NOTE TO USERS

This reproduction is the best copy available.

UMI®

UNIVERSITÉ DE MONTRÉAL

APPLICATION DES ALGORITHMES GÉNÉTIQUES POUR L'OPTIMISATION D'UN CODE D'ANALYSE DES SOUS-CANAUX D'UNE GRAPPE DE COMBUSTIBLE NUCLÉAIRE

ARMANDO NAVA DOMINGUEZ
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

GÉNIE ÉNERGÉTIQUE

(JUILLET 2004)

©Armando Nava Dominguez, 2004.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-97971-7
Our file Notre référence
ISBN: 0-612-97971-7

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

APPLICATION DES ALGORITHMES GÉNÉTIQUES POUR L'OPTIMISATION D'UN CODE D'ANALYSE DES SOUS-CANAUX D'UNE GRAPPE DE COMBUSTIBLE NUCLÉAIRE

présenté par: Armando Nava Dominguez

En vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

M. TAPUCU Altan, D.Sc.A., président

M. TEYSSEDOU Alberto, PhD., membre et directeur de recherche

M. GIRARD René, PhD., membre

DÉDICACE

J'aimerai dédier ce mémoire à tout le monde qui a aidé d'une façon directe ou indirecte à réaliser ce projet

Merci à mes parents : *Carolina et Armando*,

à mes soeurs : *Ana et Katy*,

à ma tante *Mary*,

à ma grand-mère *Alfonsina*

toute la famille

Et à tous mes amis :

Armando, Gnomo, Laura, Jorge, Ricardo, Negro, Charly, Willy, Marbel, Rosalia, Eduardo, Annie, Saris, Sukran, Eric, Raphaël, Emily, Richard, Frank, Alexandra, Christine, Khaled, Nihan, Chiheb, Noreddine, Sun, Jean, Sor Amalia, David, Allison, Emilie... et Thales.

REMERCIEMENTS

Je tiens à remercier tous mes amis qui m'ont d'une façon ou d'autre aidé à réaliser ce projet.

J'aimerai remercier monsieur Alberto Teyssedou, qui a dédié tout le temps nécessaire pour que ce projet soit bien fini, tout le long de cette recherche.

À monsieur Altan Tapucu, pour les bons conseils qu'il m'a donnés pour rédiger ce mémoire, à monsieur René Girard pour avoir dédié son temps et pour ses conseils pour faciliter la tâche de ce projet.

A tous mes amis... Gracias, Merci, Thanks, Teşekkürler.

RÉSUMÉ

Le but de cette recherche est d'optimiser un code d'analyse de sous-canal en utili sant un algorithme génétique.

La prédiction de la distribution du taux de vide, les échanges du liquide et de gaz entre les sous-canaux est très importante pour assurer des conditions de transfert de chaleur sécuritaires dans le canal de combustible. D'un autre côté, elle sert aussi pour l'amélioration du design thermohydraulique des réacteurs.

Pour simuler les limites d'exploitation d'une grappe de combustible nucléaire, il est nécessaire de connaître les paramètres thermohydrauliques locaux, c'est-à-dire, établir un modèle théorique fondé sur les équations locales de conservation de la masse, de l'énergie et de la quantité de mouvement.

La géométrie complexe de la section de passage de la grappe, le manque d'information sur les conditions d'interface entre la phase liquide et gazeuse, les hypothèses de base et les relations empiriques pour la fermeture du système sont les difficultés majeures pour la résolution du modèle. Pour faciliter la modélisation de la grappe, la méthode plus utilisée consiste à diviser la section de passage en petites sous-sections élémentaires, appelées sous-canaux .

La plus grande simplification de cette méthode est l'hypothèse d'un écoulement à une seule dimension. Pour représenter l'écoulement latéral, une équation constitutive est utilisée. Pour fermer le système d'équations, un modèle de mélange est utilisé pour tenir compte des mécanismes responsables des échanges latéraux.

Les quatre mécanismes de mélange les plus importantes pour un écoulement diphasique sont : la diffusion turbulente du vide, l'écoulement latéral forcé, la dérive du vide vers les régions à haute vitesse et la dérive due à la gravité.

Les mécanismes de diffusion turbulente du vide et la dérive du vide vers les régions à haute vitesse sont modélisés en utilisant des coefficients de diffusion. Durant les trois dernières décennies des efforts considérables ont été déployés pour déterminer la valeur de ces coefficients.

Dans ce projet nous avons employé une nouvelle méthodologie. Elle comprend l'utilisation d'un code de calcul de type sous-canaux déjà existant couplé à un algorithme génétique qui effectue l'optimization des différents coefficients qui gouvernent les mécanismes de mélange latéral.

La méthode d'optimisation a nécessité, cependant, l'élaboration d'une fonction objectif appropriée. Après plusieurs essais numériques nous avons développé une fonction objectif basée sur le principe de la dissipation de l'énergie mécanique. L'utilisation d'une telle fonction nous a permis d'obtenir des simulations thermo-hydrauliques très proches des résultats expérimentaux.

ABSTRACT

The main objective of this research is to develop a Genetic Algorithm in order to optimize the predictions of a subchannel code.

The prediction of different flow variables like void fraction, cross-flow between subchannels, enthalpies, flow pressures are very important to control the heat transfer conditions of a power nuclear reactor. In order to simulate the operating limits of the fuel bundle, it is necessary to know local variables, thus, the model must be based on the local mass, energy and momentum conservation equations.

The complex geometry of the cross-section of the fuel bundle, the lack of information about the interfacial conditions between phases, the hypothesis used, and the use of empiric correlations are the major difficulties encountered in solving the equations in the model.

To facilitate the modeling of a rod fuel bundle, the most common used method consist in dividing the complex cross-sectional area in small subsections called subchannels. The main simplification of this method is to assume the flow to be one-dimensional, with one constitutive equation modelling of the cross flow between the interconnected subchannels. To close the system equations, a mixture model is used to represent the intersubchannel interactions. These interactions are as follows: diversion cross-flow, turbulent void diffusion, void drift and buoyancy drift.

Amongst these mechanisms, the turbulent void diffusion and void drift are frequently modelled using diffusion coefficients. During the last three decades, major efforts have been devoted to the determination of these coefficients. In this work, a novel approach has been employed where an existing subchannel code coupled to a genetic algorithm code which were used to optimize these coefficients.

In this project, a subchannel code in conjunction with a genetic algorithm is used to optimize the principals mechanisms that control the lateral mixing across the subchannels.

Genetic algorithms imply the use of an appropriate objective function to optimize. After several numerical simulations, a new objective function based in the principle of minimum dissipated energy was developed. The use of this function in the genetic algorithm coupled to the subchannel code, gave results in good agreement with the experimental data.

TABLE DES MATIÈRES

DÉDICACE . . . . . *iv*

REMERCIEMENTS . . . . . *v*

RÉSUMÉ . . . . . *vi*

ABSTRACT . . . . . *viii*

TABLE DES MATIÈRES . . . . . *x*

LISTE DES ANNEXES . . . . . *xiv*

LISTE DES FIGURES . . . . . *xv*

LISTE DES TABLEAUX . . . . . *xx*

LISTE DES SIEGLES ET ABRÉVIATIONS . . . . . *xxi*

Chapitre 1 INTRODUCTION À L'ANALYSE DE SOUS-CANAL *1*

1.0 Objectif. *3*

1.1 Organisation de ce mémoire *3*

Chapitre 2 REVUE BIBLIOGRAPHIQUE *4*

2.1 Modèles d'échange de masse pour une écoulement monophasique. . . . *8*

2.2 Modèles d'échange de masse pour une écoulement diphasique. . . . *11*

Chapitre 3 MODÈLE DE SOUS-CANAL *19*

3.1 Les équations de conservation. *19*

3.1.1 Équation de conservation de masse. *19*

3.1.2 Équation de conservation de la quantité de mouvement axiale. 21

3.1.3 Équation de conservation de la quantité de mouvement latérale. 24

3.2 Le système d'équations. 25

3.3 Le modèle de mélange. 26

3.4 Le modèle numérique. 29

3.4.1 Équation discrétisée de conservation de la masse. 31

3.4.2 Équation discrétisée de conservation de la quantité de mouvement axiale. 32

3.4.3 Équation discrétisée de conservation de la quantité de mouvement latérale. 35

3.4.4 Équation de pression axiale. 36

3.4.5 Correction de flux de masse. 37

Chapitre 4 LES ALGORITHMES GÉNÉTIQUES 41

4.1 Vocabulaire. 42

4.2 L'exploration et l'exploitation. 43

4.3 La description générale d'un algorithme génétique. 44

4.3.1 La représentation des paramètres ou codification. 45

4.3.2 La génération aléatoire de la population initiale. 47

4.3.3 L'évaluation. 48

4.3.4 Principes de sélection. 48

4.3.5 L'opérateur de croisement. 53

4.3.6 L'opérateur de mutation. 53

4.3.7 La convergence. 55

4.4. Exemple. 55

Chapitre 5 LA FONCTION OBJECTIF 64

5.1 L'erreur quadratique moyenne. 64

5.2 La dissipation de l'énergie. 65

5.3 La méthode de minimisation de la création d'entropie. 68

5.3.1 Création d'entropie d'un fluide. 68

5.4 La minimisation de la différence de pression latérale. 69

5.5 Sélection de la fonction objectif. 70

5.6 Le couplage de deux codes : L'AG et le code d'analyse de sous-canal. . 72

Chapitre 6 LES RÉSULTATS 78

6.1 Conditions d'entrée. 78

6.2 Analyse des résultats. 79

6.2.1 La perte de pression axiale. 79

6.2.2 La différence de pression latérale. 80

6.2.3 Le taux de vide. *81*

6.2.4 Le débit du gaz. *81*

6.2.5 Le débit du liquide. *81*

6.3 Analyse des paramètres optimisés. *82*

6.4 Comparaison des corrélations. *84*

Chapitre 7 LES CONCLUSIONS ET LES RECOMMANDATIONS *126*

7.1 Les conclusions. *126*

7.2 Les recommandations. *128*

BIBLIOGRAPHIE *130*

ANNEXES *134*

LISTE DES ANNEXES

ANNEXE A Relations constitutives et la géométrie des sous-canaux . . . *134*

ANNEXE B Organisation du code *139*

ANNEXE C Code *143*

LISTE DES FIGURES

FIGURE 1.1 La section de passage d'une grappe de combustible. *3*

FIGURE 2.1 Représentation de l'écoulement latéral. *7*

FIGURE 2.2 Dérive causée par la gravité. *7*

FIGURE 2.3 La représentation du modèle d'échange de masse volume à volume. *11*

FIGURE 3.1 Volume de contrôle pour le bilan de conservation de la masse. *21*

FIGURE 3.2 Volume de contrôle pour le bilan de conservation de la quantité de mouvement axiale. . . . *22*

FIGURE 3.3 La discrétisation. *29*

FIGURE 4.1 Le schéma de procédé d'un AG *45*

FIGURE 4.2 Les différentes types de sélection *49*

FIGURE 4.3 La représentation de la fonction objectif *56*

FIGURE 5.1 Le schéma du couplage du code d'analyse de sous-canal avec l'AG *73*

FIGURE 5.2 La représentation du chromosome *75*

FIGURE 5.3 La première population *75*

FIGURE 5.4 L'inter-relation entre l'AG et le code d'analyse de sous-canal *76*

FIGURE 6.1a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* *86*

FIGURE 6.1b (Suite) *86*

FIGURE 6.1c (Suite) *86*

FIGURE 6.1d (Suite) *87*

FIGURE 6.1e (Suite) *87*

FIGURE 6.2a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *89*

FIGURE 6.2b (Suite) *89*

FIGURE 6.2c (Suite) *89*

FIGURE 6.2d (Suite) *90*

FIGURE 6.2e (Suite) *90*

FIGURE 6.3a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* *92*

FIGURE 6.3b (Suite) *92*

FIGURE 6.3c (Suite) *92*

FIGURE 6.3d (Suite) *93*

FIGURE 6.3e (Suite) *93*

FIGURE 6.4a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *95*

FIGURE 6.4b (Suite) *95*

FIGURE 6.4c (Suite) *95*

FIGURE 6.4d (Suite) *96*

FIGURE 6.4e (Suite) *96*

FIGURE 6.5a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* *98*

FIGURE 6.5b (Suite) *98*

FIGURE 6.5c (Suite) *98*

FIGURE 6.5d (Suite) *99*

FIGURE 6.5e (Suite) *99*

FIGURE 6.6a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *101*

FIGURE 6.6b (Suite) *99*

FIGURE 6.6c (Suite) *99*

FIGURE 6.6d (Suite) *102*

FIGURE 6.6e (Suite) *102*

FIGURE 6.7a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* *104*

FIGURE 6.7b (Suite) *104*

FIGURE 6.7c (Suite) *104*

FIGURE 6.7d (Suite) *105*

FIGURE 6.7e (Suite) *105*

FIGURE 6.8a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *105*

FIGURE 6.8b (Suite) *107*

FIGURE 6.8c (Suite) *107*

FIGURE 6.8d (Suite) *107*

FIGURE 6.8e (Suite) *108*

FIGURE 6.9a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* *110*

FIGURE 6.9b (Suite) *110*

FIGURE 6.9c (Suite) *110*

FIGURE 6.9d (Suite) *111*

FIGURE 6.9e (Suite) *111*

FIGURE 6.10a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *111*

FIGURE 6.10b (Suite) *113*

FIGURE 6.10c (Suite) *113*

FIGURE 6.10d (Suite) *115*

FIGURE 6.10e (Suite) *115*

FIGURE 6.11a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *Erreur quadratique moyenne* . . . *114*

FIGURE 6.11b (Suite) *116*

FIGURE 6.11c (Suite) *116*

FIGURE 6.11d (Suite) *117*

FIGURE 6.11e (Suite) *117*

FIGURE 6.12a Comparaison des prédictions avec les donnés expérimentales fonction objectif : *La dissipation de l'énergie* *117*

FIGURE 6.12b (Suite) *119*

FIGURE 6.12c (Suite) *119*

FIGURE 6.12d (Suite) *120*

FIGURE 6.12e (Suite) *120*

FIGURE 6.13 Coefficients de diffusion D_α *121*

FIGURE 6.14 Coefficients de distribution C_0 . . . . . *121*

FIGURE 6.15 Coefficients de frottement latéral K_{tr} *122*

FIGURE 6.16a Comparaison de la corrélation de Shoukri et le modèle basé sur la dissipation de l'énergie. *124*

FIGURE 6.16b (Suite) *124*

FIGURE 6.16c (Suite) *124*

FIGURE 6.16d (Suite) *125*

FIGURE 6.16e (Suite) *125*

FIGURE a.1 La représentation de la géométrie du système *131*

LISTE DES TABLEAUX

TABLEAU 6.1 Les valeurs d'entrée et conditions initiales. *79*

LISTE DES SIEGLES ET DES ABRÉVIATIONS

AG	Algorithme Génétique
EQM	Erreur Quadratique Moyenne
f.o.	Fonction Objectif
FCC	Flux de Chaleur Critique

A	Section de passage	$[m^2]$
A_k	Section de passage de la phase k	$[m^2]$
$[AG]$	Matrices tridiagonales	
a	Largeur du sous-canal	$[m]$
a_j	Coefficients des équations discrétisées	
$[B]$	Matrice Jacobienne	
C_o	Coefficient de distribution de l'écoulement latéral	
c	Espacement de l'interconnexion	$[m]$
D	Coefficient de diffusion	$[m^2/s]$
D_{eq}	Coefficient de diffusion à l'équilibre	$[m^2/s]$
D_α	Coefficient de diffusion par le mélange turbulent	$[m^2/s]$
D_h	Diamètre hydraulique	$[m]$
d_A	Élément de surface	$[m^2]$
d	Coefficient des diffusion du vide local	$[m^2/s]$
dV	Élément de volume	$[m^3]$
$F_{i,j}$	Terme de pertes de pression latérale	$[N/m^2]$
G	Flux de masse	$[kg/m^2\,s]$
$[G]$	Vecteur colonne contenant les termes du flux de masse	
$\hat{G}$	Pseudo flux de masse	$[kg/m^2\,s]$
G'	Flux de masse latérale	$[kg/m^2\,s]$
$\hat{G}'$	Pseudo flux de masse latérale	$[kg/m^2\,s]$
g	Accélération de la gravité	$[m/s^2]$
J	Flux volumétrique	$[m/s]$
K_{tr}	Coefficient de frottement latéral	
ℓ	Longueur de mélange	$[m]$
m	Débit massique	$[kg/s]$

P_c	Probabilité de croisement	
P_m	Probabilité de mutation	
$[P]$	Vecteur colonne contenant les termes de pression	
$[PC]$	Vecteur colonne contenant les termes de pression de correction des pressions	
p	Pression	$[N/m^2]$
Q	Débit volumétrique	$[m^3/s]$
Q	La chaleur	$[kJ/kg]$
Re	Nombre de Reynolds	
S	Terme source	
S	Entropie	$[kJ/K\,kg]$
T	Température absolue	$[K]$
t	Température	$[C]$
U	Coefficient des équations discrétisées	$[m^3/s]$
u	Composante axiale de la vitesse d'écoulement	$[m/s]$
V	Coefficient des équations discrétisées	$[m^3/s]$
V	Vecteur de vitesse d'écoulement	$[m/s]$
V_k	Vecteur vitesse de la phase k	$[m/s]$
v	Composante latérale de la vitesse d'écoulement	$[m/s]$
W	Travail mécanique	$[kJ/kg]$
W'_{ij}	Échange de masse entre les sous-canaux par unité de longueur d'interconnexion	$[kg/m\,s]$
$W'_{i\leftrightarrow j}$	Transfert net de masse entre les sous-canaux par unité de longueur d'interconnexion	$[kg/m\,s]$
$\hat{x}$	Titre de l'écoulement	
y	Coordonnée latérale	$[m]$
z	Coordonnée axiale	$[m]$
α	Taux de vide local	
β	Titre volumétrique	
P_w	Périmètre mouillé	$[m]$
P_{acce}	Perte de pression par accélération	$[N/m^2]$
P_b	Perte de pression causée par l'obstruction	$[N/m^2]$

CHAPITRE 1

INTRODUCTION À L'ANALYSE DE SOUS-CANAL

Un des éléments les plus importants dans la conception et la sûreté des réacteurs nucléaires est la capacité du système caloporteur primaire d'évacuer la chaleur générée par le combustible nucléaire afin de prévenir son assèchement. La condition du flux de chaleur critique (FCC) limite cette capacité d'évacuation. Le FCC représente une détérioration du coefficient de transfert de chaleur entre les éléments de combustible et le caloporteur, qui a pour conséquence une augmentation de température de la gaine et du combustible nucléaire mettant en risque leur intégrité.

La prédiction des conditions d'écoulement que peuvent provoquer le FCC nécessite la connaissance des variables locales de l'écoulement comme : l'enthalpie, le taux de vide, le titre, la pression, le flux de masse, etc. Parmi les différentes techniques utilisées pour la détermination de ces paramètres, on peut citer la méthode d'analyse de sous-canal. Celle-ci consiste à diviser la géométrie complexe de la section de passage de la grappe (voir figure 1.1) de combustible en petites sections appelées sous-canaux.

La représentation mathématique des sous-canaux est fondée sur les équations de conservation de la masse, de l'énergie et de la quantité de mouvement, écrites pour une seule dimension. Étant donné que la vraie nature de l'écoulement est tridimensionnelle, une équation constitutive est utilisée pour considérer les échanges latéraux entre les sous-canaux interconnectés.

Une des caractéristiques la plus importante de cette analyse de sous-canal est l'utilisation d'un modèle de mélange pour compléter le système d'équations. Ce modèle de mélange tient compte des échanges latéraux de masse, d'énergie et de quantité de mouvement.

Lahey et Moody (1993) ont déterminé que les mécanismes les plus importants sont :

-Écoulement latéral forcé : Le même phénomène se produit pour un écoulement diphasique que pour un écoulement monophasique, c'est-à-dire, qu'il y a un échange des fluides lié à une différence de pression entre deux sous-canaux adjacents.

-Mélange turbulent : En écoulement diphasique, ce mécanisme produit un échange de masse causé par la turbulence des fluides. Pour les écoulements à une seule phase ce phénomène produit un transfert de masse nul.

-Dérive du vide vers les régions à haute vitesse : Ce mécanisme est occasionné par la tendance de l'écoulement de migrer vers une distribution d'équilibre.

-Dérive causée par la gravité : Dans les canaux interconnectés et placés horizontalement, la phase moins dense est naturellement poussé vers le haut, perpendiculairement à la direction axiale de l'écoulement.

Les modèles de mélange développés jusqu'à présent utilisent des corrélations empiriques, lesquelles ont un domaine d'applicabilité; ceci limite le modèle; car au moment où la corrélation est extrapolée, les résultats obtenues ne seront pas fiables. Tye (1992) mentionne que l'utilisation d'une seule corrélation pour la modélisation des paramètres de mélange ne peut pas représenter adéquatement toutes les expériences analysées. Ce problème se traduit par l'impossibilité de prédire correctement le FCC, ou dans le cas de la conception, le code est limité à être utilisé seulement dans le domaine d'applicabilité de ces corrélations.

Basée sur cette observation, nous proposons d'utiliser un algorithme génétique (AG) pour chercher les meilleurs paramètres de mélange qui permettront de reproduire la bonne distribution de l'écoulement dans les sous-canaux interconnectés. Le motif de l'utilisation d'un AG est sa facilité d'adaptation avec d'autres codes et la possibilité d'explorer un ample espace de solutions sans rencontrer des problèmes de stabilité numérique.

1.0 Objectif

L'objectif de cette recherche est d'optimiser les paramètres de mélange dans un modèle de sous-canaux élaboré pour des écoulements diphasiques verticaux (Teyssedou, 1992). Pour atteindre notre objectif, il est nécessaire de développer une fonction objectif qui permet le lien entre le code de sous-canaux et l'AG.

Pour atteindre ce but, le modèle est divisé en deux parties. La première partie se penche sur la résolution des équations de conservation développées pour décrire les écoulements dans les sous-canaux interconnectés. Une méthode numérique appelée "SIMPLER" a été utilisée (Patankar, 1980). La deuxième partie fait le point sur un algorithme génétique pour déterminer le point optimal (minimum) de la fonction objectif.

1.1 Organisation de ce mémoire

En résumé, le présent chapitre démontre la justification de la réalisation de ce projet. Le chapitre deux donne la revue bibliographique des expériences effectuées sur les sous-canaux, ainsi que les modèles de mélange existants. Le chapitre trois présente les équations de base du modèle de sous-canal, ainsi que la discrétisation de ces équations pour l'application numérique. Le chapitre quatre est dédié à l'étude des AG. Le chapitre cinq est consacré à la détermination de la fonction objectif, le chapitre six présente les résultats obtenus et leur analyse, et finalement, le chapitre sept contient les conclusions et les recommandations.

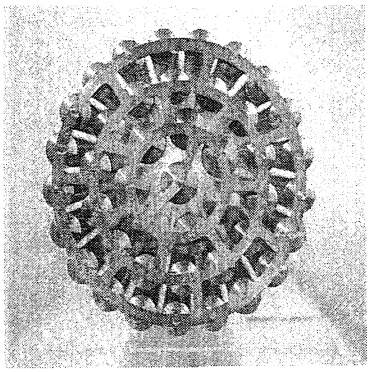


Figure 1.1: La section de passage d'une grappe de combustible.

CHAPITRE 2

REVUE BIBLIOGRAPHIQUE

Ce chapitre a pour objectif de faire une revue des expériences et des études réalisées pour l'analyse des mécanismes de mélange entre les sous-canaux et les différents modèles de mélange qui ont été proposés à partir de ces résultats.

Plusieurs chercheurs (Teyssedou, 1987; Tye, 1991; Lahey et Moody, 1993) ont déterminé que les mécanismes de mélange les plus importants sont :

-Écoulement latéral forcé : Ce phénomène est causé par la différence de pression latérale, ainsi un échange latéral de masse entre les sous-canaux permet une diminution de cette différence. La différence de pression latérale est causée, entre autres, par les conditions à l'entrée et les asymétries dans la distribution du flux de chaleur. Le gonflement et déformation des éléments de combustible peuvent entraîner une obstruction partielle ou totale d'un ou plusieurs sous-canaux. Ces derniers favorisent l'écoulement latéral forcé et entraînent un transfert de masse d'un canal vers ceux qui lui sont adjacents. L'écoulement latéral est souvent modélisé par un coefficient de perte de pression latérale (Rouhani, 1974; Tapucu *et al.,* 1988; Teyssedou *et al.,* 1992).

-Mélange turbulent : Ce mécanisme de mélange est causé par les fluctuations stochastiques des propriétés scalaires et vectorielles de l'écoulement, c'est-à-dire, pression, température, vitesse. Le mélange turbulent a été étudié par plusieurs chercheurs. Parmi les dernières études et expériences se trouvent : Sadatomi *et al.,* 1996, Kawahara *et al.,* 2000, Sadatomi *et al.,* 2004. Ces auteurs ont proposé de séparer le mélange turbulent selon le type d'écoulement :

Premièrement, *dans un écoulement monophasique* le mélange turbulent est occasionné principalement par la diffusion turbulente (Sadatomi *et al.,* 1996). Les sous-canaux peuvent avoir une température différente, ce qui produit un échange

net d'énergie du sous-canal plus chaud vers le canal plus froid, cependant si la variation de la densité est négligeable, le transfert net de masse sera nul.

Deuxièment, la classification pour un système diphasique est faite selon le type d'écoulement :

Dans l'écoulement du type à bulles le transfert turbulent de la phase liquide est très similaire à celui d'un écoulement monophasique, tandis que pour le transfert de la phase gazeuse est insignifiant.

Dans l'écoulement du type bouchon, il y a une relation très proche entre le taux du mélange turbulent de la phase liquide et les fluctuations de la différence de pression latérale entre les sous-canaux interconnectés. Cependant il n'existe pas un bon modèle pour représenter cette relation. Pour la phase gazeuse, le taux de mélange du gaz augmente approximativement dix fois par rapport à un écoulement gazeux monophasique.

Kawahara *et al.*, (2000) ont proposé un modèle pour prédire le taux de mélange turbulent dans un écoulement du type bouchon, son modèle tient compte des trois éléments suivants : la diffusion turbulente, le transfert par convection et les fluctuations de la pression latérale entre les sous-canaux interconnectés. Cependant son modèle est basé sur des corrélations empiriques élaborées dans des conditions de pression et de température atmosphérique, il faudra déterminer le domaine d'applicabilité de ce modèle.

Dans l'écoulement du type annulaire, le taux de transfert pour la phase gazeuse peut être considéré comme celui d'un écoulement gazeux monophasique, ce taux de transfert peut être calculé comme la somme de trois différentes composantes qui sont : la diffusion turbulente, le transfert par convection et les fluctuations de la pression latérale entre les sous-canaux interconnectés. Pour la phase liquide, le taux de transfert diminue de façon importante.

