrants distincts qui composent notre équation. Pour chacun de ces intégrants, nous devons choisir un identificateur :

$$\begin{aligned}
 \cdot TE &= \left[\iint_{\Omega} \sigma \alpha_i \alpha_j \frac{dr dz}{r} \right] \\
 \cdot SE &= \left[\iint_{\Omega} \nu \nabla \alpha_i \cdot \nabla \alpha_j \frac{dr dz}{r} \right] \\
 \cdot VER &= \left[\iint_{\Omega} \alpha_i JR dr dz \right] \\
 \cdot VEI &= \left[\iint_{\Omega} \alpha_i JI dr dz \right] \\
 \cdot LER &= \left[\int_{\Gamma} \alpha_i HTR d\Gamma \right] \\
 \cdot LEI &= \left[\int_{\Gamma} \alpha_i HTI d\Gamma \right]
 \end{aligned}$$

Nous rappelons qu'un intégrant peut être calculé par des opérateurs différents selon le type de l'une des caractéristiques qui interviennent dans sa formule : la caractéristique discriminante.

Notre système permet en outre de fournir des opérateurs de calcul d'intégrants différents selon que la caractéristique discriminante est variable sur l'élément ou constante (car dans ce dernier cas la propriété correspondante peut être sortie du calcul de l'intégrale ce qui permet d'optimiser le nombre d'opérations.)

Pour chaque intégrant et selon le type de sa caractéristique discriminante il faut décrire la formule correspondante à l'aide de la combinaison de deux opérateurs choisis dans le NOYAU-OPERATEURS.

L'un parmi les opérateurs EXPRESSION-INTEGRANTE.

L'autre parmi les opérateurs POIDS-DIFFERENTIEL.

Pour ce faire il faut donc consulter au préalable la base NOYAU-OPERATEUR.

Dans le cas présent, nous avons sélectionné parmi tous les opérateurs ~~présents~~ ^{programmés} les cas ci-dessous :

Expressions intégrantes

INAASC	calculant	$P1 \iint \alpha_i \alpha_j dG$
INAASV	"	$\iint \alpha_i \alpha_j dG$
INGGSC	"	$P1 \iint \nabla \alpha_i \cdot \nabla \alpha_j dG$
INGGSV	"	$\iint \nabla \alpha_i \cdot \nabla \alpha_j dG$
INTAC	"	$P1 \iint \alpha_i dG$
INTAV	"	$\iint \alpha_i dG$

Poids différentiel

POIAXV	: $dG = \frac{drdz}{r}$
POAVV1	: $dG = P3 \frac{drdz}{r}$
POICV1	: $dG = P3 drdz$
POIVID	: $dG = drdz$

Ce qui permet de définir :

$$TE = INAASC \otimes POIAXV = P1 \iint \alpha_i \alpha_j \otimes \frac{drdz}{r} = SIG \iint \alpha_i \alpha_j \frac{drdz}{r}$$

avec $P1 = SIG_{cst}$

ou bien

$$TE = INAASV \otimes POAVV1 = \iint \alpha_i \alpha_j \otimes P3 \frac{drdz}{r} = \iint \alpha_i \alpha_j SIG \frac{drdz}{r}$$

avec $P3 = SIG_{var}$

$$\text{SE} = \text{INGGSC} \otimes \text{POIAXV} = P1 \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j \otimes \frac{dr dz}{r} = NU \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j \frac{dr dz}{r}$$

avec $P1 = NU_{cst}$

ou bien

$$\text{SE.} = \text{INGGSV} \otimes \text{POAVV1} = \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j \otimes P3 \frac{dr dz}{r} = \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j NU \frac{dr dz}{r}$$

avec $P3 = NU_{var}$

$$\text{VER} = \text{INTAC} \otimes \text{POIVID} = P1 \iint_{\Omega} \alpha_i \otimes dr dz = DJR \iint_{\Omega} \alpha_i dr dz$$

avec $P1 = DJR_{cst}$

ou bien

$$\text{VER} = \text{INTAV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \otimes P3 dr dz = \iint_{\Omega} \alpha_i DJR dr dz$$

avec $P3 = DJR_{var}$

$$\text{VEI} = \text{INTAC} \otimes \text{POIVID} = P1 \iint_{\Omega} \alpha_i \otimes dr dz = DJI \iint_{\Omega} \alpha_i dr dz$$

avec $P1 = DJI_{cst}$

ou bien

$$\text{VEI} = \text{INTAV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \otimes P3 dr dz = \iint_{\Omega} \alpha_i DJI dr dz$$

avec $P3 = DJI_{var}$

$$\text{LER} = \text{INTAC} \otimes \text{POIVID} = P1 \int_{\Gamma} \alpha_i \otimes d\Gamma = HTR \int_{\Gamma} \alpha_i d\Gamma$$

avec $P1 = HTR_{cst}$

ou bien

$$\text{LER} = \text{INTAV} \otimes \text{POICV1} = \int_{\Gamma} \alpha_i \otimes P3 d\Gamma = \int_{\Gamma} \alpha_i HTR d\Gamma$$

avec $P3 = HTR_{var}$

$$\text{LEI} = \text{INTAC} \otimes \text{POIVID} = P1 \int_{\Gamma} \alpha_i \otimes d\Gamma = HTI \int_{\Gamma} \alpha_i d\Gamma$$

avec $P1 = HTI_{cst}$

ou bien

$$\text{LEI} = \text{INTAV} \otimes \text{POICV1} = \int_{\Gamma} \alpha_i \otimes P3 d\Gamma = \int_{\Gamma} \alpha_i HTI d\Gamma$$

avec $P3 = HTI_{var}$

Après la description de chacun des intégrants, il faut définir les matrices constitutives nécessaires aux opérateurs de construction de problèmes potentiels que l'on désire utiliser.

En ce qui concerne le traitement des inconnues de type complexe, une seule méthode de construction de problèmes est

disponible dans notre système :

l'opérateur CSLIDY qui construit le problème suivant :

$$\{ j\omega[L] + [M] \} \dot{I} = K_S + j K_V$$

où L , M , K_S et K_V sont les matrices et vecteurs constitutifs déjà présentés au cours de la deuxième partie de notre mémoire.

Compte-tenu de l'équation discrétisée, nous déduisons immédiatement que :

$$L = TE$$

$$M = SE$$

$$K_S = VER + LER$$

$$K_V = VEI + LEI$$

Ces formules sont décrites conversationnellement à l'aide de notre système qui demande tout d'abord le nombre d'intégrants composant chacune des matrices constitutives, puis les noms de ces intégrants.

A l'issue de cette phase l'équation à résoudre est totalement décrite, il est alors possible de traiter des problèmes effectifs en utilisant le système objet ainsi créé.

Remarque

La description d'une telle équation à l'aide de FLUX EXPERT demande un maximum d'une heure de travail sur un terminal alphanumérique, le coût de cette phase étant extrêmement faible (inférieur à quelques minutes d'utilisation d'unité centrale).

L'utilisation de ce système objet pour résoudre un problème réel est présentée dans l'annexe (1) où nous décrirons le calcul d'un champ magnétique dans un problème de chauffage par induction.

C.III.2 UTILISATION DE FLUX EXPERT POUR RÉSOUDRE UN PROBLÈME THERMODYNAMIQUE TRANSITOIRE

Dans cet exemple nous reprenons l'équation de la diffusion de la chaleur que nous avons déjà traitée au chapitre (IV) de la 1ère partie :

$$\begin{aligned} & \cdot \iint_{\Omega} \beta_i (\rho C)_n T_n - \iint_{\Omega} \beta_i (\rho C)_n T_{n-1} + \left\{ \iint_{\Omega} \beta_i (\rho C)_n V \cdot \nabla T_n + \right. \\ & \cdot \iint_{\Omega} \nabla \beta_i \cdot (K_n \nabla T_n) + \int_{\Gamma} \beta_i h_n T_n + \int_{\Gamma} \beta_i (\epsilon \sigma)_n T_n^4 - \\ & \cdot \left. \iint_{\Omega} \beta_i q - \int_{\Gamma} \beta_i (h)_n T_{an} - \int_{\Gamma} \beta_i (\epsilon \sigma)_n T_{an}^4 \right\} \Delta t = 0 \end{aligned}$$

Pour décrire cette équation, nous devons tout d'abord choisir les identificateurs des variables que nous allons utiliser :

l'inconnue :

- TN : température au pas N
- TP : température au pas N-1

les propriétés :

- RC : valeur de ρC
- V : vitesse des fluides ou des pièces en mouvement (vecteur à 2 composantes)
- K : conductibilité thermique réelle ou tensorielle
- H : coefficient d'échange convectif de surface
- ES : $\epsilon \sigma$ coefficient d'échange rayonnant des surfaces
- QS : densité des sources thermiques statiques
- QV : densité des sources thermiques évolutives
- TA : température de l'ambiante au voisinage des surfaces

Certaines dérivées sont nécessaires pour évaluer la matrice Jacobienne de l'équation non linéaire obtenue :

DRC : dérivée de RC(T)

DK : dérivée de K(T) valeurs
réelles ou tensorielles

Nous avons supposé que les propriétés H et ES étaient indépendantes de la température.

*Nous pouvons remarquer que les propriétés :
RC, V, K, Q concernent des éléments de surface
alors que :
H, ES, TA concernent les valeurs aux limites donc des éléments
linéïques.*

Pour traiter ces propriétés lors de la résolution de problèmes réels nous avons défini pour chacune la possibilité d'utiliser des valeurs constantes, tabulées, ou des fonctions. Dans ce dernier cas 2 modèles ont été définis :

LINEAIRE approximation par une formule du type $A + BT$

QUADRATIQUE approximation par une formule du type
 $A + BT + CT^2$.

Après avoir décrit les variables il faut choisir les options générales :

le type de l'inconnue : REEL

la dimension de l'espace d'étude : 2

le type de coordonnées : CARTESIEN

les polynômes utilisés : ISOPARAMETRIQUES de degré 2

Nous avons choisi les mêmes polynômes pour les fonctions de projection β_i et les fonctions d'approximation α_j .

Il faut alors décrire chacun des intégrants nécessaires au calcul de l'équation et à l'évaluation de la matrice

Jacobienne :

$$\begin{aligned}
 TES &= \left[\iint_{\Omega} RC \alpha_i \alpha_j dx dy \right] \\
 DTES &= \left[\iint_{\Omega} TN \left(\frac{\partial RC}{\partial T} \right)_n \alpha_i \alpha_j dx dy \right] \\
 DTEO &= \left[\iint_{\Omega} TP \left(\frac{\partial RC}{\partial T} \right)_n \alpha_i \alpha_j dx dy \right] \\
 CE &= \left[\iint_{\Omega} RC \alpha_i V. \nabla \alpha_j dx dy \right] \\
 DCE &= \left[\iint_{\Omega} V. \nabla TN \left(\frac{\partial RC}{\partial T} \right)_n \alpha_i \alpha_j dx dy \right] \\
 SE &= \left[\iint_{\Omega} \nabla \alpha_i . (k \# \nabla \alpha_j) dx dy \right] \\
 DSE &= \left[\iint_{\Omega} \left(\left(\frac{\partial k}{\partial T} \right)_n \# \nabla TN \right) . \nabla \alpha_i \right) \alpha_j dx dy \right] \\
 CV &= \left[\int_{\Gamma} H. \alpha_i \alpha_j d\Gamma \right] \\
 RY1 &= \left[\int_{\Gamma} ES. TN^3 \alpha_i \alpha_j d\Gamma \right] \\
 VES &= \left[\iint_{\Omega} QS. \alpha_i dx dy \right] \\
 VEV &= \left[\iint_{\Omega} QV. \alpha_i dx dy \right] \\
 VEC &= \left[\int_{\Gamma} H. TA \alpha_i d\Gamma \right] \\
 VER &= \left[\int_{\Gamma} ES. TA^4 \alpha_i d\Gamma \right]
 \end{aligned}$$

Pour décrire ces intégrants il faut choisir les couples d'opérateurs EXPRESSION INTEGRANTE et POIDS DIFFERENTIELS qui peuvent les reconstituer. L'étude des opérateurs qui composent le NOYAU nous permet de sélectionner :

$$\begin{aligned}
 INAASV &= \iint_{\Omega} \alpha_i \alpha_j dG \\
 INAVGV &= \iint_{\Omega} \alpha_i \vec{P1} . \nabla \alpha_j dG \\
 INAAVV &= \iint_{\Omega} \alpha_i \alpha_j \vec{P1} . \vec{P2} dG
 \end{aligned}$$

$$\text{INGGTV} = \iint_{\Omega} \nabla \alpha_i \cdot (\nabla \alpha_j \otimes \overline{P1}) dG \quad \overline{P1} \text{ tenseur}$$

$$\text{INGGSV} = \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j dG$$

$$\text{INVGAV} = \iint_{\Omega} (\overline{P1} \cdot \nabla \alpha_i) \alpha_j dG$$

$$\text{INTAV} = \iint_{\Omega} \alpha_i dG$$

et les opérateurs de POIDS DIFFERENTIEL§ :

$$\text{POIVID} \quad dG = dx dy \text{ ou } dy$$

$$\text{POICV1} \quad dG = P3 \, dx dy \text{ ou } P3 \, dy$$

$$\text{POICV2} \quad dG = P3 \, P4 \, dx dy \text{ ou } P3 \, P4 \, dy$$

$$\text{POI3V2} \quad dG = (P3)^3 P4 \, dx dy \text{ ou } (P3)^3 P4 \, dy$$

$$\text{POI4V2} \quad dG = (P3)^3 P3 \, dx dy \text{ ou } (P3)^4 P4 \, dy$$

$$\text{POINV2} \quad dG = -P3 \, P4 \, dx dy \text{ ou } -P3 \, P4 \, dy$$

$$\text{POIN42} \quad dG = -(P3)^4 P4 \, dx dy \text{ ou } -(P3)^4 P4 \, dy$$

A l'aide de ces opérateurs nous pouvons décrire les intégrants nécessaires au calcul :

$$\text{TES} = \text{INAASV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \alpha_j \otimes P3 dx dy$$

avec $P3 = RC$ variable

$$\text{ou bien si } RC = \text{cst} \quad \text{INAASC} \otimes \text{POIVID}$$

$$\text{DTES} = \text{INAASV} \otimes \text{POICV2} = \iint_{\Omega} \alpha_i \alpha_j \otimes P3 P4 \, dx dy$$

avec $P3 = DRC$ variables, $P4 = TN$

$$\text{ou bien si } DRC = \text{cst} \quad \text{INAASC} \otimes \text{POICV1}$$

$$\text{DTEO} = \text{INAASV} \otimes \text{POICV2} = \iint_{\Omega} \alpha_i \alpha_j \otimes P3 P4 \, dx dy$$

avec $P3 = \text{DRC variable}$, $P4 = \text{TP}$
 ou bien si $\text{DRC} = \text{cst}$ $\text{INAVGC} \otimes \text{POICV1}$

$$\text{CE} = \text{INAVGV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \vec{P1} \cdot \nabla \alpha_j \otimes P3 \, dx dy$$

avec $\vec{P1} = V$ et $P3 = \text{RC variable}$
 ou bien si $\text{RC} = \text{cst}$ $\text{INAVGC} \otimes \text{POIVID}$

$$\text{DCE} = \text{INAAVV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \alpha_j \vec{P1} \cdot \vec{P2} \otimes P3 \, dx dy$$

avec $\vec{P1} = V$, $\vec{P2} = \nabla \text{TN}$, $P3 = \text{DRC variable}$
 ou bien si $\text{DRC} = \text{cst}$ $\text{INAAVV} \otimes \text{POICC1}$

$$\text{SE} = \text{INGGTV} \otimes \text{POIVID} = \iint_{\Omega} \nabla \alpha_i \cdot (\nabla \alpha_j \# \vec{P1}) \otimes dx dy$$

avec $\vec{P1} = K$ tenseur variable, produit tensoriel
 ou bien si $K = \text{cst}$ $\text{INGGTC} \otimes \text{POIVID}$
 ou bien pour K scalaire

$$\text{SE} = \text{INGGSV} \otimes \text{POICV1} = \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j P3 \otimes dx dy$$

avec $P3 = K$ scalaire variable
 ou bien si $K = \text{cst}$ $\text{INGGSC} \otimes \text{POIVID}$

$$\text{DSE} = \text{INDGTV} \otimes \text{POIVID} = \iint_{\Omega} ((\vec{P1} \# \vec{P2}) \cdot \nabla \alpha_i) \alpha_j \otimes dx dy$$

avec $\vec{P1} = DK$ tenseur variable, $\vec{P2} = \nabla \text{TN}$
 ou bien pour K scalaire

$$\text{DSE} = \text{INVGAV} \otimes \text{POICV1} = \iint_{\Omega} (\vec{P1} \cdot \nabla \alpha_i) \alpha_j \otimes P3 \, dx dy$$

avec $\vec{P1} = \nabla \text{TN}$ et $P3 = DK$ scalaire variable
 ou bien si $DK = \text{cst}$ $\text{INVGAC} \otimes \text{POIVID}$

$$\text{CV} = \text{INAASV} \otimes \text{POICV1} = \int_{\Gamma} \alpha_i \alpha_j \otimes P3 \, d\gamma$$

avec $P3 = H$ variable
 ou bien si $H = \text{cst}$ $\text{INVAASC} \otimes \text{POIVID}$

$$\text{DCV} = \text{INAASV} \otimes \text{POICV2} = \int_{\Gamma} \alpha_i \alpha_j \otimes P3 P4 \, d\gamma$$

avec $P3 = DH$ variable, $P4 = \text{TN}$
 ou bien si $DH = \text{cst}$ $\text{INAASC} \otimes \text{POICC2}$

$$\begin{aligned} \text{DRY} &= \text{INAASV} \otimes \text{POI3V2} = \int_{\Gamma} \alpha_i \alpha_j \otimes P3^3 P4 \, dy \\ &\text{avec } P3 = \text{TN et } P4 = \text{ES variable} \\ &\text{ou bien si } ES = \text{cst } \text{INAASC} \otimes \text{POI3V1} \end{aligned}$$

$$\begin{aligned} \text{VES} &= \text{INTAV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \otimes P3 \, dx \, dy \\ &\text{avec } P3 = \text{QS variable} \\ &\text{ou bien si } QS = \text{cst } \text{INTAC} \otimes \text{POIVID} \end{aligned}$$

$$\begin{aligned} \text{VEV} &= \text{INTAV} \otimes \text{POICV1} = \iint_{\Omega} \alpha_i \otimes P3 \, dx \, dy \\ &\text{avec } P3 = \text{QV variable} \\ &\text{ou bien si } QV = \text{cst } \text{INTAC} \otimes \text{POIVID} \end{aligned}$$

$$\begin{aligned} \text{VER} &= \text{INTAV} \otimes \text{PCPC4V} = \int_{\Gamma} \alpha_i \otimes P3^4 P4 \, dy \\ &\text{avec } P3 = \text{TA et } P4 = \text{ES variable} \\ &\text{ou bien si } ES = \text{cst } \text{INTAC} \otimes \text{PCPC3} \end{aligned}$$

L'opérateur de construction de problème potentiel que l'on a choisi pour ce problème thermodynamique met en oeuvre une méthode implicite compatible avec la non-linéarité des termes de diffusion (voir 1ère partie chapitre IV).

Cet opérateur : "CSNRDR" calcule le résidu du système non linéaire à résoudre sous la forme standard :

$$R = L \cdot (\bar{I} - \bar{I}_0) + \Delta t (M \cdot \bar{I} - KS - KV)$$

et construit la matrice Jacobienne de ce système sous la forme :

$$L + DL - DLO + \Delta t (M + DM)$$

expressions dans lesquelles :

L, DL, DLO, M, DM, KS, KV sont les vecteurs et matrices constitutives.

En comparant terme à terme avec l'expression littérale de l'équation thermique et de sa matrice Jacobienne, on en

déduit, avec $I = TN$ et $I_0 = TQ$:

$$L = \iint_{\Omega} \alpha_i \alpha_j RC = TES$$

$$M = \iint_{\Omega} \alpha_i (V \cdot \nabla \alpha_j) RC + \iint_{\Omega} \nabla \alpha_i \cdot (K \# \nabla \alpha_j) + \int_{\Gamma} \alpha_i \alpha_j H + \int_{\Gamma} \alpha_i \alpha_j ES \cdot T^3 = CE + SE + CV + DRY$$

$$KS = \iint_{\Omega} \alpha_i QS = VES$$

$$KV = \iint_{\Omega} \alpha_i QV + \int_{\Gamma} \alpha_i H \cdot TA + \int_{\Gamma} \alpha_i ES \cdot TA^4 = VEV + VEC + VER$$

Les vecteurs et matrices constitutives L , M , KS et KV définissent totalement le calcul du résidu.

Dans l'évaluation de la matrice Jacobienne $J = L + DL - DLO + \Delta t (M + DM)$ les matrices L et M interviennent à nouveau, mais d'autres termes apparaissent. Pour les évaluer il suffit de procéder par élimination en comparant l'expression littérale donnée au chapitre A.IV et l'expression constitutive proposée ici. On obtient alors directement :

$$DL = \iint_{\Omega} \alpha_i \alpha_j \frac{\partial RC}{\partial T} TN = DTES$$

$$DLO = \iint_{\Omega} \alpha_i \alpha_j \frac{\partial RC}{\partial T} TP = DTET$$

et en regroupant tous les termes facteurs de Δt :

$$M + DM = \iint_{\Omega} \alpha_i (V \cdot \nabla \alpha_j) RC + \iint_{\Omega} V \cdot \nabla TN \frac{\partial RC}{\partial T} \alpha_i \alpha_j + \iint_{\Omega} \nabla \alpha_i \cdot K \# \nabla \alpha_j + \iint_{\Omega} \left(\left(\frac{\partial K}{\partial T} \# \nabla TN \right) \cdot \nabla \alpha_i \right) \alpha_j + \int_{\Gamma} \alpha_i \alpha_j H + 4 \int_{\Gamma} ES \cdot TN^3 \alpha_i \alpha_j$$

Or, nous avons déjà posé :

$$M = \iint_{\Omega} \alpha_i (V \cdot \nabla \alpha_j) RC + \iint_{\Omega} \nabla \alpha_i \cdot (K \# \nabla \alpha_j) + \int_{\Gamma} \alpha_i \alpha_j H + \int_{\Gamma} \alpha_i \alpha_j ES \cdot TN^3$$

d'où nous déduisons par élimination l'expression de DM :

$$DM = \iiint_{\Omega} (V \cdot \nabla TN) \frac{\partial RC}{\partial T} \alpha_i \alpha_j + \iiint_{\Omega} \left(\left(\frac{\partial K}{\partial T} \neq \nabla TN \right) \cdot \nabla \alpha_i \right) \alpha_j + 3 \int_{\Gamma} ES \cdot TN^3 \alpha_i$$

d'où avec les intégrants déjà définis :

$$M = CE + SE + CV + DRY$$

$$DM = DCE + DSE + 3 DRY$$

On peut remarquer que le terme :

$$4 \iint_{\Omega} ES \cdot TN^3 \alpha_i \alpha_j$$

de la matrice Jacobienne a été décomposé en :

$$\iint_{\Omega} ES \cdot TN^3 \alpha_i \alpha_j + 3 \iint_{\Omega} ES \cdot TN^3 \alpha_i \alpha_j$$

afin de rester compatible avec la définition de la matrice M qui comporte déjà le terme :

$$\iint_{\Omega} ES \cdot TN^3 \alpha_i \alpha_j = DRY$$

Remarque

La description d'une telle équation demande une préparation assez soignée car l'expression de la matrice Jacobienne est assez complexe. Sa description conversationnelle à l'aide de notre système expert demande plusieurs séances d'utilisation. 2h à 4h d'interaction homme-machine sont nécessaires pour mener à bien cette tâche dont le coût global est négligeable (quelques minutes d'unité centrale).

Nous présentons dans l'annexe (2) l'utilisation d'un

tel système objet pour la résolution d'un problème concret bidimensionnel transitoire.

C.III.3 UTILISATION DE FLUX EXPERT POUR DÉCRIRE L'ÉQUATION DE NAVIER-STOCKES BIDIMENSIONNELLE

Les problèmes numériques que pose la résolution de cette équation sont très délicats. (Non linéarité, dissymétrie, décentrage amont nécessaire à haute vitesse, coexistence de termes d'ordre différent qui imposent des fonctions de projection sous-paramétriques). C'est pourquoi dans le cadre de cet exemple nous conduisons l'étude dans un espace bidimensionnel et nous supposons la viscosité constante.

Les équations s'écrivent alors :

$$(1) \iint_{\Omega} \rho N_i \frac{\partial u}{\partial t} + \iint_{\Omega} \rho N_i U \cdot \nabla u + \eta \iint_{\Omega} \nabla N_i \cdot \nabla u - \iint_{\Omega} \frac{\partial N_i}{\partial x} P = \iint_{\Omega} N_i F_x + \int_{\Gamma} N_i \tau_x \cdot n$$

$$(2) \iint_{\Omega} \rho N_i \frac{\partial v}{\partial t} + \iint_{\Omega} \rho N_i U \cdot \nabla v + \eta \iint_{\Omega} \nabla N_i \cdot \nabla v - \iint_{\Omega} \frac{\partial N_i}{\partial y} P = \iint_{\Omega} N_i F_y + \int_{\Gamma} N_i \tau_y \cdot n$$

$$(3) \iint_{\Omega} M_i \frac{\partial u}{\partial x} + \iint_{\Omega} M_i \frac{\partial v}{\partial y} = 0$$

Appliquons à ce problème une discrétisation implicite.

En ayant posé :

$$I = \begin{pmatrix} u \\ v \\ p \end{pmatrix} \quad \text{et} \quad U = \begin{pmatrix} u \\ v \\ p \end{pmatrix}$$

le système d'équations (1), (2), (3) s'écrit :

$$(4) \{ [L] + \Delta t [M_n] \} I_n = [L] I_{n-1} + \Delta t (KS + KV)$$

Le système (4) est un système non linéaire que nous résoudrons pas la méthode de Newton-Raphson ce qui impose de calculer :

- le résidu sous la forme standard :

$$(5) \quad [L] (I - I_0) + \Delta t ([M] I - KS - KV)$$

- la matrice Jacobienne suivant l'expression :

$$(6) \quad [L] + \Delta t ([M] + [DM])$$

Dans les formules (5) et (6) les matrices et vecteurs L, M, KS, KV, DM sont les matrices et vecteurs constitutifs standards définis par rapport à l'inconnue vectorielle :

$$\dot{I} \begin{vmatrix} u \\ v \\ p \end{vmatrix}$$

Compte-tenu du système d'équations (1), (2), (3) et de l'opérateur retenu (5) et (6), nous en déduisons l'expression littérale de chaque terme constitutif :

L, M, DM, KS, KV

Par identification nous obtenons :

$$L_{uu} = L_{vv} = \left[\iint_{\Omega} \rho N_i N_j \right] = TE$$

$$L_{lm} = 0 \text{ si } l \neq m \text{ et } L_{pp} = 0$$

$$M_{uu} = \left[\iint_{\Omega} \rho N_i N_j \frac{\partial u}{\partial x} + \eta \iint_{\Omega} \nabla N_i \cdot \nabla N_j \right] = TUX + SE$$

$$M_{uv} = \left[\iint_{\Omega} \rho N_i N_j \frac{\partial u}{\partial y} \right] = TUY$$

$$M_{up} = \left[- \iint_{\Omega} \frac{\partial N_i}{\partial x} M_j \right] = DXP$$

$$M_{vu} = \left[\iint_{\Omega} \rho N_i N_j \frac{\partial v}{\partial x} \right] = TVX$$

$$\begin{aligned}
 M_{vv} &= \left[\iint_{\Omega} \rho N_i N_j \frac{\partial v}{\partial y} + \eta \iint_{\Omega} \nabla N_i \cdot \nabla N_j \right] = TVY + SE \\
 M_{vp} &= \left[- \iint_{\Omega} \frac{\partial N_i}{\partial y} M_j \right] = DYP \\
 M_{pu} &= \left[\iint_{\Omega} M_i \frac{\partial N_j}{\partial x} \right] = PDX \\
 M_{pv} &= \left[\iint_{\Omega} M_i \frac{\partial N_j}{\partial y} \right] = PDY \\
 M_{pp} &= 0
 \end{aligned}$$

et pour la matrice Jacobienne $[J] = [L] + \Delta t ([M] + [DM])$

$$DM_{uu} = DM_{vv} = \left[\iint_{\Omega} \rho N_i \frac{\partial N_j}{\partial x} u + \iint_{\Omega} \rho N_i \frac{\partial N_j}{\partial y} v \right] = TDX + TDY$$

Tous les autres termes de Dm ^{sont} étant nuls.

Les vecteurs constitutifs KS , KV , s'écrivent :

$$\begin{aligned}
 KS_u &= \left[\iint_{\Omega} N_i FSX + \int_{\Gamma} N_i TX \right] = QSX + QTX \\
 KS_v &= \left[\iint_{\Omega} N_i FSY + \int_{\Gamma} N_i TY \right] = QSY + QTY \\
 KS_p &= 0 \\
 KV_u &= \left[\iint_{\Omega} N_i FVX \right] = QVX \\
 KV_v &= \left[\iint_{\Omega} N_i FVY \right] = QVY \\
 KV_p &= 0
 \end{aligned}$$

Afin de décrire ces intégrants à l'aide de notre système il faut tout d'abord choisir les identificateurs des variables et des grandeurs physiques mises en œuvre.

L'inconnue principale :

Au pas N

UN : composante en x de la vitesse

VN : composante en y de la vitesse

PN : pression

Au pas N-1

UN1		: composantes au pas précédent
VN1		
PN1		

Les grandeurs physiques :

ETA : la viscosité du fluide

TX et TY : les composantes en x et en y du ~~tension~~ ^{tension} des conditions aux limites évalué à l'instant N

FVX et FVY : les composantes en x et en y des forces variables appliquées au liquide à l'instant N

FSX et FSY : les composantes en x et en y des forces constantes appliquées aux ~~liquides~~ ^{liquides}

RHO : la masse volumique du fluide

Notre système demande ensuite de préciser :

- . le type de l'inconnue : REEL
- . la dimension de l'espace : 2
- . le type de coordonnées utilisées : CARTESIEN
- . le degré des polynômes choisis : ISOPARAMETRIQUE ^S
de degré 2 et SOUS-PARAMETRIQUE ^S de degré 1

Le choix de deux types de polynômes est nécessaire car la vitesse et la pression n'apparaissent pas dans l'équation avec le même ordre de dérivation.

La vitesse est une inconnue qui intervient par sa dérivée seconde. Elle sera ~~approximée~~ ^{approchée} par des polynômes de degré 2 que nous noterons Ni.

La pression est une inconnue qui n'intervient que par son gradient. Elle sera interpolée par des fonctions de degré inférieur : polynômes du 1er degré que nous noterons Mi ^[101].

Afin de pouvoir décrire les matrices constitutives il faut définir tous les intégrants nécessaires dont elles dépendent :

$$\begin{aligned}
 TE &= \left[\iint_{\Omega} N_i N_j \text{ RHO} \right] \\
 TUX &= \left[\iint_{\Omega} N_i N_j \frac{\partial(uN)}{\partial x} \text{ RHO} \right] \\
 TUY &= \left[\iint_{\Omega} N_i N_j \frac{\partial(uN)}{\partial y} \text{ RHO} \right] \\
 TVX &= \left[\iint_{\Omega} N_i N_j \frac{\partial(vN)}{\partial x} \text{ RHO} \right] \\
 TVY &= \left[\iint_{\Omega} N_i N_j \frac{\partial(vN)}{\partial y} \text{ RHO} \right] \\
 SE &= \left[\text{ETA} \iint_{\Omega} \nabla N_i \cdot \nabla N_j \right] \\
 TDX &= \left[\iint_{\Omega} \text{RHO} N_i \frac{\partial N_j}{\partial x} uN \right] \\
 TDY &= \left[\iint_{\Omega} \text{RHO} N_i \frac{\partial N_j}{\partial y} vN \right] \\
 DXP &= \left[- \iint_{\Omega} \frac{\partial N_i}{\partial x} M_j \right] \\
 DYP &= \left[- \iint_{\Omega} \frac{\partial N_i}{\partial y} M_j \right] \\
 PDX &= \left[\iint_{\Omega} M_i \frac{\partial N_j}{\partial x} \right] \\
 PDY &= \left[\iint_{\Omega} M_i \frac{\partial N_j}{\partial y} \right] \\
 QSX &= \left[\iint_{\Omega} N_i \text{ FSX} \right] \\
 QSY &= \left[\iint_{\Omega} N_i \text{ FSY} \right] \\
 QVX &= \left[\iint_{\Omega} N_i \text{ FVX} \right] \\
 QVY &= \left[\iint_{\Omega} N_i \text{ FVY} \right] \\
 QTX &= \left[\int_{\Gamma} N_i \text{ TX} \right]
 \end{aligned}$$

$$QTY = \left[\int_{\Gamma} N_i T \gamma \right]$$

Chacun de ces intégrants doit être défini comme dans les exemples précédents par un couple d'opérateurs EXPRESSION INTEGRANTE et POIDS DIFFERENTIEL.

L'examen de la base de données NOYAU-ALGORITHMES permet de sélectionner les opérateurs suivants :

$$TE = P1 \iint_{\Omega} \alpha_i \alpha_j \otimes dx dy = INAASC \otimes POIVID = RHO \iint_{\Omega} \alpha_i \alpha_j dx dy$$

avec P1 = RHOCST

$$TUX = P2 \iint_{\Omega} \alpha_i \alpha_j \vec{P1}_x \otimes dx dy = INAAXC \otimes POIVID = RHO \iint_{\Omega} \alpha_i \alpha_j \frac{\partial u}{\partial x} dx dy$$

avec $\vec{P1} = \nabla u$
P2 = RHOCST

$$TUY = P2 \iint_{\Omega} \alpha_i \alpha_j \vec{P1}_y \otimes dx dy = INAAYC \otimes POIVID = RHO \iint_{\Omega} \alpha_i \alpha_j \frac{\partial u}{\partial y} dx dy$$

avec $\vec{P1} = \nabla u$
P2 = RHOCST

$$TVX = P2 \iint_{\Omega} \alpha_i \alpha_j \vec{P1}_x \otimes dx dy = INAAXC \otimes POIVID = RHO \iint_{\Omega} \alpha_i \alpha_j \frac{\partial v}{\partial x} dx dy$$

avec $\vec{P1} = \nabla v$
P2 = RHOCST

$$TVY = P2 \iint_{\Omega} \alpha_i \alpha_j \vec{P1}_y \otimes dx dy = INAAYC \otimes POIVID = RHO \iint_{\Omega} \alpha_i \alpha_j \frac{\partial v}{\partial y} dx dy$$

avec $\vec{P1} = \nabla v$
P2 = RHOCST

$$SE = P1 \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j \otimes dx dy = INGGSC \otimes POIVID = ETA \iint_{\Omega} \nabla \alpha_i \cdot \nabla \alpha_j dx dy$$

avec P1 = ETACST

$$TDX = \iint_{\Omega} \alpha_i \frac{\partial \alpha_j}{\partial x} \otimes P3.P4 dx dy = INTADX \otimes POICC2 = \iint_{\Omega} \alpha_i \frac{\partial \alpha_j}{\partial x} UN. RHO dx dy$$

avec P3 = UN
P4 = RHO CST

$$TDY = \iint_{\Omega} \alpha_i \frac{\partial \alpha_j}{\partial y} \otimes P3.P4 dx dy = INTADY \otimes POICC2 = \iint_{\Omega} \alpha_i \frac{\partial \alpha_j}{\partial y} VN. RHO dx dy$$

avec P3 = VN
P4 = RHO CST

$$DXP = \iint_{\Omega} \frac{\partial \alpha_i}{\partial x} \beta_j \otimes (-dx dy) = INTDXB \otimes PCNVID$$

$$DYP = \iint_{\Omega} \frac{\partial \alpha_i}{\partial y} \beta_j \otimes (-dx dy) = INTDYB \otimes PCNVID$$

$$PDX = \iint_{\Omega} \beta_i \frac{\partial \alpha_j}{\partial x} \otimes dx dy = INTBDX \otimes POIVID$$

$$PDY = \iint_{\Omega} \beta_i \frac{\partial \alpha_j}{\partial y} \otimes dx dy = INTBDY \otimes POIVID$$

$$QSX = P1 \iint_{\Omega} \alpha_i \otimes dx dy = INTAC \otimes POIVID = FSX \iint_{\Omega} \alpha_i dx dy$$

avec P1 = FSX cst

ou bien $\iint_{\Omega} \alpha_i \otimes P3 dx dy = INTAV \otimes POICV1 = \iint_{\Omega} \alpha_i FSX dx dy$

$$QSY = P1 \iint_{\Omega} \alpha_i \otimes dx dy = INTAC \otimes POIVID = FSX \iint_{\Omega} \alpha_i dx dy$$

avec P1 = FSX cst

ou bien $\iint_{\Omega} \alpha_i \otimes P3 dx dy = INTAV \otimes POICV1 = \iint_{\Omega} \alpha_i FSX dx dy$

$$QVX = P1 \iint_{\Omega} \alpha_i \otimes dx dy = INTAC \otimes POIVID = FVX \iint_{\Omega} \alpha_i dx dy$$

avec P1 = FVX cst

ou bien $\iint_{\Omega} \alpha_i \otimes P3 dx dy = INTAV \otimes POICV1 = \iint_{\Omega} \alpha_i FVX dx dy$

$$QVY = P1 \iint_{\Omega} \alpha_i \otimes dx dy = INTAC \otimes POIVID = FVY \iint_{\Omega} \alpha_i dx dy$$

avec P1 = FVY cst

ou bien $\iint_{\Omega} \alpha_i \otimes P3 dx dy = INTAV \otimes POICV1 = \iint_{\Omega} \alpha_i FVY dx dy$

$$QTX = P1 \int_{\Gamma} \alpha_i \otimes d\Gamma = INTAC \otimes POICID = TX \int_{\Gamma} \alpha_i d\Gamma$$

avec P1 = TX cst

ou bien $\int_{\Gamma} \alpha_i \otimes P3 d\Gamma = INTAV \otimes POICV1 = \int_{\Gamma} \alpha_i TX d\Gamma$

$$QTY = P1 \int_{\Gamma} \alpha_i \otimes d\Gamma = INTAC \otimes POIVID = TY \int_{\Gamma} \alpha_i d\Gamma$$

avec P1 = TY cst

ou bien $\int_{\Gamma} \alpha_i \otimes P3 d\Gamma = INTAV \otimes POICV1 = \int_{\Gamma} \alpha_i TY d\Gamma$

Pour construire une telle équation à l'aide de notre générateur, une journée de travail interactif est nécessaire

surtout à cause des vérifications minutieuses qu'il faut effectuer lors de la description des intégrants.

Le coût d'une telle description demeure néanmoins très faible quelques minutes de temps unité centrale.

Nous présenterons en annexe un problème réel traité à l'aide de cette équation : écoulement dans un chenal comportant une cavité en forme de T.

C.IV SYNOPTIQUE DES COMMANDES DU PROTOTYPE .-

Nous présentons brièvement ci-dessous l'ensemble des commandes que notre système met à la disposition des utilisateurs.

Nous n'aborderons pas ici la description de la tâche de découpage effectuée par le système industriel ENTREE de FLUX2D, la notice d'utilisation de ce système [58] le décrit de manière très détaillée.

C.IV.1 LES COMMANDES DE TÂCHES

Elles permettent de lancer chacune des tâches indépendantes du système :

- . Genere appelé le générateur
- . modpro lance la tâche de modélisation des propriétés des matériaux
- . entrée pour décrire la discrétisation géométrique
- . prophy pour construire un problème réel
- . iareel pour résoudre un problème dont les inconnues sont de type réel
- . iarplx pour résoudre un problème dont les inconnues sont du type complexe
- . xexploit pour exploiter des résultats

C.IV.2 LES COMMANDES DE GESTION

Chacune des 7 tâches précédentes débute par le même dialogue.

Nom du problème = ?

Voulez-vous récupérer la base précédente = ?

Si oui : nom de la base concernée = ?

En effet, chaque tâche permet de construire une base de données qui lui est propre :

PROBLEME-OBJET	<i>pour</i>	<i>genere</i>
MATERIAUX	<i>pour</i>	<i>modpro</i>
PROBLEME-GEOMETRIE	<i>pour</i>	<i>entrée</i>
PROBLEME-OBJET	<i>pour</i>	<i>prophy</i>
RESULTATS-NOD AUX	<i>pour</i>	<i>iareel et iarplx</i>
RESULTATS-POUR-COUP LAGE	<i>pour</i>	<i>xploit</i>

Après la définition du nom de la base de données concernée, et après avoir récupéré d'éventuelles sauvegardes provenant d'utilisations précédentes, chacune des tâches offre les commandes de gestion suivantes :

STOP	<i>pour arrêter la tâche en cours</i>
SAUVEGARDER	<i>pour sauvegarder la base de données en cours de construction</i>
TRANSMISSION	<i>pour transmettre aux autres tâches du système un travail achevé.</i>
IMPRESSION	<i>pour imprimer sur liste l'état du travail réalisé.</i>

C.IV.3 LES COMMANDES PROPRES À LA TÂCHE GÉNÉRATEUR

a) Choix de l'étape concernée

Une commande permet d'activer chacune des quatre étapes nécessaires à la construction d'un système objet.

NV : NOYAU-VARIABLE pour définir les variables globales
NO : NOYAU-OPERATEUR pour définir les opérateurs
OV : OBJET-VARIABLE pour définir les variables d'un système objet
OE : OBJET-EQUATION pour définir les équations caractérisant le système objet en cours de construction

b) Détail des commandes NOYAU-VARIABLE

AJOUTER		<i>qui traitent de chacune des variables à décrire</i>
MODIFIER		
SUPPRIMER		
EDITER		<i>qui permet de visualiser :</i> . NOMS des variables déjà écrites . DETAIL d'une variable . LISTE de toutes les variables décrites

c) Détail des commandes NOYAU-OPERATEUR

Choix de la classe d'opérateur concernée.

- . FORMULES-D'INTEGRATION
- . POLYNOMES
- . SYSTEMES-LINEAIRES (résolution de système)
- . CONSTRUCTION-SYSTEMES-LINERARISES
- . EXPRESSIONS-INTEGRANTES
- . POIDS-DIFFERENTIELS
- . PROPRIETE

Pour chacun de ces opérateurs, on retrouve les commandes déjà rencontrées.

AJOUTER
MODIFIER

| adapté à chaque opérateur

SUPPRIMER
EDITER

| identiques pour tous les opérateurs

. Description d'un opérateur FORMULE-D'INTEGRATION

Notre système demande successivement :

. Nom du programme matérialisant l'opérateur :

. Dimension de l'espace de calcul : à choisir parmi :

- . Ponctuelle
- . Linéique
- . Surfactive
- . Volumique
- . Macro-élémentaire

. Description d'un opérateur POLYNOME

L'utilisateur doit donner les informations suivantes :

. Nom du programme matérialisant l'opérateur :

. Degré du polynome calculé : à choisir parmi :

- . SOUS-paramétrique degré 1
- . ~~SOUS~~-paramétrique degré 2
- . SUR -paramétrique degré 4

. Dimension de l'espace de calcul : à choisir parmi :

- . Ponctuelle
- . Linéique
- . Surfactive
- . Volumique
- . Macro-élémentaire

- . Nom de l'opérateur FORMULE-D'INTEGRATION associé :
on doit fournir alors le nom d'un opérateur défini au préalable et dont la dimension de l'espace de calcul doit être compatible.
- . Y a-t-il un polynôme sous-paramétrique associé :
quand le traitement d'une équation nécessite le calcul de polynômes de degrés différents (cf Navier-Stokes) les opérateurs qui les évaluent doivent être compatibles entre-eux et sont alors accouplés. Si la réponse à la question posée est oui le programme demande alors :
- . Nom de l'opérateur sous-paramétrique associé
- . Y a-t-il un polynôme sur-paramétrique associé :
 - . Nom de l'opérateur sur-paramétrique concerné :

Puis le système demande
- . Nombre de noeuds de l'élément de référence sur lequel est calculé le polynôme :
- . Description d'un opérateur de RESOLUTION de SYSTEME LINEAIRE
- . Nom du programme matérialisation l'opérateur :
- . Type des variables : à choisir entre :
 - . REEL
 - . COMPLEXE
- . Structure de la matrice : à choisir entre :
 - . SYMETRIQUE
 - . DISSYMETRIQUE

- . Mode de stockage de la matrice : à choisir entre :
 - . SYMBOLIQUE
 - . BANDE
 - . PLEINE
- . Description d'un opérateur de CONSTRUCTION de SYSTEMES LINEARISE
- . Nom du programme matérialisant l'opérateur :
- . Type des variables : à choisir entre :
 - . REEL
 - . COMPLEXE

Puis le système dresse la liste des matrices constitutives et pour chacune demande :

- . Matrice utilisée ? oui-non
- . Entrez la formule correspondante /

Enfin, l'utilisateur doit fournir la formule correspondant à l'opérateur en cours de description afin que celle-ci apparaisse en clair dans la documentation (c'est en effet une formule mathématique qui caractérise un tel opérateur beaucoup mieux qu'un nom dans une base de données). Aucune analyse syntaxique ou sémantique n'est effectuée sur une telle formule.

- . Description d'un opérateur EXPRESSION-INTEGRANTE
- . Nom du programme matérialisant l'opérateur :
on rappelle que chaque opérateur de cette classe peut mettre en oeuvre deux types de polynômes distincts ; c'est pourquoi le système demande :
 - . Degré du 1er polynôme : à choisir entre :
 - . Vide (le 1er polynôme n'est pas utilisé)
 - . Sous-paramétrique

- . Iso-paramétrique
- . Sur-paramétrique
- . Nature du 1er polynôme : à choisir parmi :
 - . Directe : on utilise le polynôme tel quel
 - . Gradient : c'est le gradient du polynôme qui sera calculé.
- . Degré du 2ème polynôme
 - . Vide
 - . Sous-paramétrique
 - . Iso-paramétrique
 - . Sur-paramétrique
- . Nature du 2ème polynôme
 - . Directe
 - . Gradient

On rappelle qu'une expression intégrante peut faire intervenir deux propriétés au sein de son interface, c'est pourquoi le système demande alors :

- . Propriété P1 : à choisir parmi :
 - . Vide : pas de propriété utilisée
 - . Utilisée : l'expression dépend d'une propriété
 - . Propriété P2
 - . Vide
 - . Utilisée
 - . Type de l'intégrant calculé : à choisir parmi :
 - . Vecteur
 - . Matrice
- Dans le cas d'une matrice il faut alors préciser :
- . Structure de l'intégrant calculé : à choisir parmi :
 - . Symétrique
 - . Dissymétrique

Puis le système demande :

- . Entrez la formule correspondante :
toujours à des fins de documentation, il faut fournir une description standard de l'expression calculée à l'aide des variables suivantes : A, B, C pour les divers types de polynômes P1, P2 pour les propriétés GRAD ROT DIV pour les opérateurs classiques de dérivation et les opérateurs arithmétiques usuels + - x / **. Cette formule servira de modèle pour le formateur spécial qui permet de présenter le résultat d'une description d'équation avec le graphisme usuel :
exemple : $(\int \int \text{Grad}(A_i) \cdot \text{Grad}(A_j) * P1 * d\Gamma$

- . Description d'un opérateur POIDS-DIFFERENTIEL

- . Nom du programme matérialisant l'opérateur :
nous rappelons qu'un opérateur poids différentiel peut faire intervenir également deux propriétés physiques ainsi que la valeur du rayon dans le cas de coordonnées axisymétriques : c'est pourquoi le système demande :

- . Propriété P3 : à choisir parmi :

- . Vide
- . Utilisée

- . Propriété P4 : à choisir parmi :

- . Vide
- . Utilisée

- . Rayon en axisymétrique : à choisir parmi :

- . Vide
- . Utilisé

- . Entrez la formule correspondante :
comme pour l'expression intégrante, cette formule servira à reconstituer les équations selon le graphisme usuel. Exemple : $dG = R * dR * dZ$.