-Dérive du vide vers les régions à haute vitesse : Ce mécanisme est occasionné par la tendance de l'écoulement vers une distribution d'équilibre.

Shirai et Ninokata (2001) supposent que ce phénomène, appelé aussi "void drift", est le mécanisme qui permet une redistribution des phases de l'écoulement pour attendre les conditions de l'équilibre hydrodynamique et qu'il est complètement indépendant des autres mécanismes de mélange.

L'équilibre hydrodynamique dans un système de sous-canaux interconnectés est caractérisé par (Bellil, 1998; Shirai et Ninokata, 2001) :

a) Aucune différence de pression entre les sous-canaux interconnectés est observée.

b) Aucun échange net de masse pour chaque phase entre les sous-canaux est observé.

Il faut dire que ces conditions sont moyennées dans le temps, c'est-à-dire qu'à chaque instant il est possible d'avoir des échanges dans les sous-canaux, mais une fois moyennés dans le temps, les échanges nets sont nuls.

Shirai et Ninokata (2001) ont conclu que la distribution de l'écoulement à l'équilibre est fortement liée au gradient de pression axial; ils montrent que lorsque le système se trouve à l'équilibre hydrodynamique ce gradient de pression est minimal. Ces auteurs ont utilisé un modèle basé sur le principe de minimisation de l'énergie dissipée pour déterminer la distribution de l'écoulement à l'équilibre.

Tahir et Carver (1982) ont utilisé le principe de minimisation de la création de l'entropie pour expliquer le comportement de la phase gazeuse qui se déplace vers les régions à haute vitesse. Pour valider leur idée, ils ont utilisé le code ASSERT. Ce code d'analyse de sous-canal peut calculer l'entropie créé par chaque sous-canal. La méthodologie utilisée a été de faire varier les coefficients de diffusion de chaque canal jusqu'au moment où la création de l'entropie est minimale. Le modèle montre que pour un système de sous-canal formé par deux sous-canaux ayant la même géométrie, les coefficients de diffusion sont égaux.

La figure 2.1 illustre une représentation de l'écoulement latéral causé par trois mécanismes : l'écoulement latéral forcé, mélange turbulent, dérive du vide vers les

régions à haute vitesse entre deux sous-canaux en position vertical.

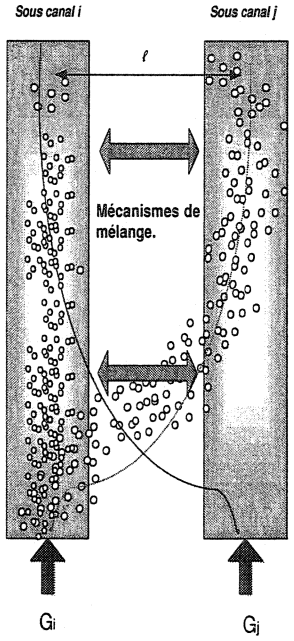


Figure 2.1: Représentation du mélange latéral

-Dérive causée par la gravité : Dans les canaux horizontaux avec un sous-canal plus élevé que l'autre, la phase la moins dense est naturellement poussée vers le haut, perpendiculairement à l'écoulement.

La figure 2.2 illustre deux sous-canaux en position horizontale. Le cas a) il n'y a pas d'échange de masse dû à la dérive causée par la gravité, car les deux sous-canaux sont à la même hauteur, tandis que pour les cas b) et c) un échange de masse est produit dû à la différence de hauteur entre les sous-canaux interconnectés.

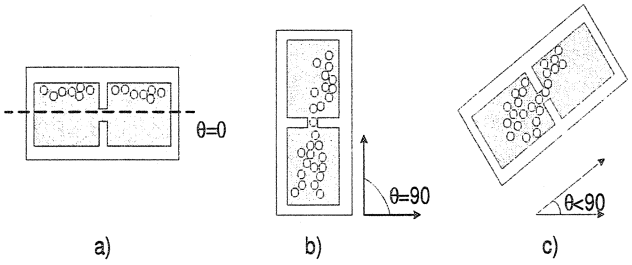


Figure 2.2: Dérive causée par la gravité

2.1 Modèles d'échange de masse pour un écoulement monophasique

Cette section est dédiée à la présentation des modèles d'échange utilisés pour la simulation des écoulements monophasiques. Avant de présenter ces modèles, il est d'abord convenable de définir la notation utilisée pour identifier les échanges latéraux entre les sous-canaux. Les sous-indices ij sont utilisés pour représenter l'écoulement qui passe du sous-canal i vers le sous-canal j, le sous-indice $i \leftrightarrow j$ est utilisé pour représenter **l'échange net** de la propriété qui va du sous-canal i vers le sous-canal j.

Le débit massique latéral par unité de longueur d'interconnexion $W'_{i\leftrightarrow j}$ est principalement dû à deux mécanismes : la contribution turbulente et la différence de pression latérale. Pour un système à une seule phase, sans variations des températures importantes (Toedras et Kazimi, 1990), la densité de la phase peut être considérée constante.

Le modèle d'échange volume à volume est basé sur l'idée d'un échange de volumes égaux entre les sous-canaux interconnectés (montré dans la figure 2.3). En termes mathématiques, on peut écrire :

$$\dot{V}'_{ij} = \dot{V}'_{ji}, \tag{2.1}$$

où :

$'$ = indice qui représente les fluctuations dans le temps,

$\dot{V}'_{ij}$ = débit volumique qui passe du canal i vers le canal j $[m^3/s]$,

$\dot{V}'_{ji}$ = débit volumique qui passe du canal j vers le canal i $[m^3/s]$.

Étant donné que :

$$\dot{V}' = v'\Omega, \tag{2.2}$$

où

Ω = la section de passage de l'interconnexion ($c\,\Delta z$, avec c = espacement de l'interconnexion) $[m^2]$,

$v^{'}$ = composante fluctuante de la vitesse de l'écoulement latérale $[m/s]$.

Si les fluctuations de vitesses latérales sont égales, on peut écrire :

$$v_{ij}^{'} = v_{ji}^{'}. \tag{2.3}$$

L'écoulement latéral qui passe d'un canal vers l'autre par unité de longueur d'interconnexion est :

$$W_{ij}^{'} = \frac{\rho_i v_{ij}^{'} c \Delta z}{\Delta z}. \tag{2.4}$$

où :

$\Delta z =$ incrément axial $[m]$.

Cette équation peut être écrite en termes d'une longueur de mélange turbulente ℓ en considérant que $W_{ij}^{'}$ est proportionnel aux variations latérales de vitesses, alors :

$$\rho_i \left[\ell \frac{dv_z}{dy} \right] = \rho_i \left[\frac{\ell^2 \frac{dv_z}{dy}}{\ell} \right]. \tag{2.5}$$

En introduisant un coefficient de diffusion ϵ, tel que :

$$\ell^2 \frac{dv_z}{dy} = \epsilon, \tag{2.6}$$

$$W_{ij}^{'} = \rho_i \frac{\epsilon}{\ell}. \tag{2.7}$$

Étant donné qu'il y a un échange des volumes égaux entre les sous-canaux interconnectés, le développement pour l'écoulement qui passe du canal j vers le canal i est le même, c'est-à-dire :

$$W_{ji}^{'} = \rho_j \frac{\epsilon}{\ell}. \tag{2.8}$$

Étant donné que la valeur de ℓ est indépendante de la séquence des sous-indices, on peut écrire :

$$W_{i \leftrightarrow j}^{'} = W_{ij}^{'} - W_{ji}^{'} = \left(\frac{\epsilon}{\ell} \right) (\rho_i - \rho_j). \tag{2.9}$$

Il faut remarquer que le développement jusqu'à l'équation 2.9 est valide pour un écoulement monophasique et diphasique.

Or, pour un système monophasique de densité constante $\rho_i = \rho_j = \rho$, le transfert net de masse qui résulte est :

$$W'_{i\leftrightarrow j} = 0. \tag{2.10}$$

Le modèle d'échange de volumes égaux pour un système monophasique où ρ est constante équivaut au modèle d'échange des masses égales, alors :

$$W'_{ij} = W'_{ji}. \tag{2.11}$$

Ce modèle a été largement utilisé dans le code de calcule COBRA IV (Stewart *et al.*, 1977). Dans ce modèle W'_{ij} est corrélé en fonction de l'espacement de l'interconnexion c et la moyenne arithmétique du flux de masse axial $\overline{G}$:

$$W'_{ij} = \beta c \overline{G}, \tag{2.12}$$

où β est un coefficient de mélange.

On présente dans la prochaine section une analyse plus détaillée pour le cas d'un écoulement diphasique avec densités différentes d'un sous-canal à l'autre.

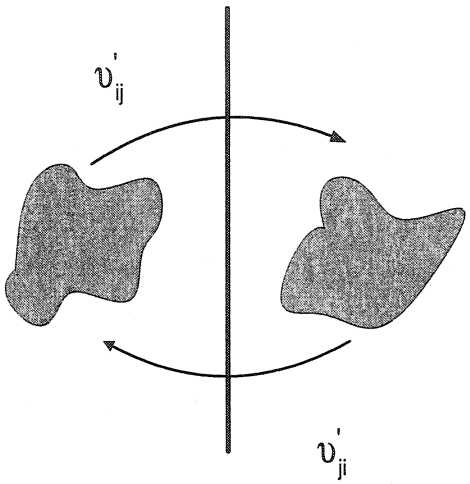


Figure 2.3: La représentation du modèle d'échange volume à volume.

2.2 Modèle d'échange de masse pour un écoulement diphasique

Modèle de Lahey

Lahey et Moody (1993) proposent un modèle de mélange basé sur l'idée d'un échange de masse par volumes égaux. En effet, les observations expérimentales des écoulements diphasiques montrent un transfert net de masse entre les sous-canaux.

En prenant l'équation 2.9 et la définition de la densité de mélange donnée par :

$$\rho_m = (1-\alpha)\rho_l + \alpha\rho_g, \tag{2.13}$$

où :

$\rho_l =$ densité de la phase liquide $[kg/m^3]$,

$\rho_g =$ densité de la phase gazeuse $[kg/m^3]$,

$\alpha =$ taux de vide.

Le transfert de masse latéral du sous-canal i vers le sous-canal j est donné par :

$$W_{ij}^{'} = \left(\frac{\epsilon}{\ell}\right)(\rho_{m_i} - \rho_{mj}) \tag{2.14}$$

où :

ρ_{m_i} = densité du mélange du canal i $[kg/m^3]$,

ρ_{m_j} = densité du mélange du canal j $[kg/m^3]$.

Si on utilise la définition de densité de mélange donné par l'équation 2.13, on obtient :

$$W_{i\leftrightarrow j}^{'} = \frac{\epsilon\rho_\ell}{\ell}[(1-\alpha_i) - (1-\alpha_j)] + \frac{\epsilon\rho_g}{\ell}[(\alpha_i - \alpha_j)]. \tag{2.15}$$

Cette équation prédit une distribution latérale vers un écoulement uniforme, i.e., même taux de vide dans les sous-canaux, bien que les expériences montrent un équilibre avec une distribution de taux de vide non-uniforme (Gonzales-Santalo, 1971). Ce modèle est évidemment incomplet. Il faudrait donc introduire un nouveau mécanisme de transfert de masse pour que cette équation puisse reproduire les tendances observées en laboratoire. Ce mécanisme, qui est proportionnel au gradient de taux de vide d'une distribution non-uniforme, est la dérive du vide vers les régions à haute vitesse. Gonzales-Santalo (1971) a proposé un modèle, pour tenir compte des résultats expérimentaux :

Modèle de Gonzales-Santalo

Le modèle développé par Gonzales-Santalo (1971) suppose que la phase liquide est continue et que la phase gazeuse est dispersée. Par analogie au processus de diffusion, le débit latéral du gaz dans un sous-canal, c'est-à-dire, canal i vers le canal j peut être exprimé comme suit :

$$\frac{dQ_g}{dy} = -cD\frac{d\alpha}{dy}, \tag{2.16}$$

où D est un coefficient de diffusion du vide et c représente l'espace de l'interconnexion.

La forme discrétisée de cette équation est donnée par :

$$\frac{\Delta Q_g}{\Delta y} = -cD\frac{(\langle\alpha_i\rangle - \langle\alpha_j\rangle)}{\Delta y}. \tag{2.17}$$

On a expérimentalement observé qu'à la condition d'équilibre, c'est-à-dire, quand il n'y a plus d'échanges latéraux entre les sous-canaux, la différence des taux de vide entre ces derniers n'est pas nulle (Bellil, 1998). Il est évident que l'équation 2.17 est insuffisante pour satisfaire une telle condition. Pour tenir compte de cette observation, cette équation doit être modifiée en définissant $(\langle\alpha_i\rangle - \langle\alpha_j\rangle)_{eq}$ comme la différence des taux de vide pour laquelle $\Delta Q_g = 0$.

$$\frac{\Delta Q_g}{\Delta y} = -\frac{cD}{\Delta y}[(\langle\alpha_i\rangle - \langle\alpha_j\rangle) - (\langle\alpha_i\rangle - \langle\alpha_j\rangle)_{eq}]. \tag{2.18}$$

De cette manière lorsque l'écoulement est en condition d'équilibre hydrodynamique, les termes $(\langle\alpha_i\rangle - \langle\alpha_j\rangle)$ et $(\langle\alpha_i\rangle - \langle\alpha_j\rangle)_{eq}$, seront égaux; à ce moment là, ΔQ_g sera égal à zéro. Comme dans l'expérience de Gonzales-Santalo, le taux de vide $\langle\alpha\rangle$ n'a pas été mesuré, une relation entre α et β a été introduite. Ainsi, l'équation 2.17 peut s'écrire en termes du titre volumétrique de l'écoulement β en posant $\langle\alpha\rangle = \gamma\beta$ avec $\gamma =$ facteur de proportionnalité (Armand, 1946) .

D'où :

$$\frac{\Delta Q_g}{\Delta y} = -\frac{\gamma Dc}{\Delta y}[(\beta_i - \beta_j) - (\beta_i - \beta_j)_{eq}]. \tag{2.19}$$

Si on regroupe les valeurs $-\frac{\gamma Dc}{\Delta y}$ dans une constante K, définie comme le coefficient de diffusion turbulente, l'équation 2.19 devient :

$$\frac{\Delta Q_g}{\Delta y} = K[(\beta_i - \beta_j) - (\beta_i - \beta_j)_{eq}]. \tag{2.20}$$

Modèle de mélange de Shoukri

Shoukri *et al.,* (1984) ont étudié la redistribution des écoulements diphasiques horizontaux dans des sous-canaux interconnectés latéralement. Ces expériences ont été effectuées de telle sorte que l'échange de masse entre les sous-canaux se

produise en absence de différence de pression latérale entre les sous-canaux. Le fait d'avoir une différence de pression latérale nulle permet d'éliminer l'échange de masse causé par l'écoulement latéral forcé. Par conséquent, les échanges qui interviennent sont seulement dus à la force de gravité, entraînant la séparation des deux phases et à la diffusion turbulente du vide qui tend à rendre la distribution des deux phases plus homogènes. L'importance relative de chaque mécanisme est fonction du flux de masse du mélange et du titre de l'écoulement. Les données obtenues concernant la distribution des deux phases présentent un comportement asymptotique se rapprochant des distributions d'équilibre au-delà desquelles aucun échange n'est observé. Ces distributions d'équilibre sont fonction du flux de masse et du titre volumique de l'écoulement à l'entrée. La distribution à l'équilibre est causée par l'action simultanée de deux mécanismes : la séparation des phases due à la gravité et la diffusion turbulente du vide. Ces mécanismes sont inclus dans l'équation de mélange en utilisant un modèle de diffusion simple.

Le modèle de Shoukri *et al.,* (1984) est basé sur le modèle à écart de vitesse. La composante latérale de la vitesse du gaz (Zuber et Findlay, 1965) s'écrit comme suit :

$$\vec{V}_g = C_0\langle j\rangle + \vec{V}_{gj}, \tag{2.21}$$

où :

C_0 = paramètre de distribution qui tient compte de la distribution non-uniforme de α et j, à travers l'interconnexion,

$\langle j\rangle$ = densité de flux volumétrique $[m/s]$,

$\vec{V}_{gj}$ = écart de vitesse $[m/s]$.

L'écart de vitesse dans la direction latérale est considéré comme une fonction de la gravité et de la diffusion turbulente du vide :

$$\vec{V}_{gj} = (1-\langle\alpha\rangle)^n V_\infty - \frac{D_\alpha}{\langle\alpha\rangle}\nabla_y\langle\alpha\rangle, \tag{2.22}$$

où :

$\langle\alpha\rangle$= taux de vide moyenné sur la section de passage,

V_∞= vitesse terminale des bulles de gaz $[m/s]$,

D_α= coefficient de diffusion turbulente du vide $[m^2/s]$.

Cependant, pour ce modèle, Shoukri propose que $n = 0$, alors :

$$\vec{V}_{gj} = V_\infty - \frac{D_\alpha}{\langle\overline{\alpha}\rangle}\nabla_y\langle\alpha\rangle. \tag{2.23}$$

La forme discrétisée de cette équation s'écrit comme suit :

$$\vec{V}_{gj} = V_\infty - \frac{D_\alpha}{\langle\overline{\alpha}\rangle}\frac{\Delta\langle\alpha\rangle}{\Delta y}. \tag{2.24}$$

En introduisant cet écart de vitesse dans l'équation 2.21, on obtient :

$$\vec{V}_g = C_0\langle j\rangle + V_\infty - \frac{D_\alpha}{\langle\overline{\alpha}\rangle}\frac{\Delta\langle\alpha\rangle}{\Delta y}. \tag{2.25}$$

En appliquant cette équation aux sous-canaux horizontaux, le débit volumétrique du gaz dans la direction latérale à travers l'espacement c et la longueur de l'interconnexion Δz peut s'écrire de la façon suivante :

$$Q_g = c\Delta z\langle\overline{\alpha}\rangle\left[C_0\langle j\rangle + V_\infty - \frac{D_\alpha}{\langle\overline{\alpha}\rangle}\left(\frac{\langle\alpha_j\rangle - \langle\alpha_i\rangle}{\ell}\right)\right] \tag{2.26}$$

avec

$$\langle\overline{\alpha}\rangle = \frac{1}{2}(\langle\alpha\rangle_j + \langle\alpha\rangle_i),$$

où :

j = densité du flux volumétrique latéral du mélange $[m/s]$,

ℓ = longueur de mélange $[m]$,

c = espacement de l'interconnexion $[m]$.

Shoukri *et al.*, (1984) considèrent que la valeur de C_0 pour l'écoulement du gaz dans la direction latérale est égale à 1. En multipliant l'équation 2.25 par la densité du gaz ρ_g, on obtient :

$$W_g = s\Delta y\rho_g\langle\overline{\alpha}\rangle\left[\langle j\rangle + V_\infty - \frac{D_\alpha}{\langle\alpha\rangle}\left(\frac{\langle\alpha_j\rangle - \langle\alpha_i\rangle}{\Delta\ell}\right)\right], \tag{2.27}$$

où la vitesse terminale est donnée par la corrélation de Wallis (1969) :

$$V_\infty = K\left(g\sigma\frac{\rho_l - \rho_g}{\rho_l^2}\right)^{\frac{1}{4}}, \tag{2.28}$$

avec :

$\sigma =$ tension superficielle de la phase liquide $[N/m]$,

$K =$ constante ajustée avec les données expérimentales.

Le premier terme à droite de l'équation 2.26 représente la contribution occasionnée par l'écoulement latéral forcé. Le deuxième terme représente la contribution liée à la séparation par gravité et enfin, le dernier terme correspond à la contribution due à la diffusion turbulente du vide.

Puisque les échanges entre les sous-canaux sont gouvernés par quatre mécanismes et que Shoukri *et al.* ne tiennent pas compte de la dérive du vide vers les régions à haute vitesse, son modèle reste alors incomplet. La contribution de ce mécanisme est causé par la différence du taux de vide à l'équilibre entre les sous-canaux.

Modèle de mélange utilisé dans le code ASSERT-4

Le modèle utilisé par le code ASSERT-4 (Carver *et al.*, 1987) est basé sur un modèle de diffusion qui débute avec l'équation 2.21 et prend en considération le terme $(1-\alpha)^n$.

Dans ce modèle, la vitesse du gaz dépend des facteurs suivants :

- la distribution de l'écoulement qui est donnée par :

$$C_0 = \frac{\langle\alpha\vec{j}\rangle}{\langle\alpha\rangle\langle\vec{j}\rangle}, \tag{2.29}$$

- le champ gravitationnel considéré dans la vitesse terminale V_∞,

- la diffusion turbulente du vide, provoquée par la différence latérale du vide, est donnée par :

$$-\frac{D_\alpha}{\langle\alpha\rangle}\overline{\nabla}\langle\alpha\rangle, \tag{2.30}$$

où D_α est le coefficient de diffusion turbulente du vide $[m^2/s]$.

Le modèle proposé par Carver *et al.*, (1987) définit la vitesse du gaz de la façon suivante :

$$\vec{V_g} = C_0\langle\vec{j}\rangle + (1-\alpha)^n\vec{V}_\infty - \epsilon_i\nabla[\langle\alpha\rangle_i - \langle\alpha\rangle_{i_{eq}}] + \epsilon_j\nabla[\langle\alpha\rangle_j - \langle\alpha\rangle_{j_{eq}}], \tag{2.31}$$

où $\langle\alpha_{eq}\rangle$ est la fraction du vide à l'équilibre dans chaque sous-canal et ϵ est un coefficient de diffusion qui dépend de la géométrie du sous-canal.

Carver *et al.*, (1987) proposent que le coefficient de diffusion pour des sous-canaux interconnectés pourrait être corrélé de la façon suivante :

$$\frac{\epsilon_i}{\epsilon_j} = f\left(\frac{d_j}{d_i}\right)^n, \tag{2.32}$$

où d_k représentent les diamètres hydrauliques des canaux considérés.

Le taux du vide à l'équilibre est évalué à partir du modèle de Lahey et Moody (1993), et est exprimée par la relation suivante :

$$[\langle\alpha\rangle_i - \langle\alpha\rangle_j]_{eq} = \frac{\langle\overline{\alpha}\rangle}{\overline{G}}[G_i - G_j] \tag{2.33}$$

avec :

$$\langle\overline{\alpha}\rangle = \frac{1}{2}(\langle\alpha\rangle_j + \langle\alpha\rangle_i) \tag{2.34}$$

et

$$\overline{G} = \frac{1}{2}(\langle G\rangle_j + \langle G\rangle_i). \tag{2.35}$$

Pour le modèle utilisé par le code ASSERT-4, les coefficients de diffusion des sous-canaux adjacents sont considérés égaux: $\epsilon_i = \epsilon_j = \epsilon_\alpha$. Ainsi, en introduisant cette modification dans l'expression de la vitesse du gaz, on obtient :

$$\vec{V_g} = C_0\langle\vec{j}\rangle + (1-\alpha)^n\vec{V}_\infty - \epsilon_\alpha\nabla[\langle\alpha\rangle_i - \langle\alpha\rangle_j] + \epsilon_\alpha\nabla[\langle\alpha\rangle_i - \langle\alpha\rangle_j]_{eq}. \tag{2.36}$$

Si on exprime le coefficient de diffusion comme suit :

$$\epsilon_\alpha = \frac{D_\alpha}{\langle\alpha\rangle\ell}, \tag{2.37}$$

où :

$D_\alpha =$ coefficient de diffusion utilisé par Shoukri *et al.*, (1984) $[m^2/s]$,

$\ell =$ longueur de mélange $[m]$,

l'équation 2.35 peut être écrite de la façon suivante :

$$\vec{V}_g = C_0\langle\vec{j}\rangle + (1-\langle\alpha\rangle)^n\vec{V}_\infty - \frac{D_\alpha}{\langle\alpha\rangle}\frac{\Delta\langle\alpha\rangle}{\ell} + \frac{D_\alpha}{\langle\alpha\rangle}\frac{\Delta\langle\alpha\rangle_{eq}}{\ell}. \tag{2.38}$$

Le modèle du code ASSERT-4 utilise les composantes de la vitesse relative (i.e., $\vec{V}_r = \vec{V}_g - \vec{V}_l$) pour exprimer les vitesses des phases :

$$\vec{V}_r = \frac{1}{\langle 1-\alpha\rangle}\Big[(C_0-1)\langle\vec{j}\rangle + (1-\alpha)^n\vec{V}_\infty - \frac{D_\alpha}{\langle\alpha\rangle}\frac{\Delta\langle\alpha\rangle}{\Delta_y} + \frac{D_\alpha}{\langle\alpha\rangle}\frac{\Delta\langle\alpha\rangle_{eq}}{\Delta_y}\Big]. \tag{2.39}$$

Dans la direction latérale, le taux de vide peut être considéré comme étant distribué de manière uniforme (*i.e.* $C_0 = 1.0$), dans ce cas la vitesse relative est donnée par :

$$\vec{V}_r = \frac{1}{\langle 1-\alpha\rangle}\Big[(1-\alpha)^n\vec{V}_\infty - \frac{D_\alpha}{\langle\alpha\rangle}\nabla\langle\alpha\rangle + \frac{D_\alpha}{\langle\alpha\rangle}\nabla\langle\alpha\rangle_{eq}\Big], \tag{2.40}$$

Le coefficient de diffusion turbulent du vide est calculé à partir du nombre de Peclet selon l'équation suivante :

$$P_e = \frac{D_\alpha}{\overline{U}\,\overline{D_h}} = 0.075\Big(\frac{\bar{\alpha}}{0.6}\Big)^6, \tag{2.41}$$

où :

$\overline{U} =$ est la moyenne des vitesses axiales du mélange des sous-canaux adjacents $[m/s]$,

$\overline{D_h} =$ est la moyenne des diamètres hydrauliques des sous-canaux adjacents $[m]$.

CHAPITRE 3

MODÈLE DE SOUS-CANAL

Le but de ce chapitre est de montrer le système d'équations utilisées pour la représentation mathématique des sous-canaux et la méthode utilisée pour résoudre ce système d'équations.

Ces équations ont été développées par Teyssedou *et al.*, (1992) pour décrire les écoulements adiabatiques en régime stationnaire dans des sous-canaux partiellement obstrués, de géométrie carrée et orientés verticalement.

3.1 Les équations de conservation

Les équations ont été développées en utilisant un modèle d'écoulement séparé. Pour le cas d'un écoulement adiabatique dans deux sous-canaux interconnectés, les équations se résument comme suit :

- deux équations de conservation de masse, pour le gaz et pour le mélange,
- une équation de conservation de la quantité de mouvement axial du mélange,
- une équation de conservation de la quantité de mouvement latérale.