. Description d'un opérateur CALCUL-PROPRIETE

. Nom du programme matérialisant l'opérateur :

- . Opérateur fonction du temps ? *oui-non*
- . Opérateur fonction de l'espace ? *oui-non*
- . Opérateur fonction de la variable principale ? *oui-non*
- . Opérateur fonction d'une autre propriété P1 ? *oui-non*
- . Opérateur fonction d'une propriété P2 ? *oui-non*

. Entrez la formule correspondante :

là encore cette formule ne sert qu'à la documentation interne et facilite le choix des divers opérateurs lors de la description d'un modèle de propriété.

d) Détail des commandes de OBJET-VARIABLE

INCONNUE-PRINCIPALE

permet de définir

- . le nombre de composante de l'inconnue principale
- . le nom donné à chacune des composantes au pas de temps courant et au pas de temps précédent pour les équations de phénomènes d'évolution.

AJOUTER

commande qui permet de sélectionner les variables utilisées par les équations en cours de description, au sein de la base NOYAU-VARIABLES.

Pour chaque variable ainsi sélectionnée, il faut également préciser les modèles de calcul de propriétés dont on autorisera l'utilisation dans les phases ultérieures de description de problèmes réels à savoir :

- . modèle nul
- . modèle constant

- . liste de valeurs nodales globales
- . liste de valeurs nodales définies par élément
- . liste de valeurs nodales définies par région
- . modèle calculé par un opérateur propriété
dans ce dernier cas le système demande :
 - . le nombre de modèles distincts envisagés et pour chacun :
 - son nom
 - le nom de l'opérateur propriété qui le calcule

EDITER

commande standard qui permet d'éditer sur console ou sur liste.

- . NOMS des variables définies
- . DETAIL d'une variable
- . LISTE détaillée de toutes les variables

e) Détail des commandes de OBJET-EQUATION

OPTIONS GENERALES

permet de définir

- . le type des variables utilisées
 - . REELLES
 - . COMPLEXE
- . le type des coordonnées utilisées
 - . CARTESIENNES
 - . AXISYMETRIQUES
- . la dimension de l'espace d'étude
1, 2 ou 3
- . Puis pour chacun des opérateurs de CONSTRUCTION de SYSTEMES-linéarisés contenus dans le NOYAU-OPERATEUR le programme demande si cet opérateur sera utilisé

par l'une des stratégies de résolution. La formule de chaque opérateur est affichée simultanément afin que le choix ne nécessite pas la consultation de documents écrits annexes.

Commande INTEGRANT-AJOUTER

Elle a pour objet de décrire chacun des intégrants qui composent l'équation en précisant

- . le nom de l'intégrant
- . la caractéristique discriminante ou si c'est un intégrant qui ne met pas en oeuvre de propriété, la dimension de cet intégrant : ponctuel, linéique, surfacique, volumique, macro-élémentaire.
- . le type de l'intégrant :
 - . Matrice
 - . Vecteur
- . Pour le cas où la propriété discriminante est une constante
 - . Nom de l'opérateur expression intégrante :
 - . Les opérandes effectifs de cet opérateur : propriétés P1 et P2
 - . Nom de l'opérateur poids différentiel :
 - . Les opérandes effectifs de cet opérateur : propriétés P3 et P4
- . La nature de l'intégrant défini :
 - . Linéaire
 - . Dynamique
 - . Non-linéaire
- . Pour le cas où la propriété discriminante est variable :
 - . Nom de l'opérateur expression intégrante
 - . les opérandes effectifs de cet opérateur : P1 et P2

- . Nom de l'opérateur poids différentiel :
- . Les opérandes effectifs de cet opérateur :
P3 et P4
- . La nature de l'intégrant défini :
 - . Linéaire
 - . Dynamique
 - . Non-linéaire

Remarque

Cette dernière information permet à l'outil de résolution de problèmes d'optimiser le nombre d'intégrants à calculer dans les stratégies itératives (un intégrant linéaire est supposé constant au cours du déroulement de toute la stratégie, un intégrant dynamique est recalculé au début de chaque pas de temps, un intégrant non linéaire est recalculé à chaque résolution).

Commande INTEGRANTS-EDITER

Elle permet d'éditer sur console :

- . Le détail d'un intégrant déjà créé
- . La liste détaillée de tous les intégrants déjà créés.

Commande MATRICE-DESCRIPTION

Elle permet de définir les matrices constitutives nécessaires pour les opérateurs de CONSTRUCTION de systèmes linéarisés qui ont été retenus au préalable par la commande OPTIONS.

Pour chacune des matrices sectionnées et pour chaque bloc de ces matrices (dans le cas d'une inconnue multi-composantes) le programme demande :

- . Le nombre d'intégrants
- . Les noms de ces intégrants

Commande MATRICE-EDITION

Elle permet de faire éditer les matrices constitutives préalablement définies.

C.IV.4 LES COMMANDES DE LA TÂCHE MODÉLISATION DES PROPRIÉTÉS

Commande AJOUTER

Elle permet d'ajouter un modèle à l'aide des informations suivantes

- . Nom du modèle à créer
- . Propriété modélisée
- . Type du modèle parmi
 - . Valeur constante : valeur de cette constante
 - . Valeurs tabulées aux noeuds globalement
 - . Valeurs tabulées aux noeuds par élément
 - . Valeurs tabulées aux noeuds par région

Pour ces trois derniers cas l'utilisateur doit fournir le nom du fichier qui contient les tables.

. Fonction

Dans ce dernier cas l'utilisateur doit indiquer

- . le nom de la fonction choisie
- . le nombre de paramètres variables définissant cette fonction
- . les valeurs de ces paramètres ou le nom du fichier qui les contient (dans le cas des courbes expérimentales interpolées).

Commande SUPPRIMER

Elle permet de supprimer un modèle devenu inutile ou erroné.

Commande EDITER

Elle permet de visualiser

- . Les noms des modèles décrits
- . Le détail d'un modèle
- . La liste détaillée de tous les modèles
 - sur console ou sur fichier*

C.IV.5 LES COMMANDES DE LA TÂCHE DE DESCRIPTION DE PROBLÈME RÉEL

Commande SYSTEME

Permet de lire le nom du système objet contenant les équations à traiter.

Commande DECOUPAGE

Permet de lire le nom de la base de données contenant l'objet discrétisé en éléments finis.

Commande MATERIAU

Permet de lire le nom de la base de données contenant les modèles des propriétés.

Commande AFFECTER

Pour chaque région du découpage

- . Tracé de la région sur terminal

Pour chaque propriété du système objet

- . Demande le nom du modèle à utiliser pour calculer la propriété

Commande LIMITES

- . Affiche les contours de l'objet
pour chaque composante de l'inconnue
- . Demande le type de conditions aux limites : à choisir
parmi :
 - . Dirichlet
 - . Cyclique
 - . Anticyclique
 - . Translation

Pour chaque type de condition aux limites on doit fournir la liste des numéros des noeuds du contour (noeud de début, et noeud de fin) et les valeurs numériques des inconnues (pour Dirichlet et Translation).

C.IV.6 LES COMMANDES DE LA TÂCHE DE RÉOLUTION DE PROBLÈME

Elles consistent à :

- . choisir la stratégie de résolution parmi les stratégies proposées :
 - . linéaire direct statique
 - . non-linéaire statique Newton-Raphson
 - . linéaire dynamique implicite
 - . linéaire dynamique Crank-Nicholson
 - . dynamique terme d'espace non linéaire, terme de temps linéaire Crank-Nicholson
 - . dynamique terme d'espace non linéaire, terme de temps linéaire implicite
 - . dynamique terme d'espace non linéaire, terme de temps linéaire prédiction-correction
 - . dynamique termes spatiaux et temporels non linéaires implicite

- . dynamique termes spatiaux et temporels non linéaires implicite
- . dynamique termes spatiaux et temporels non linéaires implicite prédiction-correction
- . Choisir la méthode de résolution selon la structure de la matrice :
 - . symétrique définie positive
 - . non définie positive bien conditionnée
 - . dissymétrique mal conditionnée
- . Fournir les pas de temps, d'espace, les précisions désirées de Newton Raphson, le pas d'échantillonnage des résultats sur disque (pour les phénomènes évolutifs sur le temps).

C.IV.7 LES COMMANDES DE LA TÂCHE D'EXPLOITATION DES RÉSULTATS

Elles consistent à :

- . Déterminer le numéro de l'échantillon à visualiser
- . Déterminer la composante de l'inconnue à traiter
- . Préparer le tracé
 - des régions
 - des lignes isovaleurs
 - des vecteurs nodaux dans le cas d'une variable vectorielle
- . Afficher les tracés préparés
- . Sauvegarder les dessins obtenus
- . Construire les fichiers de valeurs tabulées pour couplage

Le système demande alors le type de modèle parmi :

- . Tabulé global*
- . Tabulé par élément*
- . Tabulé régional*

dans ce dernier cas le nom de la région concernée doit être fourni.

C.V FONDEMENTS INFORMATIQUES DU SYSTEME

Nous avons déjà justifié l'organisation de notre système en 6 phases auxquelles correspondent six programmes principaux. A l'exécution, ces programmes sont totalement indépendants ; par contre, ils sont constitués à l'aide de sous-programmes appartenant à des bibliothèques communes et ils échangent des données au moyen de bases de données partagées. Nous allons détailler cette organisation interne.

C.V.1 BASES DE DONNÉES ET BANQUE DE SAVOIR-FAIRE

Nous avons vu, dans le paragraphe précédent, que le système gère de nombreuses bases de données. Leur dimensionnement ne peut en aucun cas être connu a priori, chacune des 6 phases du système accède donc à des bases de données dynamiques.

Or, le système peut être réparti sur des machines différentes, le problème de la communication entre phases se pose donc. On doit remarquer, en outre, que l'utilisateur ne peut accéder aux données d'une phase précédente que lorsque le travail correspondant à cette phase est totalement achevé. Ceci autorise donc le mode de gestion suivant :

- . Lorsque l'utilisateur estime qu'il a achevé une phase la commande "TRANSMISSION" lui permet de faire construire un fichier formaté portable : "la base de donnée statique" qui contient l'état de la base de données qui vient d'être construite.*

Dès que la phase suivante est démarrée, il faut bien entendu récupérer le fichier correspondant et les programmes reconstruisant automatiquement la base de donnée dynamique.

L'expérience nous a montré qu'il était très délicat de gérer avec fiabilité des bases de données ainsi traitées par deux jeux de programmes distincts l'un écrivant la base, l'autre la lisant et la reconstituant. C'est pourquoi chaque base n'est gérée que par un seul jeu de programmes qui assurent la lecture ou l'écriture de façon totalement symétrique (un paramètre d'entrée permettant l'aiguillage selon l'option choisie).

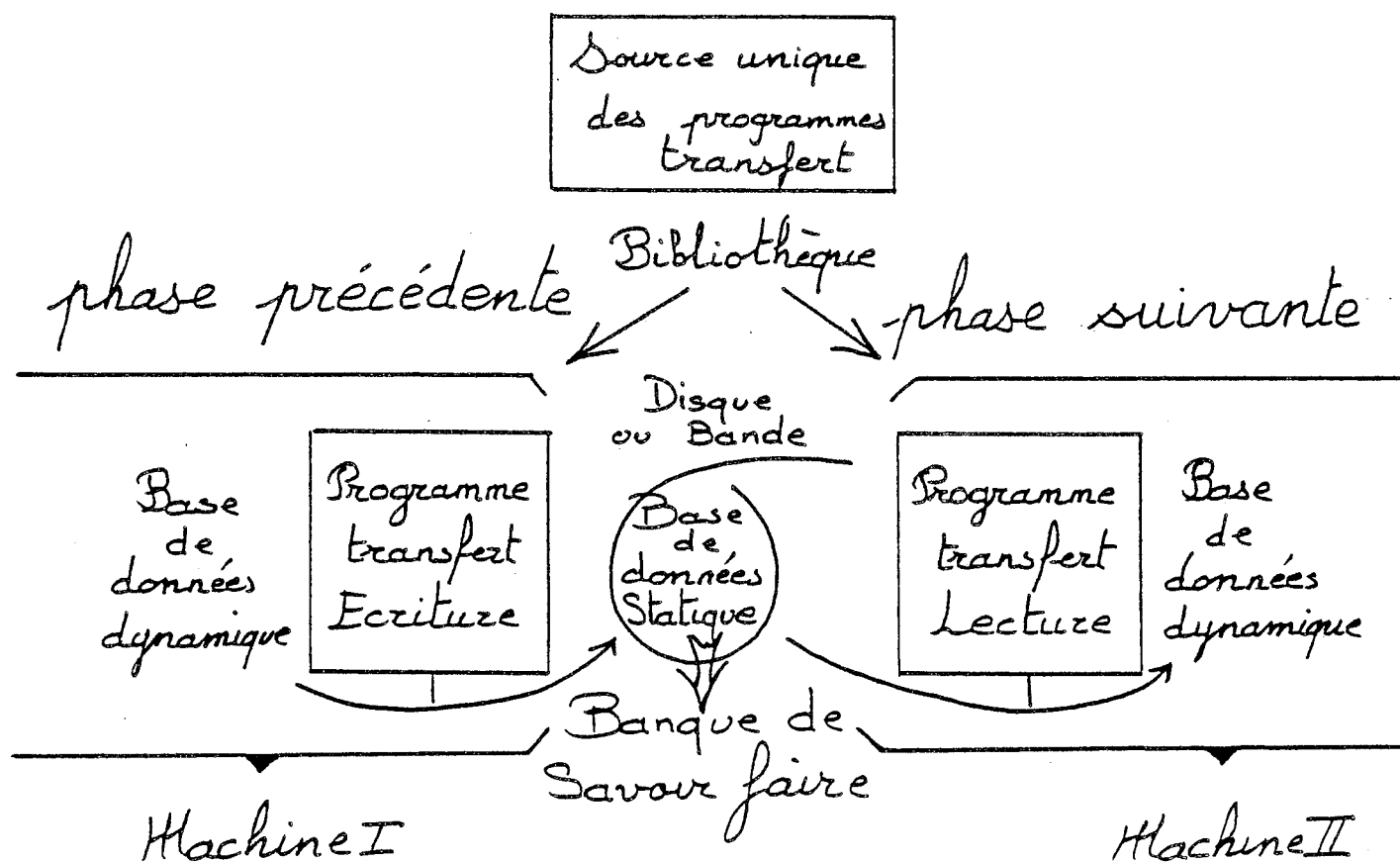


Figure 11 Bases de données statique et dynamique.

L'intérêt de pouvoir utiliser des bases de données sous forme dynamique ou statique ne réside pas seulement dans la possibilité de répartir le système sur des machines différentes, cela permet également de constituer des archives fort utiles notamment pour les bases de données "système-objet". En outre, le fait d'utiliser des fichiers formatés rend ces archives parfaitement lisibles. L'utilisateur peut donc ainsi se constituer une banque de savoir faire qu'il peut gérer à son gré et surtout qu'il peut réactiver aisément grâce aux programmes de transfert.

La banque de savoir-faire de notre système se compose des informations suivantes :

NOYAUX
SYSTEMES-OBJETS
MODELES-MATERIAUX
DECOUPAGES
PROBLEME
RESULTATS-NOD AUX
RESULTATS-POUR-COUPLAGES

C.V.2 BILBIOTHÈQUES D'OPÉRATEURS

La notion d'ensemble d'opérateurs d'une même classe a déjà été présentée dans la deuxième partie de ce mémoire, nous exposons ici le détail de la structure informatique qui les gère.

A chaque classe d'opérateur correspond une bibliothèque qui doit être écrite en fortran. C'est l'utilisateur qui doit programmer un algorithme qui ferait défaut dans le noyau ; c'est pourquoi la documentation prend dans ce cas une grande importance.

La procédure à suivre pour programmer un opérateur est

fort simple :

- chaque classe d'opérateur comporte un opérateur particulier : l'opérateur VIDE, composé d'une seule instruction : RETURN. Mais dans la subroutine correspondant l'utilisateur peut trouver la description détaillée de l'interface commune à tous les opérateurs de la classe, la déclaration des types des paramètres et leur signification respectives, et des conseils sur la programmation de tout opérateur. Il suffit donc de recopier l'"opérateur vide" dans un fichier, d'insérer la programmation du nouvel opérateur et de modifier le nom de la subroutine pour obtenir le programme désiré sans courir aucun risques d'erreur d'interface.

Le système comportant plusieurs centaines de sous-programmes, le risque d'obtenir des noms homonymes est assez important ; c'est pourquoi les deux premières lettres de tous les noms des programmes d'une même bibliothèque sont imposées. Ceci restreint la lisibilité des noms de programmes (6 caractères), mais la description détaillée fournie dans le générateur et les commentaires dans le texte des opérateurs pallient cet inconvénient.

Le principe de notre système repose sur la possibilité d'ajouter des opérateurs chaque fois que cela est nécessaire sans être contraint de modifier le code des programmes déjà écrits et même sans avoir à refaire une édition de lien de ces programmes. Un chargeur dynamique prépare l'interface des programmes opérateurs en consultant d'une part le dictionnaire d'interface, associé à chaque bibliothèque et d'autre part la base de données "PROBLEME-REEL" qui contient les valeurs numériques demandées. La réalisation de ces outils spécifiques a été grandement facilitée par l'utilisation du système MULTICS qui met en oeuvre un chargeur dynamique de programmes, accessible en PL1.

C.V.3 BIBLIOTHÈQUES FONCTIONNELLES

Tout notre système est organisé en bibliothèques que l'on peut classer en trois catégories (outre les opérateurs déjà présentés).

, Les outils non portables : ils sont en fait des extensions apportées au langage FORTRAN et ~~qui~~ sont propres à chacune des machines :

- . les outils de base : écrits en langage standard (mais utilisant les outils non portables), ils résolvent des problèmes qui peuvent apparaître dans plusieurs phases du système,
- . les bibliothèques structurelles : qui ne font que regrouper sous des rubriques logiques les routines générales de notre programme.

C.V.3.1 Les outils non portables

Ils concernent :

- . la gestion des allocations dynamiques de variables
- . les appels dynamiques d'opérateurs
- . la gestion des fichiers séquentiels
- . la gestion des fichiers à accès direct

La plupart ^{de} ces outils ^{de} avaient été déjà mis ^{en} oeuvre dans le système FLUX, ils ont néanmoins été spécifiés à nouveau pour les besoins de notre système.

Il faut reprogrammer ces outils sur chaque nouvelle machine mais l'expérience en ce domaine (acquise dans le transport de FLUX sur au moins 5 machines différentes) nous a montré que c'était un travail fastidieux mais posant peu de problèmes.

Seul l'enchaînement dynamique des opérateurs peut poser des problèmes sur des machines qui ne disposent pas de chargeur dynamique.

C.V.3.2 Les outils de bases

Ils sont composés de :

- . la gestion des dialogues
- . la gestion des dessins
- . la gestion des structures de listes

Cette dernière partie est particulièrement importante car notre système traite de nombreuses listes :

- . les éléments sont décrits par une liste de noeuds,
- . les régions sont constituées de listes d'éléments,
- . les valeurs de chaque caractéristique sont calculées pour tous les points d'intégration d'un élément et tabulées sous forme de listes,
- . les intégrants calculés pour chaque élément sont regroupés sous forme de listes de matrices ou de vecteurs.
- . les matrices constitutives sont également stockées sous forme de listes
- . enfin chaque classe d'opérateur forme une liste qui comporte pour chaque item le nom du programme FORTRAN effectuant le travail et la description de son interface formelle.

Toutes les listes mises en oeuvre par notre système sont directement gérées par un outil commun écrit en FORTRAN, compatible avec les primitives de déclarations dynamiques de mémoire et la gestion des fichiers à accès direct. Ceci permet notamment de pouvoir aisément sauvegarder sur disque ces

structures de liste et d'accéder à volonté à un item sélectionné qu'il soit résidant en mémoire centrale ou stocké en mémoire de masse.

La gestion des dessins est assurée par les programmes que nous avons développés lors de l'écriture du système FLUX2D ¹⁵⁶.

La gestion des dialogues a été conçue par J.L. COULOMB pour le système pédagogique Didact-Flux []. Sa robustesse et son utilisation très souple en font un outil particulièrement performant (il a été écrit en Fortran 77).

^{fh} C.V.3.3 Les bibliothèques structurelles

Tout notre système est organisé en bibliothèques qui réunissent chacune les sous-programmes concernant une même phase de traitement de l'information.

L'ensemble des quelques 500 routines de notre système est organisé en une trentaine de bibliothèques qui comportent donc en moyenne une vingtaine de programmes. Les plus volumineuses sont bien sûr les bibliothèques d'opérateurs (par exemple les opérateurs calcul d'expressions intégrantes sont au nombre de 45, les poids différentiels atteignent le même nombre).

Afin de pouvoir gérer efficacement ces bibliothèques, des outils de documentation automatique ont été mis au point. Ils assurent :

- . Une présentation standard du code (indentation des boucles et des instructions logiques, numérotation des étiquettes par valeurs croissantes, écriture des commentaires en minuscules).
- . Un contrôle des déclarations d'interface sous une forme standardisée.

- . La construction des chaînages d'appels directs et inverses ainsi que la liste des communs référencés.
- . L'édition automatique d'un recueil comportant les interfaces et la description de tous les programmes.

Ces outils ont été entièrement réalisés par MM Bernard MOREL et Patrick GIRARD en deux versions : langage PL1 ~~et puis~~ langage FORTRAN 77. Leur utilisation s'est montrée très efficace dans la gestion de notre prototype.

VI. CONCLUSION .-

Par la réalisation de ce prototype, nous pensons avoir démontré que les techniques de résolution automatiques de problèmes peuvent être efficaces pour le traitement des équations aux dérivées partielles utilisant la méthode des éléments finis.

L'utilisation d'un tel outil permet de réduire à quelques jours le temps nécessaire à la conception et à la réalisation d'un code d'éléments finis adapté à des besoins bien particuliers.

Le langage de description que nous avons proposé n'est qu'un premier jet et nous estimons qu'il devra subir de nombreuses modifications pour s'adapter au mieux aux besoins de ses utilisateurs. Cette phase d'adéquation devrait logiquement faire suite à la phase de mise au point de notre prototype à laquelle nous nous employons actuellement.

Nous pouvons cependant dès à présent remarquer que l'utilisation d'une telle méthode pour décrire un système ne va pas sans difficultés. En effet, la technique informatique prend à sa charge presque toutes les tâches fastidieuses que l'on rencontrait dans l'écriture d'un code aux éléments finis traditionnel. La description d'un nouveau système, à l'aide de

nos outils, demande en conséquence une concentration extrême de la part de l'utilisateur. Toutes les décisions importantes concernant le nouveau système à construire doivent avoir été prises avant le début de la programmation, car le temps de réponse extrêmement rapide de la machine n'est pratiquement pas compatible avec une réflexion "on line".

D'autre part, la programmation assistée d'un système informatique n'exclut pas la dure nécessité de tester le produit obtenu. Certes quelques vérifications (souvent triviales) sont effectuées par notre générateur, mais, dans ce domaine, nous sommes très loin d'avoir fourni toutes les sécurités qu'il aurait été possible de concevoir. Nous pensons qu'il serait utile de conduire une recherche dans cette direction afin de tester, dans le détail, la cohérence de toute programmation réalisée à l'aide de notre système expert.