3.1.1 Équation de conservation de la masse

Les équations de conservation de masse ont été développées à partir de l'équation générale de conservation (Teyssedou, 1987).

Équation de conservation de la masse pour le liquide

$$\frac{\partial}{\partial t}(1-\alpha)A\rho_l + \frac{\partial}{\partial z}AG_l = (A\Gamma_l)_{l,g} - cG_l^{'}. \tag{3.1}$$

Équation de conservation de la masse pour le gaz

$$\frac{\partial}{\partial t}\alpha A\rho_g + \frac{\partial}{\partial z}AG_g = (A\Gamma_g)_{g,l} - cG_g^{'}. \tag{3.2}$$

où

A = section de passage du canal $[m^2]$,

c = largeur de l'interconnexion $[m]$,

$G^{'}$ = flux de masse latéral $[kg/m^2s]$,

G = flux de masse axial $[kg/m^2s]$,

ρ_l = densité du liquide $[kg/m^3]$,

ρ_g = densité du gaz $[kg/m^3]$,

α = taux de vide moyenné sur la surface.

Les sous-indices l et g représentent la phase liquide et la phase gazeuse respectivement.

Le premier terme à gauche des équations 3.1 et 3.2 représente la variation temporelle de la densité de la phase. Le deuxième terme représente la variation spatiale du flux de masse de la phase. Le premier terme à droite des équations 3.1 et 3.2 représente la condensation ou l'évaporation à l'interface (nul dans notre cas, car il s'agit d'un mélange air-eau). Quant au dernier terme de droite, il représente le flux de masse échangé à travers l'interconnexion.

Équation de conservation de la masse pour le mélange

La somme de ces deux équations donne :

$$\frac{\partial}{\partial t}A[(1-\alpha)\rho_l + \alpha\rho_g] + \frac{\partial}{\partial z}A(G_l + G_g) = -c(G_l^{'} + G_g^{'}). \tag{3.3}$$

En utilisant la définition de la densité de mélange donnée par :

$$\rho_m = (1-\alpha)\rho_l + \alpha\rho_g, \tag{3.4}$$

l'équation 3.3 peut s'écrire sous la forme suivante :

$$\frac{\partial}{\partial t}A\rho_m + \frac{\partial}{\partial z}A(G_l + G_g) = -c(G_l^{'} + G_g^{'}). \tag{3.5}$$

Il faut mentionner que dû à la condition de transfert de masse à l'interface, la somme des termes d'évaporation et condensation devient nulle, c'est-à-dire : ($\Gamma_g + \Gamma_l = 0$)

La figure 3.1 représente les termes de l'équation 3.5 dans un volume de contrôle.

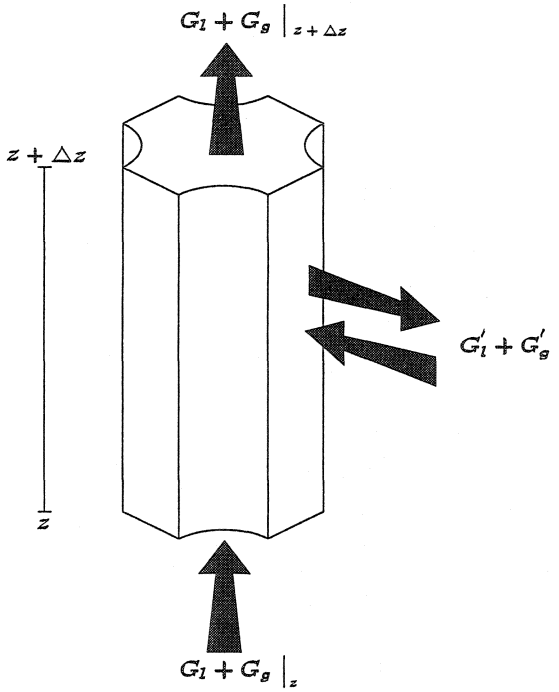


Figure 3.1: Volume de contrôle pour le bilan de conservation de la masse.

3.1.2 Équation de conservation de la quantité de mouvement axiale

L'équation de conservation de la quantité de mouvement axiale pour un écoulement vertical est donnée par :

$$\frac{\partial}{\partial t}AG + \frac{\partial}{\partial z}\left[A\left(\frac{\hat{x}}{\alpha\rho_g}G_g + \frac{(1-\hat{x})}{(1-\alpha)\rho_l}G_l\right)G\right]$$
$$+cF_m\left[\frac{\hat{x}^\star}{\alpha^\star\rho_g^\star}G_g' + \frac{(1-\hat{x})^\star}{(1-\alpha^\star)\rho_l^\star}G_l'\right]G^\star + c\lambda G\left\{\hat{x}\left[\left(\frac{G_g}{\alpha\rho_g}\right)_r - \left(\frac{G_g}{\alpha\rho_g}\right)_d\right]\right.$$
$$\left.+(1-\hat{x})\left[\left(\frac{G_l}{(1-\alpha)\rho_l}\right)_r - \left(\frac{G_l}{(1-\alpha)\rho_l}\right)_d\right]\right\}$$

$$= -A\frac{\partial}{\partial z}p - A\rho_m g + \gamma_\omega \tau_\omega + c(\tau_{dr})_z, \tag{3.6}$$

où :

* = indique des valeurs qui sont évaluées à l'interconnexion ,

r, d = représentent le canal receveur et donneur respectivement,

G = flux massique du mélange $(G_g + G_l)$ $[kg/m^2 s]$,

$\hat{x}$ = titre de l'écoulement,

α = taux de vide moyenné sur la surface,

λ = coefficient de mélange turbulent,

γ_ω = contrainte de cisaillement à la paroi $[N/m^2]$,

$(\tau_{dr})_z$ = contrainte de cisaillement à l'interconnexion $[N/m^2]$.

La figure 3.2 représente le bilan de conservation de la quantité de mouvement axiale dans un volume de contrôle.

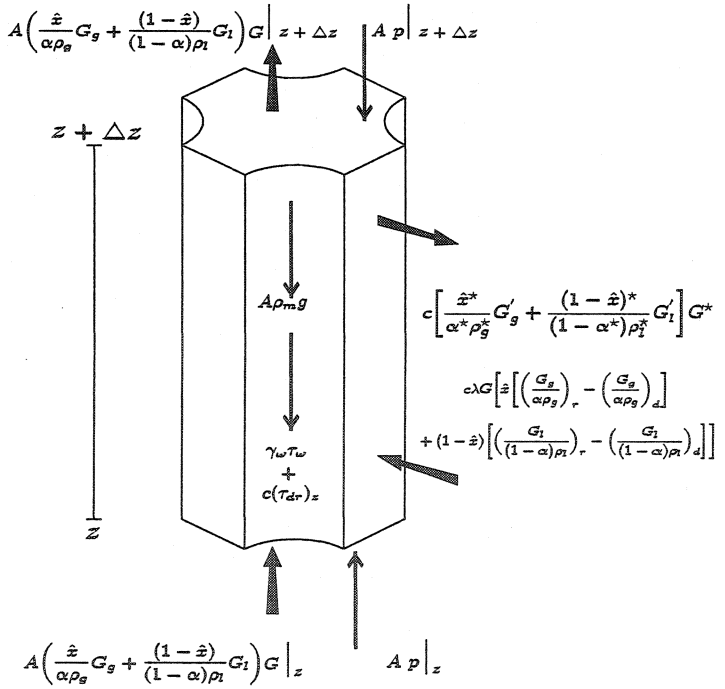


Figure 3.2: Volume de contrôle pour le bilan de conservation de la quantité de mouvement axiale.

Le premier terme du côté gauche de l'équation 3.6 représente l'accélération locale de l'écoulement. Le deuxième terme représente la quantité de mouvement axiale transférée par convection par l'écoulement axial. Le troisième terme représente la quantité de mouvement axiale transférée par convection due à l'écoulement latéral. Le quatrième terme représente la quantité de mouvement axiale échangée par le mélange turbulent. L'intensité du mélange turbulent est ajustée par le coefficient λ; sa valeur est inférieure à 1. Comme dans notre cas ce modèle de mélange turbulent n'est pas utilisé, la valeur de λ a été fixée à zéro.

Le premier terme à droite de l'égalité représente la perte de pression axiale. Le deuxième terme représente la perte de pression par gravité. Enfin, le troisième et le quatrième terme représentent des pertes de pression par frottement.

Le terme $\gamma_\omega \tau_\omega$ est souvent exprimé en termes d'un coefficient de frottement et d'un multiplicateur de deux phases. Cependant, le présent modèle peut simuler des cas d'obstructions donc ce terme est modifié de façon à considérer les pertes irreversibles causées par l'obstruction, alors le terme $\gamma_\omega \tau_\omega$ a la forme suivante :

$$\gamma_\omega \tau_\omega = \Phi_L^2 \left(f_0 \frac{G^2}{2\rho_l} \frac{A}{D_h} \right) + K_{form} \frac{G^2}{2\rho'} \frac{A}{D_h}, \tag{3.7}$$

où :

f_0 = coefficient de frottement en simple phase,

Φ_l^2 = multiplicateur diphasique,

D_h = diamètre hydraulique $[m]$,

ρ' = densité de la quantité de mouvement $(= \frac{\hat{x}^2}{\alpha \rho_g} + \frac{(1-\hat{x})^2}{(1-\alpha)\rho_l})$.

Le terme τ_{dr}, qui représente la contrainte de cisaillement à l'interconnexion, n'a pas été modélisé. Ce terme peut être important pour certaines géométries de l'interconnexion.

Pour des conditions d'écoulement latéral forcé important, c'est-à-dire, obstruction très sévère, l'accélération de l'écoulement latéral à travers l'interconnexion produit une fraction d'énergie mécanique non-négligeable qui est convertie en pression. Pour prendre en considération cet effet, un modèle simple a été développé par

Teyssedou (1987) :

$$\delta p = \beta \frac{(G^{'})^2}{2\rho_m^{\star}} \Big[1 - \Big(\frac{c}{a}\Big)^2\Big], \tag{3.8}$$

où:

$\delta p =$ correction de la pression axiale due à l'accélèration de l'écoulement latéral $[N/m^2]$,

$\rho_m^{\star} =$ densité du mélange à l'interconnexion $[kg/m^3]$,

$a =$ largeur équivalente d'un sous-canal $[m]$,

$\beta =$ est un coefficient empirique, positif pour le canal receveur et négatif pour le canal donneur.

En introduisant l'équation 3.7 et 3.8 dans l'équation 3.6, on obtient :

$$\begin{aligned}
&\frac{\partial}{\partial t} AG + \frac{\partial}{\partial z}\Big[A\Big(\frac{\hat{x}}{\alpha\rho_g} G_g + \frac{(1-\hat{x})}{(1-\alpha)\rho_l} G_l\Big) G\Big] \\
&+c\,F_m\Big[\frac{\hat{x}^{\star}}{\alpha^{\star}\rho_g^{\star}} G_g^{'} + \frac{(1-\hat{x})^{\star}}{(1-\alpha^{\star})\rho_l^{\star}} G_l^{'}\Big] G^{\star} + c\lambda G\Big[\hat{x}\Big[\Big(\frac{G_g}{\alpha\rho_g}\Big)_r - \Big(\frac{G_g}{\alpha\rho_g}\Big)_d\Big] \\
&+(1-\hat{x})\Big[\Big(\frac{G_l}{(1-\alpha)\rho_l}\Big)_r - \Big(\frac{G_l}{(1-\alpha)\rho_l}\Big)_d\Big]\Big] \\
&= -A\frac{\partial}{\partial z} p - A\rho_m g - \Phi_L^2 (f_0 \frac{G^2}{2\rho_l}\frac{A}{D_h}) - K_{form}\frac{G^2}{2\rho^{'}}\frac{A}{D_h} \\
&-A\frac{\partial}{\partial z}\beta\frac{(G^{'})^2}{2\rho_m^{\star}}\Big[1 - \Big(\frac{c}{a}\Big)^2\Big].
\end{aligned} \tag{3.9}$$

Il faut mentionner que dans notre étude nous n'avons pas simulé des cas avec obstructions, donc les termes K_{form} et le terme de correction de la pression axiale δp ont été mis à zéro.

3.1.3 Équation de la conservation de la quantité de mouvement latérale

L'équation de la quantité de mouvement latérale a le caractère d'une équation constitutive. Cette équation sert à coupler les équations de la quantité de mouvement axiale de chaque sous-canal. L'équation de la quantité de mouvement latérale s'écrit comme suit :

$$\frac{\partial}{\partial t}c(G_g^{'}+G_l^{'})+\frac{\partial}{\partial z}c\left[G_g^{'}\frac{G_g^{\star}}{\alpha^{\star}\rho^{\star}}+G_l^{'}\frac{G_l^{\star}}{(1-\alpha^{\star})\rho_l}\right]+\frac{\partial}{\partial y}c\left[\frac{G_g^{'\,2}}{\alpha^{\star}\rho_g^{\star}}+\frac{G_l^{'\,2}}{(1-\alpha^{\star})\rho_l}\right]$$
$$=-c\frac{\partial}{\partial y}p-\tau_{\omega y}\frac{dA_\omega}{dzdy}, \tag{3.10}$$

où :

$\tau_{\omega y}$ = contrainte de cisaillement visqueuse latérale.

Le terme de contrainte de cisaillement $\tau_{\omega y}(dA_\omega/dzdy)$ est modélisé en utilisant la loi de Darcy et un coefficient de perte de pression latérale par frottement (K_{tr}), ce qui peut se résumer par l'équation suivante :

$$\tau_{\omega y}\frac{dA_\omega}{dzdy}\simeq K_{tr}\frac{G^{'2}}{2\rho_m^{\star}}, \tag{3.11}$$

où :

$\rho_m^{\star}$ = densité de mélange à l'interconnexion $[kg/m^3]$,

$G^{'}$ = flux de masse total transféré à travers l'interconnexion $[kg/m^2s]$.

En introduisant l'équation 3.11 dans l'équation 3.10 on obtient :

$$\frac{\partial}{\partial t}c(G_g^{'}+G_l^{'})+\frac{\partial}{\partial z}c\left[G_g^{'}\frac{G_g^{\star}}{\alpha^{\star}\rho^{\star}}+G_l^{'}\frac{G^{\star}}{(1-\alpha^{\star})}\rho_l\right]+\frac{\partial}{\partial y}c\left[\frac{G_g^{'\,2}}{\alpha^{\star}\rho^{\star}}+\frac{G_l^{'\,2}}{(1-\alpha^{\star})}\rho_l\right]$$
$$=-c\frac{\partial}{\partial y}p-K_{tr}\frac{G^{'2}}{2\rho_m^{\star}}. \tag{3.12}$$

3.2 Le système d'équations

L'ensemble des équations données par :

1.- l'équation de conservation de masse du gaz (éq. 3.2),

2.- l'équation de conservation de masse du mélange (éq. 3.3),

3.- l'équation de conservation de la quantité de mouvement axiale du mélange (éq. 3.9),

4.- l'équation de conservation de la quantité de mouvement latérale (éq. 3.12).

forme un système de quatre équations avec cinq inconnues $(G_l, G_g, G_l^{'}, G_g^{'}, p)$. Il est donc nécessaire d'introduire une autre équation pour fermer le système. Le développement d'un modèle de mélange approprié permettra d'obtenir une équation supplémentaire.

3.3 Le modèle de mélange

À partir du modèle à écart de vitesse (Ishii, 1975), la vitesse latérale de la phase gazeuse moyennée sur la section de passage de l'interconnexion peut s'écrire de la façon suivante :

$$\vec{v}_g^{\alpha} = \frac{\ll \alpha(\alpha\vec{V}_g + (1-\alpha)\vec{V}_l \gg}{\ll \alpha \gg} + \frac{\ll \alpha(1-\alpha)(\vec{V}_g - \vec{V}_l) \gg}{\ll \alpha \gg}, \tag{3.13}$$

où $\vec{v}_g^{\alpha}$ est la vitesse moyenne du gaz pondérée par le taux de vide local α. Cette vitesse est alors définie comme suit :

$$\vec{v}_g^{\alpha} = \frac{\ll \alpha\vec{V}_g \gg}{\ll \alpha \gg}. \tag{3.14}$$

Si on introduit la définition du flux volumétrique et l'écart de vitesse pour la phase gazeuse qui sont respectivement donnés par :

$$\vec{J} = \alpha\vec{V}_g + (1-\alpha)\vec{V}_l, \tag{3.15}$$

$$\vec{V}_{g_j} = (1-\alpha)(\vec{V}_g - \vec{V}_l) = \vec{V}_g - \vec{J}, \tag{3.16}$$

l'équation 3.13 peut être écrite sous la forme suivante :

$$\vec{v}_g^{\alpha} = \frac{\ll \alpha\vec{J} \gg}{\ll \alpha \gg} + \frac{\ll \alpha\vec{V}_{g_j} \gg}{\ll \alpha \gg}. \tag{3.17}$$

Le terme $\ll \alpha\vec{V}_{g_j} \gg$ représente un flux moyenné qui traverse une surface qui se déplace avec la vitesse du centre de volume $(\vec{J})$. La différence $(\vec{V}_g - \vec{J})$ représente la

vitesse relative entre la vitesse de la phase gazeuse et celle du centre de volume. Si on introduit le coefficient de distribution de Zuber et Findlay (1965) défini comme suit:

$$C_0 = \frac{\ll \alpha \vec{J} \gg}{\ll \alpha \gg \ll \vec{J} \gg}, \tag{3.18}$$

l'équation 3.17 prend la forme suivante :

$$\vec{v}_g^{\alpha} = C_0 \ll \vec{J} \gg + \vec{v}_{g_j}^{\alpha}. \tag{3.19}$$

Le second terme à droite de cette équation peut être considéré proportionnel au gradient du taux du vide, alors :

$$\vec{V}_{g_j} = -d_\alpha \vec{\nabla}(\alpha), \tag{3.20}$$

où:

$$\vec{v}_{g_j}^{\alpha} \approx -\frac{\ll d_\alpha \alpha \vec{\nabla} \alpha \gg}{\ll \alpha \gg} = -D_\alpha \ll \vec{\nabla} \alpha \gg, \tag{3.21}$$

avec :

$d_\alpha =$ coefficient local de diffusion du vide $[m^2/s]$,

$D_\alpha =$ coefficient de diffusion moyen du vide $[m^2/s]$.

En introduisant l'équation 3.21 dans l'équation 3.19 et en faisant la moyenne sur la section de passage latérale au niveau de l'interconnexion, on obtient :

$$\vec{v}_g^{\alpha} = C_0 J_{dr} - D_\alpha \frac{\Delta_{dr}(\alpha)}{\ell}, \tag{3.22}$$

où α est le taux de vide moyenné sur la surface et Δ_{dr} représente la différence entre le canal donneur et le canal receveur.

Si on suppose que la densité des phases est constante dans les sous-canaux ainsi que dans la région de l'interconnexion, alors à partir de l'équation 3.22 le flux de masse du gaz à travers l'interconnexion s'écrit comme suit :

$$G_g^{'} = \alpha^{\star}\rho_g^{\star}\left[C_0 J_{dr} - D_{\alpha}\frac{\triangle_{dr}(\alpha)}{\ell}\right]. \tag{3.23}$$

Comme il a déjà été mentionné dans le chapitre précèdent dans la section des modèles de mélange, cette équation ne permet pas de représenter les distributions du taux de vide sur l'état d'équilibre hydrodynamique dans les sous-canaux interconnectés. En effet cette équation ne tient pas compte du mécanisme de mélange appelé "void drift" (le lecteur peut se référer au chapitre antérieur pour plus de détails sur ce mécanisme de mélange). Pour tenir compte de ce mécanisme, le modèle de mélange employé a un terme que représente le "void drift", ce terme est basé sur l'idée d'un processus de diffusion. L'équation 3.23 peut être alors écrite comme suit :

$$G_g^{'} = \alpha^{\star}\rho_g^{\star}\left[C_0 J_{dr} - D_{\alpha}\frac{\triangle_{dr}(\alpha)}{\ell} + D_{eq}\frac{\triangle_{dr}(\alpha)_{EQ}}{\ell}\right], \tag{3.24}$$

où D_{eq} est le coefficient de diffusion à l'équilibre. En se basant sur le modèle de Lahey et Moody (1993), la différence du vide à l'équilibre est calculée par :

$$\triangle_{dr}(\alpha)_{EQ} = \frac{\bar{\alpha}(G_d - G_r)}{\bar{G}}, \tag{3.25}$$

où $\bar{\alpha}$ et $\bar{G}$ sont les valeurs moyennes dans les sous-canaux correspondants.

Même si cette équation a été montrée dans le chapitre précèdent, il sera plus facile pour le lecteur de suivre le développement du modèle de mélange utilisé dans ce modèle de sous-canal.

Le flux de masse du gaz à travers l'interconnexion, donné par l'équation 3.24, est la résultante de trois mécanismes de mélange différents qui agissent simultanément. Le premier est le transfert du gaz causé par l'écoulement latéral forcé. Le deuxième

est la diffusion turbulente du vide causé par la présence d'un gradient du taux de vide à l'interconnexion. Le dernier est la dérive du vide vers les régions à haute vitesse ("void drift").

3.4 Le modèle numérique

La méthode numérique utilisée pour résoudre le système d'équations présentées dans la section précedente est basée sur la méthode SIMPLER (Semi-Implicit Method for Pressure-Linked Equation Revised) donné par Patankar (1980). Afin de permettre l'utilisation de cette technique, les équations sont écrites en fonction du flux de masse total (G). Ces équations sont écrites sous la forme de différences finies et par la suite, discrétisées selon une grille décalée (voir figure 3.3). Les flux de masse axiaux sont évalués aux nœuds principaux situés aux frontières des volumes de contrôle (où i représente la position des interfaces du volume de contrôle) . Quant à la pression, la fraction du vide et la densité, elles sont évaluées au centre des volumes de contrôle.

Patankar (1980) a montré que le schéma de différence centrée obtenue à partir d'un développement en séries de Taylor n'est pas la méthode adéquate pour traiter des problèmes de diffusion combinée avec la convection. Un schéma numérique en amont est alors utilisé pour discrétiser les différentes équations.

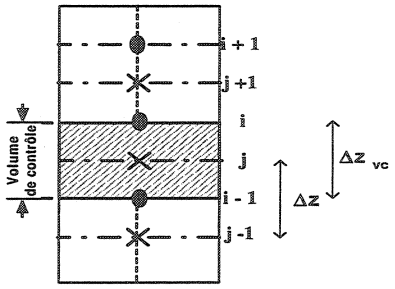


Figure 3.3: La discrétisation.

Le schéma numérique en amont stipule que la propriété transférée par convection à l'interface du volume de contrôle est égale à la moyenne des propriétés aux

interfaces avoisinantes. Cette formulation préliminaire présente un point faible en assumant que la propriété de convection à l'interface du volume de contrôle est égale à la moyenne des propriétés situées aux interfaces avoisinantes. Une amélioration a été apportée à cette formulation : elle consiste à garder le terme de diffusion inchangé tandis que le terme de convection est calculé de la façon détaillée ci-dessous.

Pour une fonction quelconque ϕ, on peut définir sa valeur à l'interface comme la valeur de ϕ située au nœud de la grille en amont de l'interface en remontant le courant (voir figure 3.3), c'est-à-dire :

si

$$G_i\phi > 0 \Longrightarrow \phi = \phi_{j-1}, \tag{3.26}$$

si

$$G_i\phi < 0 \Longrightarrow \phi = \phi_j. \tag{3.27}$$

Ces relations peuvent être écrites sous une forme compacte en introduisant la notation suivante :

$$\| \psi, 0 \| = \text{la plus grande valeur entre } \psi \text{ et } 0,$$

et qui satisfait :

$$\| \psi, 0 \| = \| -\psi, 0 \| + \psi. \tag{3.28}$$

Alors les équations 3.26 et 3.27 peuvent être écrites sous la forme compacte sui vante :

$$G_i\phi \equiv \| G_i, 0 \| \phi_{j-1} - \| -G_i, 0 \| \phi_j, \tag{3.29}$$

$$G_{i+1}\phi \equiv \| G_{i+1}, 0 \| \phi_j - \| -G_{i+1}, 0 \| \phi_{j+1}. \tag{3.30}$$

On utilise cette notation pour discrétiser les équations de conservation. Ces équations sont d'abord linéarisées et intégrées sur le volume de contrôle. Elles sont finalement présentées sous la forme d'un système d'équations linéaires général suivant :

$$a_i\phi_i = \sum_j a_i\phi_j + S. \tag{3.31}$$

La variable ϕ_i au point (i) est représentée en fonction des valeurs ϕ_j situées au point avoisinant (j). Les coefficients sont évalués aux interfaces des volumes de contrôle.

L'algorithme SIMPLER est utilisé pour résoudre les équations de conservation discrétisées. Pour ce cas particulier (modélisation des sous-canaux), l'algorithme utilisé se résume comme suit :

a) On débute avec un flux de masse estimé, c'est-à-dire, un flux de masse égal à celui à l'entrée du canal et un flux massique nul partout à travers l'interconnexion.

b) À partir de l'équation de conservation de la quantité de mouvement axiale, on calcule les pressions.

c) On utilise ces pressions pour corriger le flux de masse axial.

d) On utilise l'équation de conservation de la quantité de mouvement, les pressions évaluées à l'étape (b) et les flux de masse corrigés à l'étape (c) pour calculer les flux massiques latéraux.

e) On utilise l'erreur sur le bilan de masse pour corriger les flux de masse axiaux et latéraux sans corriger les pressions.

f) On vérifie si le critère de convergence sur la masse est satisfait, sinon on recommence à partir de l'étape (b).

Dans le cas du présent travail, on simule seulement des écoulements en deux phases eau-air en régime permanent; alors, seuls les termes des équations qui décrivent l'écoulement en régime permanent seront exposés en détail.

3.4.1 Équation discrétisée de conservation de la masse

Équation de conservation de la masse pour le gaz

Puisque dans ce travail on s'intéresse à un écoulement air-eau, le terme de génération Γ_g dans l'équation 3.1 est mis à zéro. Pour un écoulement en régime permanent, la discrétisation de l'équation 3.1 prend la forme suivante :

$$A_j G_{g_j} - A_{j-1} G_{g_{j-1}} = -c\Delta z G'_{g_i}, \tag{3.32}$$

où :

$A =$ section de passage du sous-canal $[m^2]$,

$\Delta z =$ incrément axial pour l'équation de masse $(z_j - z_{j-1})$ $[m]$,

$\Omega =$ surface de l'interconnexion (=c Δz) $[m^2]$.

Le flux massique à travers l'interconnexion (G'_{g_i}) est calculé pour chaque point (i) le long de l'axe en utilisant le modèle de mélange donné par l'équation 3.24.

Équation discrétisée de conservation de la masse du mélange

Pour un écoulement en régime permanent l'équation 3.3 s'écrit comme suit :

$$A_j G_j - A_{j-1} G_{j-1} = -\Omega G'_i. \tag{3.33}$$

En combinant 3.32 et l'équation 3.33, on obtient :

$$G'_{l_j} = G'_j - G'_{g_j}. \tag{3.34}$$

3.4.2 Équation discrétisée de conservation de la quantité de mouvement axiale

La discrétisation de l'équation 3.6 pour un régime permanent prend la forme suivante :

$$A\left[\left(\sum_{k=g,l}\frac{\hat{x}_k G_k}{\alpha_k \rho_k}\right)_{i+1} G_j - \left(\sum_{k=g,l}\frac{\hat{x}_k G_k}{\alpha_k \rho_k}\right)_i G_{j-1}\right] + cF_m(j)\Delta z_{cv}\left(\sum_{k=g,l}\left(\frac{\hat{x}_k}{\alpha_k \rho_k}\right)^{\star} G'_k\right)_j G_j$$

$$c\lambda\Delta z_{cv}\sum_{k=g,l}\hat{x}_k\left\{\left(\frac{G_k}{\alpha_k\rho_k}\right)_r - \left(\frac{G_k}{\alpha_k\rho_k}\right)_d\right\} G_j = A_j(P_i - P_{i+1}) - \left[f_0\Phi_l^2\frac{A}{2}\frac{\Delta z_{cv}}{\rho_l D_h}\right]_j G_j^2 -$$

$$\left[k_{form}\frac{A}{2}\frac{\Delta z_{cv}}{\rho' D_h}\right]_j G_j^2 + \beta\left[1-\left(\frac{c}{a}\right)^2\right]A_j\frac{1}{2}\Delta\frac{G'^2}{\rho_m} - \rho_m g A_j \Delta z_{cv}. \tag{3.35}$$

En introduisant les notations suivantes :

$$U = A\sum_{k=g,l}\left(\frac{\hat{x}_k}{\alpha_k\rho_k}\right)G_k, \tag{3.36}$$

$$V = \Omega_{cv}F_m(j)\sum_{k=g,l}\left(\frac{\hat{x}_k}{\alpha_k\rho_k}\right)^{\star}G'_k, \tag{3.37}$$

$$D = \Omega_{cv}\lambda\sum_{k=g,l}\hat{x}_k\left\{\left(\frac{G_k}{\alpha_k\rho_k}\right)_r - \left(\frac{G_k}{\alpha_k\rho_k}\right)_d\right\}, \tag{3.38}$$

$$S_o = f_o\phi_l^2\frac{G^2}{2\rho_l}A\frac{\Delta z_{cv}}{D_h}, \tag{3.39}$$

$$S_1 = K_{form}\frac{G^2}{2\rho'}A\frac{\Delta z_{cv}}{D_h}, \tag{3.40}$$

$$S_2 = \frac{1}{2}\beta\left\{1-\left(\frac{c}{a}\right)^2\right\}\Delta\frac{G'^2}{\rho_m}. \tag{3.41}$$

L'équation discrétisée aura la forme suivante :

$$U_{i+1}G_j - U_iG_{j-1} + [V+D]_jG_j =$$

$$A_j(P_i - P_{i+1}) - S_o - S_1 + S_2 + \rho_m g A_j \Delta z_{cv}. \tag{3.42}$$

Le terme source associé aux pertes de pression par frottement est linéarisé autour du point j. Étant donné que le terme source qui représente les pertes de pression irréversibles (S_j), ne nécessite pas de linéarisation, il n'y a pas d'obstruction dans notre cas. La linéarisation des termes sources est effectuée en utilisant un développement de Taylor de premier ordre, c'est-à-dire :

$$S_o = S_{oj} + \frac{\partial S_o}{\partial G}dG. \tag{3.43}$$

En utilisant les valeurs discrétisées $dG \approx G^n - G^o$ ou n représente la nouvelle valeur et o l'ancienne valeur de G, l'équation 3.39 devient :

$$S_o = -\left(\frac{1}{2}f_o\phi_l^2\frac{G^o}{\rho_e}A\frac{\Delta z_{cv}}{D_h}\right)G^o + \left(f_o\phi_l^2\frac{G^o}{\rho_e}A\frac{\Delta z_{cv}}{D_h}\right)G^n. \tag{3.44}$$

Cette expression est substituée dans l'équation 3.42. Le terme contenant la nouvelle valeur de G est gardé sous sa forme implicite. En utilisant les notations données par les expressions 3.28-3.30, l'équation de conservation de la quantité de mouvement axiale aura la forme suivante :

$$a_0 G_j - a_1 G_{j-1} - a_2 G_{j+1} = A_j(P_i - P_{i+1}) + b. \tag{3.45}$$

En définissant les nouvelles variables,

$$U^+ = \frac{1}{2}(U_{i+1} + U_i), \tag{3.46}$$

$$U^- = \frac{1}{2}(U_i + U_{i-1}), \tag{3.47}$$

alors les coefficients de l'équation 3.45 s'écrivent comme suit :

$$a_0 = \| -U^+, 0 \| + \| U^-, 0 \| - (U^+ - U^-) + \| V, 0 \| + D + f_o \Phi_l^2 \frac{G^o}{\rho_l} A \frac{\Delta z_{cv}}{D_h}, \tag{3.48}$$

$$a_1 = \| U^-, 0 \|, \tag{3.49}$$

$$a_2 = \| U^+, 0 \|, \tag{3.50}$$

$$b = \frac{1}{2} f_o \Phi_l^2 \frac{| G^o | G^o}{\rho_l} A \frac{\Delta z_{cv}}{D_h} - \rho_m g A_j \Delta z_{cv} + \| -V, 0 \| G_{d_j} - S_1 + S_2. \tag{3.51}$$

En utilisant des conditions aux frontières appropriées, on écrit ces équations sous la forme matricielle suivante :

$$[AG][G] = [DG], \tag{3.52}$$

où $[AG]$ est une matrice tridiagonale contenant les coefficients a_0, a_1, a_2, et $[G]$ est le vecteur contenant les flux de masse axiaux. Le vecteur $[DG]$ est une fonction de la différence de pression qui est exercée sur la facette du volume de contrôle. En connaissant la distribution des pressions, la distribution axiale du débit peut être alors calculée à partir de l'équation 3.52.

3.4.3 Équation discrétisée de conservation de la quantité de mouvement latérale

La discrétisation de l'équation de conservation de la quantité de mouvement latérale s'écrit sous la forme suivante :

$$\left\{A^{'}\sum_{k=g,l}\left(\frac{\hat{x}_k G_k}{\alpha_k \rho_k}\right)^{\star}\right\}_{j}^{+} G^{'} - \left\{A^{'}\sum_{k=g,l}\left(\frac{\hat{x}_k}{\alpha_k \rho_k}\right)^{\star}\right\}_{j-1}^{-} G^{'} + \left\{\Omega G \sum_{k=g,l}\left(\frac{\hat{x}_k}{\alpha_k \rho_k^{\star}}\right) G_k^{'}\right\}$$

$$= \Omega(P_d - P_r)_i - K_{tr}\frac{G^{'2}}{2\rho_m^{\star}}\ell\frac{\Omega}{c}, \tag{3.53}$$

avec:

$$A^{'} = c\ell,$$

$$\Omega = c(x_j - x_{j-1}). \tag{3.54}$$

En définisant :

$$U^{'} = A^{'}\sum_{k=g,l}\left(\frac{G_k}{\alpha_k \rho_k}\right)^{\star}, \tag{3.55}$$

$$V^{'} = \Omega\sum_{k=g,l}\left(\frac{1}{\alpha_k \rho_k}\right)^{\star} G_k^{'}, \tag{3.56}$$

$$S_o = K_{tr}\frac{G^{'2}}{2\rho_m^{\star}}, \tag{3.57}$$

en linéarisant le terme source S_o autour du point i et en utilisant l'équation 3.43, on obtient :

$$S_o = -\left[\frac{1}{2}K_{tr}\frac{\mid G^{'o} \mid}{\rho_m^{\star}}\frac{\ell\Omega}{c}\right]G^{'o} + \left[\frac{1}{2}K_{tr}\frac{G^{'o}}{\rho_m^{\star}}\frac{\ell\Omega}{c}\right]G^{'n}, \tag{3.58}$$

on introduit maintenant 3.58 dans 3.53 et on garde $G^{'n}$ sous la forme implicite tout en utilisant les notations 3.28-3.30, l'équation 3.53 devient :

$$a_0^{'} - a_1^{'}G_{i-1}^{'} - a_2^{'}G_{i+1}^{'} = \Omega(P_d - P_r)_i + b^{'}, \tag{3.59}$$

avec :

$$a_0^{'} = \| U^{'+}, 0 \| + \| -U^{'-}, 0 \| + V^{'} + K_{tr} \frac{G^{'o}}{\rho_m^{\star}} \frac{l^{\star}}{c} \Omega, \tag{3.60}$$

$$a_1^{'} = \| U^{'-}, 0 \|, \tag{3.61}$$

$$a_2^{'} = \| -U^{'+}, 0 \|, \tag{3.62}$$

$$b^{'} = \left[\frac{1}{2} K_{tr} \frac{| G^{'o} |}{\rho_m^{\star}} \frac{\ell}{c} \right] G^{'o}. \tag{3.63}$$

L'équation 3.59 représente un système linéaire tridiagonal qui permet de calculer les flux de masse latéraux à l'interface en fonction de la différence de pression latérale entre les sous-canaux. Pour des raisons de stabilité numérique, ce système est résolu en utilisant une méthode dite "point Jacobi" au lieu d'utiliser une solution simultanée comme c'est le cas pour l'écoulement axial.

3.4.4 Équation de pression axiale

L'algorithme "SIMPLER" nécessite la connaissance de la distribution de la pression pour permettre le calcul de la distribution du flux massique de l'écoulement. Cependant, il est impératif de trouver une équation convenable pour les calculs des pressions. Cette équation est obtenue en combinant l'équation de conservation de masse pour le mélange (éq. 3.33) et les équations de conservation de la quantité de mouvement axiale et latérale (éq. 3.45 et éq. 3.59).

À partir des équations de conservation de la quantité de mouvement axiale et latérale on peut écrire :

$$G_j = \frac{a_1 G_{j-1} + a_2 G_{j+1} + b}{a_{0_j}} + \frac{A_j}{a_{0_j}} (P_i - P_{i+1}), \tag{3.64}$$

$$G_i = \frac{a_1^{'} G_{i-1}^{'} + a_{2'} G_{i+1}^{'} + b^{'}}{a_{0_i}^{'}} + \frac{\Omega}{a_{0_i}^{'}} (P_d - P_r)_i. \tag{3.65}$$

Si on introduit ces deux équations dans l'équation de conservation de masse (3.33) du mélange, on obtient :

$$(a_{P0})P_i - (a_{P1})P_{i-1} - (a_{P2})P_{i+1}$$
$$= A_{j-1}\hat{G}_{j-1} - A_j\hat{G}_{j-1} - \Omega\hat{G}' + \Omega d_i P_{ri}, \tag{3.66}$$

les expressions des coefficients sont :

$$a_{P_1} = A_{j-1}d_{j-1}, \tag{3.67}$$

$$a_{P_2} = A_j d_j, \tag{3.68}$$

$$a_{P0} = (a_{P1} + a_{P2} + \Omega d_i'), \tag{3.69}$$

$$d_j = \left(\frac{A}{a_0}\right)_j, \tag{3.70}$$

$$d_i = \left(\frac{\Omega}{a_0}\right)_i, \tag{3.71}$$

$$\hat{G}_j = \frac{a_0 G_{j-1} + a_2 G_{j+1} + b}{a_0}, \tag{3.72}$$

$$\hat{G}_i = \frac{a_i G_{i-1}' + a_2 G_{i+1}' + b'}{a_0}. \tag{3.73}$$

Avec les pressions comme valeurs aux frontières, l'équation 3.66 peut être écrite sous la forme d'une matrice tridiagonale :

$$[AP][P] = [DP]. \tag{3.74}$$

Les éléments de la matrice $[AP]$ sont une fonction des flux de masse et de la géométrie des sous-canaux. Le vecteur source $[DP]$ contient les pseudo-flux de masse $\hat{G}_j$. La résolution de ce système donne les distributions des pressions de l'écoulement pour chaque sous-canal nécessaire pour exécuter la deuxième étape de l'algorithme SIMPLER.

3.4.5 Correction de flux de masse

Durant la première itération, la méthode numérique utilisée dans le présent modèle évalue la pression à partir d'un flux de masse égal aux conditions d'entrées. Par

ailleurs, à cause de la nature non-linéaire des équations de conservation de la quantité de mouvement, il faut s'attendre à ce que les pressions ne puissent pas satisfaire les vraies conditions d'écoulement. Alors, les distributions des débits évaluées à partir des équations 3.45 et 3.59 ne satisferont pas l'équation de conservation de masse. Par conséquent, il faut introduire une technique corrective pour l'écoulement. On élabore une équation qui suppose que la pression devrait aussi être corrigée. En supposant que :

$$G = G^o + G_c, \tag{3.75}$$

$$G^{'} = G^{'o} + G^{'}_c, \tag{3.76}$$

$$P = P^o + P_c. \tag{3.77}$$

où l'indice ^{o} indique la valeur non-corrigée et l'indice c la correction apportée à chaque quantité. En introduisant ces expressions dans l'équation de conservation de la quantité de mouvement axiale et latéral, on obtient respectivement :

$$\begin{aligned} &a_0(G^o_j + G_{c_j}) - a_1(G^o_{j-1} + G_{c_{j-1}}) - a_2(G^o_{j+1} + G_{c_{j+1}}) \\ &\qquad = A_j(P^o_i + P_{c_i} - P^o_{i+1} + P_{c_{i+1}}) + b, \end{aligned} \tag{3.78}$$

$$\begin{aligned} &a^{'}_0(G^{'o}_i + G^{'}_{c_i} - a^{'}_1(G^{'o}_{i-1} + G^{'}_{c_{i-1}}) - a^{'}_2(G^{'o}_{i+1} + G^{'}_{c_{i+1}}) \\ &\qquad = \Omega(P^o_{di} + P_{c_{di}} - P^o_{ri} - P_{c_{ri}}) + b^{'}. \end{aligned} \tag{3.79}$$

Si on soustrait les deux équations de conservation de la quantité de mouvement axiale et latéral avant et après correction, on obtient :

$$a_0 G_{c_j} - a_1 G_{c_{j-1}} - a_2 G_{c_{j+1}} = A_j(P_{c_i} - P_{c_i+1}) + b, \tag{3.80}$$

$$a^{'}_0 G^{'}_{c_i} - a^{'}_1 G^{'}_{c_{i-1}} - a^{'}_2 G_{c_{i+1}} = \Omega(P_{c_d} + P_{c_r})_i. \tag{3.81}$$

À partir de ces équations, on peut donc écrire :

$$G_{c_j} = \frac{a_1 G_{c_{j-1}} + a_2 G_{c_{j+1}} + b}{a_o} + \left(\frac{A}{a_o}\right)_j (P_{c_i} - P_{c_{i+1}}), \tag{3.82}$$

$$G'_{c_i} = \frac{a'_1 G'_{c_{i-1}} + a'_2 G'_{c_{i+1}} + b'}{a'_o} + \left(\frac{\Omega}{a'_o}\right)(P_{c_d} - P_{c_r}). \tag{3.83}$$

Si on néglige le premier terme à droite dans ces équations, on obtient :

$$G_{c_j} \cong d_j(P_{c_i} - P_{c_{i+1}}), \tag{3.84a}$$

$$G'_{c_i} \cong d'_i(P_{c_d} - P_{c_r})_i. \tag{3.85}$$

Où d_j et d'_i sont définies par 3.70 et 3.71 respectivement. À partir de ces deux équations, le terme de correction peut être exprimé en fonction de la pression de telle sorte que les flux de masse corrigés deviennent :

$$G_j = G^0_j + d_j(P_{c_i} - P_{c_{i+1}}), \tag{3.86a}$$

$$G'_i = G'^0_i + d'_i(P_{c_d} - P_{c_r})_i. \tag{3.87}$$

Ceci nous donne la procédure pour la correction du flux de masse. Pour utiliser cette procédure corrective pour l'écoulement, il est nécessaire de connaître la valeur de la quantité P_c. On utilise pour cela l'équation de conservation de masse dans laquelle on introduit les expressions du flux de masse corrigé. On obtient alors :

$$-A_{j-1}d_{j-1}P_{c_{i-1}} + (d_j A_j + d_{j-1}A_{j-1}P_{c_i} + d'_i\Omega)P_{c_i} - d_j A_j P_{c_{i+1}} =$$

$$A_{j-1}G^o_{j-1} - A_j G^o_j - \Omega G'^o_i + \Omega d'_i P_{c_{r_i}}. \tag{3.88}$$

Les conditions aux frontières des termes correctifs de la pression P_c sont utilisées pour écrire cette équation sous la forme d'un système matriciel tridiagonal, donné sous la forme suivante :

$$[AP][PC] = [DC]. \tag{3.89}$$

La matrice tridiagonale $[AP]$ possède les mêmes éléments que la matrice donnée par l'équation 3.74, seul le vecteur source est différent. Ce système est résolu simultanément pour chaque sous-canal. Les éléments du vecteur $[PC]$ sont utilisés pour corriger les flux de masse. Cependant, les pressions axiales pour chaque nouvelle itération sont calculées à partir du flux de masse corrigé précédemment,

évitant ainsi la correction de la pression à chaque itération, car la pression sera calculée à partir du flux de masse corrigé. Les éléments des systèmes matriciels 3.52, 3.74 et 3.89 sont calculés à partir des expressions developpées prècédement. La distribution de la pression axiale est résolue à partir du système lineaire 3.45. Étant donné la nature non-lineaire du problème, une attention bien spéciale doit être apportée pour obtenir une solution stable. Pour assurer la convergence, le calcul du flux de masse axial est soumis à une sous-relaxation, on introduit alors un facteur de sous-relaxation ϖ dans le système linéaire 3.80 :

$$a_0 G_j - \varpi a_1 G_{j-1} - \varpi a_2 G_{j+1} = (1-\varpi) G_j^o + \left(A_j + \frac{Ab_j}{2} \right) (P_i - P_{i+1}) + b. \quad (3.90)$$

L'indice ^{o} représente la valeur ancienne du flux de masse et Ab_j est la section de passage de l'obstruction (dans notre cas la section de passage est constante, car nous n'avons pas simulé des cas avec d'obstruction). Il est recommandé pour ϖ une valeur inférieure à 0.5 (Patankar, 1980).

Avant de terminer ce chapitre il est important de mentionner que l'on doit fournir au modèle différents paramètres d'entrée tel que les coefficients de diffusion D_α et D_{eq}, le facteur de frottement latéral K_{tr} et le coefficient de distribution C_0 (pour l'écoulement latéral). Très souvent des corrélations empiriques sont utilisées pour déterminer ces valeurs. Cependant ces corrélations peuvent être en dehors de leurs domaines d'applicabilité, ce qui peut entraîner des erreurs importantes dans les résultats de la modélisation. Tye (1991) a démontré qu'il n'existe pas une seule corrélation qui permet de simuler toutes les conditions d'écoulement auxquelles peut se trouver une grappe de combustible. Pour résoudre ce problème, nous avons couplé un "Algorithme Génétique" (AG) au code d'analyse de sous-canal pour déterminer les meilleures valeurs de D_{eq}, D_α, K_{tr}, C_0. Le chapitre suivant expose l'application des Algorithmes Génétiques pour l'évaluation optimale des coefficients D_{eq}, D_α, K_{tr} et C_0.

CHAPITRE 4

LES ALGORITHMES GÉNÉTIQUES

Comme il a déjà été mentionné, le but principal de ce projet est l'optimisation des différents paramètres de mélange d'un modèle de sous-canaux. Ainsi, nous avons utilisé un Algorithme Génétique (AG) couplé à une code de sous-canaux pour effectuer une telle optimisation. Ce chapitre a alors comme objectif d'expliquer le fonctionnement de la méthode numérique utilisée pour aboutir à ce projet, i.e., le couplage du code sous-canal avec un AG.

L'utilisation du modèle des sous-canaux, discuté dans le chapitre précèdent, nécessite des différentes valeurs d'entrée comme D_α, D_{eq}, C_0, K_{tr}. Ces valeurs sont généralement obtenues en utilisant des corrélations empiriques. Cependant, ces corrélations ont des contraintes ou des domaines finis; leur utilisation en dehors de leur domaine d'application peut produire des résultats complètement erronés.

Un algorithme génétique a été élaboré pour déterminer quels sont les meilleurs coefficients D_α, D_{eq}, C_0, K_{tr} qui optimisent les prédictions du code. Il y a deux raisons principales pour laquelle on a choisi ce type d'algorithme :

La première est que les algorithmes génétiques ne requièrent pas d'hypothèse particulière sur la régularité de la fonction objectif. L'algorithme génétique n'utilise pas notamment les dérivées de la fonction objectif pour la recherche de l'optimum; ce qui rend leur domaine d'application très vaste.

La deuxième est que le peu d'hypothèses requises permettent de traiter des problèmes très complexes. La fonction à optimiser *peut ainsi être le résultat d'une simulation.*

L'objectif de ce chapitre est de montrer succinctement, les caractéristiques d'un AG ainsi que les différentes techniques existantes pour élaborer cet algorithme. Toutefois, pour montrer le fonctionnement complet du code génétique, il faut avoir une

fonction objectif. Le chapitre suivant est dédié complètement à la détermination de cette fonction.

Une fois la fonction objectif élaborée, une analyse plus détaillée sur l'adaptation de l'AG et le code d'analyse de sous-canal sera expliquée.

Avant d'entrer plus en détail dans la description des AG, il est utile d'introduire le vocabulaire que nous emploierons.

4.1 Vocabulaire

Le chromosome.- Un AG classique traite des individus constitués d'une seule séquence de gènes, i.e., un chromosome, alors que les organismes biologiques peuvent en avoir plusieurs. L'approche informatique idéalise le chromosome en le réduisant à un vecteur de gènes. Du point de vue informatique, il faut mentionner qu'un *chromosome représente une solution potentielle du problème mathématique.*

Le gène.- Unité élémentaire qui caractérise la solution particulière (i.e., le chromosome) du problème que l'AG traite. Il faut noter que lorsqu'on parle d'organisme vivant, un gène peut être un complexe de séquences génétiques qui coopèrent pour assurer la solution d'une fonction donnée. Par exemple, un tel complexe (ou opéron) peut regrouper des séquences qui codent une protéine et d'autres qui en assurent la régulation.

Le locus.- Signifie lieu en latin. Le locus est la position du gène dans le chromosome. Ce terme est valable pour l'AG aussi bien que pour les systèmes biologiques.

L'allèle.- Du grec allos, autre. Alors que l'alphabet du code génétique naturel est composé de quatre lettres (Adénine, Thymine, Cytosine, Guanine), nous pouvons choisir n'importe quel alphabet pour un AG. Le système binaire est couramment utilisé.

La population.- Un groupe d'organismes naturels ou artificiels forment la population. Pour un AG, la population est un ensemble de chromosomes, c'est-à-dire, un

groupe de solutions possibles d'un problème mathématique.

4.2 L'exploration et l'exploitation

Un des problèmes le plus important de l'ingénierie est l'optimisation et la recherche des points optimaux. La recherche des points optimaux peut s'effectuer en utilisant les stratégies suivantes :

1.- *La stratégie Aveugle* : Cette stratégie n'utilise pas l'information contenue dans le domaine du problème.

2.- *La stratégie Heuristique* : Cette stratégie utilise l'information additionnelle pour orienter la recherche vers la direction plus appropriée où peut se trouver la solution du problème.

Il y a deux aspects importants qui caractérisent toute stratégie de recherche de solutions optimales : l'exploitation de meilleures solutions et l'exploitaton de l'espace d'exploration (Michalewiz, 1996; Dipama, 2003). Les méthodes de recherche peuvent être classifiées selon les trois types de stratégies suivantes :

Recherche de l'extremum : Stratégie exploitant les meilleures solutions susceptibles d'améliorer le résultat, tout en ignorant l'évaluation de l'espace d'exploration.

Recherche aléatoire : Stratégie évaluant l'espace d'exploration tout en ignorant les régions les plus prometteuses.

Algorithme génétique : Stratégie robuste combinant les éléments d'une recherche dirigée et stochastique, et pouvant faire un équilibre entre l'exploitation et l'explo ration de l'espace de recherche.

Les AGs diffèrent des autres méthodes conventionnelles d'optimisation et procédures de recherche en plusieurs aspects, d'abord ils manipulent des solutions codées et non les solutions elles-mêmes, ils cherchent la solution optimale à partir d'une population de solutions et non à partir d'une seule solution. Les AGs utilisent l'information sur une fonction objectif (fonction d'adaptation) sans faire appel aux dérivés de la fonction. Finalement, ils utilisent des règles de transition prob-

abilistes plutôt que déterministes.

4.3 La description générale d'un algorithme génétique

La structure générale de fonctionnement d'un algorithme génétique décrite entre autres par Melanie (1996) et Dipama (2003) est la suivante :

1.- Génération aléatoire d'une population de chromosomes, i.e., génération d'un espace initial des solutions.

2.- Évaluation de l'adaptation (aptitude) de chaque chromosome.

3.- Création d'une nouvelle population en répétant les étapes suivantes jusqu'à ce que la nouvelle population soit complète :

   a) Sélection de deux chromosomes parents en fonction de leur adaptation.

   b) Avec une probabilité de croisement P_c, croiser les deux parents pour former deux nouveaux chromosomes (enfants). Si les deux parents ne sont pas croisés, les enfants sont une réplique exacte des parents.

   c) Avec un probabilité de mutation P_m, muter les nouveaux enfants.

   d) Placer les nouveaux enfants dans la population.

4.- Utilisation de la nouvelle population pour la prochaine exécution de l'algorithme.

5.- Si la condition d'arrêt est satisfaite, alors arrêter la recherche et retenir la meilleure solution. Dans le cas contraire, continuer à l'étape 2.

La figure 4.1 montre le schéma de procédé général d'un AG.