Enfin, le temps de réponse de nos outils de résolution de problèmes peut certainement être amélioré considérablement ; nous pensons que l'application de notre technique aux géométries tridimensionnelles dépendra beaucoup des progrès que nous pourrions réaliser dans ce domaine, soit en optimisant finement la phase de résolution de problèmes, soit en mettant en oeuvre des machines multi-processeurs bien adaptées à ces structures arborescentes.

C O N C L U S I O N

C O N C L U S I O N

Les méthodes de la recherche appliquée dans le domaine de la conception de nouveaux prototypes ont été véritablement révolutionnées par la puissance conjuguée de la C.A.O. et des méthodes de résolution numérique des équations aux dérivées partielles. La simulation du fonctionnement d'ensembles complexes est actuellement réalisable dans des conditions très proches de leur mode de fonctionnement réel. La conception assistée par ordinateur et la méthode des éléments finis ont obtenus définitivement leurs lettres de noblesse.

C'est dans ce contexte qu'il nous a paru nécessaire de conduire une réflexion sur le fond : peut-on dégager une méthodologie générale qui rende compte le plus fidèlement possible de la grande majorité des modélisations de phénomènes physiques qui mettent en oeuvre à la fois des techniques de C.A.O. et la méthode des éléments finis ? Cette analyse détaillée nous a permis de concevoir de nouvelles techniques qui peuvent d'une part intégrer un grand nombre d'outils de modélisation et d'autre part alléger au maximum le travail informatique que nécessite la réalisation d'un logiciel de C.A.O. mettant en oeuvre la méthode des éléments finis.

Dans les deux premières parties de notre mémoire, nous avons exposé la méthodologie générale que nous proposons pour aborder une modélisation numérique de phénomènes physiques à l'aide de la méthode des éléments finis. Cette méthodologie permet de prendre en compte un grand nombre de techniques actuellement disponibles. Elle peut également être appliquée à

de nombreux phénomènes physiques régis par des équations aux dérivées partielles, elliptiques, paraboliques ou hyperboliques. Nous avons choisi de prendre comme sujet d'analyse des phénomènes magnétiques, thermiques et hydrauliques mais les résultats que nous avons obtenus sont ~~indépendants~~ ^{indépendants} de ces exemples qui nous ont servi de support et ~~peuvent~~ ^{peuvent} être appliqués à d'autres phénomènes.

Dans la deuxième partie de ce mémoire nous avons montré comment les techniques usuellement utilisées en Intelligence Artificielle permettent de matérialiser sous forme de programme la méthodologie que nous avons dégagée. Nous nous sommes attachés cependant à ne pas utiliser un langage propre à l'intelligence artificielle ou même trop proche de l'informatique pour concrétiser la méthodologie proposée. En effet notre but était de fournir des logiciels destinés aux physiciens et non aux informaticiens. Nous avons essayé d'éviter toute formalisation qui aurait pu paraître ésotérique ou trop informatique pour demeurer à un niveau pragmatique qui nous a paru plus efficace. C'est le même souci qui nous a conduit à insister sur l'aspect physique de notre approche au sein de ce mémoire ; à l'analyse déductive théorisante, nous avons préféré la démarche inductive caractéristique de la physique.

Enfin, nous avons présenté le prototype de système EXPERT que nous avons conçu et réalisé. La description d'un système informatique est toujours délicate et là encore nous avons préféré une illustration à l'aide d'exemples d'utilisation plutôt qu'une présentation plus systématique.

Grâce à ce système nous avons pu montrer que le temps nécessaire à la construction d'un logiciel de C.A.O. qui utilise la méthode des éléments finis peut être ramené à quelques jours de travail de modélisation et quelques heures de travail informatique. Les exemples concrets que nous avons pu réaliser à l'aide de notre prototype ont également prouvé que le coût

en temps d'unité centrale de la construction d'un logiciel à l'aide de notre générateur est pratiquement négligeable, alors que les coûts d'utilisation des systèmes construits demeurent comparables aux performances de systèmes industriels optimisés.

Mais notre approche n'est que la première étape dans un domaine de recherche qui reste très ouvert. En effet, de gros progrès peuvent encore être envisagés dans de nombreux domaines :

- . L'analyse méthodologique doit être encore plus fouillée pour permettre de mettre en oeuvre un plus grand nombre d'outils performants (macro-éléments, couplage avec les méthodes intégrales, ...)

- . L'étude des phénomènes couplés conduit à définir un postprocesseur de traitement des résultats plus souples et qui peut être décrit à l'aide de notre générateur.

- . Les techniques d'intelligence artificielle utilisées permettent d'envisager des contrôles syntaxiques et sémantiques beaucoup plus poussés que ceux que nous avons implantés sur notre prototype.

Les interactions hommes-machines doivent être assouplies et adaptées aux desiderata des utilisateurs de notre système.

- . Les performances enfin doivent être améliorées afin que le coût d'utilisation des systèmes générés soit compatible avec des analyses tridimensionnelles de taille conséquente.

- . Un travail technique considérable doit donc encore être réalisé, mais c'est l'aspect humain de notre recherche qui est certainement le plus ouvert. En effet, nous pensons avoir créé avec ce prototype une nouvelle classe d'outils dont l'utilisation demeure encore délicate. C'est dans le domaine de la formation des

utilisateurs de tels systèmes qu'il faudra à notre avis investir le plus de travail. Cet effort devrait permettre d'approfondir nos connaissances dans la méthodologie de la modélisation et ainsi de mieux adapter ces nouveaux outils aux besoins des enseignants, des chercheurs et des ingénieurs.

B I B L I O G R A P H I E

MODELISATION PHYSIQUE (OUVRAGES GÉNÉRAUX)

- [1] BATCHELOR G.K.
"An introduction to fluid dynamics". Cambridge University Press (1967).
- [2] CAMBEL A.B.
"Plasma physics and Magnetofluid mechanics". Mc Graw Hill Ed (1963).
- [3] DURAND E.
Electrostatique. T. I, II et III, Masson et Cie, Ed. Paris (1964).
- [4] DURAND E.
Magnétostatique. Masson et Cie, Ed. Paris (1968).
- [5] FOURNET G.
"Electromagnétisme à partir des équations locales. Masson ed. Paris (1979).
- [6] GOSMAN A.D., PUN W.M., RUNCHAL A.K., SPALDING D.B., WOLFHSTEIN M.
"Heat and mass transfer in recirculation flows". Academic Press London N.Y. (1969).
- [7] JACKSON J.D.
"Classical Electrodynamics". J. Wiley & Son Ed. (1967).
- [8] KNUDSEN J.G., KATZ D.L.
"Fluid dynamics and heat transfer". Mac Graw-Hill Ed. (1958).

- [9] MORSE FESHBACH
"Methods of theoretical physics. Mc Graw-Hill (1953).
- [10] ROBERTS P.H.
"An introduction to M. H. D.". American Elsevier publishing company inc N.Y. (1967).
- [11] STRATTON J.A.
"Théorie de l'électromagnétisme". Dunod (1961).
- [12] WOODSON J.R., MELCHER H.H.
"Electromechanical dynamics". Part III "Elastic and Fluid media". J. Wiley & Son Editors (1968).
- [13] YUAN S.W.
"Foundations of fluid mechanics". Prentice Hall Ed. (1967).

MODELISATION NUMERIQUE, ELEMENTS FINIS

(OUVRAGES GÉNÉRAUX)

- [14] AZIZ
"Mathematical foundations of the Finite Element Method with applications to partial equations". Acad-press N.Y. (1972).
- [15] BOUJOT J.P., MORERA J.P., TEMAM R.
"Colloque International sur les méthodes scientifiques et techniques". IRIA France (1973).
- [16] BREBBIA C.
"Integration of area and volume coordinates in finites elements". A.I.A.A Journal, vol. 7, n°6, p 1212 (1969).
- [17] BREBBIA C.A., FERRANTE A.J.
"Computation Methods for the Solution of Engineering Problems". Pentech Press, London (1978).
- [18] CHARI M.V.K., SILVESTER P.P.
"Finite Elements in Electrical and Magnetic Field Problems". Wiley, New-York (1960).
- [19] CHUNG T.J.
"Finite element analysis in fluid dynamics". Mc Graw Hill (1978).
- [20] FINLAYSON B.A.
"The method of weighted residuals and variational principles". Academic Press N.Y. (1972).

- [21] FINLAYSON B.A.
"Weighted residual methods and their relations to finite element methods in flow problems". Finite elements in fluids, vol. 2, p 1.29, J. Wiley & Son, Ed.
- [22] FIX G.F., NASSIF N.
"Finite element approximations to time dependent problems". Numerische Mathematik, vol. 19, n°2 (1972).
- [23] FORTIN M.
"Approximation des fonctions à divergence nulle par la méthode des éléments finis". In Proc 3rd Int. Conf. Num. Meth. in Fluid Mec . (ed by CABANNES TEMAM) (1972).
- [24] GALLEGHER R.H.
"Finite element analysis : fundamentals". Prentice Hall, Englewood Cliffs, New Jersey (1975).
- [25] IRONS B.M.
"Quadrature rules for brick based finite elements". Int. J. Num. Meth. Eng. 3, pp 293-294 (1971).
- [26] S.G. MIKHLIN
"Variational methods in mathematical physics". Bergamon Press (1964).
- [28] MITCHELL A.R., WAIT R.
"The finite element method in partial differential equations". Wiley, London (1977).
- [29] MORTON K.W.
"Finite difference and finite element methods". Computer Phys. Com., vol. 12, numb 1, Sept-Oct (1976).
- [30] NORRIE D.H., Dr VRIES G.
"A finite element bibliography". Reports 57,58,59, University of Calgary Canada (May 1975).

- [31] ODEN J.T.
"Finite elements of non linear continua". Mc Graw Hill
(1972).
- [32] OWEN D.R.J., HINTON E.
"Finite elements in plasticity". Pineridy Press (1980).
- [33] RAVIART P.A.
"Les méthodes d'éléments finis en mécanique des fluides".
CEA-EDF-INRIA Ecole d'été d'analyse numérique. Col. de la
Direction des Etudes et Recherches E.D.F.
- [34] SILVESTER P.
"High-order polynomial triangular finite elements for
potential problems". Int. J. Enong Sci., vol. 7, pp 481-
497 (1969).
- [35] SILVESTER P., KONRAD A.
"Axisymmetric triangular finite elements for scalar helm-
holtz equation". Int. J. For Num. Meth. In Engng, vol. 5,
pp 481-497 (1973).
- [36] TEMAM R.
"Navier-Stokes equations". North Holland, Amsterdam (1977).
- [37] VARGA R.S.
"Matrix iterative analysis". Prentice Hall N.Y. (1962).
- [38] VICHNEVETSKY R.
"Stability charts in the numerical approximation of partial
differential equations : a review/mathematics and computers
in simulation XXI, pp 170-177 (1979).
- [39] ZIENKIEWICZ OC.
"The finite element method in engineering Sciences". Mc
Grw Hill, London (1971).

[40]

ZIENKIEWICZ O.C., KELLY D.W., BETTESS P.

"The coupling of the finite element method and boundary solution procedures". Int. J. For Num. Meth. in Engineering, vol. 11, pp 355-375 (1977).

152

ZIENKIEWICZ O.C.

"Coupled problems and their numerical solutions". International Conference on coupled problems (Déc. 1981).

MODELISATION NUMERIQUE DES PHENOMENES ELECTROMAGNETIQUES

- [36] ABDEL-RAZEK A., COULOMB J.L., FELIACHI M., SABONNADIERE J.C.
"Conception of an air gap element for the dynamic analysis of the electromagnetic field in electrical machines".
Proc. of COMPUMAG Conf., Chicago (1981).
- [42] ARMOR A.F., CHARI M.V.K.
"Heat flow in the stator core of large turbine-generators by the method of three-dimensional finite elements".
Parts I & II (1976).
- [43] ARMOR A.F.
"Transient, three-dimensional, finite element analysis of heat flow in turbine-generator rotors". IEEE Trans., PAS 99, n°3, (1980).
- [57] BARTON M.L.
"Loss calculation in laminated steel utilizing anisotropic magnetic permeability". IEEE Trans., PAS-99, n°3, pp 1280-1287 (1980).
- [58] CALLEGHER E., MOREL B., SABONNADIERE J.C., ANCELLE B., COULOMB J.L., MASSE Ph., MEUNIER G.
"Mode d'emploi du système FLUX". Rapport interne, Lab. Electrotechnique de Grenoble (1980).
- [59] CARPENTER C.J.
"Comparison of alternative formulations of 3-dimensional magnetic-field and eddy-current problems at power frequencies". Proc. IEE, vol. 124, n°11, pp 1926-1934 (1977).

- [60] CHARI M.V.K., CSENDES Z.J., SILVESTER P., KONRAD A.,
PAIMO M.A.
"Three-dimensional magnetostatic field analysis of electrical machinery by the finite element method". IEEE PAS Winter Meeting, paper 81, WM 175-9 A, Atlanta, Georgia (1981).
- [61] COULOMB J.L.
"Finite element three dimensional magnetic field computation". Intermag, Grenoble, invited paper (1981).
- [62] DEMERDASH N.A., NEHI T.W., FOUAD F.A.
"Finite element formulation and analysis of three dimensional magnetic field problems". IEEE Trans., MAG-16, n°5, pp 1092-1094 (1980).
- [63] FOGGIA A., SABONNADIÈRE J.C., SILVESTER P.
"Finite element solution of saturated travelling magnetic field problems". IEEE Trans., PAS-94, n°3 (1975).
- [64] HAMMOND P.
"Use of potentials in calculation of electromagnetic fields". IEE Proc., vol. 129, n°2, pp 106-112 (March 1982).
- [65] KONRAD A., SILVESTER P.
"Triangular finite elements for the generalized Bessel equation of order n". Int. J. for Num. Math. in Eng., vol. 7, pp 43-55, (1973).
- [66] KONRAD A., SILVESTER P.
"A finite element program package for axisymmetric scalar field problems". Computer Physics Communications, pp 437-455 (1973).
- [67] KOTIUGA P.R., SILVESTER P.P.
"Vector potential formulation for three-dimensional magnetostatics".

- [68] LAJOIE-MAZENC M., HECTOR J., FAUCHER
"Notice d'utilisation de DIFIMEDI code conversationnel de calcul du champ dans les structures magnétiques". Lab. d'Electrotechnique et d'Electronique Industrielle, Toulouse France (1980).
- [69] LOWTHER D.A., SILVESTER P.P., FREEMAN E.D.
"The use of interactive postprocessing in the design of electromagnetic devices". IEEE Trans., MAG-16, n°5, pp 803-805 (1980).
- [70] MAGNET 78
"User's Manual". Mc Gill University, Montréal, Canada (1978).
- [71] MARROCCO A.
"Analyse numérique de problèmes tridimensionnels en magnétostatique". Lecture note in Physics, Springer-Verlag, pp 217-236 (1975).
- [72] PALMO M.A., CZENDES Z.J., CHARI M.V.K.
"Axisymmetric and three dimensional electrostatic field solutions by the finite element method". Elect. Mach. Electromech. Q, pp 235-244 (1979).
- [73] POLAK S.J., WATCHERS
"Maggy 2 and Paddy program packages for 2 and 3 dimensional magnetostatic problems". Proc. of COMPUMAG Conf. Grenoble (1978).
- [74] RAFINEJAD P.
"Adaptation de la méthode des éléments finis à la modélisation des systèmes électromécaniques de conversion d'énergie". Thèse de Doctorat d'Etat ès-Sciences, Grenoble (1977).

- [75] RAFINEJAD P., COULOMB J.L., MEUNIER G.
"Permanent magnet 3D field computation by finite elements".
Proc of COMPUMAG Conf Grenoble (1978).
- [76] SABONNADIÈRE J.C., NICOLAS A.
"A three dimensional analysis method for induction machine".
IEEE Winter Meeting, New-York, paper A 76 241-7 (1976).
- [77] SILVESTER P., RAFINEJAD P.
"Curvilinear finite elements for two-dimensional saturable
magnetic fields". IEEE Trans., PAS-93, n°6, pp 1861-1870
(1974).
- [78] SIMKIN J., TROWBRIDGE C.W.
"On the use of the total scalar potential in the numerical
solution of field problems in electromagnetics". Int. J.
for Numerical methods in Engineering, 14, pp 423-440
(1979).
- [79] SIMKIN J., TROWBRIDGE C.W.
"Three dimensional computer program (TOSCA) for non-linear
static electromagnetic fields". Rutherford Laboratory
Oxon U.K. (
- [80] YOUNG F.J.
"On the validity of characterizing eddy current phenomena
by parabolic partial equations". IEEE Trans., MAG-14,
n°6, pp 1185-1186 (1978).
- [81] ZAKY S.G., ROBERTSON S.D.T.
"Integral equation formulation for the solution of magne-
tic field problems". Parts I & II, IEEE Trans., PAS-92,
n° 2, pp 808-823 (1973).
- [82] ZIENKIEWICZ O.C., LYNESS J., OWEN D.R.J.
"Three-dimensional magnetic field determination using a
scalar potential. A finite element solution". IEEE Trans.
MAG-13, n° 5, pp 1649-1656 (1977).

PHENOMENES HYDRAULIQUES

- [83] BABA K., TABATA M.
"On a conservative upwind finite element scheme for convective diffusion equations". RAIRO Numerical Analysis, vol. 15, n°1, pp 3-25 (1981).
- [84] BAKER A.J.
"Navier-Stokes solutions using finite element algorithm". In Proc. of 4th Conf. on Num. Meth. in fluid dynamics June Univ. of Colorado (1974).
- [85] BREBBIA A., CONNOR J.J.
"Numerical methods in fluid dynamics". Proc. of int. conf. in Southampton, London Pentech Press (1973).
- [86] BREBBIA C.A., WROBEL L.C.
"Steady and unsteady potential problems using the boundary integral method in recent advances in numerical methods in fluids pp 1-26, edited by TAYLOR & MORGAN Pineridge Press (1980).
- [87] BRISTEAU M.O., GLOWINSKI R., MANTEL B., PERIAUX J., PERRIER P., PIRONNEAU O.
"A finite element approximation of Navier-Stokes equations for incompressible viscous fluid, iterative methods of solution". Lecture notes in mathematics Proc of Approximation Methods for Navier-Stokes Problems : Paderborn Germany, pp 78-128 (1979).

- [88] CALLAGHER, RICHARD
"International Conference on the Finite elements in fluids".
Univerty College of Wales (Swansea), J. Wiley & Son
Editors, (Jan 1974).
- [89] CHRISTIE, GRIFFITHS D.F., MITCHELL A.R., ZIENKIEWICZ O.C.
"Finite element methods for second order differential
equations with significant first derivatives". Int. J. Num.
Meth. Eng., 10, pp 1389-1396 (1976).
- [90] CHUNG T.J., CHIOU J.N.
"Analysis of unsteady compressible boundary layer flow
via finite elements". Computer and fluids, vol. 4, pp 1-12,
(1976).
- [91] COOKE H., BLANCHARD D.K.,
"A higher order finite element algorithm for the unsteady
Navier-Stokes equations". Mathematics and computers in
simulation XXII, pp 127-132 (1980).
- [92] CROUZEIX M., RAVIART P.A.
"Conforming and non-conforming finite element methods for
solving the stationary Stokes equations". Rev. Franc. Autom.
Inf. Rech. Oper., V7, pp 33-75 (1973).
- [93] DARCY G.P., SCHMIDT P.S.
"MHD Entry flow for a plane channel in an axial magnetic
field". Journal of Fluid Mechanics, vol. 80, part 2,
p 209 (April 25 1977).
- [94] DE SOCIO, SPARROW, ECKERT
"The contrived transient explicit method for solving
steady-state flows : applications to a rotating, recircu-
lating flow". Computers and Fluids, vol. 1, pp 273-287
(1973).

[95] FALK R.S.

"An analysis of the penalty method and extrapolation for the stationary stokes equations". Advances in Comp. Meth. for Partial Diff. Equations ed by : Vichnevelsky (Proc AICA) Lehigh University, pp 66-69 (June 1975).

[96] FAKL R.S.

"A finite element method for the stationary stokes equations using trial functions which DO NOT have to satisfy $\text{DIVU}=0$ ". Mathematics of computation V30, n°136, pp 698-702 (Oct 1976).

[97] FAKL R.S., KING J.T.

"A penalty and extrapolation method for the stationary stokes equations". SIAM Journal Numer. Ann. V13 (1976).

[98] FAKL R.S.

"An analysis of the finite element method using Lagrange multipliers for the stationary stokes equation". Math. Comp. V30, pp 241-249 (1976).

[99] FAUTRELLE Y.

"Deux aspects de la magnéto-hydrodynamique la dynamo convective, le brassage dans les fours à induction". Thèse d'Etat Grenoble (1982).

[100] FAUTRELLE Y.,

"Analytical and numerical aspects of the electromagnetic stirring induced by alternating magnetic fields". J. fluid Mec., 102, pp 405-430 (1981).

[101] GALLAGHER, ODEN, TAYLOR, ZIENKIEWICZ

"Finite elements in fluids". J. Wiley & Son Editors, vol. 1 & 2 (1974).

- ~~102~~ GETSELEV Z.M., MARTYNOV G.I.
"Calculation of the velocity induced in the liquid phase of a casting by electromagnetic forces". Magnetohydrodynamics V.11, pp 217-221 (1975).
- ~~103~~ GRAND D.
"Contribution à l'étude des courants de recirculation". Thèse d'Etat, Grenoble (1975).
- ~~104~~ GRESHO P.M., LEE R.L.
"On the time dependant solution of the incompressible Navier-Stokes equations in two and three dimensions". In recent advances in numerical methods in fluids, pp 27-80, edited by TAYLOR & MORGAN Pineridge Press (1980).
- ~~105~~
~~106~~ HEINRICH J.C., ZIENKIEWICZ O.C.
"Quadratic finite element schemes for two-dimensional convective transport problems". Int. J. Num. Meth. Engn 11, pp 1831-1844 (1977).
- ~~107~~
~~108~~ Ikegawa M.
"A new finite element technique for the analysis of steady viscous flow problems". Int. Jour. for Num. Meth. in Eng., vol. 14, pp 103-113 (1979).
- ~~109~~ IKENOUCHI DIMURA
"An approximate numerical solution of the Navier-Stokes equations by Galerkin method (in finite element in flow problems).
- ~~110~~ JAMET P., RAVIART P.A.
"Numerical solution of the stationary Navier-Stokes equations by finite element methods". Lecture Note in Computer Science, vol. 10 (computing methods in applied sciences and engineering Par 1 of Internat. Symposium Versailles (1973)). Ed by LIONS J.L. GLOWINSKI R., pp 193-223 Springer Verlag N.Y. (1974).

- [111] KAWAHARA M., YOSHIMURA N., NAKAGAWA K.
"Steady flow analysis of incompressible viscous fluid by finite element method". In theory and practice in finite element structural analysis (ed by YAMADA & GALLAGHER) univ of TOKIO Press, pp 557-572 (1973).
- [112] KAWAHARA M., TAKEUCHI N.
"Mixed finite element method for analysis of viscoelastic fluid flow". Computer and fluids, vol. 5, n°1, pp 33-45 (1977).
- [113] KHALETZKY D.
"Etude théorique du brassage électromagnétique dans les fours à induction. Application à l'analyse d'un procédé de séparation d'inclusions". Thèse de 3ème Cycle, U.S.M. Grenoble (1976).
- [114] KOCHETKOVA G.Y., STOLOV M.Y., TIR L.L., CHAIKIN P.M.
"The circulation of metal in an induction furnace". Magneto-hydrodynamics, vol. 2, n°2 (1966).
- [115] LAUNDER B.E., SPALDING D.B.
"The numerical computation of turbulent flows". Computer methods in applied mechanics and engineering, 3, pp 269-289 (North holland publishing company) (1974).
- [116] LEE C.H.
"Finite element method for transient linear viscous flow problems". In Num. Meth in fluid dynam. Proceed. Int. Conf. Southampton, pp 140-152 (1973 Sept).
- [117] MASLIYAH J.H., KUMAR D.
"Application of orthogonal collocation on finite elements to a flow problem". Mathematics and Computers in Simulation XXII, pp 49-54 (1980).

- [118] MOULT A., BURLEY D., RAWSON H.
"The numerical solution of two-dimensional steady flow problems by the finite element method". Int. Jour. Num. Meth. in Eng., vol. 14, pp 11-35 (1979).
- [119] NORRIE D.H., DE VRIES, BERARD
"Application of the finite element technique to compressible flow problems". Mech Eng Rep 18 and 51, Dep of Mech Eng univ of Calgary (1973 & 1974).
- [120] ODEN J.T.
"Finite element analogue of Navier-Stokes equations". Journal Eng Mech Div Proc. ASCE 96, n° EM4, pp 529-534, (1970).
- [121] ODEN J.T., WELLFORD L.C.
"Analysis of flow of viscous fluids by the finite element method". J. of AIAA 10.12, pp 1590-1599 (1972).
- [122] OLSON M.D.
"A variational finite element method for two-dimensional steady viscous flows". Mc GILL EIC Conf on finite element method in civil eng, MONTREAL, pp 585-616 (June 1972).
- [123] ROACHE P.J.
"Computational fluid dynamics". Hermosa Publishers.
- [124] ROBERT D., RICHTMYER
"Proceedings of the fourth international Conference on Numerical Methods in fluid dynamics". University of Colorado in Lecture Notes in Physics SPRINGER VERLAG, (June 1974).
- [125] SKIBA E.
"A finite element solution of general fluid dynamics problems natural convection in rectangular cavities". M. App. Sci Civ Eng thesis University of Waterloo Ontario, Canada (April 1970).

- [126] TARAPORE E.D., EVANS J.W.
"Fluid velocities in induction melting furnace". Part I
Theory and Laboratory Experiments (Met. Trans.) vol. 7,
B p 343 (Septembre 1976).
- [127] TAYLOR C., HOOD P.
"A numerical solution of the Navier-Stokes equations
using the finite element technic". Computer and fluids,
vol. 1, pp 73-100 (1973).
- [128] TAYLOR C., HUGHES T.G., MORGAN K.
"A finite element model of one and two equations models
of turbulent flow". In recent advances in numerical methods
in fluids, pp 311-334, edited by TAYLOR & MORGAN Pineridge
Press (1980).
- [129] TEMAM R., THOMASSET F.
"Solution of Navier-Stokes equations by finite elements
method". In Proc of 4th Int Conf on Num Metho in Fluid
Dynamics, Univ of Colorado, pp 392-395 (1974).
- [130] TIR L.L.
"Modélisation du mouvement d'un métal liquide dans un four
à induction". Magnytnaya Guidrodinamica, vol. 1, n°4,
(1965).
- [131] TIR L.L.
"Vortex forces in a liquid metal in the field of a single
phase inductor". Magnetohydrodynamics, vol. 10, pp 101-107.
- [132] TONG P.
"The finite element method for fluid flow". In recent
advances in matrix methods of structural analysis and
design (ed by GALLAGHER) Univ of Alabama Press, pp 787-
808 (1971).

- [133] TSEBES A.O.
"Interfacial stresses in the hydrodynamics of liquids with internal rotation". Magnetohydrodynamics, vol. 11, n° 1, pp 63-65 (1975).
- [134] SIPLIVYI B.N.
"Approximate calculation of velocity field in flat and cylindrical MHD channels". Magnetohydrodynamics, vol. 11, pp 164-170 (1975).
- [135] SZEKELY J., YADOYA R.T.
"The physical and mathematical modelling of the flow field in the mold region in continuous casting systems". Part II "The mathematical representation of the turbulent flow field", metallurgical trans, vol. 4 (May 1973).
- [136] SZEKELY J., NAKANISHI K.
"Part 2 "Mathematical representation of the turbulent flow field and of tracerdispersion". Metallurgical trans, vol. 6B (June 1975).
- [137] VOLCHEK B.B., GEKHT G.M., TOLMACH I.M., EL'KIN A.I.
"Stationary flows in coaxial channel of MHD induction pump caused by hydrodynamic instability". Magnetohydrodynamics, vol. 12, n° 2 (1976).
- [138] WENGER N.C.
"Variationnal principle for MHD channel flow". J. Fluid. Mech, vol. 43, n°1, pp 211-224 (1970).
- [139] WU S.T.
"Unsteady MHD duct flow by the finite element method". Int J. Numer Meth Eng, vol. 6, n°1, pp 3-10 (1973).
- [140] YAMADA ITO, YOKOUCHI, TAMANO, OHTSUBO
"Finite element analysis of steady fluid and metal flow". In finite elements in fluids, vol. 1, J. Wiley & Son Ed. pp 73-94.

PROBLEMES COUPLES ET M H D

- [141] APPERT, BERGER, GRUBER, TRAYON, ROBERT
"THALIA a one-dimensional magnetohydrodynamics stability program using the method of finite elements". Computer physics communications, vol. 10, n° 1, pp 11-29 (July 1975).
- [142] APPERT, TRAN, VACLAVIK
"Finite element approximation for the wave-particle interaction in weakly turbulent plasmas". Computer physics communications, vol. 12, n°2, pp 135-144 (Nov 1976).
- [143] BERGER, GRUBER, TRAYON
"A finite element approach to the computation of the MHD spectrum of straight non circular plasma equilibria". Computer physics communications, vol. 11, n°3, pp 313-232 (1976).
- [144] BOUJOT J.P., MERCIER C., SOUBBARAMAYER
"Numerical simulation of the evolution of a plasma in confinement device". First European Conference, Geneve (1972).
- [145] GRIMM R.C., JOHNSON J.L.
"Recent developments in the computational aspects of MHD stability". Computer physics communications, vol. 12, n°1, pp 45-52.
- [146] LACKNER K.
"Computation of ideal MHD equilibria". Computer physics communications, vol. 12, n°1, pp 33-44 (Sept/Oct 1976).

[147] LEWIS, GARNER

"Finite element solution of coupled electrokinetic and hydrodynamic flow in porous media". Int. J. Numer. Meth. Eng, vol. 5, n°11, pp 41-55 (Sept/Oct 1972).

[148] MIYA A.N., ANDO

"Application of finite element method on electro-magneto-mechanical dynamics". In proc of 6th Symposium on Eng Prob of Fusion Research San Diego, pp 927-934 (Nov 1975).

[149] MIYA K.

"Dynamic analysis of electro-magneto-mechanical field by finite element method". Journal of the faculty of engineering, university of Tokio (B), vol XXXIV, n°1, pp 137-157 (1977).

[150] NEMIROVSKII Y.U.V., KHEINLOO Y.A.L.

"Rotational anisotropic turbulent flows of electrically conducting media in external electric and magnetic fields". Magnetohydrodynamics, vol. 12, n°2 (1976).

[151] SAMSONOV A.M.

"Asymptotic solution of the problem of heat exchange in an induction magnetohydrodynamic apparatus". Magneto-hydrodynamics, vol. 11, pp 480-484 (1975).

152

RESOLUTION DE PROBLEME
ET INTELLIGENCE ARTIFICIELLE

- [153] ALLAN J.J. III
"Foundations of the many manifestations of CAD". Proceedings of IFIP working conf on principles of CAD, Eindhoven, pp 27-58 (Oct 1972).
- [154] ANCELLE B., COULOMB J.L., MASSE Ph., MEUNIER G., SABONNADIERE J.C.
"Computer methods for electrical and magnetic devices designed by field analysis". IEEE, MAG-15, n°6, pp 1671-1673 (1979).
- [155] ANCELLE B., COULOMB J.L., MASSE Ph., MEUNIER G., SABONNADIERE J.C.
"Software for computed-aided analysis of electromagnetic fields". IEEE, MAG-16, n°6, pp 1435-1437 (1980).
- [156] ANCELLE B., MASSE Ph., CALLEGHER E.
"IMPUT : a fully parametric pre-processor for computer aided design of magnetic devices". Proc. of COMPUMAG Conf Chicago (1981).
- [157] BONNET A..
"Transfert interactif d'expertise : expérience avec EMYCIN pour la construction du programme LITHO". Journée sur les systèmes EXPERT, Avignon, pp 1-14 (Mai 1981).
- [158] COULOMB J.L., MASSE Ph., SABONNADIERE J.C.
"How to build your own CAD system : the CASCADE program and its application to finite elements program design". Proc. of Numerical Methods in Electrical and Magnetic Field Problems, Santa Margherita Ligure (1976).

[159] DAVID B.

"Pour une généralisation des systèmes CAO approches et applications". Thèse 3ème Cycle, Grenoble (Oct 1975).

[160] JACQUART R., REGNIER P., VALETTE F.R.R.

"GERMINAL toward a general and integrated system for CAD". Journées sur les systèmes généraux pour la CAO, Toulouse (1974).

[161] JACQUART R., REGNIER P., VALETTE F.R.R.

"GERMINAL un système intégré et conversationnel pour la CAO". Automatisme, T XIX, n°3 (Mars 1974).

[162] LATOMBE J.C.

"ESPACE : a computer aided instruction system oriented towards educational problem solving activities". R.G.U. 74, Conference on computer based learning system, Hambourg p 10 (August 1974).

[163] LATOMBE J.C.

"Artificial intelligence in computer aided design : the TROPIC system". Proc. IFIP working conference 5.2 on CAD systems (pub by North Holland Publish CO) (Feb 1976).

[164] LOWTHER D., REA K., FREEMAN E., TROWBRIDGE C., SILVESTER P., SIMKIN J., NEWMAN M.

"A stack configured vector calculator for electromagnetic field evaluation".

[165] LOWTHER D.A.

"A microprocessor based electromagnetic field analysis system".

[166] MASSE Ph., SABONNADIÈRE J.C.

"A computer aided instruction system for network engineering education". Conference on "The future progress of electrical engineering", Prague, p 9 (Oct 1975).

- [167] MASSE Ph., SABONNADIÈRE J.C.
"CASCADE une aide à la conception de système d'enseignement assisté par ordinateur". Panorama de la nouveauté informatique en France, p 8 (Nov 1976).
- [168] MASSE Ph., ANCELLE B., COULOMB J.L.
"System methodology in CAD programs based on finite element method". Proc of COMPUMAG Conf, Chicago, invited paper (1981).
- [169] NAU D.S.
"Expert computer systems". IEEE Computer, pp 63-85 (February 1983).
- [170] NILSSON N.J.,
"Problem solving methods in artificial intelligence". McGraw Hill, New-York (1971).
- [171] PONCET A.
"Autour de l'écriture d'un code d'éléments finis". Thèse Doctorat ès Sciences Mathématiques, Grenoble (1979).
- [172]
[173] ROOS D., RODRIGUEZ J.E.
"Theoretical foundations for the CAD systems". AFIPS, proc SJCC, vol. 23, (1963).
- [174] ROOS D.
"ICES System : general description". MIT Department of civil engineering report, R 67-49, (Sept 1967).
- [175] SCHREM E.
"Trends and aspects of the development of large finite element software systems". Computers and structures, pp 419-426 (1979).

A N N E X E S

EXEMPLES D'UTILISATION

C115 10-11-12
Atelier Micro Informatique
S.P. 53 X 20041 GRENOBLE CEDEX
Tel. (73) 41.55.90

Dans ce document, annexe à notre mémoire de thèse, nous présentons plusieurs utilisations de systèmes construits à l'aide de notre logiciel FLUX-EXPERT :

- a) .Un extrait des bases de données éditées par notre système lors de la construction d'un système objet.
(Nous avons choisi le système résolvant des problèmes thermodynamiques dont le détail de la construction a été fourni au chapitre C.III.2.).
- b) .Un exemple d'utilisation du système objet destiné à traiter des problèmes magnétodynamiques linéaires en coordonnées axisymétriques :
Etude magnétique d'un dispositif de chauffage par induction.
- c) .Un exemple d'utilisation du système objet destiné à traiter des problèmes thermodynamiques bidimensionnels :
Etude transitoire du comportement d'un fusible soumis à une surintensité.
- d) .Un exemple d'utilisation du système objet destiné à traiter les problèmes d'hydrodynamique :
Etude bidimensionnelle d'un écoulement visqueux dans un chenal en forme de T.

EXTRAIT DES BASES DE DONNEES EDITEES
LORS DE LA CONSTRUCTION D'UN SYSTEME OBJET

Dans la troisième partie de notre mémoire de thèse, nous avons présenté des exemples de construction de systèmes objets à l'aide de notre générateur. Nous avons indiqué qu'il était nécessaire que le constructeur d'un système prépare soigneusement son travail avant d'aborder la phase de description interactive. Pour l'aider dans cette tâche, notre système permet l'édition de nombreux documents soit sur écran soit sur imprimante. Ce sont des extraits de ces documents que nous présentons ci-après ; ils ont été réalisés lors de la conception du système objet destiné à la thermodynamique dont nous avons décrit la construction au chapitre C.III.2 ; ils comportent trois parties :

- Le contenu de la base NOYAU-OPERATEUR.

On pourra trouver dans ce document la description de tous les opérateurs qui étaient alors disponibles dans le NOYAU de notre générateur.

On remarquera que chaque opérateur est décrit dans le détail et que les formules des opérateurs de construction, des expressions intégrantes et des poids différentiels sont explicites et facilitent beaucoup la lecture de cette documentation.

- Le contenu de la base OBJET-VARIABLE.

Cette base de données contient les variables utilisées dans la construction du système objet thermodynamique. On pourra remarquer que chaque propriété peut être modélisée avec

une grande souplesse grâce aux opérateurs de calcul de propriétés.

Les commentaires qui explicitent chaque variable permettent une lecture plus aisée de ce document et sont utilisés dans notre système pour présenter le détail des menus à choix multiples dans les dialogues mis en oeuvre par le système objet construit.

- Le contenu de la base OBJET-EQUATION.

On retrouvera dans cette base de données le détail de la description des équations de la modélisation thermodynamique présentée aux chapitres A.IV et C.III.2.

Les expressions intégrantes y sont décrites avec le plus grand soin. L'utilisation du formateur spécialement conçu pour retranscrire en clair ces formules facilite grandement les vérifications minutieuses qui sont nécessaires lors de la description de ces expressions assez complexes.

On remarquera la simplicité de la description des matrices constitutives et l'utilité des formules fournies dans les opérateurs de construction.

NOYAU-OPERATEURS

Operateurs calcul de polynomes

Programme polynome :POLLN2 : lagrange 1ier degre sur un segment
Degre :SOUS-parametrique
Nature :LINEIQUE
Programme coord. poids associe :GAUL1D : poids et abscisses sur un segment
Nombre de noeuds : 2

Programme polynome :POLLN3 : lagrange 2ieme degre sur un segment
Degre :ISO-parametrique
Nature :LINEIQUE
Programme coord. poids associe :GAUL1D : poids et abscisses sur un segment
Programme sous-parametrique associe POLLN2 : lagrange 1ier degre sur un segment
Nombre de noeuds : 3

Programme polynome :POLQU4 : lagrange 1ier degre sur un carre
Degre :SOUS-parametrique
Nature :SURFACIQUE
Programme coord. poids associe :GAUQ2D : poids et abscisses sur un quadrilatre
Nombre de noeuds : 4

Programme polynome :POLQU8 : lagrange 2ieme degre incomplet sur un carre
Degre :ISO-parametrique
Nature :SURFACIQUE
Programme coord. poids associe :GAUQ2D : poids et abscisses sur un quadrilatre
Programme sous-parametrique associe POLQU4 : lagrange 1ier degre sur un carre
Nombre de noeuds : 8

Programme polynome :POLQU9 : lagrange 2ieme degre complet sur un carre
Degre :ISO-parametrique
Nature :SURFACIQUE
Programme coord. poids associe :GAUQ2D : poids et abscisses sur un quadrilatre
Programme sous-parametrique associe POLQU4 : lagrange 1ier degre sur un carre
Nombre de noeuds : 9

Programme polynome :POLTR3 : coordonnees d aire sur un triangle ordre 1
Degre :SOUS-parametrique
Nature :SURFACIQUE
Programme coord. poids associe :GAUT2D : poids et abscisses sur un triangle
Nombre de noeuds : 3

Programme polynome :POLTR6 : coordonnees d aire sur un triangle ordre 2
Degre :ISO-parametrique
Nature :SURFACIQUE
Programme coord. poids associe :GAUT2D : poids et abscisses sur un triangle
Programme sous-parametrique associe POLTR3 : coordonnees d aire sur un triangle
Nombre de noeuds : 6

Operateurs de construction

Nom du programme :CSLIST : cas lineaire statique resolution directe
 $M * I = KS$
 Type des variables :REEL

Nom du programme :CSKAST : lineaire statique affinage par kantorovitch
 $J = M ; R = M * I - KS$
 Type des variables :REEL

Nom du programme :CSNRST : non lineaire statique newton raphson
 $J = M + DM ; R = M * I - KS$
 Type des variables :REEL

Nom du programme :CSLIDX : dynamique lineaire complexe resolution directe
 $(M + j.\omega.L) * I = KS + j.KV$
 Type des variables :COMPLEXE

Nom du programme :CSKADX : lineaire dynamique complexe affinage par kantorovi
 $J = (M + j.\omega.L) ; R = (M + j.\omega.L) * I - (KS + j.KV)$
 Type des variables :COMPLEXE

Nom du programme :CSLIDR : lineaire dynamique implicite resolution directe
 $(L + dt.M) * I = L * IO + dt.(KS + KV)$
 Type des variables :REEL

Nom du programme :CSKADR : lineaire dynamique implicite affinage kantorovitch
 $J = L + dt.M ; R = L * (I - IO) + dt.(M * I - KS - KV)$
 Type des variables :REEL

Nom du programme :CSNRDI : dynamique L non lineaire implicite demarrage
 $(L + dt.M) * I = L * IO + dt.(KS + KV)$
 Type des variables :REEL

Nom du programme :CSNRDR : dynamique L non lineaire implicite raphson
 $J = L + DL - DLO + dt.(M + DM) ; R = L * (I - IO) + dt.(M * I - KS - KV)$
 Type des variables :REEL

Nom du programme :CSCNIN : crank.nich. L lineaire demarrage direct
 $(L + dt.r.MO) * I = L * IO - dt.(s.MO * IO - KS - s.KVO - r.KV)$
 Type des variables :REEL

Nom du programme :CSCNTL : crank.nich. L lineaire iteration d espace raphson
 $J = L + dt.r.(M + DM) ; R = L * (I - IO) + dt.(r.M * I + s.MO * IO - r.KV - s.KVO - KS)$
 Type des variables :REEL

Nom du programme :CSCNIL : crank.nich. (M+L)lineaire demarrage direct
 $(L + dt.r.M) * I = L * IO - dt.(s.M * IO - KS - s.KVO - r.KV)$
 Type des variables :REEL

Nom du programme :CSCNKL : crank.nich.(M+L)lineaire affinagew par kantorovit
 $R = L * (I - IO) + dt.(M * (r.I + s.IO) - r.KV - s.KVO - KS)$
 Type des variables :REEL

Operateurs formules d integration

Nom du programme :GAUL1D : poids et abscisses sur un segment
Nature :LINEIQUE

Nom du programme :GAUT2D : poids et abscisses sur un triangle
Nature :SURFACIQUE

Nom du programme :GAUQ2D : poids et abscisses sur un quadrilatere
Nature :SURFACIQUE

Operateurs de resolution

Nom du programme :RESSRE : gradient conjuge avec preconditionnement
Type de la matrice :REEL
Structure de la matrice : SYMETRIQUE
Stockage de la matrice :SYMBOLIQUE

Nom du programme :RESBRE : gradient conjuge avec preconditionnement
Type de la matrice :REEL
Structure de la matrice : DISSYMETRIQUE
Stockage de la matrice :SYMBOLIQUE

Nom du programme :RESSCO : iccg pour nombres complexes
Type de la matrice :COMPLEXE
Structure de la matrice : SYMETRIQUE
Stockage de la matrice :SYMBOLIQUE

Operateurs de calcul des poids differentiels

Nom du programme :POIVID : sans parametre element differentiel seul
DW
1iere variable auxiliaire :VIDE : pas de variable auxiliaire
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POIAXS : axisymetrique scalaire sans propriete
R * DW
1iere variable auxiliaire :VIDE : pas de variable auxiliaire
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POIAXV : axisymetrique vectoriel sans propriete
/ R * DW
1iere variable auxiliaire :VIDE : pas de variable auxiliaire
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POAVC1 : axisym vectoriel propriete constante
P3 / R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POICC1 : cartesien propriete constante
P3 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POAVV1 : axisym vectoriel propriete variable
P3 / R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POASV1 : axisym scalaire propriete variable
P3 * R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POICV1 : cartesien propriete variable
P3 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POAVC2 : axisym vectoriel 2 propri : variable constante
P3 * P4 / R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POASC2 : axisym scalaire 2 propri : variable constane
P3 * P4 * R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee--
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POICC2 : cartesien 2 proprietes : variable constante
P3 * P4 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POAVV2 : axisym vectoriel 2 proprietes variables
P3 * P4 / R * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:UTILISEE : variable auxiliaire utilisee

Nom du programme :POINV1 : une propriete variable negative
(-P3)*DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POINV2 : 2 proprietes variables (valeur negative)
(-P3*P4)*DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POI4V1 : une propriete variable **4
P3*P3*P3*P3 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POI3V1 : une propriete variable au cube
P3*P3*P3 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :POI4V2 : P3VAR ** 4 * P4VAR
P3*P3*P3*P3 * P4 * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :PCNCV : = - P3cst * P4var
(-P3*P4)*DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :PCNC4 : = - P3cst ** 4
(-P3*P3*P3*P3) * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme :PCNC : = - P3cst
(-P3) * DW
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire
Rayon en axisymetrique:VIDE : pas de variable auxiliaire

Nom du programme : POASC1 : axisym scalaire propriete constante
P3 * R * DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : VIDE : pas de variable auxiliaire
Rayon en axisymetrique: UTILISEE : variable auxiliaire utilisee

Nom du programme : POASV2 : axisym scalaire 2 proprietes variables
P3 * P4 * R * DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique: UTILISEE : variable auxiliaire utilisee

Nom du programme : POICV2 : cartesien 2 proprietes variables
P3 * P4 * DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique: VIDE : pas de variable auxiliaire

Nom du programme : POI3V2 : derivee rayonnement (P3)**3 / P4 variable
P3*P3*P3 * P4 * DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique: VIDE : pas de variable auxiliaire

Nom du programme : POIN41 : rayonnement seul (valeur negative)
(-P3*P3*P3*P3)*DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : VIDE : pas de variable auxiliaire
Rayon en axisymetrique: VIDE : pas de variable auxiliaire

Nom du programme : POIN42 : rayonnement(valeur negative) et P4 variable
(-P3*P3*P3*P3)*P4*DW
1iere variable auxiliaire : UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire : UTILISEE : variable auxiliaire utilisee
Rayon en axisymetrique: VIDE : pas de variable auxiliaire

Operateurs de calcul d expressions integrantes

Nom du programme :INAASC : terme de diffusion propriete constante
/
! P1 * A * A
/ I J
Degre du 1ier polynome :ISO parametrique
Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
Degre du 2ieme polynome :VIDE le polynome n'est pas utilise
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
Type de l'integrand calcule :MATRICE
Structure du resultat :SYMETRIQUE

Nom du programme :INAASV : terme de diffusion sans propriete
/
! A * A
/ I J
Degre du 1ier polynome :ISO parametrique
Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
Degre du 2ieme polynome :VIDE le polynome n'est pas utilise
1iere variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
Type de l'integrand calcule :MATRICE
Structure du resultat :SYMETRIQUE

Nom du programme :INAVGC : transport vitesse variable P2 constante
/
! A * (P1 . GRAD(A)) * P2
/ I J
Degre du 1ier polynome :ISO parametrique
Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
Degre du 2ieme polynome :ISO parametrique
Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
Type de l'integrand calcule :MATRICE
Structure du resultat :DISSYMETRIQUE