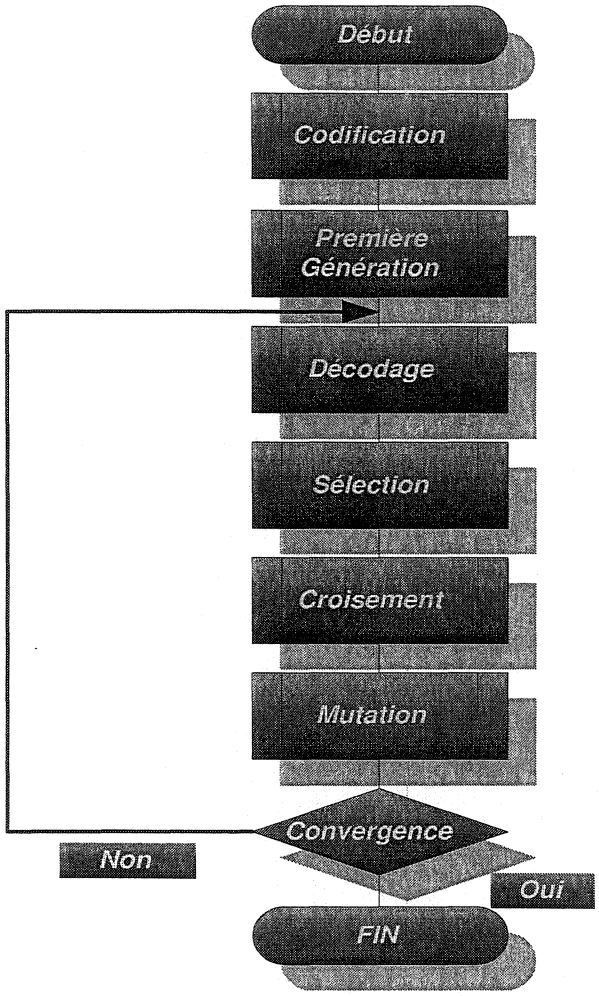


Figure 4.1: Le schéma de procédé d'un AG.

4.3.1 La représentation des paramètres ou codification

Avant de générer la première population, il faut connaître la longueur du chromosome, c'est-à-dire, savoir combien de symboles ou de nombres il doit contenir. Pour déterminer cette valeur, il faut connaître quelle est la précision demandée pour le résultat final du problème, le nombre de paramètres à chercher et l'espace d'exploration (domaine de la variable).

Supposons qu'on doit optimiser la fonction $f(x) \in [-1..2]$, le résultat doit avoir pour le moins six chiffres significatifs. La longueur du domaine de la variable x

est | $(-1-2)$ |=3, alors le domaine doit être divisé en 3×10^6 intervalles de même taille.

Écrit d'une façon mathématique cette condition devient :

$$2^{m-1} < 3 \times 10^6 \leq 2^m, \tag{4.1}$$

où m représente le nombre d'éléments que doit contenir le chromosome pour avoir la précision demandée. En résolvant cette inégalité, on obtient :

$$2097152 = 2^{21} < 3000000 < 2^{22} = 4194304. \tag{4.2}$$

Alors, le chromosome doit contenir au moins 22 allèle pour satisfaire la condition de la précision demandée.

Dans le cas où le problème contient plusieurs paramètres ou variables à optimiser, Gen et Cheng (1997) suggèrent :

$$2^{m_j-1} < (b_j - a_j) \times 10^n \leq 2^{m_j} \tag{4.3}$$

où :

$m =$ longueur du gène,

$10^n =$ précision demandée,

$[a_j, b_j] =$ domaine de la variable j,

$j =$ représente le nombre de la variable ou du paramètre à optimiser.

Ainsi la valeur décimale exprimé par le chromosome sera donnée par :

$$x = a_j + valeur \ \ décimal \ \ (chromosome)\frac{b_j - a_j}{2^{m_j}}. \tag{4.4}$$

Il est très important d'utiliser la taille adéquate du chromosome, car un chiffre ou un symbole de moins peut rendre l'algorithme incapable de trouver une bonne

solution. D'un autre part, des gènes de plus auront comme conséquence une perte de temps de calcul. Une fois la taille du chromosome connue, une première population peut être générée.

4.3.2 La génération aléatoire de la population initiale

La première population est un ensemble d'individus (solutions possibles codées en système binaire) avec laquelle on démarre l'algorithme. Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace d'exploration est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'exploration. En revanche, dans les cas où des informations *à priori* sur le problème sont disponibles, il semble être naturel de générer les individus dans un sous-domaine particulier afin d'accélérer la convergence.

Dans le cas où on connaît l'espace où se trouve la solution, c'est-à-dire, le domaine où se trouve le point optimal (par exemple le résultat ne doit pas être plus petit que zéro, le résultat doit être toujours positif). Gen et Cheng (1997) ont proposé de créer une première population où les résultats de l'évaluation soient à l'intérieur de ce domaine. Il est clair que s'il y a plus d'individus (solutions) qui tombent en dehors de cet espace, la vitesse de convergence diminuera ou dans le pire des cas, le code aboutira à une convergence prématurée. La taille de la population est un paramètre très important. En effet, une population avec un faible nombre d'individus peut accélérer l'algorithme vers une solution incorrecte. Ainsi, il y a un risque d'arriver à un minimum local, car l'espace d'exploration est limité par le nombre d'individus. Toutefois, une population très grande peut occasionner la perte de temps de calcul, puisqu'il faut évaluer plus d'individus (chromosomes, solutions). Donc, la taille du nombre d'individus de la population n'est pas une valeur prédéterminée; elle dépend du type de problème. En général la taille doit être suffisamment grande pour permettre d'examiner l'espace d'exploration et elle ne doit pas être trop grande pour ne pas perdre trop de temps de calcul.

4.3.3 L'évaluation

Cette procédure consiste à évaluer la qualité du chromosome en utilisant la fonction objectif à optimiser, c'est-à-dire, déterminer son niveau d'adaptabilité à l'environnement (aptitude). Cette évaluation est effectuée en utilisant une fonction objectif. Dans notre cas, la fonction objectif n'était pas connue. Une étude des différentes fonctions basées sur les lois de minimisation de l'énergie et de l'entropie a été effectuée. Le chapitre V est dédié à l'étude et à la sélection de la fonction objectif.

4.3.4 Principes de sélection

La théorie Darwinienne est basée sur les deux postulats suivants :

"Dans chaque environnement, seuls les espèces les mieux adaptées perdurent au cours du temps, les autres étant condamnées à disparaître".

"Au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce".

En se basant sur ces deux principes, nous pouvons facilement conclure que les chromosomes (solution codée en système binaire) qui ont une adaptation plus élevée seront sélectionnés pour continuer dans le processus d'évolution, les autres seront condamnés à disparaître.

Le processus de sélection aide à identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. Michalewicz (1996) mentionne que les deux facteurs les plus déterminants d'un algorithme génétique sont : la pression de sélection et la diversification de la population. La pression de sélection représente le niveau de qualification des chromosomes pour continuer dans la génération suivante, tandis que la diversification de la population représente son uniformité dans l'espace d'exploration.

Une pression stricte laisse passer seulement les meilleurs chromosomes, alors la probabilité d'avoir une population peu diversifiée augmentera; tandis que pour

une sélection moins stricte, les chromosomes qui passeront augmenteront en nombre et la population sera complètement non-uniforme. Une population avec une petite diversification peut produire une convergence vers un point optimal local; l'AG n'aura aucune convergence si la population est très diversifiée. Le plus grand problème des algorithmes génétiques est la convergence prématurée. Cette dernière est causée par une pression de sélection très élevée (donc, seuls les meilleurs chromosomes passeront) et une population peu diversifiée. La population serait rapidement complétée par les meilleurs chromosomes qui se reproduiront entre eux. Ils donnent ainsi aux autres moins de possibilités d'explorer des solutions au-delà de leur espace.

Pour faire la sélection, il faut se poser les questions suivantes :

1.- *Combien de chromosomes vont passer à la génération suivante ?*

2.- *Comment peut-on les sélectionner ?*

3.- *Comment peut-on déterminer la frontière entre les bons et les mauvais chromosomes ?*

Pour répondre à ces questions, nous avons dessiné un schéma (figure 4.2) montrant les différentes techniques et les méthodes trouvés dans la littérature (Michalewicz, 1996; Gen et Cheng, 1997 et Haupt R. et Haupt S., 1998).

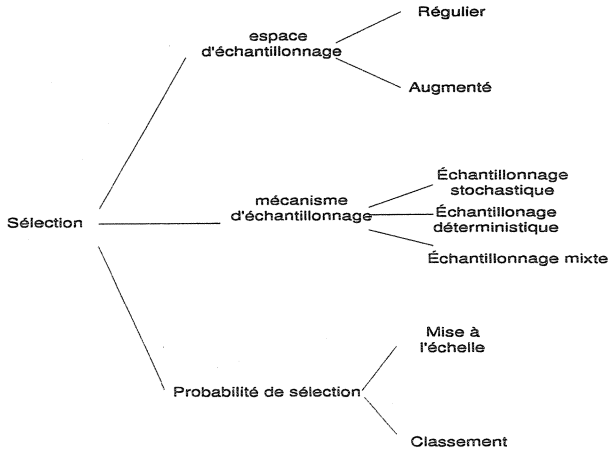


Figure 4.2: Les différents types de sélection.

L'espace d'échantillonnage répond à la première question. On peut définir l'espace d'échantillonnage comme le nombre des individus qui vont passer à la génération suivante. Lors de la sélection une nouvelle population sera formée par les parents et les nouveaux enfants, la taille de la population dépendra alors du nombre de parents et des enfants qui passeront a la nouvelle génération. Il y a trois possibilités : tous les parents et tous les enfants vont passer à la génération suivante, une partie des parents et une partie des enfants vont passer à la génération suivante, ou bien les enfants vont remplacer complètement leurs parents.

Dan le cas où la taille de la population reste constante pendant toute l'évolution du code, la stratégie s'appelle "population régulière", dans le cas où la population change selon l'évolution du code, la stratégie s'appelle "population augmentée".

Le mécanisme d'échantillonnage répond à la deuxième question. Ce mécanisme représente la façon de sélectionner les individus. Il y a trois options : la première est l'échantillonnage stochastique où les individus sont choisis aléatoirement, la deuxième est l'échantillonnage déterministe où les individus sont sélectionnés d'une façon préétablie, et enfin, la troisième est l'échantillonnage mixte qui est une combinaison entre les deux premières options.

Dans le type de sélection stochastique, la roulette biaisée est la plus connue, cette méthode consiste à élaborer une roulette divisée en n sections de différente taille (où n représente le nombre des chromosomes dans la population).

La taille de chaque section est calculée de manière que le meilleur chromosome aura un espace plus grand, tandis que le pire chromosome aura l'espace le plus petit. Pour déterminer la dimension de chaque espace, il est très courant d'utiliser la probabilité de chaque chromosome P_k, c'est-à-dire :

$$P_k = \frac{Eval(C_k)}{F}, \tag{4.5}$$

avec :

$$F = \sum_{k=1}^{n} Eval(C_k). \tag{4.6}$$

L'étape suivante calcule la probabilité cumulative et son assignation à chaque

chromosome de la population :

$$q_k = \sum_{j=1}^{k} P_k. \tag{4.7}$$

En suite, on peut élaborer la roulette avec différents espaces de taille q_k. Maintenant, il faut générer n nombres aléatoires r_k entre 0 et 1 et vérifier la condition suivante :

$$q_{k-1} < r < q_k. \tag{4.8}$$

Le chromosome qui satisfait cette condition sera sélectionné pour passer à la nouvelle population. Il est clair que le chromosome qui a l'espace le plus grand aura plus de probabilités de passer au niveau suivant.

La dernière question peut être répondue par la **probabilité de sélection**, elle détermine la frontière ou le seuil entre les bons chromosomes et les mauvais. Cette probabilité est un paramètre critique, car le processus de sélection est très sensible aux écarts de performance (aptitude). Dans certains cas, un très bon individu risque d'être reproduit trop souvent et il peut même provoquer l'élimination complète de ses congénères. On obtient alors une population homogène contenant un seul type d'individu. Dans ce cas, il y a un risque d'obtenir un seul représentant pour la génération suivante et seule l'opération de mutation pourra aider à atteindre l'objectif global au prix de nombreux essais successifs.

Pour éviter ce comportement, il existe d'autres modes de sélection (classement) ainsi que des principes (mise à l'échelle) qui empêchent les individus "forts" d'éliminer complètement les plus "faibles".

La méthode de classement (*Ranking*) permet de corriger le problème du super individu. Cette méthode consiste à modifier l'aptitude originale afin de réduire ou d'amplifier artificiellement les écarts entre les individus. La méthode du classement procède d'abord à un classement de chromosomes en fonction de leurs aptitudes. En suite chaque chromosome reçoit la valeur 1, le suivant la valeur 2 et ainsi de suite jusqu'au meilleur qui reçoit la valeur de l'aptitude de classement égale au nombre de chromosomes dans la population.

La sélection n'opère plus sur l'aptitude réelle du chromosome, sinon sur l'aptitude de classement, ce qui empêche les effets des super-individus. La méthode de mise à l'échelle consiste en transformer la valeur de l'aptitude (originale) f en $f^{'}$, de telle manière que l'aptitude des chromosomes les plus forts soit diminuée. Ce qui donne plus d'opportunités aux autres chromosomes d'être sélectionnés pour se reproduire. Cette transformation est réalisée en utilisant une fonction $g()$ appliquée sur f, c'est-à-dire : $f_k^{'} = g(f)$ où $f_k^{'}$ représente la nouvelle valeur de l'aptitude.

La fonction $g()$ peut avoir différentes formes; ceci produit un ensemble de différentes méthodes de mise à l'échelle, par exemple, mise à l'échelle linéaire, logarithmique, de puissance, etc.

Parmi les méthodes de mise à l'échelle, la normalisation et le fenêtrage ont été utilisés dans ce projet.

La fonction de normalisation pour un problème de maximisation, proposé par Gen et Cheng (1997) est :

$$f_k^{'} = \frac{f_k - f_{max} + \zeta}{f_{max} - f_{min} + \zeta}, \tag{4.9}$$

et pour un problème de minimisation :

$$f_k^{'} = \frac{f_{max} - f_k + \zeta}{f_{max} - f_{min} + \zeta}, \tag{4.10}$$

où :

f_{max} = aptitude du meilleur chromosome,

f_{min} = aptitude du pire chromosome,

ζ = constante pour éviter la division par zéro.

La méthode de fenêtrage est basée sur l'équation suivante :

$$f_k^{'} = f_k - f_{min_w}, \tag{4.11}$$

où w est la taille de la fenêtre, par exemple si $w = 2$ la taille de la fenêtre sera de 2 générations. Cheng et Gen (1997) mentionnent une valeur de w de 2 à 10.

4.3.5 L'opérateur de croisement

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Traditionnellement, les croisements sont envisagés avec deux parents qui génèrent deux enfants. Initialement, le croisement associé au codage par chaînes de bits est le croisement à découpage de chromosomes ("slicing crossover"). Pour effectuer ce type de croisement sur des chromosomes constitués de M gènes, on détermine aléatoirement une position de croisement dans chacun des parents. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants avec un mélange de l'information génétique de chaque parent.

En continuant avec les lois de la nature, les chromosomes les plus adaptés auront une possibilité plus élevée pour se reproduire et créer une nouvelle génération mieux adaptée. Le taux de reproduction (croisement) est nommé la probabilité de croisement Pc. Elle détermine le niveau de reproduction. Un niveau très bas de réproduction occasionnera une population peu diversifiée et une convergence prématurée. Le croisement se fait de la façon suivante :

Pour chaque chromosome, nous générerons un nombre aléatoire entre 0 et 1; si cette valeur est plus petite à Pc, ce chromosome est sélectionné pour se reproduire. Cette procédure se répète pour tous les chromosomes qui forment la population.

Le point de coupure se détermine aussi aléatoirement. Il indique à quelle position le chromosome sera divisé.

4.3.6 L'opérateur de mutation

Cette procédure consiste à changer l'allèle d'un ou de plusieurs gènes du chromosome. La mutation a différents buts :

a) sortir d'un minimum local,

b) remplacer les gènes perdus pendant la sélection pour être évalués dans un autre contexte,

c) générer des gènes qui n'ont pas été présents dans la première génération.

Si le rapport de mutation est trop bas, plusieurs chromosomes ne seront jamais évalués, mais si la mutation est trop élevée, les prochaines générations perdront leur ressemblance avec leurs parents et l'algorithme perdra son habileté d'apprendre à partir des générations passées.

Michalewicz (1996) identifie deux types de mutation; la mutation uniforme où la valeur de la probabilité de mutation reste constante pendant toute l'évolution du programme, et la mutation non-uniforme où la probabilité de mutation dépend du nombre de générations. Pour la mutation non-uniforme, le même auteur propose que la probabilité de mutation soit calculée selon :

$$P_m = (1 - r^{(1-t/T)^b}), \tag{4.12}$$

où

$r =$ est un nombre aléatoire entre [0 et 1],

$T =$ est le nombre maximal des générations,

$b =$ représente un paramètre pour déterminer le degré de l'uniformité,

$t =$ représente la génération actuelle.

Cette équation est décroissante, c'est-à-dire, au début de l'évolution du code la probabilité de mutation P_m sera plus grande ainsi l'espace d'exploration sera plus vaste. Lorsque le code continu à évoluer, la valeur de P_m diminuera et l'espace d'exploration sera plus local permettant ainsi de préciser la solution et d'accélérer la convergence.

L'utilisation d'une probabilité non-uniforme a montré être plus efficace par rapport à la mutation uniforme, en effet il est mieux de diminuer le taux de mutation à mesure que le code avance, de cette manière les individus ayant une adaptation meilleure resteront à exploiter un espace plus petit tout en raffinant la précision

de la solution.

4.3.7 La convergence

Il n'y a pas de preuve mathématique sur la convergence des algorithmes génétiques, ni aucune garantie de trouver un minimum global (Haupt et Haupt, 1998). Cependant, nous pouvons utiliser les possibilités suivantes :

1. Si l'algorithme génétique continue avec la même solution pendant X générations sans amélioration, il faut alors l'arrêter. Dans une telle situation l'algorithme a soit trouvé une bonne solution ou il est tombé dans un minimum local.

2. Si la déviation standard de la population atteint un niveau spécifique, il faut alors arrêter l'algorithme, car les valeurs de performance ne changent plus.

3. Si l'algorithme n'arrête pas pour une des raisons mentionnées antérieurement, alors il faut fixer un nombre maximal d'itérations.

Dans notre cas, nous avons fixé un nombre d'itérations et étant donné que la mutation est fonction du nombre de générations, la dernière population contiendra le même chromosome.

4.4 Exemple d'application d'un algorithme génétique

Supposons que nous devions trouver le maximum absolue de la fonction $f(x, y) = 21.5 + xsin(4\pi x) + ysin(20\pi y)$ (Michalewicz 1996), représentée par la figure 4.3.

De façon mathématique on peut écrire :

$$Max \quad f(x, y) = 21.5 + xsin(4\pi x) + ysin(20\pi y) \tag{4.13}$$

avec les variables indépendantes x et y bornées entre :

$$-3.0 \leq x \leq 12.1 \; et \; 4.1 \leq y \leq 5.8.$$

Cette fonction est composée de plusieurs maximums locaux; pour une méthode

d'optimisation basée sur la dérivé de la fonction, sera très difficile, sinon impossible de trouver la solution optimale.

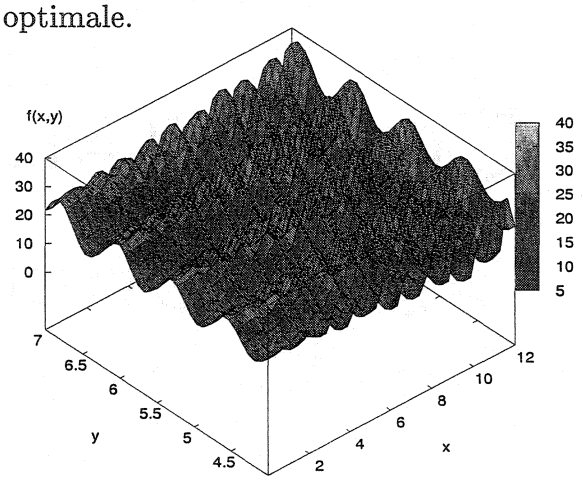


Figure 4.3: La représentation de la fonction objectif
$f(x, y) = 21.5 + xsin(4\pi x) + ysin(20\pi y)$.

Supposon que la précision demandée est de 10^{-5}, d'après l'équation 4.3 chaque intervalle doit être divisé au moins en $(b_j - a_j) \times 10^5$, c'est-à-dire :

$$2^{m_j - 1} < (b_j - a_j) \times 10^5 \leq 2^{m_j},$$

donc, pour la variable x le nombre de bits est donnée par :

$$12.1 - (-3.0) \times 10000 = 151000$$

$$2^{17} < 151000 < 2^{18} \implies m_1 = 18$$

et pour la variable y on a :

$$(5.8 - 4.1)10000 = 17000$$

$$2^{14} < 17000 < 2^{15} \implies m_2 = 15.$$

Ainsi la taille du chromosome sera donnée par $m_1 + m_2 = 18 + 15 = 33$. Le chromosome peut être représenté par 33 bits de la façon suivante :

$$\underbrace{000001010100101001}_{18}\ \underbrace{101111011111110}_{15}$$
$$\underbrace{\qquad\qquad\qquad}_{gène_1}\qquad\underbrace{\qquad\qquad\qquad}_{gène_2}$$

	Binaire	Décimal
$gène_1$	000001010100101001	5417
$gène_2$	101111011111110	24318

Les valeurs décimals des variables indépendantes sont déterminées en utilisant l'équation 4.4, c'est-à-dire :

$$x = -3.0 + 5417\frac{12.1 - (-3.0)}{2^{18} - 1} = -2.687969,$$

$$y = 4.1 + 24318\frac{5.8 - 4.1}{2^{15} - 1} = 5.361653.$$

Population initiale

Avec ces valeurs on pourra effectuer une première évaluation de la fonction. Cependant, de manière générale la première population est générée de façon complètement aléatoire.

Pour une population de 10 individus une telle génération aléatoire produit les chromosomes suivantes :

$$C_1 = [00000101010010100110111011111110]$$

$$C_2 = [00111010111001100000010101001000]$$

$$C_3 = [11100011100000100001010100100011 0]$$

$$C_4 = [10011011010010110100000010111001]$$

$$C_5 = [00001011110110001000111000110100 0]$$

$$C_6 = [11111010101101100000010110011001]$$

$$C_7 = [11010001001111100010011001110 1101]$$

$$C_8 = [00101101010000110001011001100 1100]$$

$$C_9 = [\,11111000101110110001110100011110 1\,]$$

$$C_{10} = [\,11110100111010101000001010110 1010\,]$$

Les valeurs décimales sont calculées en utilisant l'équation 4.4 :

$$C_1 = [x, y] = [-2.687969, 5.361653]$$

$$C_2 = [x, y] = [0.474101, 4.170144]$$

$$C_3 = [x, y] = [10.419457, 4.661461]$$

$$C_4 = [x, y] = [6.159951, 4.109598]$$

$$C_5 = [x, y] = [-2.301286, 4.477282]$$

$$C_6 = [x, y] = [11.788084, 4.174346]$$

$$C_7 = [x, y] = [9.342067, 5.121702]$$

$$C_8 = [x, y] = [-0.330256, 4.694977]$$

$$C_9 = [x, y] = [11.671267, 4.873501]$$

$$C_{10} = [x, y] = [11.446273, 4.171908]$$

L'évaluation du chromosome consiste à transformer la valeur du chromosome (en système binaire) vers son équivalente en valeur décimale, et par la suite de remplacer x et y dans la fonction objectif, i.e., l'équation donnée initialement (éq. 4.13).

Ainsi on obtient :

$$eval(C_1) = f[-2.687969, 5.361653] = 19.80519$$

$$eval(C_2) = f[0.474101, 4.170144] = 17.370896$$

$$eval(C_3) = f[10.419457, 4.661461] = 9.590546$$

$$eval(C_4) = f[6.159951, 4.109598] = 29.406122$$

$$eval(C_5) = f[-2.301286, 4.477282] = 15.686091$$

$$eval(C_6) = f[11.788084, 4.174346] = 11.900541$$

$$eval(C_7) = f[9.342067, 5.121702] = 17.958717$$

$$eval(C_8) = f[-0.330256, 4.694977] = 19.763190$$

$$eval(C_9) = f[11.671267, 4.873501] = 26.401669$$

$$eval(C_{10}) = f[11.446273, 4.171908] = 10.252480$$

Sélection

Dans ce cas, nous allons utiliser l'approche de la roulette biaisée. Pour commencer, il faut calculer la probabilité de sélection pour chaque chromosome, c'est-à-dire :

$$P_k = \frac{Eval(C_k)}{F}$$

avec

$$F = \sum_{k=1}^{taille\ \ pop} Eval(C_k)$$

la probabilité cumulative est donnée par :

$$q_k = \sum_{j=1}^{k} P_k$$

En utilisant les chromosomes C_1 - C_{10} antérieurs, nous avons comme résultat :

$P_1 = 0.111180$ $P_2 = 0.097515$ $P_3 = 0.053839$ $P_4 = 0.165077$ $P_5 = 0.088057$

$P_6 = 0.066806$ $P_7 = 0.100815$ $P_8 = 0.110945$ $P_9 = 0.148211$ $P_{10} = 0.057554$

$q_1 = 0.111180$ $q_2 = 0.208695$ $q_3 = 0.262534$ $q_4 = 0.421761$ $q_5 = 0.515668$

$q_6 = 0.582475$ $q_7 = 0.683290$ $q_8 = 0.794234$ $q_9 = 0.942446$ $q_{10} = 1.000000$

Pour sélectionner les chromosomes qui passeront à la génération suivante, nous devons assigner un numéro aléatoire r (entre 0 et 1) à chaque chromosome et les comparer aux probabilitées de survie de chacun.

Si le nombre r est plus petit que q_1, il faut donc sélectionner le chromosome C_1, sinon il faut sélectionner le chromosome C_k qui vérifie la condition :

$$q_{k-1} < r < q_k.$$

Supposons que les dix nombres aléatoires r_n soient :
0.301431 0.322062 0.766503 0.881893 0.350871 0.583392 0.177618
0.343242 0.032685 0.197577

Le premier nombre est 0.301431 qui est plus grand que q_3 et plus petit que q_4, par conséquent le chromosome C_4 est sélectionné pour passer vers la prochaine génération. Le deuxième nombre est le 0.322062 qui est plus grand que q_3 et plus petit que q_4, c'est-à-dire, que le chromosome C_4 est choisi de nouveau à passer à la nouvelle population. Nous devons répéter la même procédure jusq'à la fin des 10 nombres aléatoires, de cette manière on aura 10 chromosomes dans la nouvelle population. Ainsi la nouvelle population est donnée par :

$$C_1^{''} = [1001101101001011010000000010111001] \qquad (C_4)$$
$$C_2^{''} = [1001101101001011010000000010111001] \qquad (C_4)$$
$$C_3^{''} = [0010110101000011000101100110011100] \qquad (C_8)$$
$$C_4^{''} = [1111100010111011000111010001111101] \qquad (C_9)$$
$$C_5^{''} = [1001101101001011010000000010111001] \qquad (C_4)$$
$$C_6^{''} = [1101000100111110001001100111011101] \qquad (C_7)$$
$$C_7^{''} = [0011101011100110000000010101001000] \qquad (C_2)$$
$$C_8^{''} = [1001101101001011010000000010111001] \qquad (C_4)$$
$$C_9^{''} = [0000010101001010011011110111111110] \qquad (C_1)$$
$$C_{10}^{''} = [0011101011100110000000010101001000] \qquad (C_2)$$

où $C_n^{'}$ représente la nouvelle population.