Nom du programme :INAVGV : transport vitesse variable seule
/
! A * (P1 . GRAD(A))
/ I J
Degre du 1ier polynome :ISO parametrique
Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
Degre du 2ieme polynome :ISO parametrique
Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
Type de l'integrand calcule :MATRICE
Structure du resultat :DISSYMETRIQUE

Nom du programme :INBDBV : derivee de la conduction P1=grad(A) ou rot(A)
/
! (P1 . GRAD(A)) * (P1 . GRAD(A))
/ I J
Degre du 1ier polynome :VIDE le polynome n'est pas utilise
Degre du 2ieme polynome :ISO parametrique
Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
Type de l'integrand calcule :MATRICE
Structure du resultat :SYMETRIQUE

Nom du programme :INGGTC : conduction tensorielle propriete constante
 /
 ! P1 \$ GRAD(A) \$ GRAD(A)
 / I J
 Degre du 1ier polynome :VIDE le polynome n'est pas utilise
 Degre du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :SYMETRIQUE

Nom du programme :INGGTV : conduction tensorielle propriete variable
 /
 ! P1 \$ GRAD(A) \$ GRAD(A)
 / I J
 Degre du 1ier polynome :VIDE le polynome n'est pas utilise
 Degre du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :SYMETRIQUE

Nom du programme :INTAC : sources constantes
 /
 ! P1 * A
 / I
 Degre du 1ier polynome :ISO parametrique
 Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
 Degre du 2ieme polynome :VIDE le polynome n'est pas utilise
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :VECTEUR
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INTAV : sources sans propriete
 /
 ! A
 / I
 Degre du 1ier polynome :ISO parametrique
 Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
 Degre du 2ieme polynome :VIDE le polynome n'est pas utilise
 1iere variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :VECTEUR
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INGGSC : conduction propriete constante
 /
 ! P1 * (GRAD(A) . GRAD(A))
 / I J
 Degre du 1ier polynome :VIDE le polynome n'est pas utilise
 Degre du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :SYMETRIQUE

Nom du programme :INTGVC : sources vectorielles constantes de type aimant
 /
 ! P1 . GRAD(A)
 / I
 Degré du 1ier polynome :VIDE le polynome n'est pas utilise
 Degré du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :VECTEUR
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INTGVV : sources vectorielle variable (type aimant)
 /
 ! P1 . GRAD(A)
 / I
 Degré du 1ier polynome :VIDE le polynome n'est pas utilise
 Degré du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :VECTEUR
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INVGAC : derivee conduc P1=GRAD(T) type thermique P2=cs
 /
 ! (P1 . GRAD(A)) * A * P2
 / I J
 Degré du 1ier polynome :ISO parametrique
 Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
 Degré du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INVGAV : derivee conduc P1=GRAD(T) type thermique sans
 /
 ! (P1 . GRAD(A)) * A
 / I J
 Degré du 1ier polynome :ISO parametrique
 Nature 1ier polynome :DIRECT : on utilise le polynome tel quel
 Degré du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :UTILISEE : variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :DISSYMETRIQUE

Nom du programme :INGGSV : conduction sans propriete
 /
 ! (GRAD(A) . GRAD(A))
 / I J
 Degré du 1ier polynome :VIDE le polynome n'est pas utilise
 Degré du 2ieme polynome :ISO parametrique
 Nature du 2ieme polynome :GRADIENT : transmission du gradient du polynome
 1iere variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 2ieme variable auxiliaire :VIDE : pas de variable auxiliaire utilisee
 Type de l'integrand calcule :MATRICE
 Structure du resultat :SYMETRIQUE

Operateurs de calcul de proprietes

ALARME : detection automatique d une propriete non programm
Formule correspondante :
affiche alarme et provoque l arret du calcul

PRONU1 : courbe spline reelle dependance:(rot(A)
Fonction de la variable principale :
Formule correspondante :
consulter la documentation

PROMUA : courbe spline tensorielle (GRAD(A))
Fonction de la variable principale :
Formule correspondante :
consulter la documentation

PRDNUI : derivee courbe spline reelle (ROT(A))
Fonction de la variable principale :
Formule correspondante :
consulter la documentation

PRDMUA : derivee courbe spline tensorielle (GRAD(A))
Fonction de la variable principale :
Formule correspondante :
consulter la documentation

PROSIN : sinusoides en t
Fonction du temps :
Formule correspondante :
 $v1 * \sin(v2 * t + v3)$

PROSIO : sinusoides en t-deltat
Fonction du temps :
Formule correspondante :
 $v1 * \sin(v2 * (t - \text{deltat}) + v3)$

PROECH : echelon
Fonction du temps :
Formule correspondante :
si npas = 1 proech = v1 si non proech = v2

PROLIN : droite
Fonction de la variable principale :
Formule correspondante :
 $v1 + v2 * i$

PROPAR : parabole
Fonction de la variable principale :
Formule correspondante :
 $v1 + v2 * i + v3 * i ** 2$

OBJET-VARIABLES

RC : capacite calorifique
Nature : SURFACIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : cas usuel
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises
Modeles de type fonction autorises

Nom de la fonction : RCLIN : variation lineaire

Cas du calcul direct

Nom du programme de calcul : PROLIN : droite

COMPOSANTE:IN : temperature au pas n
Type de dependance:DIRECT : la variable est utilisee telle quelle

Cas du calcul du gradient
Non utilise

Cas du calcul du rotationnel
Non utilise

Nom de la fonction : RCPAR : variation parabolique de rc

Cas du calcul direct

Nom du programme de calcul : PROPAR : parabole

COMPOSANTE:IN : temperature au pas n
Type de dependance:DIRECT : la variable est utilisee telle quelle

Cas du calcul du gradient
Non utilise

Cas du calcul du rotationnel
Non utilise

V : vitesse de deplacement d un corps ou d un fluide

Nature : SURFACIQUE
Nombre de types : 1
TYPE NUMERO : 1
VECTEUR : dans le plan d etude
Nombre de composantes : 2
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises

H : echange convectif de surface

Nature : LINEIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : cas usuel
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises

ES : coeficient de rayonnement

Nature : LINEIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : echange radiatif
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises

QV : sources thermiques evolutives

Nature : SURFACIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : r a s
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises
Modeles de type fonction autorises

Nom de la fonction : QECH : source en echelon

Cas du calcul direct

Nom du programme de calcul : PROECH : echelon

Cas du calcul du gradient
Non utilise

Cas du calcul du rotationnel
Non utilise

TA : temperature de l ambiante

Nature : LINEIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : r a s
Nombre de composantes : 1
Modeles de type constant autorises

DRC : derivee de rc / temperature

Nature : SURFACIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : cas usuel non lineaire
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises
Modeles de type fonction autorises

Nom de la fonction : DRCLIN : variation lineaire de drc derivee de rc paraboliqu

Cas du calcul direct

Nom du programme de calcul : PROLIN : droite

COMPOSANTE:TN : temperature au pas n
Type de dependance:DIRECT : la variable est utilisee telle quelle

Cas du calcul du gradient
Non utilise

Cas du calcul du rotationnel
Non utilise

QS : sources thermiques statiques

Nature : SURFACIQUE
Nombre de types : 1
TYPE NUMERO : 1
REEL : cas usuel
Nombre de composantes : 1
Modeles de type constant autorises
Modeles de type table globale autorises
Modeles de type table elementaire autorises
Modeles de type table regionale autorises

DK : derivee de k / temperature

Nature : SURFACIQUE

Nombre de types : 1

TYPE NUMERO : 1

REEL : cas isotrope

Nombre de composantes : 1

Modeles de type constant autorises

Modeles de type table globale autorises

Modeles de type table elementaire autorises

Modeles de type table regionale autorises

Modeles de type fonction autorises

Nom de la fonction : DKLIN : variation lineaire de dk derivee de k parabolique

Cas du calcul direct

Nom du programme de calcul : PROLIN : droite

COMPOSANTE:TN : temperature au pas n

Type de dependance:DIRECT : la variable est utilisee telle quelle

Cas du calcul du gradient

Non utilise

Cas du calcul du rotationnel

Non utilise

OBJET-EQUATIONS

OPTIONS GENERALES CHOISIES

Dimension de l'espace de travail : 2
Type de coordonnees : CARTESIENNES
Types de polynomes necessaires :
1 : Isoparametriques

LISTE DES INTEGRANTS UTILISES

Nom de l'integrand:DTES : derivee du terme de stockage

Nom de la caracteristique discriminante:DRC : derivee de rc / temperature
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : cas usuel non lineaire

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INAASC : terme de diffusion propriete constante
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la 1iere caracteristique auxiliaire:DRC : derivee de rc / temperature
Mode d'utilisation:Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POICV1 : cartésien propriete variable
Nom de la 1iere caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Direct
/ /
! ! DRC * A * A * TN * DX DY
/ / I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INAASV : terme de diffusion sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POICV2 : cartésien 2 proprietes variables
Nom de la 1iere caracteristique auxiliaire:DRC : derivee de rc / temperature
Mode d'utilisation:Direct
Nom de la 2ieme caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Direct
/ /
! ! A * A * DRC * TN * DX DY
/ / I J

Nom de l'integrand:TES : stockage

Nom de la caracteristique discriminante:RC : capacite calorifique
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : cas usuel

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INAASC : terme de diffusion propriete constante
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la liere caracteristique auxiliaire:RC : capacite calorifique
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/ /
! ! RC * A * A * DX DY
/ / I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INAASV : terme de diffusion sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POICV1 : cartesien propriete variable
Nom de la liere caracteristique auxiliaire:RC : capacite calorifique
Mode d'utilisation:Direct
/ /
! ! A * A * RC * DX DY
/ / I J

Nom de l'integrand: DTEO : derivee terme de stockage au pas precedent

Nom de la caracteristique discriminante: DRC : derivee de rc / temperature
Nature de l'integrand: Matrice

CAS NUMERO : 1
Caracteristique de type : REEL : cas usuel non lineaire

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand: INAASC : terme de diffusion propriete constante
Polynome 1 : Iso-parametrique
Mode : Direct
Nom de la 1iere caracteristique auxiliaire: DRC : derivee de rc / temperature
Mode d'utilisation: Direct
Type de l'integrand: Non lineaire
Nom du programme poids differentiel: POICV1 : cartésien propriete variable
Nom de la 1iere caracteristique auxiliaire: TP : temperature au temps precedent
Mode d'utilisation: Direct
/ /
! ! DRC * A * A * TP * DX DY
/ / I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand: INAASV : terme de diffusion sans propriete
Polynome 1 : Iso-parametrique
Mode : Direct
Type de l'integrand: Non lineaire
Nom du programme poids differentiel: POICV2 : cartésien 2 proprietes variables
Nom de la 1iere caracteristique auxiliaire: DRC : derivee de rc / temperature
Mode d'utilisation: Direct
Nom de la 2ieme caracteristique auxiliaire: TP : temperature au temps precedent
Mode d'utilisation: Direct
/ /
! ! A * A * DRC * TP * DX DY
/ / I J

Nom de l'integrand:SE : conduction

Nom de la caracteristique discriminante:K : conductibilite thermique
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : cas isotrope

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INGGSC : conduction propriete constante
Polynome 2 :Iso-parametrique
Mode :Gradient
Nom de la liere caracteristique auxiliaire:K : conductibilite thermique
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/ /
! ! K * (GRAD(A) . GRAD(A)) * DX DY
/ / I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INGGSV : conduction sans propriete
Polynome 2 :Iso-parametrique
Mode :Gradient
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POICV1 : cartésien propriete variable
Nom de la liere caracteristique auxiliaire:K : conductibilite thermique
Mode d'utilisation:Direct
/ /
! ! (GRAD(A) . GRAD(A)) * K * DX DY
/ / I J

Nom de l'integrand:DSE : derivee des termes de conduction

Nom de la caracteristique discriminante:DK : derivee de k / temperature
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : cas isotrope

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INVGAC : derivee conduc P1=GRAD(T) type thermiq
Polynome 1 :Iso-parametrique
Mode :Direct
Polynome 2 :Iso-parametrique
Mode :Gradient
Nom de la 1iere caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Gradient
Nom de la 2ieme caracteristique auxiliaire:DK : derivee de k / temperature
Mode d'utilisation:Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/ /
! ! (GRA(TN) . GRAD(A)) * A * DK * DX DY
/ / I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INVGAV : derivee conduc P1=GRAD(T) type thermiq
Polynome 1 :Iso-parametrique
Mode :Direct
Polynome 2 :Iso-parametrique
Mode :Gradient
Nom de la 1iere caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Gradient
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POICV1 : cartésien propriete variable
Nom de la 1iere caracteristique auxiliaire:DK : derivee de k / temperature
Mode d'utilisation:Direct
/ /
! ! (GRA(TN) . GRAD(A)) * A * DK * DX DY
/ / I J

Nom de l'integrand:CV : echange convectif de surface

Nom de la caracteristique discriminante:H : echange convectif de surface
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : cas usuel

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INAASC : terme de diffusion propriete constante
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la liere caracteristique auxiliaire:H : echange convectif de surface
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/
! H * A * A * DX
/ I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INAASV : terme de diffusion sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POICV1 : cartésien propriete variable
Nom de la liere caracteristique auxiliaire:H : echange convectif de surface
Mode d'utilisation:Direct
/
! A * A * H * DX
/ I J

Nom de l'integrand:RY1 : rayonnement

Nom de la caracteristique discriminante:ES : coefficient de rayonnement
Nature de l'integrand:Matrice

CAS NUMERO : 1
Caracteristique de type :REEL : echange radiatif

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INAASC : terme de diffusion propriete constante
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la 1iere caracteristique auxiliaire:ES : coefficient de rayonnement
Mode d'utilisation:Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POI4V1 : une propriete variable **4
Nom de la 1iere caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Direct
/
! ES * A * A * TN*TN*TN*TN * DX
/ I J

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INAASV : terme de diffusion sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Non lineaire
Nom du programme poids differentiel:POI4V2 : P3VAR ** 4 * P4VAR
Nom de la 1iere caracteristique auxiliaire:TN : temperature au pas n
Mode d'utilisation:Direct
Nom de la 2ieme caracteristique auxiliaire:ES : coefficient de rayonnement
Mode d'utilisation:Direct
/
! A * A * TN*TN*TN*TN * ES * DX
/ I J

Nom de l'integrand:VES : sources statiques

Nom de la caracteristique discriminante:QS : sources thermiques statiques
Nature de l'integrand:Vecteur

CAS NUMERO : 1
Caracteristique de type :REEL : cas usuel

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INTAC : sources constantes
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la liere caracteristique auxiliaire:QS : sources thermiques statiques
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/ /
! ! QS * A * DX DY
/ / I

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INTAV : sources sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POICV1 : cartisien propriete variable
Nom de la liere caracteristique auxiliaire:QS : sources thermiques statiques
Mode d'utilisation:Direct
/ /
! ! A * QS * DX DY
/ / I

Nom de l'integrand:VEV : sources evolutives

Nom de la caracteristique discriminante:QV : sources thermiques evolutives
Nature de l'integrand:Vecteur

CAS NUMERO : 1
Caracteristique de type :REEL : r a s

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INTAC : sources constantes
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la liere caracteristique auxiliaire:QV : sources thermiques evolutives
Mode d'utilisation:Direct
Type de l'integrand:Dynamique
Nom du programme poids differentiel:POIVID : sans parametre element differentiel seul
/ /
! ! QV * A * DX DY
/ / I

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INTAV : sources sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Dynamique
Nom du programme poids differentiel:POICV1 : cartisien propriete variable
Nom de la liere caracteristique auxiliaire:QV : sources thermiques evolutives
Mode d'utilisation:Direct
/ /
! ! A * QV * DX DY
/ / I

Nom de l'integrand:VEC : terme convectif de surface ambiante

Nom de la caracteristique discriminante:H : echange convectif de surface
Nature de l'integrand:Vecteur

CAS NUMERO : 1
Caracteristique de type :REEL : cas usuel

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INTAC : sources constantes
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la 1iere caracteristique auxiliaire:H : echange convectif de surface
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POICC1 : cartisien propriete constante
Nom de la 1iere caracteristique auxiliaire:TA : temperature de l ambiante
Mode d'utilisation:Direct
/
! H * A * TA * DX
/ I

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INTAV : sources sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:POICC2 : cartisien 2 proprietes : variable constant
Nom de la 1iere caracteristique auxiliaire:H : echange convectif de surface
Mode d'utilisation:Direct
Nom de la 2ieme caracteristique auxiliaire:TA : temperature de l ambiante
Mode d'utilisation:Direct
/
! A * H * TA * DX
/ I

Nom de l'integrand:VER : rayonnement ambiante

Nom de la caracteristique discriminante:ES : coefficient de rayonnement
Nature de l'integrand:Vecteur

CAS NUMERO : 1
Caracteristique de type :REEL : echange radiatif

Cas caracteristique discriminante constante
Nom du programme calculant l'integrand:INTAC : sources constantes
Polynome 1 :Iso-parametrique
Mode :Direct
Nom de la 1iere caracteristique auxiliaire:ES : coefficient de rayonnement
Mode d'utilisation:Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:PCPC4 : = + P3cst ** 4
Nom de la 1iere caracteristique auxiliaire:TA : temperature de l ambiante
Mode d'utilisation:Direct
/
! ES * A * TA*TA*TA*TA * DX
/ I

Cas caracteristique discriminante variable
Nom du programme calculant l'integrand:INTAV : sources sans propriete
Polynome 1 :Iso-parametrique
Mode :Direct
Type de l'integrand:Lineaire
Nom du programme poids differentiel:PCPC4V : = + P3cst ** 4 * P4var
Nom de la 1iere caracteristique auxiliaire:TA : temperature de l ambiante
Mode d'utilisation:Direct
Nom de la 2ieme caracteristique auxiliaire:ES : coefficient de rayonnement
Mode d'utilisation:Direct
/
! A * TA*TA*TA*TA * DX
/ I

Nom de l'integrand: DRY : rayonnement residu et matrice jacobienne

Nom de la caracteristique discriminante: ES : coefficient de rayonnement
 Nature de l'integrand: Matrice

CAS NUMERO : 1
 Caracteristique de type : REEL : echange radiatif

Cas caracteristique discriminante constante
 Nom du programme calculant l'integrand: INAASC : terme de diffusion propriete constante
 Polynome 1 : Iso-parametrique
 Mode : Direct
 Nom de la 1iere caracteristique auxiliaire: ES : coefficient de rayonnement
 Mode d'utilisation: Direct
 Type de l'integrand: Non lineaire
 Nom du programme poids differentiel: POI3V1 : une propriete variable au cube
 Nom de la 1iere caracteristique auxiliaire: TN : temperature au pas n
 Mode d'utilisation: Direct
 /
 ! ES * A * A * TN*TN*TN * DX
 / I J

Cas caracteristique discriminante variable
 Nom du programme calculant l'integrand: INAASV : terme de diffusion sans propriete
 Polynome 1 : Iso-parametrique
 Mode : Direct
 Type de l'integrand: Non lineaire
 Nom du programme poids differentiel: POI3V2 : derivee rayonnement (P3)**3 , P4 variable
 Nom de la 1iere caracteristique auxiliaire: TN : temperature au pas n
 Mode d'utilisation: Direct
 Nom de la 2ieme caracteristique auxiliaire: ES : coefficient de rayonnement
 Mode d'utilisation: Direct
 /
 ! A * A * TN*TN*TN * ES * DX
 / I J

LISTE DES OPERATEURS DE CONSTRUCTION RETENUS

CSLIST : cas lineaire statique resolution directe
 $M * I = KS$

CSNRST : non lineaire statique newton raphson
 $J = M + DM ; R = M * I - KS$

CSLIDR : lineaire dynamique implicite resolution directe
 $(L + dt.M) * I = L * IO + dt.(KS + KV)$

CSNRDI : dynamique L non lineaire implicite demarrage
 $(L + dt.M)*I = L*IO + dt (KS + KV)$

CSNRDR : dynamique L non lineaire implicite raphson
 $J = L+DL-DLO + dt.(M+DM) ; R = L*(I-IO) + dt.(M*I-KS-KV)$

MATRICE CONSTITUTIVE : M
BLOC : 1 1
TERME DE COUPLAGE ENTRE LES INCONNUES :
TN : temperature au pas n
ET :
TN : temperature au pas n

SE : conduction
CV : echange convectif de surface
DRY : rayonnement residu et matrice jacobienne

MATRICE CONSTITUTIVE : DM
BLOC : 1 1
TERME DE COUPLAGE ENTRE LES INCONNUES :
TN : temperature au pas n
ET :
TN : temperature au pas n

DSE : derivee des termes de conduction
DRY : rayonnement residu et matrice jacobienne
DRY : rayonnement residu et matrice jacobienne
DRY : rayonnement residu et matrice jacobienne

MATRICE CONSTITUTIVE : L
BLOC : 1 1
TERME DE COUPLAGE ENTRE LES INCONNUES :
TN : temperature au pas n
ET :
TN : temperature au pas n
TES : stockage

MATRICE CONSTITUTIVE : DL
BLOC : 1 1
TERME DE COUPLAGE ENTRE LES INCONNUES :
TN : temperature au pas n
ET :
TN : temperature au pas n
DTES : derivee du terme de stockage

MATRICE CONSTITUTIVE : DLO
BLOC : 1 1
TERME DE COUPLAGE ENTRE LES INCONNUES :
TN : temperature au pas n
ET :
TN : temperature au pas n
DTEO : derivee terme de stockage au pas precedent

VECTEUR CONSTITUTIF : KS
BLOC : 1
TERME CONCERNANT L'INCONNUE :
TN : temperature au pas n
VES : sources statiques

VECTEUR CONSTITUTIF : KV
BLOC : 1
TERME CONCERNANT L'INCONNUE :
TN : temperature au pas n
VEV : sources evolutives
VEC : terme convectif de surface ambiante
VER : rayonnement ambiante

EXEMPLE D'UTILISATION DU SYSTEME OBJET MAGNETODYNAMIQUE

Ce système objet traite l'équation magnétodynamique linéaire en coordonnées axisymétriques et utilise des variables de type complexe.