Le croisement

Pour réaliser cette procédure, nous avons utilisé la méthode d'un simple point ("slicing crossover") avec une probabilité de 0.25, c'est-à-dire, en moyenne les 25 % de la population se croisera. Pour présenter la procédure de croisement, nous allons générer 10 nombres aléatoires qui seront assignés à chaque chromosome. Les chromosomes ayant une valeur inférieure à la probabilité de croisement P_c, se croiseront. Imaginons que les dix nombres aléatoires soient :

$P_{C_1} = 0.625721$ $P_{C_2} = 0.266823$ $P_{C_3} = 0.288644$ $P_{C_4} = 0.295114$
$P_{C_5} = 0.163274$ $P_{C_6} = 0.567461$ $P_{C_7} = 0.085940$ $P_{C_8} = 0.392865$
$P_{C_9} = 0.770714$ $P_{C_{10}} = 0.548656$

Il y a seulement deux nombres inférieurs à 0.25 qui se trouvent dans la position 5 et 7. Donc, les chromosome C_5' et C_7' seront sélectionnés pour se croiser.

$$C_5'' = [1001101101001011010000000010111001]$$
$$C_7'' = [0011101011100110000000010101001000]$$

Pour déterminer le point de coupure pour le croisement, il faut générer un autre nombre aléatoire entre 1 *et* $C_{m_1+m_2-1}$. Supposons que le point de coupure soit 7, les nouveaux chromosomes seront :

$$C_5''' = [\mathbf{1001101}\,0111001100000001010100 1000]$$
$$C_7''' = [0011101\,\mathbf{1010010110100000010111001}\,]$$

Mutation

Cette procédure altère un ou plusieurs allèles d'un chromosome. Pour déterminer l'allèle qui sera modifié, il faut assigner un nombre aléatoire à chaque allèle de la population. Si la valeur assignée est inférieure à la probabilité de mutation P_m, cet allèle passera de 0 à 1 ou vice-versa. Supposons que le gène 6 du chromosome 4 a un nombre inférieur à P_m, alors cet allèle sera muté comme le montre la séquence suivante :

$$C_4'' = [11111000101110110001110100011110 1]$$
$$C_4''' = [11111100101110110001110100011110 11]$$

Imaginons une probabilité de mutation P_m de 0.01 et qu'il y ait dix chromosomes dans la population et chacun est composé de 33 allèles, c'est-à-dire qu'il y a $33 \times 10 = 330$ allèles dans toute la population, nous atteindrons en moyenne entre 3 et 4 mutations pour chaque génération ($P_m \times m \times taille\ \ population$). Il faut

mentionner que chaque allèle a la même probabilité d'être modifié. Alors on a besoin de générer une séquence des nombres aléatoires $r_k(k = 1, ..., 330)$ entre 0 et 1 et les assigner à chaque allèle.

Supposons que les gènes suivants doivent être mutés:

Position	Chromosome	Gène	Nombre aléatoire
105	4	6	0.009857
164	5	32	0.003113
199	7	1	0.000946
329	10	32	0.001282

Après la mutation, la population finale devient :

$$C_1''' = [10011011010010110100000010111001]$$

$$C_2''' = [10011011010010110100000010111001]$$

$$C_3''' = [00101101010000110001011001100]$$

$$C_4''' = [11111100101110110001110100011101]$$

$$C_5''' = [10111010111001100000010101001010]$$

$$C_6''' = [11010001001111100010011001101101]$$

$$C_7''' = [10111011010010110100000010111001]$$

$$C_8''' = [10011011010010110100000010111001]$$

$$C_9''' = [00000101010010100110111011111110]$$

$$C_{10}''' = [00111010111001100000010101001010]$$

Les valeurs correspondantes en système décimal ainsi que l'évaluation de la fonction objectif sont :

$$C_1''' = [x_1, x_2] = [6.159951, 4.109598] \Rightarrow f(6.159951, 4.109598) = 29.406122$$
$$C_2''' = [x_1, x_2] = [6.159951, 4.109598] \Rightarrow f(6.159951, 4.109598) = 29.406122$$
$$C_3''' = [x_1, x_2] = [-0.330256, 4.694977] \Rightarrow f(-0.330256, 4.694977) = 19.763190$$
$$C_4''' = [x_1, x_2] = [11.907206, 4.873501] \Rightarrow f(11.907206, 4.873501) = 5.702781$$
$$C_5''' = [x_1, x_2] = [8.024130, 4.170248] \Rightarrow f(8.024130, 4.170248) = 19.91025$$
$$C_6''' = [x_1, x_2] = [9.342067, 5.121702] \Rightarrow f(9.342067, 5.121702) = 17.958717$$
$$C_7''' = [x_1, x_2] = [8.047458, 4.109598] \Rightarrow f(8.047458, 4.109598) = 25.373219$$
$$C_8''' = [x_1, x_2] = [6.159951, 4.109598] \Rightarrow f(6.159951, 4.109598) = 29.406122$$
$$C_9''' = [x_1, x_2] = [-2.687969, 5.361653] \Rightarrow f(-2.687969, 5.361653) = 19.805119$$
$$C_{10}''' = [x_1, x_2] = [0.474101, 4.170248] \Rightarrow f(0.474101, 4.170248) = 17.370896$$

Constatez que nous avons montré une seule génération de l'algorithme génétique, qui est la toute première. Le programme sera terminé après 1000 générations.

La solution trouvée après les 1000 générations est de $f(11.631507, 5.714824) = 37.678605$, ce résultat se compare très bien avec le résultat analytique qui est $Max\, f(11.631407, 5.724824) = 38.818208$.

Dans le chapitre suivante, nous présenterons toutes les caractéristiques ainsi que les paramètres et techniques utilisées pour élaborer l'algorithme génétique.

Comme on peut constater, le critère de convergence a été choisie comme le nombre maximal des itérations. Présentement, il existe de nouvelles procédures pour le perfectionnement de ce type de code, comme l'utilisation des nombres réels au lieu de binaires, mutations non-uniformes, etc. Il faut souligner que l'exemple donné avait seulement pour but d'éclaircir l'idée générale d'un AG.

L'utilisation de l'AG dans ce projet ne peut pas se réaliser sans une fonction objectif, le chapitre suivant est alors dédié à l'analyse de différentes fonctions objectif.

CHAPITRE 5

LA FONCTION OBJECTIF

Comme déjà mentionné dans le chapitre précédent, l'implantation de l'AG n'est pas possible sans l'évaluation de la fonction objectif car elle permet le lien entre le code d'analyse de sous-canal et l'algorithme génétique.

Comme première fonction objectif nous avons minimisé l'erreur quadratique moyenne (EQM). Une fois toutes les expériences analysées, une étude des différentes fonctions basées sur la minimisation de la dissipation de l'énergie et de la création de l'entropie a été effectuée.

Ce chapitre est consacré à l'analyse des fonctions étudiées complété par une description de leurs avantages et leurs inconvénients. L'implantation de la fonction objectif dans l'AG ainsi que le couplage avec le code d'analyse de sous-canal seront également expliqués.

5.1 L'erreur quadratique moyenne

La première fonction objectif utilisée a été l'EQM, cette fonction détermine la dispersion des valeurs prédites par rapport aux valeurs expérimentales. Pour construire cette fonction il faut déterminer les résidus, c'est-à-dire; la différence entre la valeur actuelle et la valeur prédite. L'EQM est calculé comme suit :

$$EQM = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y} - y)_i^2}{N - 1}}. \tag{5.1}$$

Il y a neuf variables à optimiser entre les deux sous-canaux interconnectés. Pour chaque sous-canal nous avons quatre variables physiques à optimiser : le débit du liquide, le débit du gaz, le taux de vide et la pression, ce qui donne huit variables au total; la neuvième variable est la différence de pressions entre les deux sous-canaux.

Étant donné que le but d'utiliser cette fonction est de faire une analyse de sen-

sibilité aux différentes variables physiques, une constante de pondération a été ajoutée à l'équation 5.1. Cette constante de pondération varie entre zéro et un.

$$EQM = \sum_{k=1}^{9} \omega_k \left(\sqrt{\frac{\sum_{i=1}^{N} (\hat{y} - y)_i^2}{N-1}} \right)_k , \tag{5.2}$$

où ω_k est une constante qui permet de choisir une ou plusieurs variables à étudier. En effet, en utilisant l'équation 5.2, nous avons analysé chaque variable d'une façon individuelle. Parmi toutes les analyses des variables, la différence de pression latérale entre les sous-canaux interconnectés a permis au code de converger dans une solution raisonnable.

5.2 La dissipation de l'énergie

Le deuxième principe de la thermodynamique implique que les processus pouvant survenir dans un système quelconque sont en général irréversibles. De la même manière que le premier principe de la thermodynamique, le deuxième principe exprime une limitation des processus permis par la nature. Le degré d'irréversibilité d'une transformation d'un système peut être caractérisé *par la dissipation de son énergie, que nous définissons pour plus de généralité comme l'augmentation de l'entropie totale.*

La fonction objectif basée sur la minimisation de la dissipation de l'énergie peut être développée à partir de l'équation de l'énergie mécanique. Pour montrer l'analyse de cette fonction, nous commencerons à partir du bilan généralisé pour un ou plusieurs fluides, donné par (Ishii, 1975) :

$$\frac{\partial \rho_k \psi_k}{\partial t} + \vec{\nabla} \cdot (\vec{v}_k \rho_k \psi_k) = -\vec{\nabla} \cdot \overline{\overline{\xi_k}} + \rho_k \phi \tag{5.3}$$

où:

$k =$ indice de la phase,

$\psi =$ une proprieté quelconque (vectorielle ou scalaire) du fluide,

$\overline{\overline{\xi_k}} =$ flux de propriété à la surface,

$\phi=$ terme source.

Le premier terme à gauche de cette équation est le taux de changement local de la propriété par unité de volume ψ_k, le deuxième terme représente la convection de la propriété ψ. Les termes à droite représentent le flux de propriété à la surface et la source par unité de volume, respectivement.

Cette équation généralisée peut être utilisée pour développer *l'équation de conservation de la masse*, avec :

$$\psi_k = 1, \tag{5.4}$$

$$\rho_k = 0, \tag{5.5}$$

$$\overline{\overline{\xi_k}} = \overline{\overline{0}}. \tag{5.6}$$

En utilisant l'équation 5.3, on obtient :

$$\frac{\partial \rho_k}{\partial t} + \vec{\nabla} \cdot (\rho_k \vec{v}_k) = 0. \tag{5.7}$$

L'équation de la conservation de la *quantité de mouvement* avec :

$$\psi_k = \vec{v}_k, \tag{5.8}$$

$$\overline{\overline{\xi_k}} = p_k \overline{\overline{I}} - \overline{\overline{\tau_k}}, \tag{5.9}$$

$$\phi_k = \vec{g}_k, \tag{5.10}$$

alors, on obtient :

$$\frac{\partial \rho_k \vec{v}_k}{\partial t} + \vec{\nabla} \cdot (\rho_k \vec{v}_k \vec{v}_k) = -\vec{\nabla} p_k \overline{\overline{I}} + \vec{\nabla} \cdot \overline{\overline{\tau_k}} + \rho_k \vec{g}_k. \tag{5.11}$$

En utilisant la transformation de la dérivée matérielle pour la vitesse, donnée par :

$$\frac{D\vec{v}}{Dt} = \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla})\vec{v}. \tag{5.12}$$

Si on applique cette transformation à l'équation de la continuité (équation 5.3) on a :

$$\frac{\partial \rho_k \psi_k}{\partial t} + \vec{\nabla} \cdot (\rho_k \vec{v}_k \psi_k) = \rho_k (\frac{\partial \psi_k}{\partial t} + \vec{v}_k \cdot \vec{\nabla} \psi_k) \equiv \rho_k \frac{D_k \psi_k}{Dt}. \tag{5.13}$$

Si la propriété ψ_k est égale à $\vec{v}_k$, l'équation précédente s'écrit :

$$\frac{\partial \rho_k \vec{v}_k}{\partial t} + \vec{\nabla} \cdot (\rho_k \vec{v}_k \otimes \vec{v}_k) = \rho_k (\frac{\partial \vec{v}_k}{\partial t} + \vec{v}_k \cdot \vec{\nabla} \vec{v}_k) \equiv \rho_k \frac{D\vec{v}_k}{Dt}. \tag{5.14}$$

Le premier terme de cette équation est égal au terme de l'équation 5.11 de la conservation de la quantité de mouvement, alors cette équation peut être écrite comme :

$$\rho_k \frac{D\vec{v}_k}{Dt} = -\vec{\nabla} p_k \overline{\overline{I}} + \vec{\nabla} \cdot \overline{\overline{\tau_k}} + \rho_k \vec{g}. \tag{5.15}$$

En multipliant scalairement avec la vitesse $\vec{v}_k$ cette équation, on obtient :

$$\frac{\partial}{\partial t}(\frac{\rho_k |\vec{v}_k|^2}{2}) + \vec{\nabla} \cdot (\rho_k \frac{|\vec{v}_k|^2}{2} \vec{v}_k) = -\vec{v}_k \cdot \vec{\nabla} p_k \overline{\overline{I}} + \vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}}) + \vec{v}_k \cdot \rho_k \vec{g}. \tag{5.16}$$

Cette équation représente le bilan de l'énergie mécanique du système. Le premier terme à gauche représente la variation locale de l'énergie cinétique, le deuxième terme est la variation spatiale de l'énergie cinétique, le premier terme à droite représente le travail effectué par le fluide contre la pression statique, le deuxième représente la force de frottement, enfin le dernier terme représente le travail de la force gravitationnelle.

Dans notre cas le terme $\vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}})$ est très important, car il représente la perte irréversible d'énergie mécanique occasionnée par la dissipation visqueuse. Ce terme peut être aussi représenté par une fonction de dissipation de la façon suivante :

$$\vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}}) = \Phi_\mu, \tag{5.17}$$

avec :

$\tau_k =$ contrainte de cisaillement $[kg/m^2]$,

$\Phi_\mu =$ fonction de la dissipation $[kg/m^2\, s^2]$.

En partant de l'observation suivante (Shirai et Ninokata 2001) :

"À mesure que l'écoulement dans de sous-canaux interconnectés s'approche à l'équilibre hydrodynamique, le système se redistribue pour minimiser la dissipation de l'énergie".

Même si les expériences utilisées dans cette recherche ont été effectuées en dehors de l'équilibre hydrodynamique, nous avons décidé de minimiser la dissipation de l'énergie car c'est la **tendance** de l'écoulement de se redistribuer de manière que l'énergie dissipée soit la plus petite.

5.3 La minimisation de la création d'entropie

La méthode de minimisation de la création d'entropie (ou optimisation thermodynamique, ou thermodynamique de temps fini) est une méthode de modélisation de procédés réels (irréversibles). Les modèles intègrent des principes de base en thermodynamique et en transfert thermique et l'optimisation est assujettie à des contraintes liées à des dimensions finies et de temps finis.

Ce développement est un aspect de l'optimisation de la thermodynamique qui permet à l'ingénieur de comprendre l'organisation du monde naturel. Dans le cadre de ces analyses, les ingénieurs peuvent déterminer l'entropie créée par le système comme fonction de sa structure, de ses paramètres physiques, incluant la taille, la forme et les matériaux. Après avoir compris -en principe- les ingénieurs essayent d'optimiser le système en termes de ses contraintes responsables de l'opération irréversible (Bejan, 1996; Bejan, 2003).

5.3.1 Création d'entropie d'un fluide

Comme déjà mentionné dans la section précédente, la dissipation de l'énergie est une augmentation de l'entropie, alors cette section a comme but de montrer la relation qu'il y a entre la dissipation de l'énergie et la création de l'entropie.

Le terme $\vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}})$ dans l'équation 5.16 représente l'augmentation de l'énergie interne du fluide par unité de volume causée par la dissipation visqueuse. À partir de la première loi de la thermodynamique, on sait que :

$$\delta Q + \delta W = dU, \tag{5.18}$$

et par la deuxième loi de la thermodynamique :

$$\delta Q = TdS. \tag{5.19}$$

Si le système ne génère ni reçoit de la puissance mécanique, le terme δW est égal à zéro, alors :

$$\delta Q = dU \tag{5.20}$$

c'est-à-dire que l'augmentation de l'énergie interne dU est occasionnée par une source de chaleur δQ. Si dans un écoulement adiabatique on considère que cette source provient de la dissipation visqueuse $\vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}})$, il résulte :

$$dU = \vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}}). \tag{5.21}$$

Alors en utilisant les équations 5.21, 5.22, 5.23 nous pouvons écrire :

$$\vec{v}_k \cdot (\vec{\nabla} \cdot \overline{\overline{\tau_k}}) = TdS. \tag{5.22}$$

Le premier terme à gauche représente la fonction de dissipation visqueuse Φ_μ, cette équation devient :

$$\Phi_\mu = TdS. \tag{5.23}$$

Cette équation montre la relation qui existe entre la dissipation visqueuse et la création d'entropie dans un écoulement adiabatique (i.e., $\vec{q}'' = \vec{0}$). L'inconvénient de cette équation est de déterminer quelle est la fonction de la dissipation visqueuse.

Selon Kolev (2002) l'utilisation de l'équation de l'entropie est la façon la plus simple de décrire l'écoulement. L'utilisation d'une autre équation plutôt que celle de l'entropie rend la description de l'écoulement plus compliquée.

Pour une analyse informatique, l'utilisation de cette équation plutôt que l'équation de l'énergie ou d'une de ses variantes est la façon la plus efficace pour la modélisation de mouvements de fluides.

5.4 La minimisation de la différence de pression latérale

Une des observations les plus importantes de l'analyse des sous-canaux est l'annulation de la différence de pression latérale entre les deux sous-canaux interconnectés. Dans toutes les expériences utilisées pour cette recherche, nous avons constaté que la différence de pression latérale s'approche rapidement vers une valeur proche de

zéro, à ce moment là l'écoulement latéral forcé est nul, et il reste seulement que deux mécanismes de mélange : le mélange turbulent et la dérive du vide vers une distribution à l'équilibre ("void drift"). Comme déjà mentionné antérieurement, la minimisation de l'erreur quadratique moyenne (EQM) appliquée seulement sur cette variable a permis de vérifier que la différence de pression latérale est un facteur très important pour la distribution de l'écoulement.

En utilisant cette information, nous avons essayé de minimiser la différence de pression latérale, le long de la région interconnectée. Cette opération a été effectuée en utilisant la relation suivante :

$$\sum_{k=1}^{N} \left(\sqrt{(p_1 - p_2)^2} \right)_k , \tag{5.24}$$

où

k= numéro de noeud,

p_1= pression du sous-canal 1 $[kPa]$,

p_2= pression du sous-canal 2 $[kPa]$.

Cette équation constitue la fonction objectif la plus facile à utiliser, mais elle est la plus incomplète, car elle ne contient aucun terme sur la différence de concentration du vide entre les sous-canaux. En plus, au début de l'interconnexion la différence de pression latérale est différente de zéro. Cette fonction objectif ne tient pas compte de ce fait, ainsi le résultat obtenu est physiquement irréel.

5.5 Sélection de la fonction objectif

La fonction objectif que nous avons sélectionnée est la dissipation de l'énergie, car cette fonction possède le plus grand nombre de termes qui peuvent avoir un effet sur la distribution de l'écoulement vers l'équilibre hydrodynamique.

La fonction de la dissipation de l'énergie tient compte des écoulements axiaux et latéraux où l'énergie transportée par l'écoulement latéral est beaucoup plus petite en comparaison avec l'énergie transportée par l'écoulement axial. Cette remarque

est très importante pour notre problème d'optimisation, car au moment d'évaluer la fonction objectif, il est possible que seulement les termes plus grands soient pris en compte; tandis que les autres auront un poids trop faible lors des calculs.

Par exemple supposons les deux cas suivants :

1) L'énergie cinétique transportée par l'écoulement axial est égale à 1000 [kJ/kg] et l'énergie cinétique transportée par l'écoulement latéral est de 10 [kJ/kg], l'énergie cinétique totale sera égale à 1010 [kJ/kg].

2) L'énergie cinétique transportée par l'écoulement axial est égale à 800 [kJ/kg] et l'énergie cinétique transportée par l'écoulement latéral est 20 fois plus grande que dans le premier cas, c'est-à-dire 200 [kJ/kg], l'énergie cinétique totale sera égale à 1000 [kJ/kg].

Entre ces deux cas, le deuxième a l'énergie cinétique totale la plus petite, néanmoins l'énergie transportée par l'écoulement latérale est la plus grande; ceci est un problème car la minimisation doit être effectuée sur les deux termes, sinon la solution ne sera pas physiquement réelle.

La première façon de résoudre ce problème consiste à ajouter une constante de pondération aux plus petits des termes, de cette manière tous les termes de la fonction objectif seront dans la même échelle. Cependant, la valeur de cette constante est inconnue et peux changer d'une expérience à une autre.

La deuxième façon consiste à additionner un autre terme à la fonction objectif. Ce terme doit diminuer à mesure que l'écoulement atteint l'équilibre hydrodynamique et surtout avoir le même ordre de grandeur que les autres termes.

Comme on l'a déjà mentionné, la minimisation de la pression latérale a donné des résultats raisonnables; cette fonction a la même tendance que la fonction basée sur la dissipation de l'énergie et a pour avantage d'avoir le même ordre de grandeur que les autres termes de la fonction objectif basée sur la dissipation de l'énergie.

La nouvelle fonction objectif à minimiser est composée de deux fonctions : la

première tient compte de la dissipation de l'énergie et la deuxième est une représentation de la tendance de l'écoulement à égaliser les pressions latérales (conséquence de la minimisation de l'énergie), c'est à dire :

$$min \;\; y = \vec{v}_k \cdot (\nabla \cdot \overline{\overline{\tau_k}}) + \chi \sum_{k=1}^{N} \left(\sqrt{(P_1 - P_2)^2} \right)_k , \tag{5.25}$$

ou bien, en utilisant l'équation 5.16, sans tenir compte du terme de la variation locale, on a :

$$min \;\; y = \nabla \cdot (\rho_k \frac{|\vec{v}_k|^2}{2} \vec{v}_k) + \vec{v}_k \cdot \vec{\nabla} p_k + \vec{v}_k \cdot \rho_k g + \chi \sum_{k=1}^{N} \left(\sqrt{(P_1 - P_2)^2} \right)_k , \tag{5.26}$$

où χ est une constante utilisée pour homogeniser les dimensions des deux fonctions.

5.6 Le couplage des deux codes : L'AG et le code d'analyse de sous-canal

Cette section a comme but de détailler la procédure utilisée pour le couplage entre l'algorithme génétique et le code d'analyse de sous-canaux.

Premièrement, on a défini les paramètres à optimiser, c'est-à-dire : le coefficient de distribution C_0, le coefficient de frottement latéral K_{tr} et les coefficients de diffusion pour le taux de vide D_{eq} et D_α.

La procédure de calcul utilisée est représentée sur la figure 5.1. L'algorithme est décrit comme suit :

1) On commence par spécifier les valeurs d'entrée. Du côté du code génétique, il faut spécifier : la taille de la population N, le nombre de générations, la probabilité de croisement P_c, la probabilité de mutation P_m, la taille du chromosome. Pour le code d'analyse de sous-canal il faut rentrer le nombre de noeuds, le critère de convergence, le numéro d'expérience, la valeur de λ, le facteur de quantité de mouvement F_m (dans notre cas $\lambda = 0$ et $F_m = 1$), la température, le nombre maximal d'itérations et le facteur de sous relaxation ϖ.

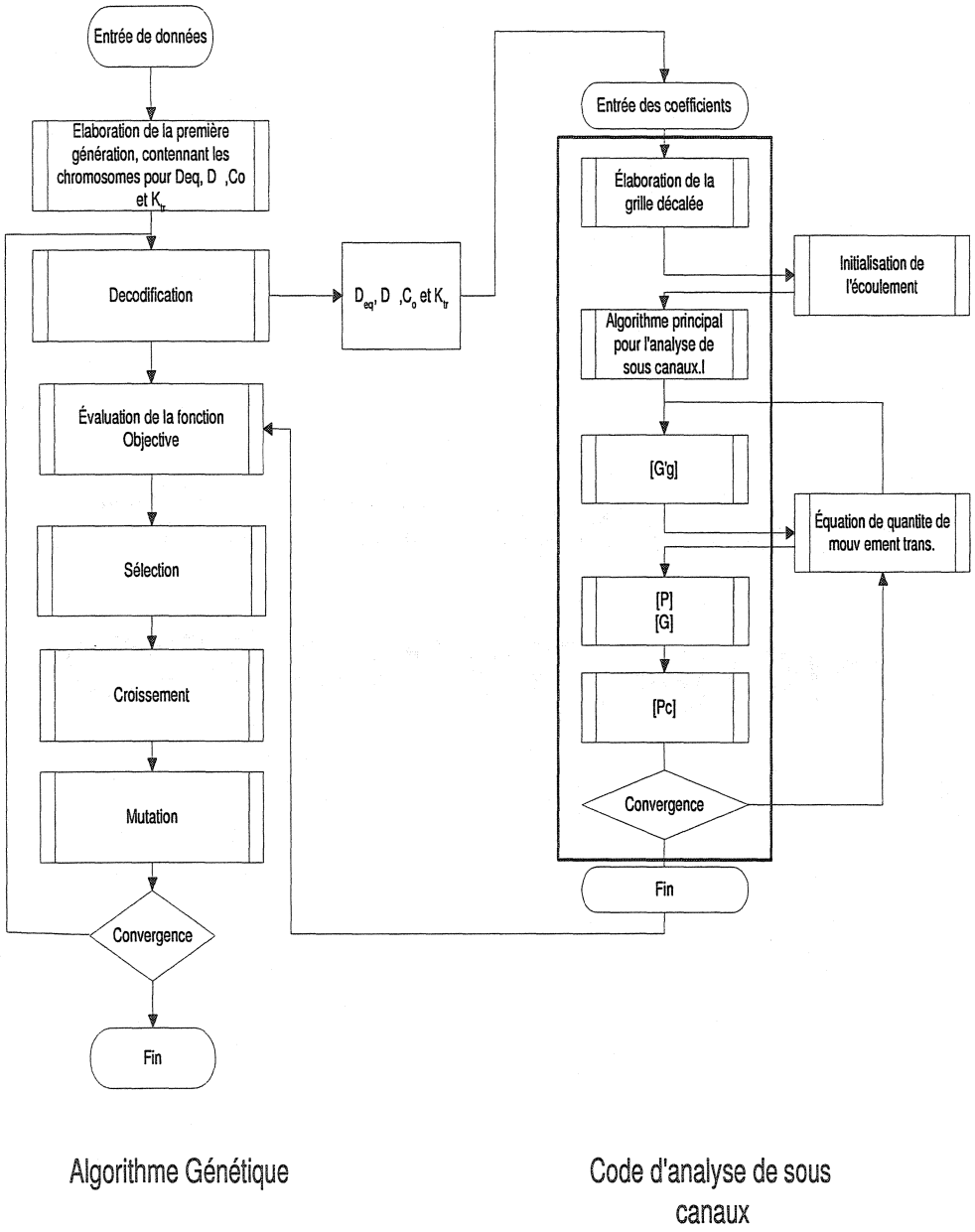


Figure 5.1: Schéma du couplage du code d'analyse de sous-canal avec l'AG.

Dans le code d'analyse de sous-canal, les valeurs K_{tr} et C_0 sont considérées constantes sur toute la longueur de l'interconnexion (c'est-à-dire : ils ne sont pas fonctions d'un autre paramètre) pendant que les valeurs de D_α et D_{eq} varient selon (Teyssedou, 1987) :

$$D_\alpha = 0.075\left(\frac{\tilde{\alpha}}{0.45}\right)^6 \ll J \gg_T \tilde{D}_h. \tag{5.27}$$

où:

$\ll J \gg_T =$ flux volumétrique transversal des deux canaux $[m/s]$,

$\tilde{\alpha}=$ moyenne arithmétique des taux de vide,

$\tilde{D}_h =$ moyenne arithmétique des diamètres hydrauliques des sous canaux $[m]$.

Pour optimiser D_α, cette équation a été réécrite de la façon suivante :

$$D_\alpha = a\left(\frac{\tilde{\alpha}}{0.45}\right)^b \ll J \gg_T \tilde{D}_h, \tag{5.28}$$

Étant donné que $\ll J \gg_T$ et $\tilde{\alpha}$ ne sont pas constantes le long des sous-canaux, la façon d'optimiser D_α a été de chercher les meilleures valeurs a, b qui nous permettrons de minimiser la fonction objectif.

Une procédure similaire a été utilisée pour trouver la valeur optimale de D_{eq}. De cette façon ce coefficient de diffusion a été exprimé par la relation suivante :

$$D_{eq} = c\left(\frac{\tilde{\alpha}}{0.45}\right)^d \ll J \gg_T \tilde{D}_h. \tag{5.29}$$

Ainsi l'AG cherche les valeurs de c et d qui permettent l'optimisation des calculs.

Donc le chromosome utilisé (montré dans la figure 5.2) est formé de six paramètres a, b, c, d, C_0, K_{tr}, chaque paramètre ou gène a 10 bits.

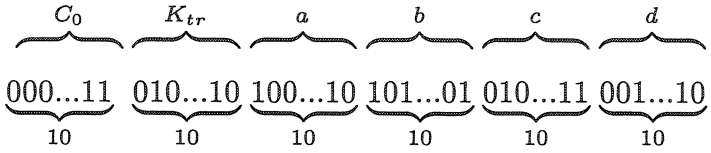


Figure 5.2: La représentation du chromosome.

2) Une fois la taille du chromosome est connue, il faut générer la première population, cette première population contiendra un nombre N de chromosomes ou individus lesquels représentent les coefficients à optimiser codés en système binaire.

   La figure 5.3 représente la première population; de cette population chaque ligne représente un chromosome. Étant donné que chaque chromosome est une possible solution codée dans un système binaire, il est évident que cette population des chromosomes ou individus est un ensemble de solutions possibles.

$$\begin{pmatrix} 000...11 & 010...10 & 100...10 & 101...01 & 010...11 & 001...10 \\ 101...00 & 000...00 & 110...01 & \dots & \dots & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \underbrace{000...11}_{K_{TR}} & \underbrace{010...10}_{C_0} & \underbrace{100...10}_{a} & \underbrace{101...01}_{b} & \underbrace{010...11}_{c} & \underbrace{001...10}_{d} \end{pmatrix}$$

Figure 5.3: La première population.

3) Chaque chromosome de la population est par la suite envoyé vers le code d'analyse de sous-canal, c'est-à-dire qu'on envoie les paramètres a, b, c, d, C_0, K_{tr} comme paramètres d'entrée du code d'analyse de sous-canal. Cependant, étant donné que les chromosomes sont codés en binaire, alors il faut d'abord les décoder en base décimal (voir figure 5.4).

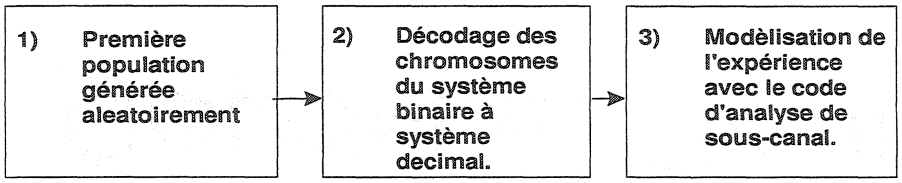


Figure 5.4: L'inter-relation entre l'AG et le code d'analyse de sous-canal.

4) Une fois la simulation finie, il faut évaluer la fonction objectif. Si la fonction objectif sélectionnée a été l'EQM, le résultat sera la minimisation de l'erreur entre les valeurs expérimentales et les valeurs calculées.

   Si la fonction objectif est basée sur la dissipation de l'énergie le résultat sera l'énergie totale dissipée par le système.

5) Lorsque toute la population est évaluée, il faut sélectionner les meilleurs chromosomes, c'est à dire, ceux qui ont donné un résultat de la simulation proche des valeurs expérimentales (minimisation de la fonction EQM) ou une dissipation de l'énergie minimale, si la f.o était la dissipation de l'énergie.

6) Pour réaliser la sélection on a utilise la méthode de la fenêtre mobile. Cette technique sélectionne les chromosomes ayant une valeur plus élevée ou inférieure à un seuil (dépendament si le problème est une minimisation ou une maximisation). Le seuil peut être la déviation standard ou simplement la moyenne de la population. dans notre cas, le seuil a été le moyen de la performance de la population.

   Une fois les chromosomes de la population sont sélectionnés, il faut les passer à la nouvelle génération, dans le cas où la nouvelle génération est incomplète il faudra la compléter avec des nouveaux chromosomes.

   Maintenant les étapes suivantes sont la source de diversification de la population, i.e., le croisement et la mutation. Dans ce cas nous avons utilisé des probabilités de croisement P_c entre 0.6 et 0.9, et pour la probabilité

de mutation P_m la fonction suivante a été choisie.

$$P_m = (1 - r^{(1-t/30)^5}), \tag{5.30}$$

où :

$r =$ nombre aléatoire entre [0 et 1],

$t =$ génération actuelle.

7) Si le nombre de générations est égal à la valeur d'entrée le code s'arrête, cas contraire la procédure continue à partir de l'étape 3 jusqu'au moment où le nombre de générations soit égal au nombre spécifié à l'entrée.

Les résultats obtenus en utilisant la fonction EQM et la minimisation de l'énergie dissipée sont montrés et analysés dans le chapitre suivant.

CHAPITRE 6

LES RÉSULTATS

Dans ce chapitre, on compare les prédictions du modèle présenté dans les chapitres précédents avec les données expérimentales obtenues pour des écoulements diphasiques verticaux sans obstruction (Tapucu et Gençay, 1980).

Comme discuté dans le chapitre V, nous avons élaboré un algorithme d'optimisation (AG) qui a été couplé à un code de sous-canal. Cet outil nous a permis d'analyser les données et de comprendre le comportement des différentes simulations en fonction des coefficients utilisés pour l'optimisation.

La première étape de ce projet était l'utilisation de la fonction de l'erreur quadratique moyenne comme fonction objectif. Cette partie a deux buts :

1) Vérifier le fonctionnement du code génétique. Une fois l'algorithme génétique fini, il a fallu le tester pour s'assurer le bon fonctionnement du code, analyser la probabilité de croisement P_c, la probabilité de mutation P_m, le nombre de générations et la taille de la population.

2) Faire une analyse de sensibilité des coefficients utilisés pour l'optimisation. L'analyse de sensibilité des coefficients a permis de connaître le comportement du système. Cette partie était la plus importante, car elle représente la base pour l'élaboration d'une fonction objectif indépendante des valeurs expérimentales.

L'étape suivante consiste en l'implantation de la fonction objectif basée sur la dissipation de l'énergie. Les détails pour cette fonction et son implantation ont été donnés dans le chapitre 5.

6.1 Conditions d'entrée

Pour permettre d'effectuer les calculs, certains paramètres particuliers doivent être

définis.

La discrétisation numérique est fixée à 90 nœuds, ce nombre est constant pour toutes les simulations effectuées. Un facteur de sous relaxation $\varpi = 0.5$ et un critère de convergence de 10^{-3} sont utilisées dans tous les cas étudiés.

Toutes les comparaisons ont été effectuées pour des écoulements caractérisés par un écoulement forcé important. Pour cette raison, nous avons décidé d'utiliser une valeur du coefficient de mélange turbulent λ égale à zéro. Ceci est justifié par le fait qu'on considère qu'il n'y pas de changement majeur dans les prédictions obtenues par le modèle quand les principaux mécanismes d'échanges impliqués sont : l'écoulement latéral forcé et la diffusion turbulente de vide. Les conditions d'entrée pour chaque expérience simulée sont résumées dans le tableau 6.1 :

Tableau 6.1 Valeurs d'entrée et initiales

Expérience	λ	F_m	α_i	α_j	G_i Liquide [kg/m^2 s]	G_j Liquide [kg/m^2 s]	G_i gaz [kg/m^{2}s]	G_j gaz [kg/m^2 s]	P_i Sortie [N/m^2]	P_j Sortie [N/m^2]
1	0	1	0.60	0.0	3175	3089	13.47	0	140086.63	140228.31
2	0	1	0.595	0.395	3125	3072	13.84	4.31	146970.19	146966.75
3	0	1	0.405	0	3127	3050	4.01	0	125724.19	125699.50
5	0	1	0.292	0	3139	3071	1.96	0	123961.88	123944.25
6	0	1	0.51	0	3125	3065	8.45	0	144285.38	144230.88
8	0	1	0.40	0.13	3119	3064	4.25	0.73	139829.25	139725.06

6.2 Analyse des résultats

Cette section est dédiée à l'analyse des résultats obtenus. Les figures 6.1(a,b,c,d,e) à 6.12(a,b,c,d,e) montrent la comparaison entre les prédictions obtenues en utilisant la valeur de l'erreur quadratique moyenne ainsi que la dissipation de l'énergie et les données expérimentales (Tapucu et Gençay, 1980).

6.2.1 La perte de pression axiale

Les simulations des pertes de pression sont montrées dans les figures 6.1(a) à 6.12(a). On peut observer que le code prédit très bien les tendances expérimentales, sauf dans la région avant l'interconnexion où les valeurs simulées sont en général surestimées. En effet, comme dans cette région, il n'y a pas d'interconnexion, le

facteur de frottement est une des principales variables qui peut affecter le comportement des pertes de pression.

6.2.2 La différence de pression latérale

La comparaison entre les prédictions de la différence de pression latérale et les données est présenté dans les figures 6.1(b) à 6.12(b). Étant donné que ces figures représentent la différence entre les pertes des pression axiales dans les sous-canaux, la différence de la pression latérale avant l'interconnexion est aussi surestimée.

Il faut remarquer que juste après l'interconnexion la différence de pression latérale entre les sous-canaux tombe très rapidement vers zéro; cette caractéristique est valide pour l'ensemble des expériences qui ont été simulées.

Comme on l'a déjà mentionné dans le Chapitre V l'écoulement tend vers l'équilibre hydrodynamique. Ceci correspond à la minimisation du gradient de pression ou à une annulation des différences de pressions latérales.

Cette remarque a permis d'établir la première fonction objectif indépendante des valeurs expérimentales, i.e., la minimisation des différences de pressions latérales. Les prédictions obtenues en utilisant cette fonction ont été proches des valeurs expérimentales. Cependant, cette fonction ne tient pas compte de plusieurs termes qui sont liés aux mécanismes de mélange (Ces résultats ne sont pas montrés dans les figures).

Gonzales-Santalo (1971) a démontré que le système d'équations qui gouvernent la distribution des écoulements dans les sous-canaux interconnectés a une infinité de solutions mathématiques possibles, mais seulement une solution physique. La difficulté consiste à trouver la bonne fonction objectif qui contient tous les termes nécessaires pour prédire le résultat physique réel; c'est-à-dire, les comportements observés expérimentalement.

La minimisation de la somme quadratique des différences de pressions latérales n'est pas suffisante comme fonction objectif pour représenter le comportement physique du système. Cependant elle a été considérée comme une fonction ad-

ditionnelle pour aider la recherche d'une solution réelle à travers l'AG (voir l'éq 5.25).

6.2.3 Le taux de vide

La comparaison entre les prédictions de la distribution du taux de vide et les données est présentée dans les figures 6.1(c) à 6.12(c). Les prédictions suivent les comportements expérimentaux sauf au proximité du début de l'interconnexion. Ces difficultés ont aussi été notées par d'autres chercheurs. Deux raisons possibles sont données par Teyssedou (1987) pour expliquer ce phénomène :

Premièrement, l'expansion de la phase gazeuse due au changement de la section de passage après l'interconnexion peut provoquer un tel écart. La deuxième raison est la perturbation de la couche limite de la phase liquide, occasionnée dans le début de l'interconnexion, qui peut donner des valeurs mesurées légèrement différentes à celles obtenues pendant la calibration du système de mesure de taux de vide. Il faut mentionner que le calibrage du système de mesure de taux de vide a été effectué en utilisant un seul sous-canal uniforme, i.e., sans changement de section.

6.2.4 Le débit du gaz

La comparaison entre les prédictions de la distribution du débit de gaz et les données est présentée dans les figures 6.1(d) à 6.12(d). On peut remarquer que les prédictions du code suivent très bien les tendances observées expérimentalement. Donc, le modèle de mélange utilisé est capable de prédire de façon remarquable les données expérimentales.

6.2.5 Le débit du liquide

La comparaison entre les prédictions de la distribution du liquide et les données est présentée dans les figures 6.1(e) à 6.12(e). Ces figures montrent que le code n'est pas capable de prédire le transfert de liquide dans la zone du début de l'interconnexion. Cependant, la simulation tend à se rapprocher aux valeurs expérimentales pour le reste du sous-canal. Une explication possible de cette difficulté est : l'analyse de sous-canal est à une seule dimension avec une équation con-

stitutive pour tenir compte des échanges de masse latéraux, tandis que la vrai nature du phénomène est 2D ou 3D. De plus la valeur de K_{tr} n'est pas nécessairement constante, elle doit dépendre de la vitesse latérale et de la géométrie du système. Dans notre cas la valeur de K_{tr} a été un paramètre à optimiser, cependant elle est restée constante le long de l'interconnexion pour chaque simulation. Une analyse plus détaillée devrait être réalisée pour déterminer son importance juste après l'interconnexion. En effet, l'échange de masse du liquide est plus élevé à ce point du sous-canal, ceci nécessite une valeur de K_{tr} plus faible.

La figure 6.2(e) ne montre pas la tendance vers l'équilibre hydrodynamique. En effet, les courbes commencent à s'approcher seulement vers la fin de l'interconnexion. Des raisons possibles de ce comportement sont : a) le code d'analyse de sous-canal n'a pas convergé de façon convenable, ou b) l'AG n'a pas trouvé le minimum global de la fonction objectif.

6.3 Analyse des paramètres optimisés

Cette section a comme but de montrer le comportement des coefficients de diffusion D_α et D_{eq} tel que déterminées par l'AG. Deux corrélations ont considérées pour comparer les résultats, la première est la corrélation de Shoukri *et al.,* (1984) et la deuxième est la corrélation utilisée par le code ASSERT (Carver *et al.,* 1987; Tye *et al.,* 2000), basée sur le nombre de Reynolds. La corrélation élaborée par Shoukri *et al.,* (1984) peut être discutable. En effet les auteurs ont supposé que dans un système de sous-canaux en position horizontal et sans différence de pression latérale entre eux, la diffusion turbulente du vide et la dérive causée par la gravité sont contrebalancées. Cette hypothèse n'est pas valide si le mécanisme de mélange "void drift" est présent pendant les expériences.

La corrélation obtenue par Shoukri *et al.,* (1984) à partir de ses expériences a la forme suivante :

$$\Omega = \frac{\epsilon}{\bar{U}\bar{D}_h}, \tag{6.1}$$

où $0.052 \leq \Omega \leq 0.074$, les auteurs ont proposé d'utiliser la valeur de Ω égale à 0.06.

La corrélation utilisée par ASSERT (Carver *et al.*, 1987; Tye *et al.*, 2000) est donnée par l'équation suivante :

$$\frac{\epsilon}{\bar{D}_h \bar{u}_m} = a\bar{R}e^b, \tag{6.2}$$

avec $a = 0.004$ et $b = 0.20$.

La valeur du nombre de Reynolds est calculée de la façon suivante :

$$\bar{R}e = \frac{\bar{D}_h \bar{G}}{\mu_m}. \tag{6.3}$$

où:

$\bar{D}_h$ = diamètre hydraulique moyen $[m]$,

$\bar{G}$ = flux massique moyen $[kg/m^2 s]$,

μ_m = viscosité de mélange $(1/\mu = x/\mu_g + (1-x)/\mu_l)$ $[kg/ms]$.

Il faut mentionner que la valeur de D_α n'est pas nécessairement un coefficient constant, en effet, l'équation 5.26 montre que D_α est dépendante du $\ll J \gg_T$ et $\tilde{\alpha}$.

La figure 6.13 présente les valeurs du coefficient D_α calculés par l'AG ainsi que ceux calculés, par les mêmes conditions, à partir des équations 6.1 et 6.2. Les valeurs présentées dans cette figure correspondent aux valeurs moyennes du coefficient calculé pour chaque expérience. Elle montre aussi qu'à mesure que le taux de vide moyen augmente le coefficient de diffusion augmente aussi, cette observation est valide pour les deux corrélations utilisées ainsi que pour les résultats prédits par le modèle basé sur la minimisation de la dissipation de l'énergie; il faut remarquer que l'écart entre ces courbes augmente à mesure que le taux de vide moyen augmente.

La figure 6.14 montre la variation du coefficient C_0 vs. la variation du taux de vide moyen. On peut remarquer que la valeur de C_0 reste toujours entre 1 et 1.1. Cette observation confirme l'hypothèse employée par plusieurs codes d'utiliser la valeur de C_0 égale à 1.0 pour l'écoulement latéral.

La figure 6.15 représente la variation du coefficient K_{tr} vs. la différence de pression latérale d'entrée des sous canaux. La plupart des valeurs sont entre 1 et 1.4, sauf pour un point (expérience 3) où la valeur est de 0.5.

Durant la première partie de ce projet, lors de l'utilisation de la fonction objectif basée sur l'EQM, les analyses ont montré que le coefficient K_{tr} n'a pas eu beaucoup d'influence sur les simulations, sauf pour l'expérience 3. Pour cette expérience le coefficient de frottement latéral était indispensable pour obtenir des résultats similaires à ceux des expériences. Ceci peut s'expliquer parce que les calculs sont plus sensibles au début de l'interconnexion où l'écoulement latéral est plus élevé.

Il faudra faire plus d'analyses pour déterminer la vrai dépendance avec les conditions de l'écoulement.

Nous avons comparé les prédictions du code en utilisant la corrélation de Shoukri *et al.,* (1984) avec celles obtenues avec l'AG.

Les figures 6.16 montrent les prédictions du code de sous-canal en utilisant le modèle basé sur la minimisation de la dissipation de l'énergie mécanique et la corrélation de Shoukri *et al.,* (1984). Pour montrer la différence de ces deux modèles on a prit l'expérience 2, or le taux de vide moyen est le plus élevé parmi toutes les expériences analysées. Il est clair que la corrélation proposée par Shoukri *et al.,* (1984) n'est pas capable de prédire adéquatement les données expérimentales. Kawahara *et al.,* (2000) ont montré que lors d'un changement d'écoulement du type à bulle à bouchon, le transfert de masse devienne très importante. La figure 6.13 montre que seulement le modèle basé sur la minimisation de l'énergie mécanique représente cette observation, tandis que la corrélation proposée par Shoukri *et al.,* (1984) et la corrélation utilisée par le code ASSERT-4 (Carver *et al.,* 1987) sont plutôt linéaires.

EXPERIENCE 1

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.60	0
Débit du liquide [kg/m² s]	3175	3089
Débit du gaz [kg/m² s]	13.47	0

Résultats

$C_0 = 1.038$

$K_{tr} = 1.010$

$D_\alpha = 0.01315\left(\frac{\alpha}{0.45}\right)^{4.575} \tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.05133\left(\frac{\alpha}{0.45}\right)^{6.938} \tilde{D}_h \ll J_T \gg$

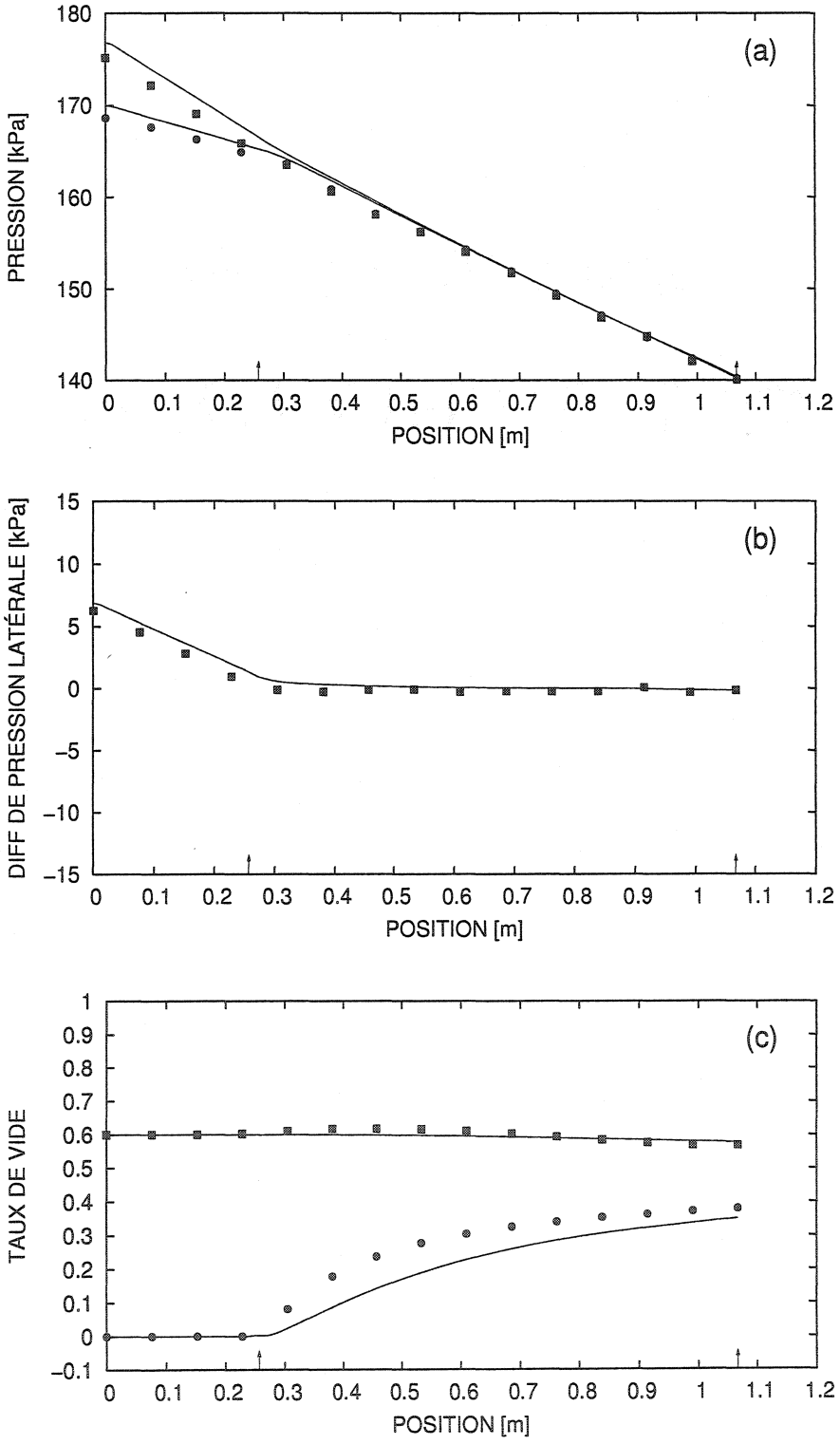


Figure 6.1(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : EQM.)