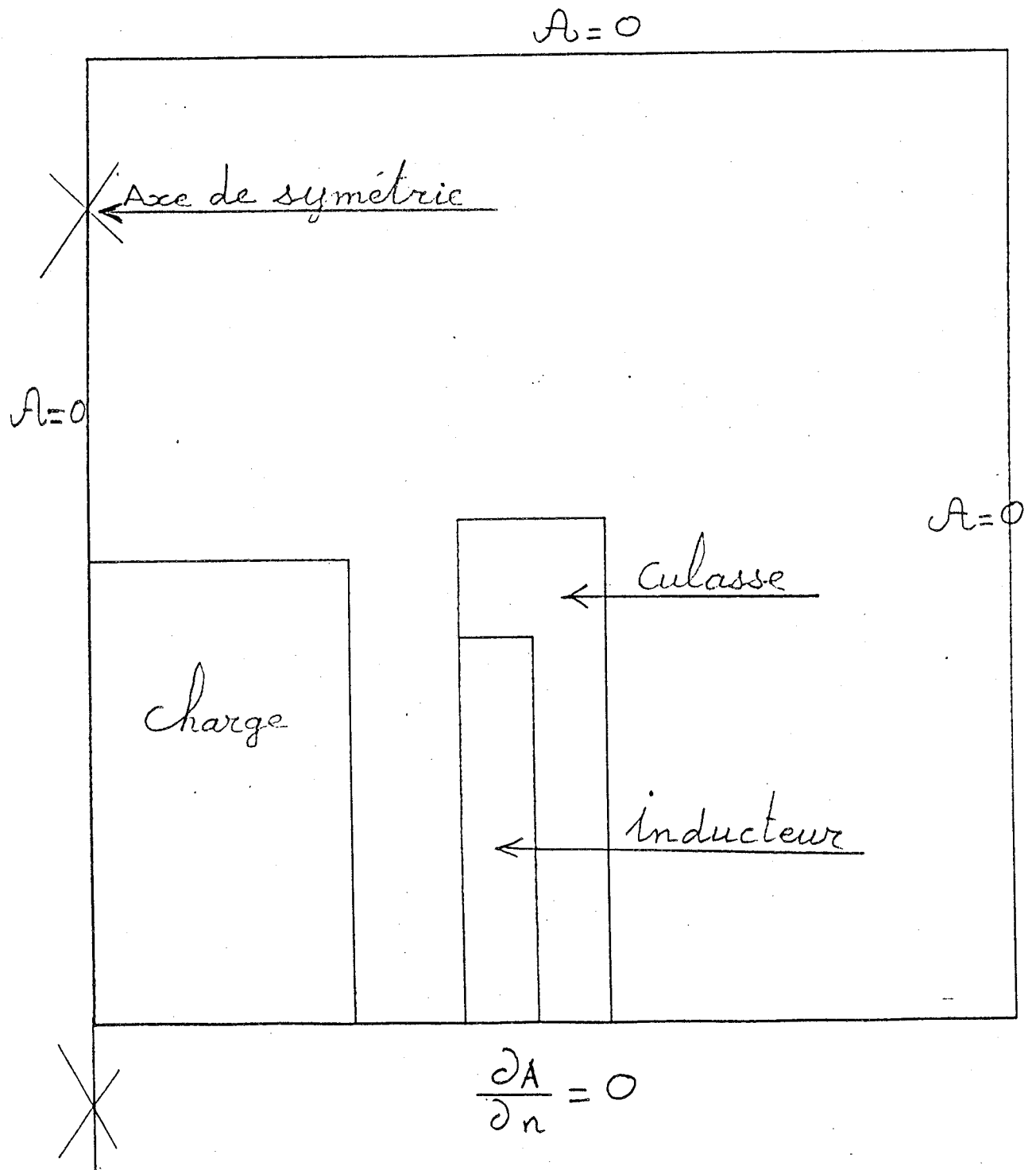
La description de cette équation a été fournie au chapitre A.III.3 et la construction de ce système objet a été décrite au chapitre C.III.1.

Nous présentons ci-dessous le dispositif de chauffage par induction étudié. La même étude a été réalisée à l'aide du système industriel FLUX2D et nous avons pu montrer la totale concordance entre les résultats obtenus.

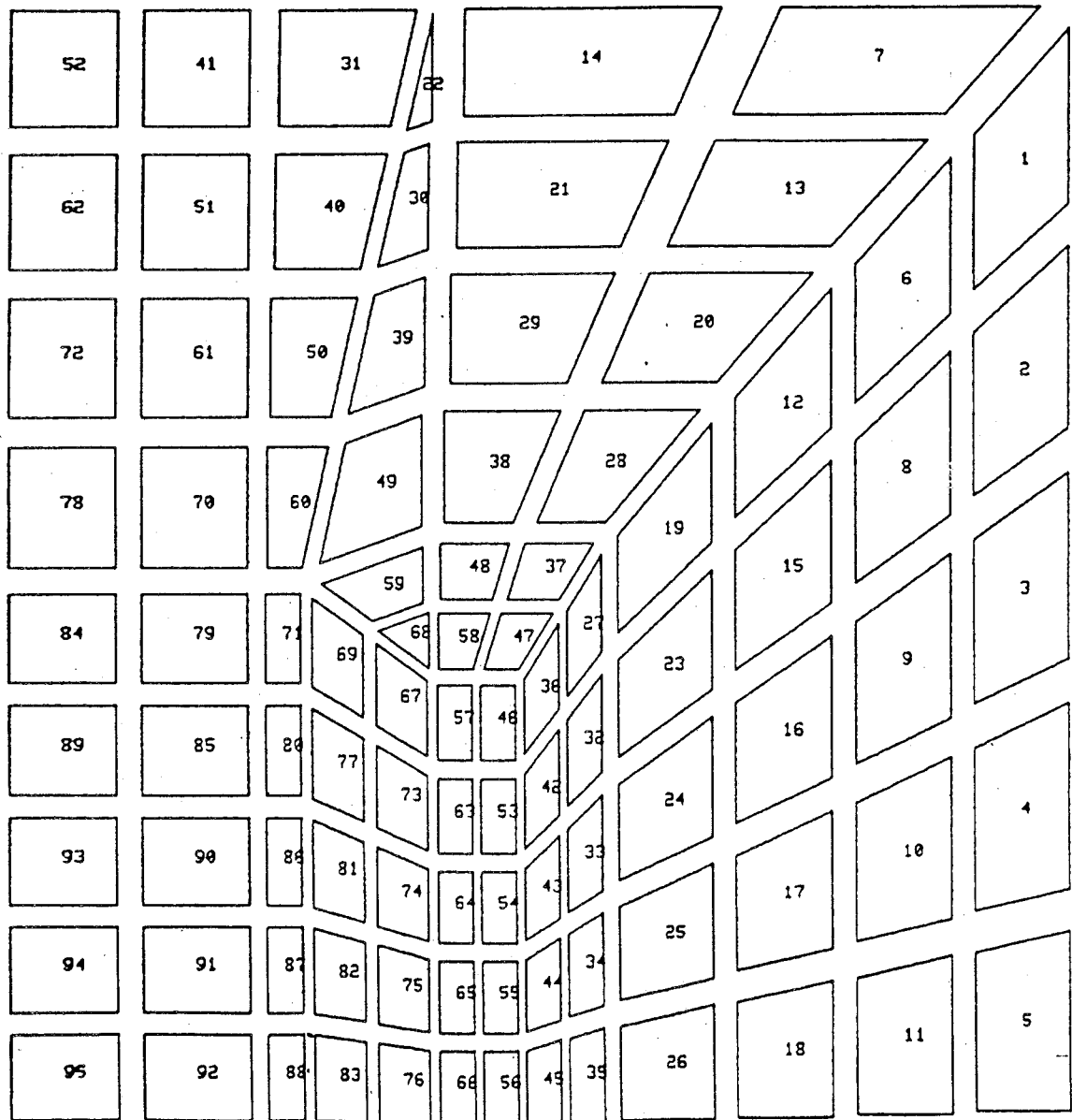
Ce problème de taille assez réduite comporte 88 éléments 316 noeuds, son coût de résolution global à l'aide de notre système est 30 % plus élevé que celui obtenu avec FLUX2D.

A l'aide de notre générateur nous avons décrit un système objet permettant de modéliser des phénomènes transitoires quelconques (non sinusoïdaux et pouvant mettre en oeuvre des matériaux non linéaires). Nous avons pu comparer cette méthode avec les résultats obtenus à l'aide des nombres complexes. Le régime permanent est pratiquement atteint après un transitoire non sinusoïdal d'une durée à peu près égale à 2 périodes sur le cas étudié.

Le coût d'une telle méthode est bien entendu proportionnel au nombre de pas de temps utilisés.



Le dispositif étudié

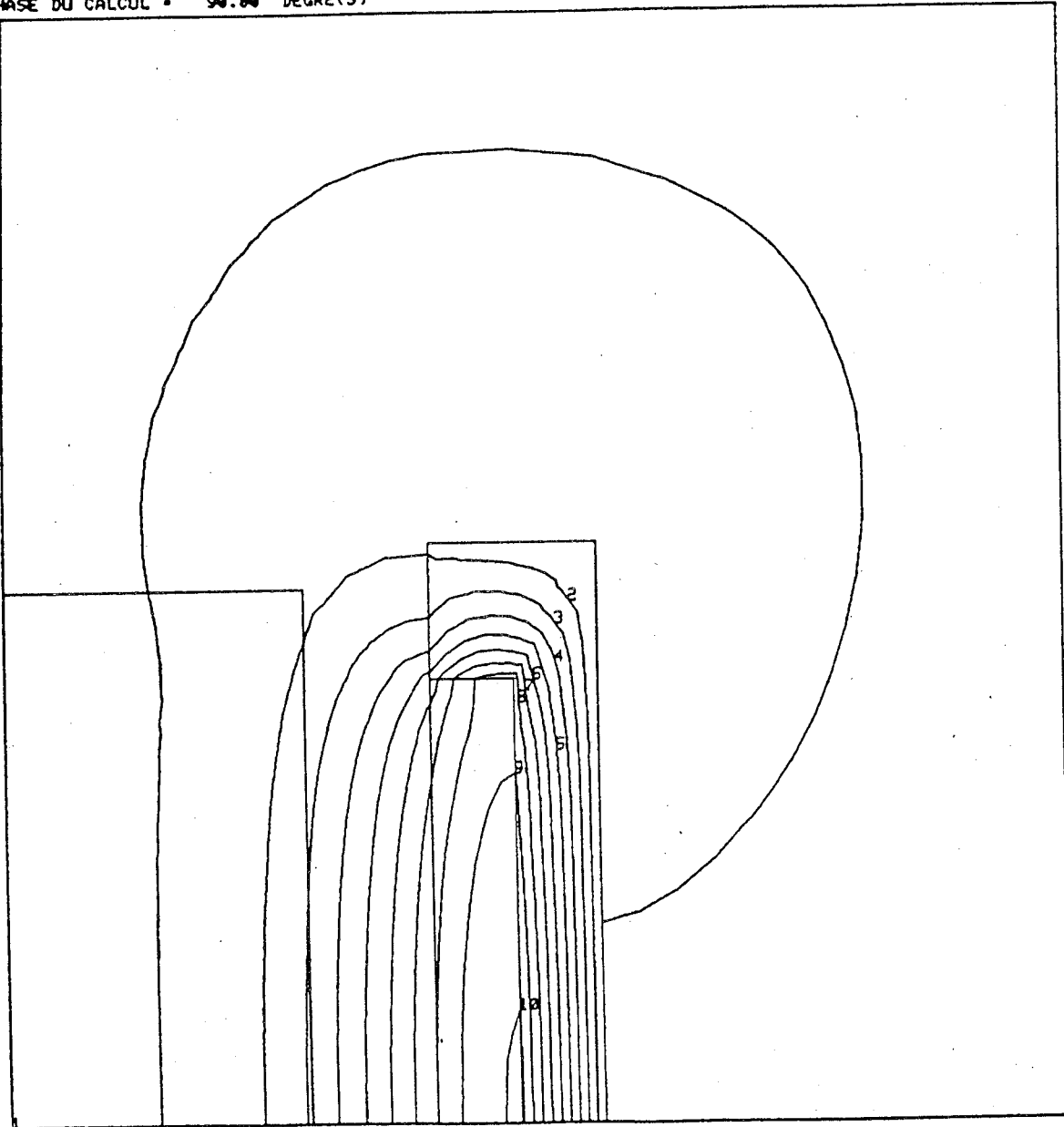


Le découpage en éléments finis

PHASE DU CALCUL = 99.99 DEGRE(S)

VALEURS

1	0.11541E-04
2	0.13817E-03
3	0.26480E-03
4	0.39142E-03
5	0.51805E-03
6	0.64468E-03
7	0.77131E-03
8	0.89793E-03
9	0.10246E-02
10	0.11512E-02



Lignes isopotentielles vecteur pour $I_{excitation} = \text{Max}$

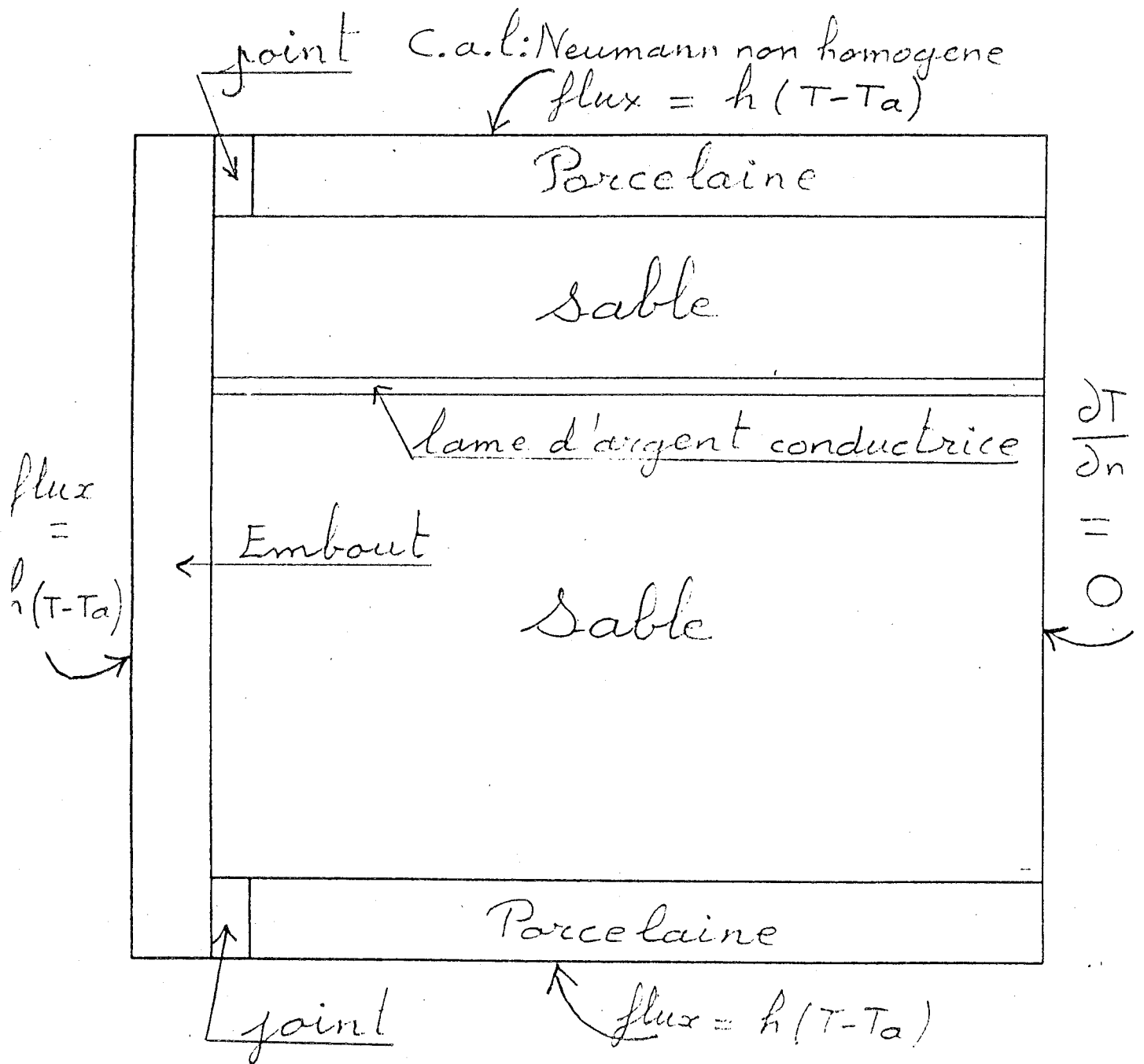
EXEMPLE D'UTILISATION DU SYSTEME OBJET THERMODYNAMIQUE

Ce système objet correspond à l'équation présentée au chapitre A.IV et sa construction a été décrite au chapitre C.III.2.

Le dispositif étudié est un fusible de forte puissance soumis à une surintensité.

Une première étude fournit la carte des températures pour le régime permanent qui sera obtenu suivant la surintensité appliquée. Cela permet de déterminer si le point de fusion du fusible sera atteint. Dans l'affirmative une deuxième étude plus détaillée, analyse alors le comportement transitoire de ce fusible.

Le découpage utilisé met en oeuvre 377 éléments et 1216 noeuds. Les valeurs très échelonnées des conductibilités thermiques des matériaux mis en oeuvre (plus de 10^6 de différence) créent des problèmes numériques qui peuvent être très délicats en étude transitoire vu la faible épaisseur de peau dans certains matériaux.

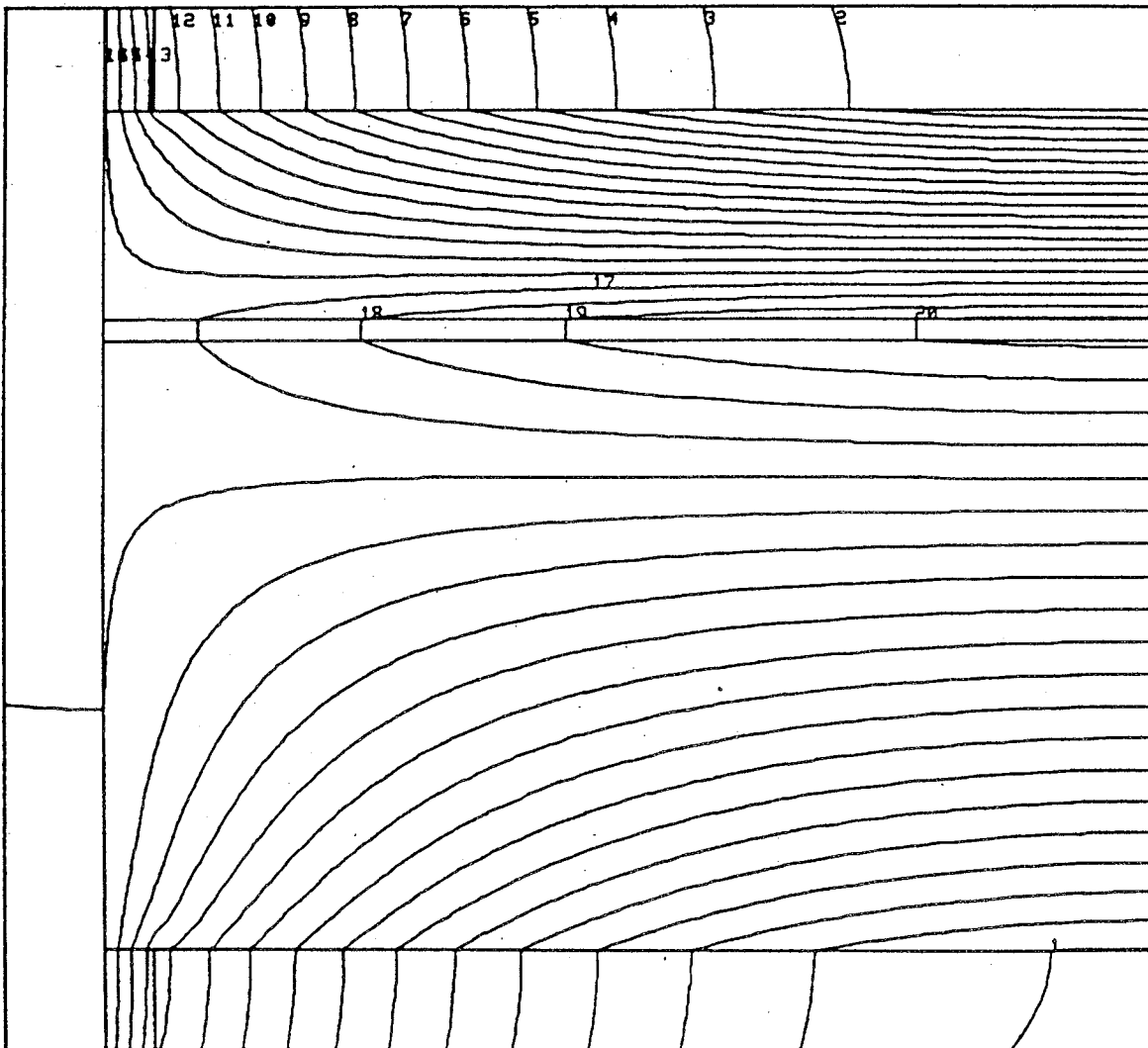


Le dispositif étudié

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156
157	158	159	160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232	233	234
235	236	237	238	239	240	241	242	243	244	245	246	247
248	249	250	251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270	271	272	273
274	275	276	277	278	279	280	281	282	283	284	285	286
287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312
313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338
339	340	341	342	343	344	345	346	347	348	349	350	351
352	353	354	355	356	357	358	359	360	361	362	363	364
365	366	367	368	369	370	371	372	373	374	375	376	377

Le découpage utilisé

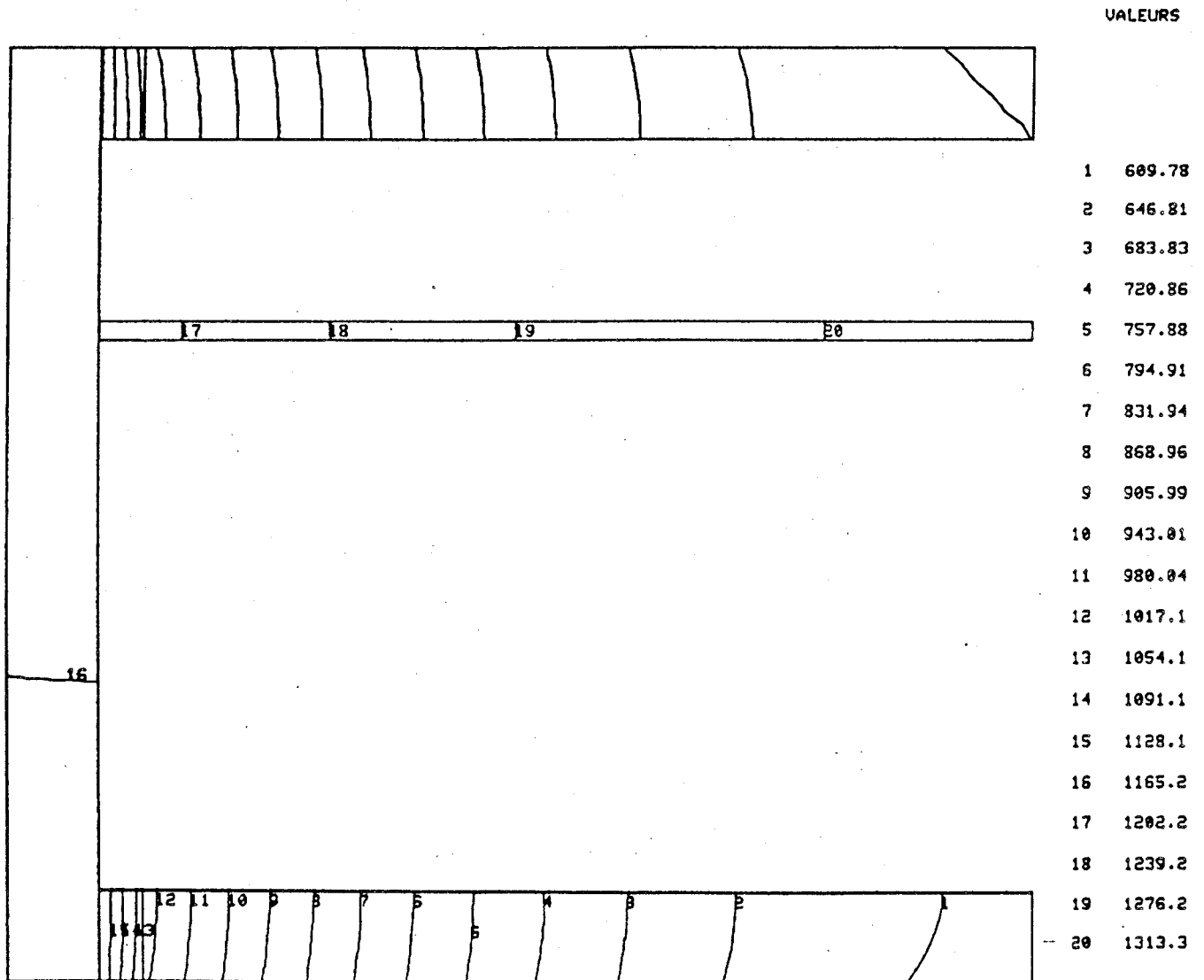
VALEURS



1	611.42
2	648.19
3	684.95
4	721.72
5	758.48
6	795.25
7	832.01
8	868.78
9	905.54
10	942.31
11	979.07
12	1015.8
13	1052.6
14	1089.4
15	1126.1
16	1162.9
17	1199.7
18	1236.4
19	1273.2
20	1310.0

Etude du régime permanent

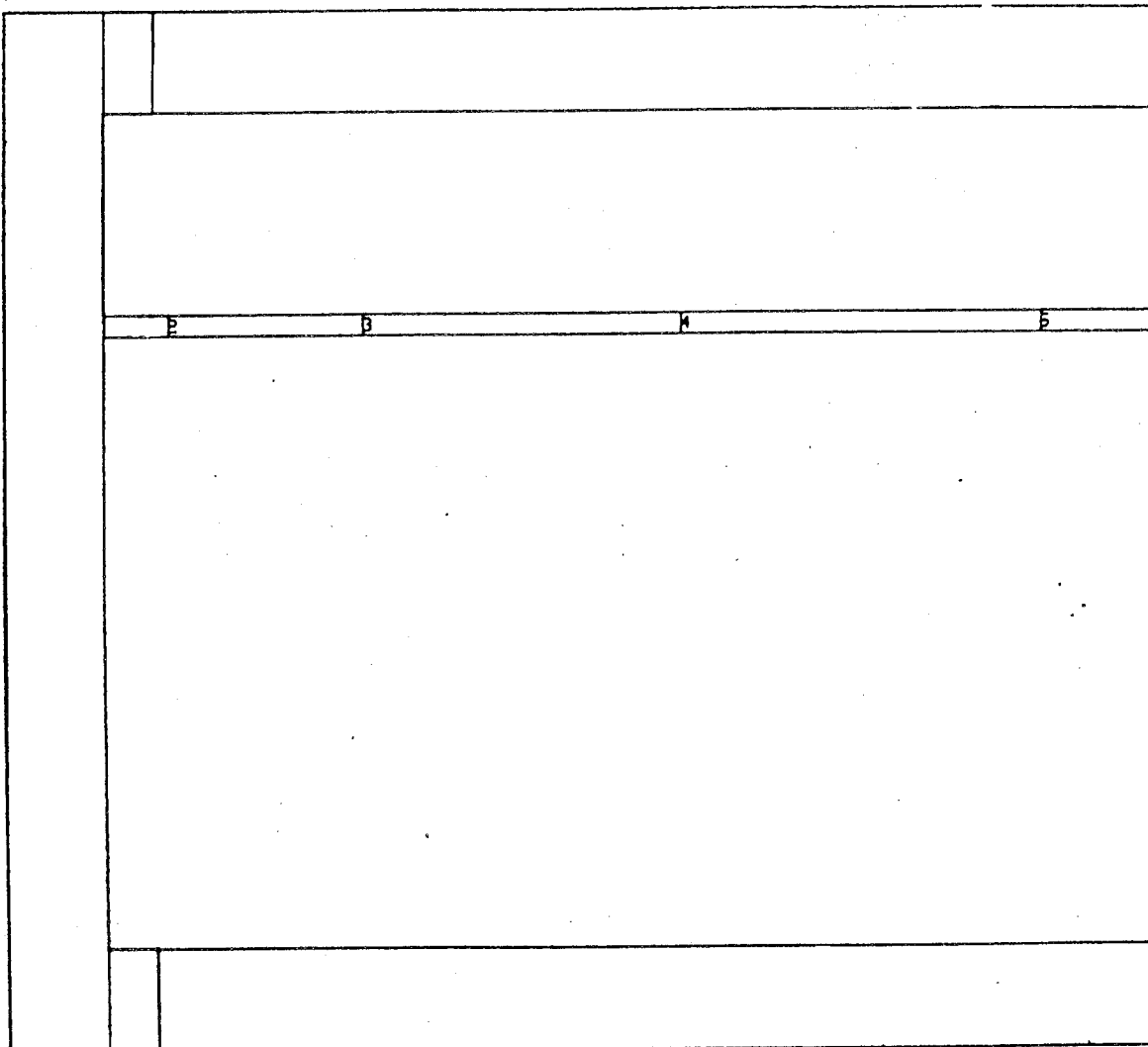
Tracé des isothermes
cas d'une faible surintensité



Modèle simplifié : Conduction
du sable négligée

résultats équivalents à moins de 1%
régime permanent (cas d'une faible
surintensité)

VALEURS



1	300.00
2	320.00
3	340.00
4	360.00
5	369.00

Etude du régime transitoire
dans le cas d'une faible surintensité
état du fusible après 20s

		VALEURS
	1	301.72
	2	310.15
	3	318.59
	4	327.02
	5	335.45
	6	343.89
	7	352.32
	8	360.75
	9	369.19
	10	377.62
	11	386.05
	12	394.48
	13	402.92
	14	411.35
	15	419.78
	16	428.22
	17	436.65
	18	445.08
	19	453.52
	20	461.95
	21	470.38

Etude du régime transitoire
dans le cas d'une faible surintensité
état du fusible après 100s

EXEMPLE D'UTILISATION DU SYSTEME OBJET HYDRODYNAMIQUE

Ce système objet correspond à l'équation présentée au chapitre A.V.2 et sa construction a été décrite au chapitre C.III.3.

La variable est représentée par trois inconnues à chaque noeuds : les deux composantes de la vitesse et la valeur de la pression.

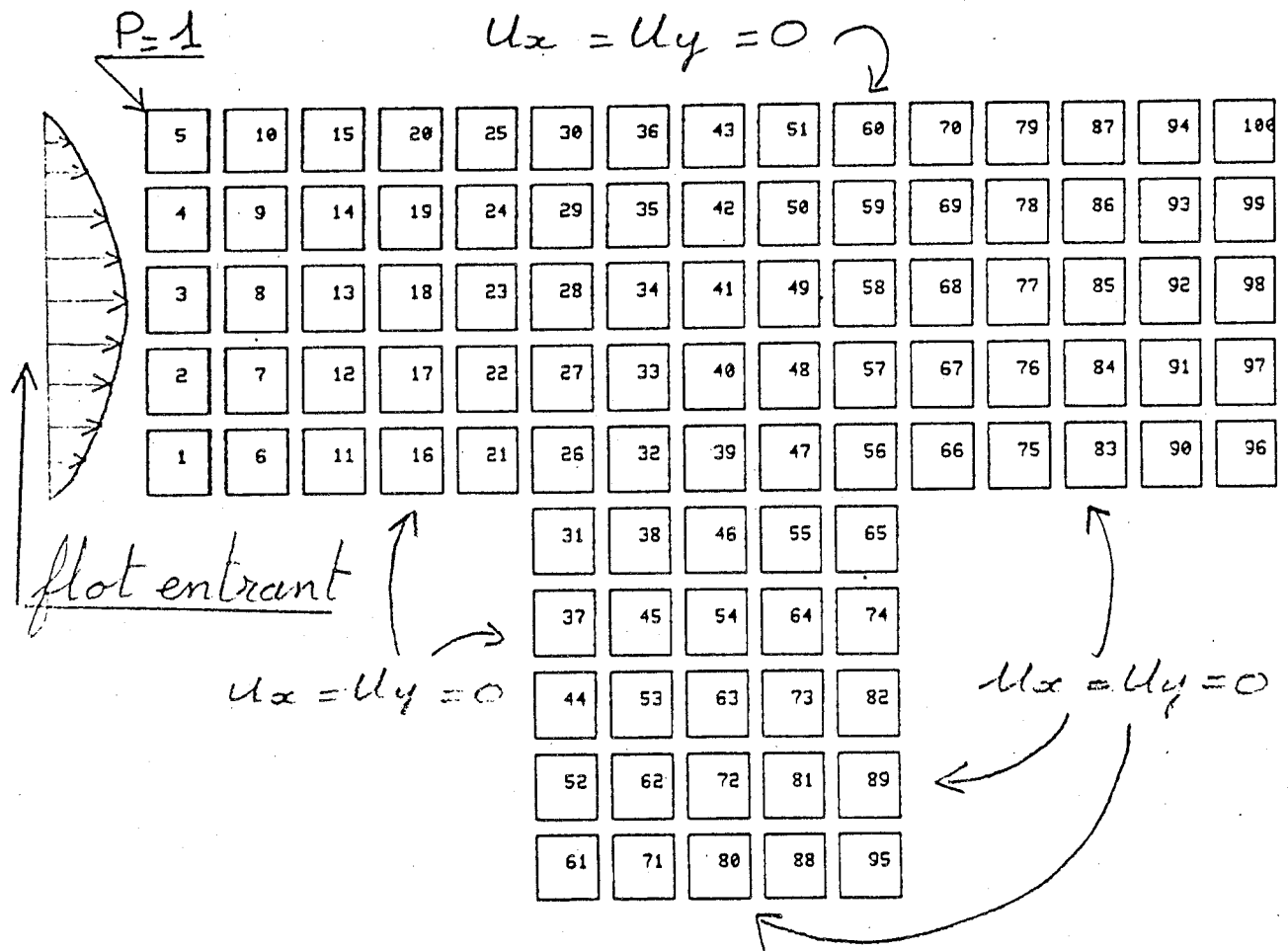
L'équation à résoudre est fortement non linéaire et 3 à 4 itérations de l'algorithme de Newton-Raphson sont nécessaires pour atteindre la convergence en régime transitoire perturbé.

L'étude réalisée concerne un chenal comportant une cavité. Le découpage effectué comporte 100 éléments et 351 noeuds ce qui conduit à un système de 1053 équations non linéaires, non symétriques à résoudre à chaque pas de temps.

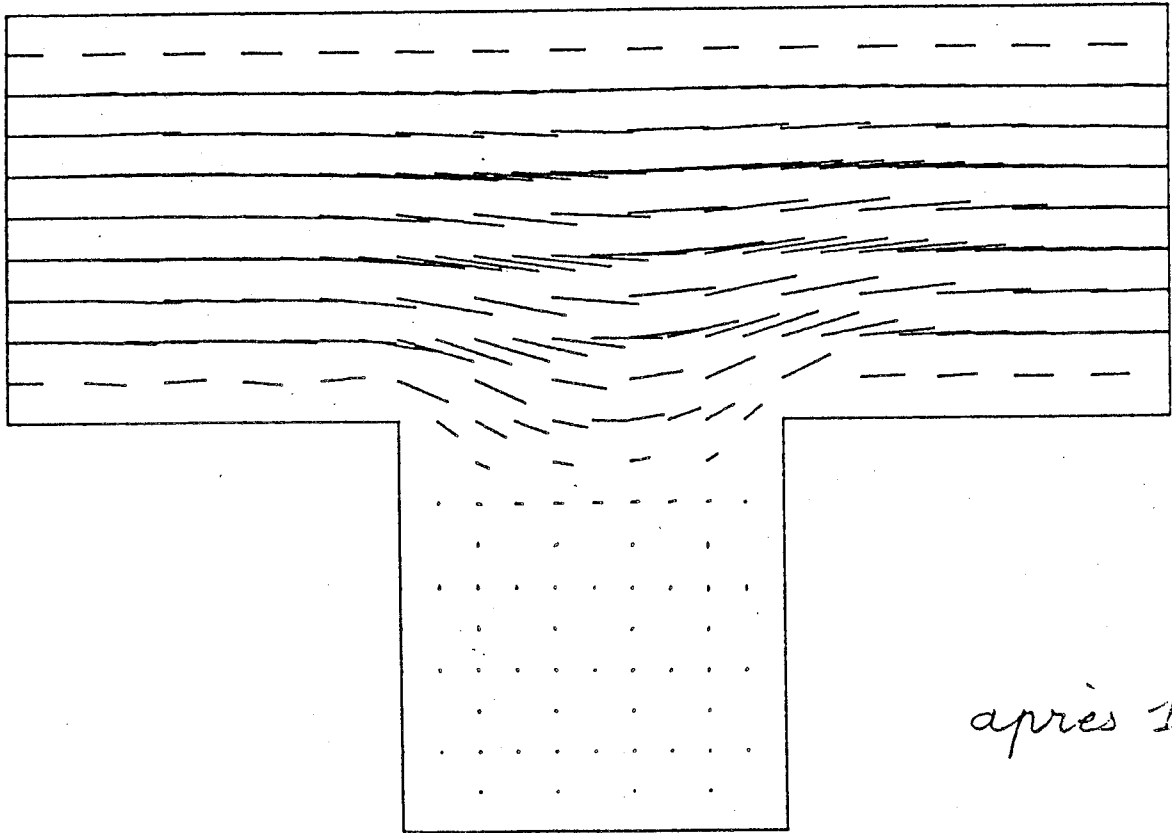
Le coût d'une telle étude est assez élevé vu la taille et le nombre des systèmes d'équations à résoudre.

20 minutes d'UC ont été nécessaires pour le traiter.

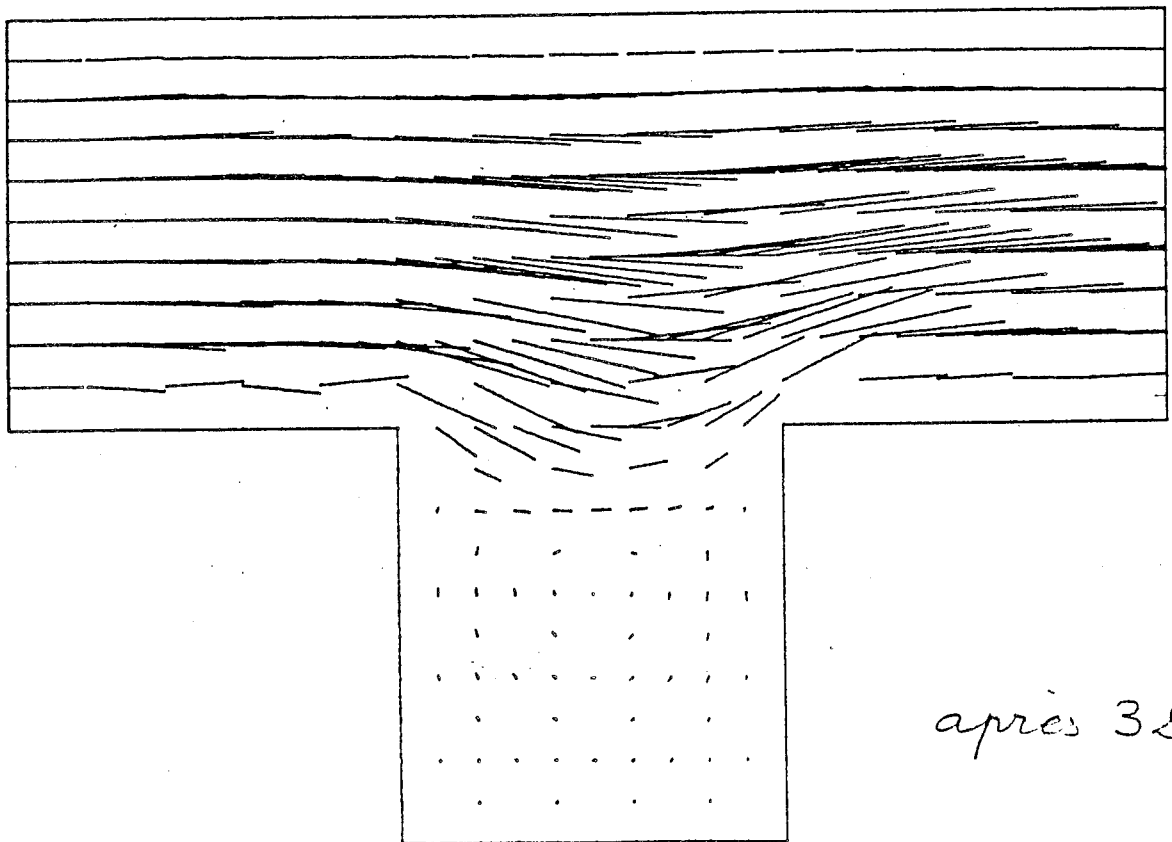
On remarquera dans cet exemple un autre mode de représentation des résultats qu'offre notre système. Affichage de grandeurs vectorielles alors que les autres exemples utilisaient tous des tracés de lignes isovaleurs.



Le découpage et les conditions aux limites

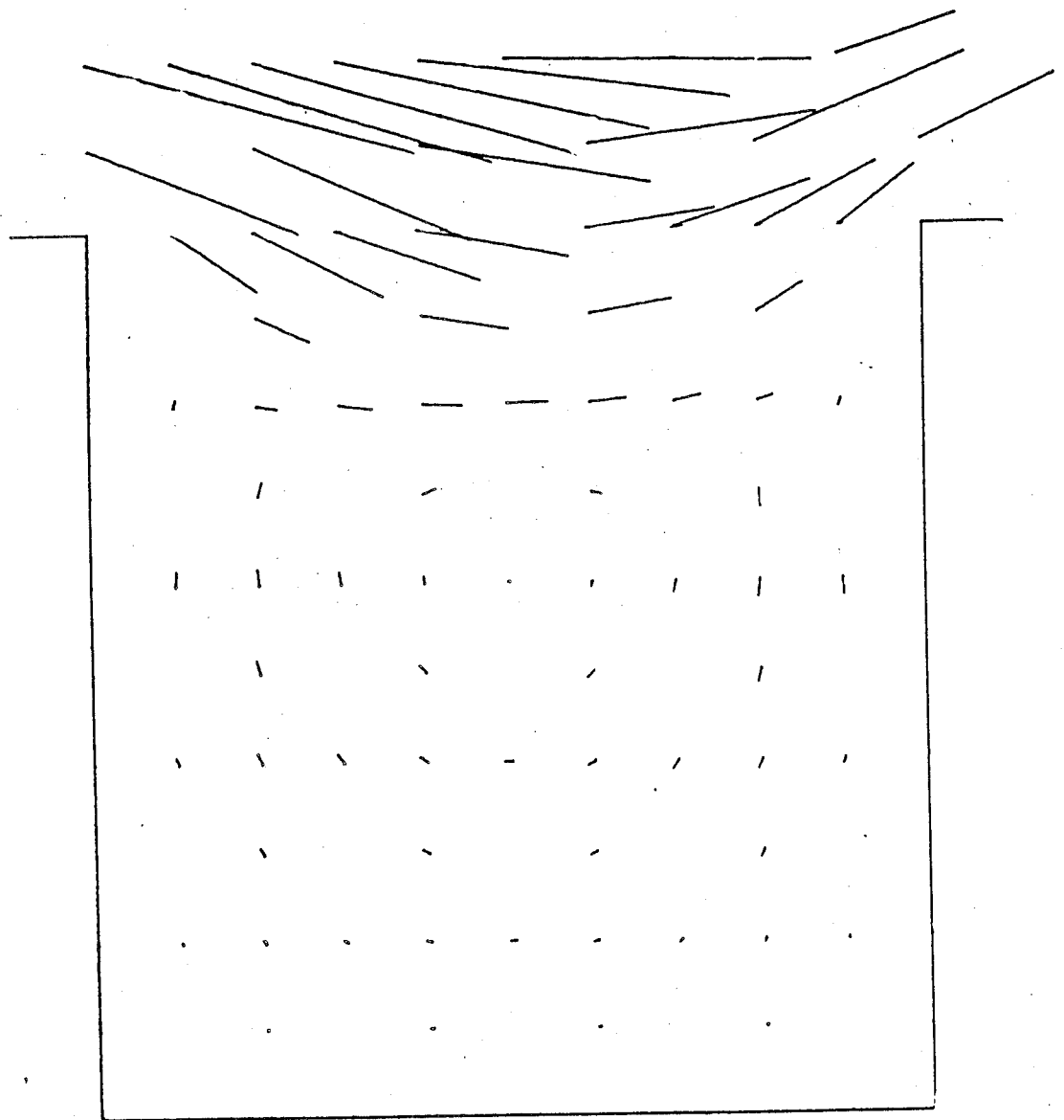


après 1s



après 3s

Simulation de la mise en vitesse
du vortex

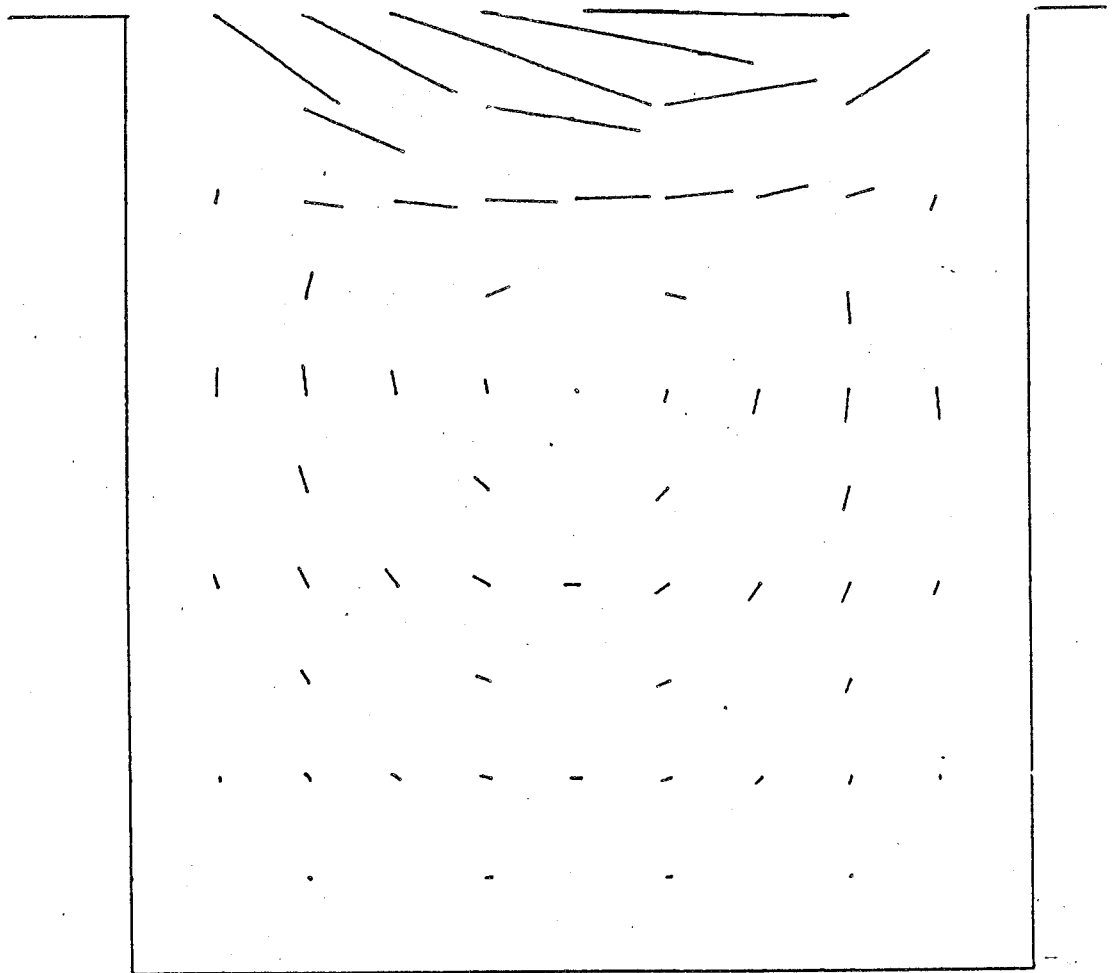


après 1 s

Détail du mouvement dans la cavité



60984 81800



après 5s

Détail du mouvement dans la cavité

ANALYSE METHODOLOGIQUE DE LA MODELISATION NUMERIQUE
DES ÉQUATIONS DE LA PHYSIQUE DES MILIEUX CONTINUS
À L'AIDE DE LA MÉTHODE DES ELEMENTS FINIS

* * *

FLUX EXPERT : UN SYSTÈME D'AIDE
À LA CONSTRUCTION DE LOGICIELS

Résumé

Une méthodologie générale de la modélisation numérique des équations aux dérivées partielles à l'aide de la méthode des éléments finis est proposée. Son application aux équations de la physique des milieux continus est exposée. Un système informatique EXPERT mettant en oeuvre ces résultats a été construit, testé, et appliqué à plusieurs études caractéristiques.