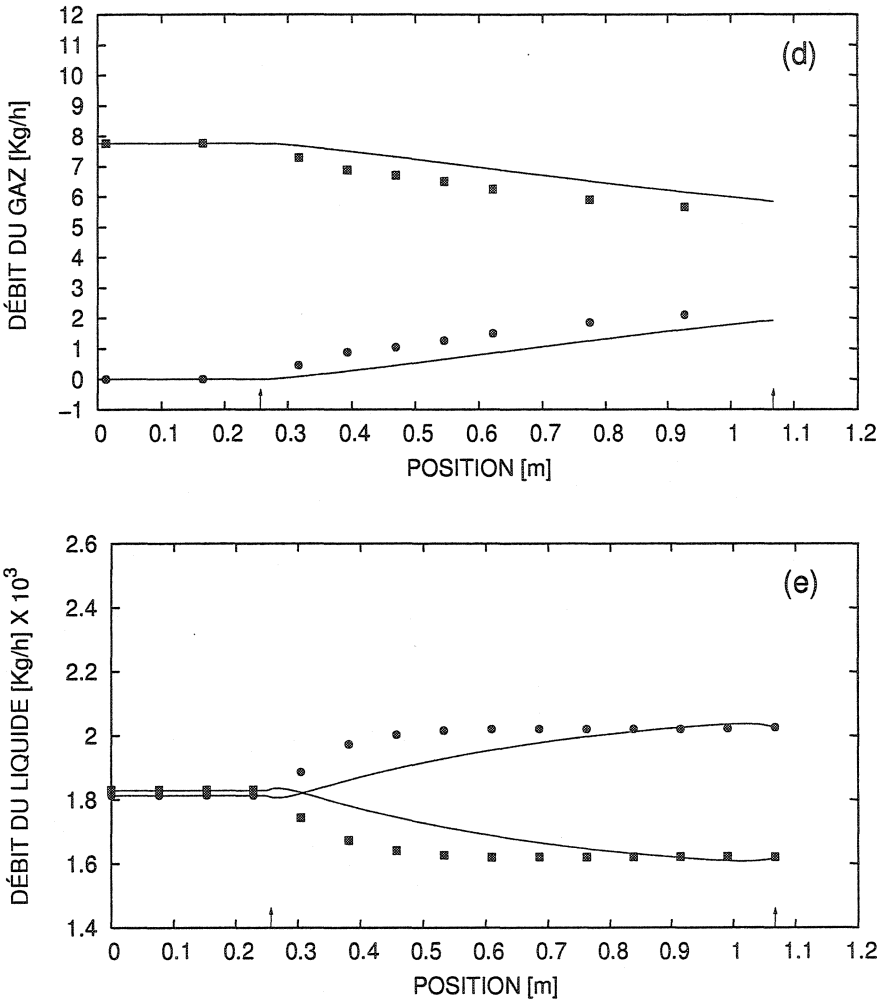


Figure 6.1(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 1

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.60	0
Débit du liquide [kg/m² s]	3175	3089
Débit du gaz [kg/m² s]	13.47	0

Résultats

$C_0 = 1.018$

$K_{tr} = 1.729$

$D_\alpha = 0.02742\left(\frac{\alpha}{0.45}\right)^{4.581}\tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.06103\left(\frac{\alpha}{0.45}\right)^{5.853}\tilde{D}_h \ll J_T \gg$

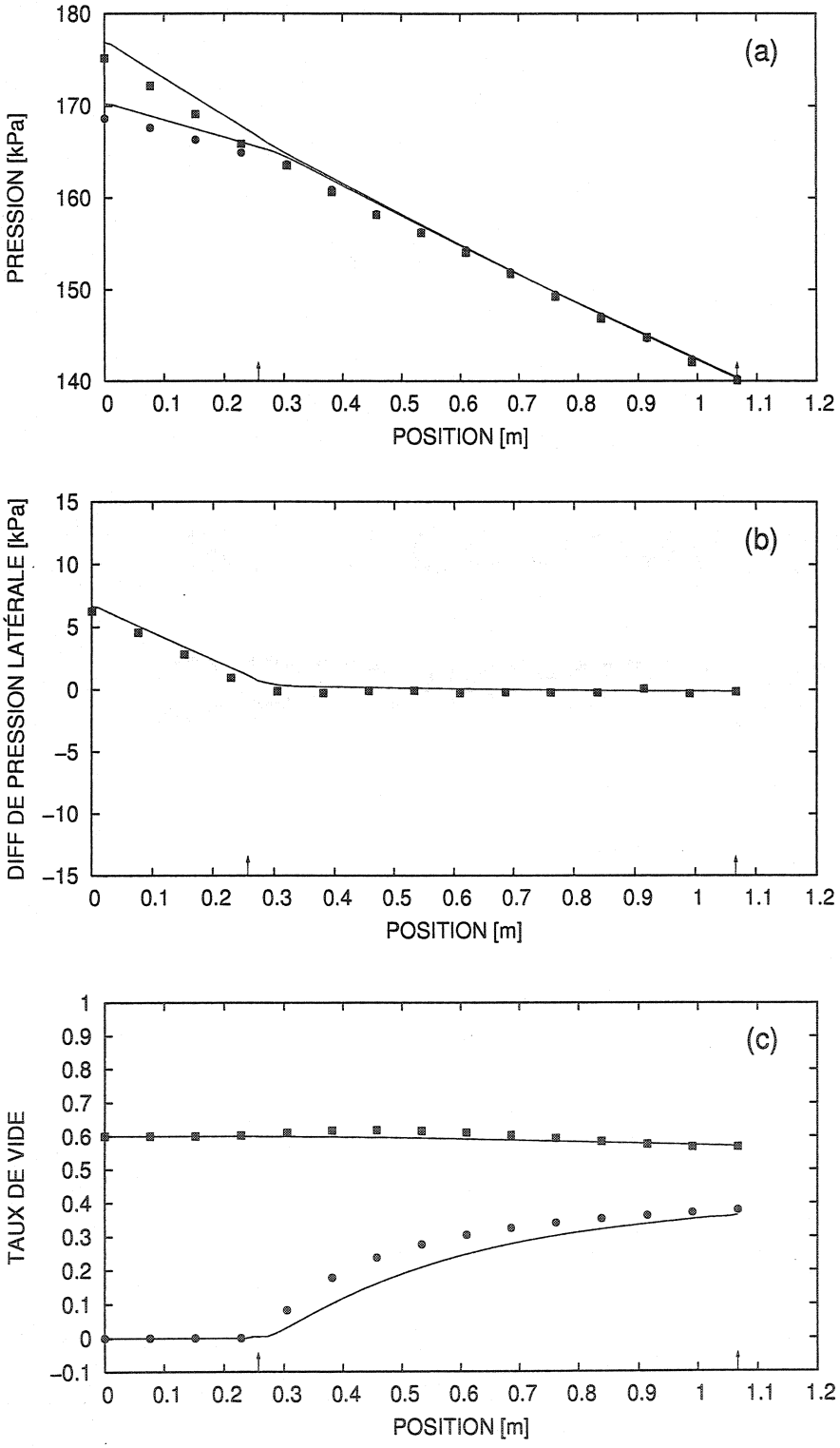


Figure 6.2(a)-(c): Comparaison des prédictions avec les données expérimentales. (fonction objectif : dissipation de l'énergie.)

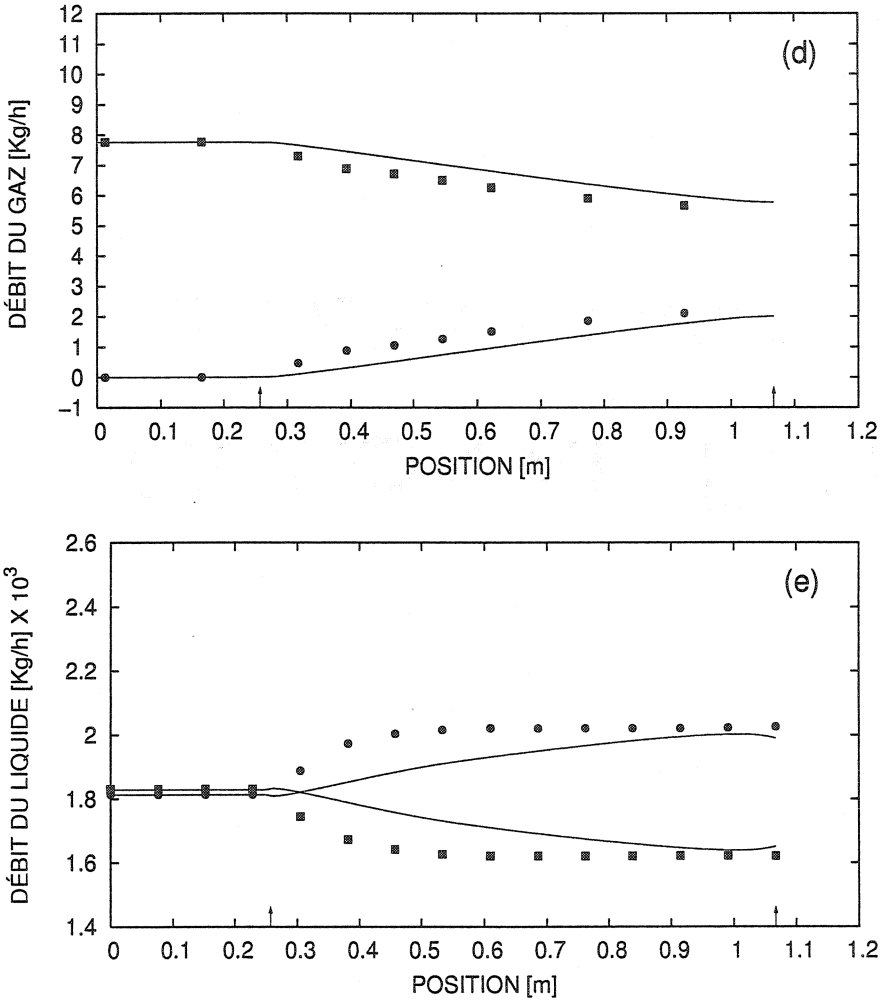


Figure 6.2(d)-(e) (Suite)
(fonction objectif : dissipation de l'énergie.)

EXPERIENCE 2

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.595	0.395
Débit du liquide [kg/m^2 s]	3125	3072
Débit du gaz [kg/m^2 s]	13.84	4.31

Résultats

$C_0 = 1.054$

$K_{tr} = 1.328$

$D_\alpha = 0.01097\left(\frac{\alpha}{0.45}\right)^{5.276}\tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.05267\left(\frac{\alpha}{0.45}\right)^{4.152}\tilde{D}_h \ll J_T \gg$

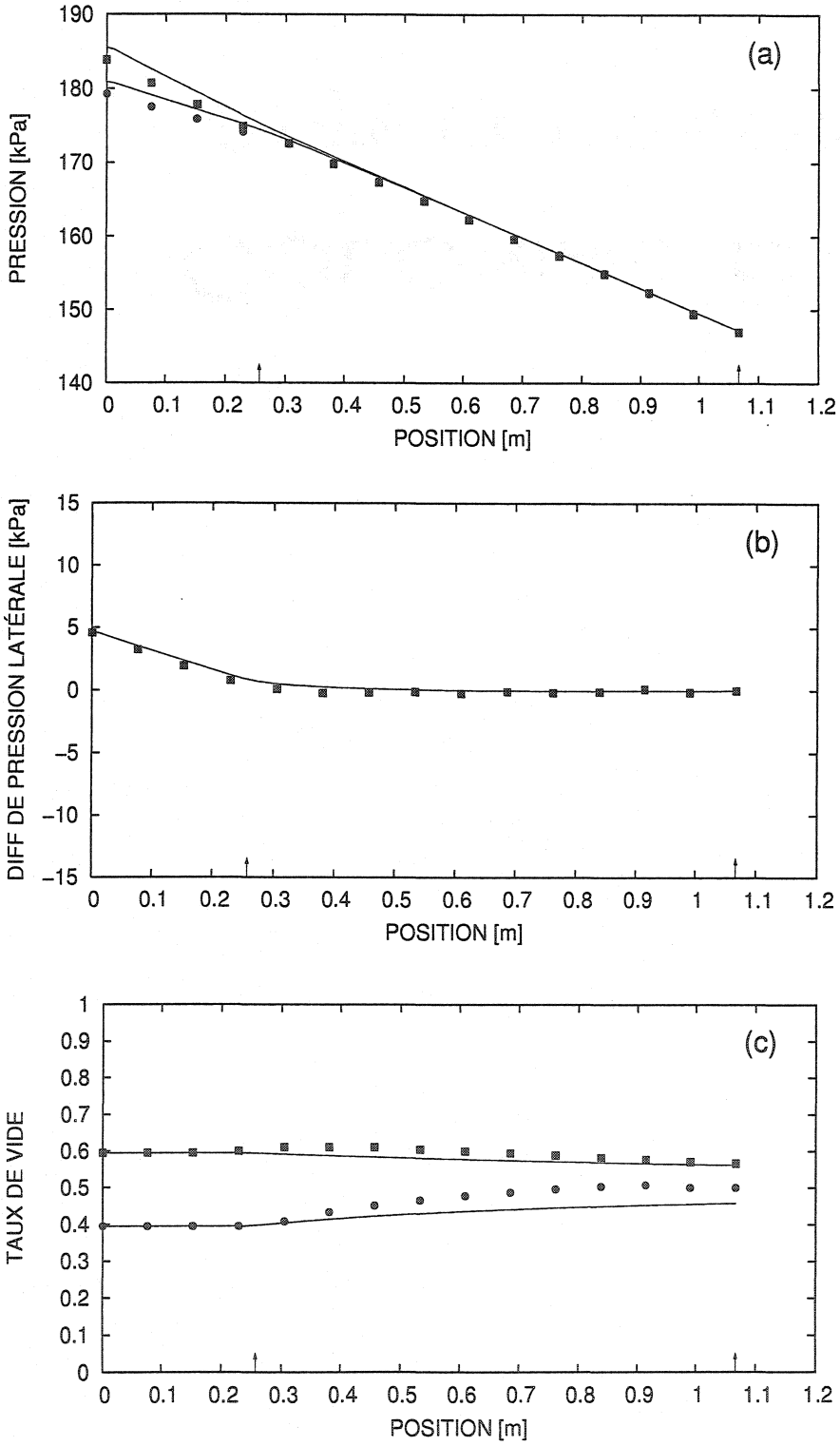


Figure 6.3(a)-(c): Comparaison des prédictions avec les données expérimentales. (fonction objectif : EQM.)

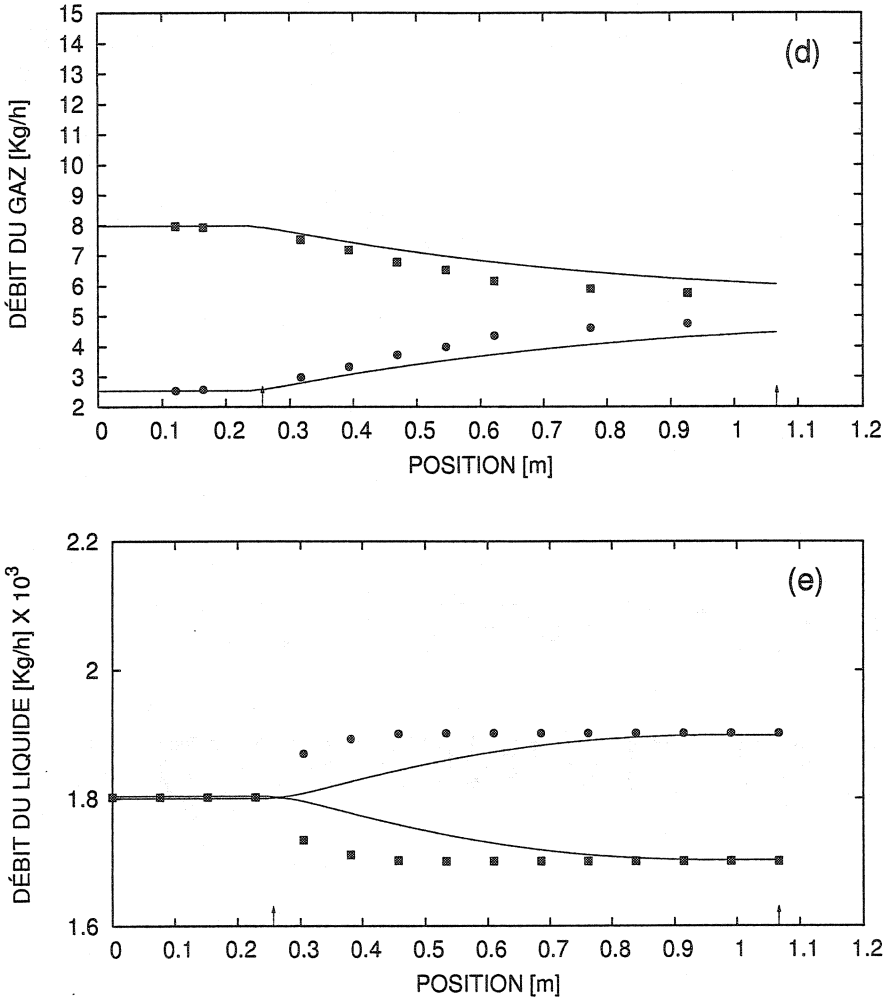


Figure 6.3(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 2

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.595	0.395
Débit du liquide [kg/m² s]	3125	3072
Débit du gaz [kg/m² s]	13.84	4.31

Résultats

$C_0 = 1.040$

$K_{tr} = 1.213$

$D_\alpha = 0.01016\left(\frac{\alpha}{0.45}\right)^{5.243}\tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.05959\left(\frac{\alpha}{0.45}\right)^{4.398}\tilde{D}_h \ll J_T \gg$

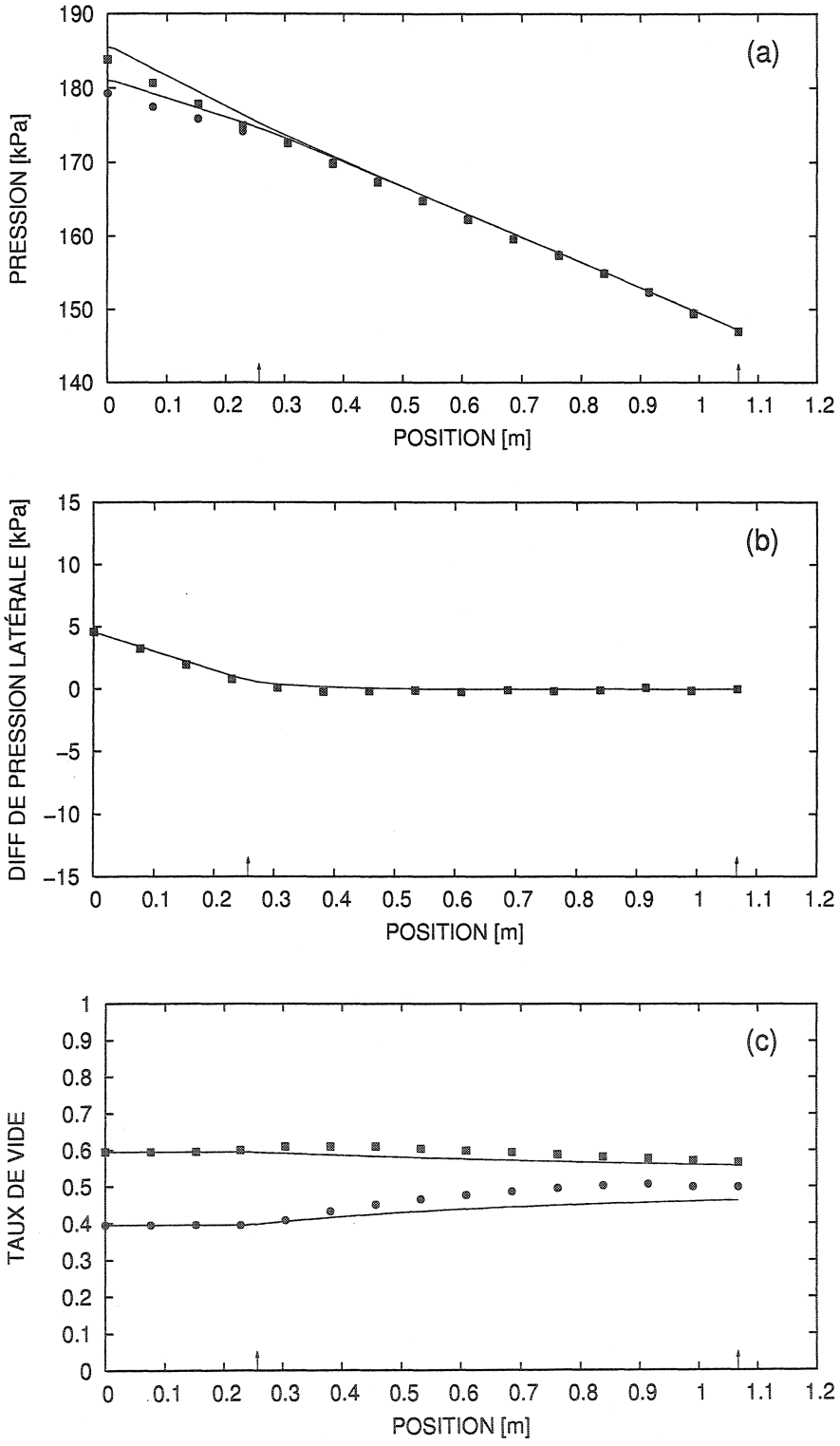


Figure 6.4(a)-(c): Comparaison des prédictions avec les données expérimentales. fonction objectif : (dissipation de l'énergie.)

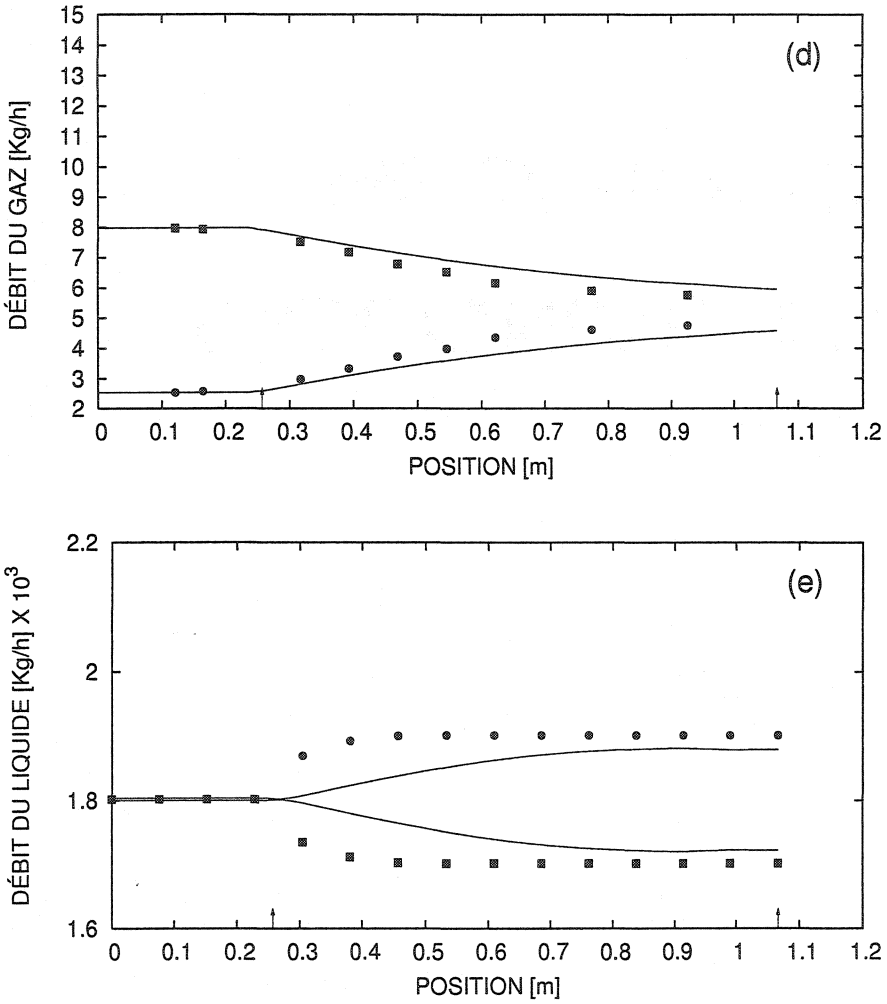


Figure 6.4(d)-(e) (Suite)
(fonction objectif : dissipation de l'énergie.)

EXPERIENCE 3

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.405	0
Débit du liquide [kg/m² s]	3127	3050
Débit du gaz [kg/m² s]	4.01	0

Résultats

$C_0 = 1.013$

$K_{tr} = 0.6641$

$D_\alpha = 0.4612\left(\frac{\alpha}{0.45}\right)^{4.012} \tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.03826\left(\frac{\alpha}{0.45}\right)^{6.757} \tilde{D}_h \ll J_T \gg$

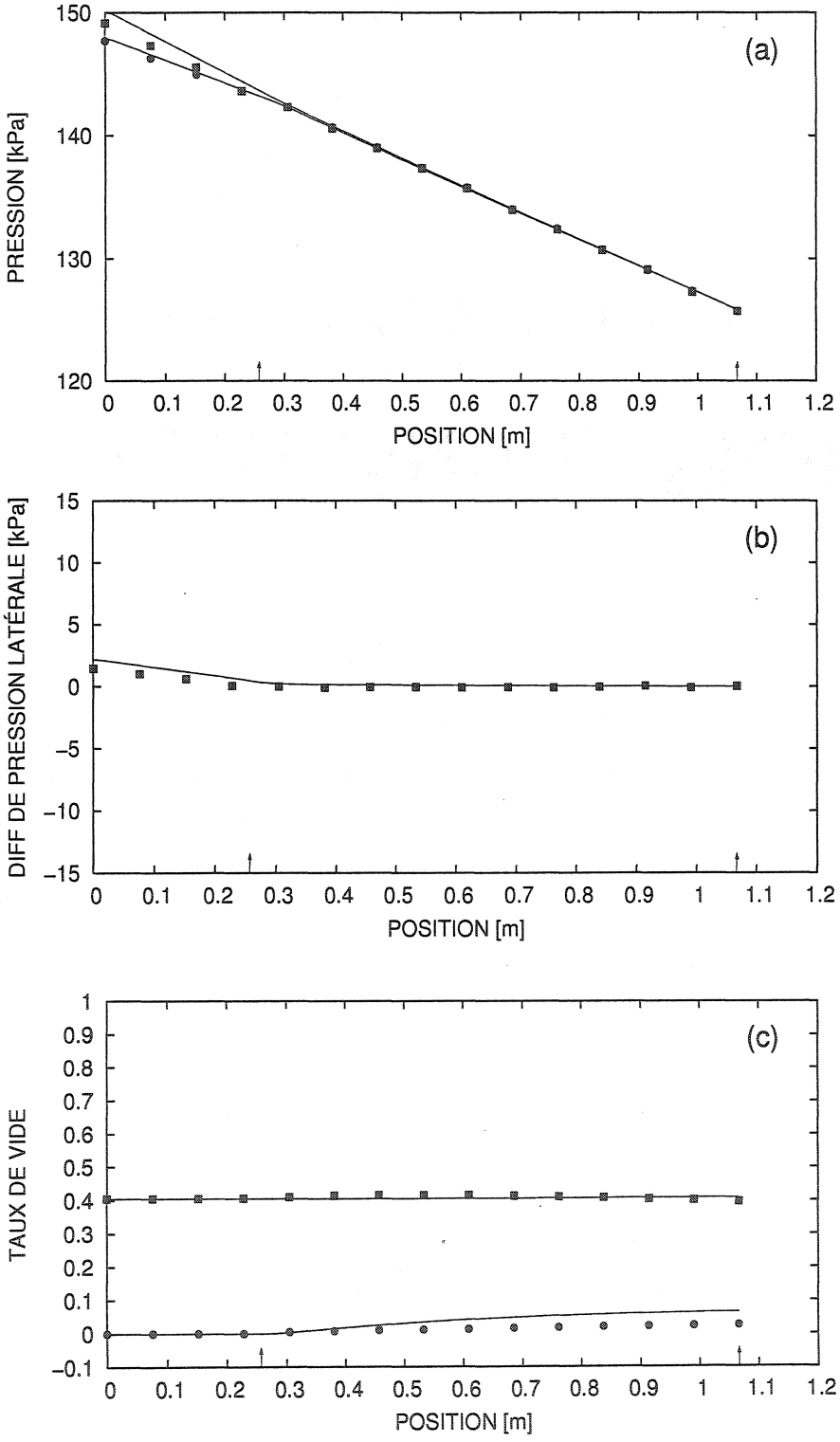


Figure 6.5(a)-(c): Comparaison des prédictions avec les données expérimentales. (fonction objectif : EQM.)

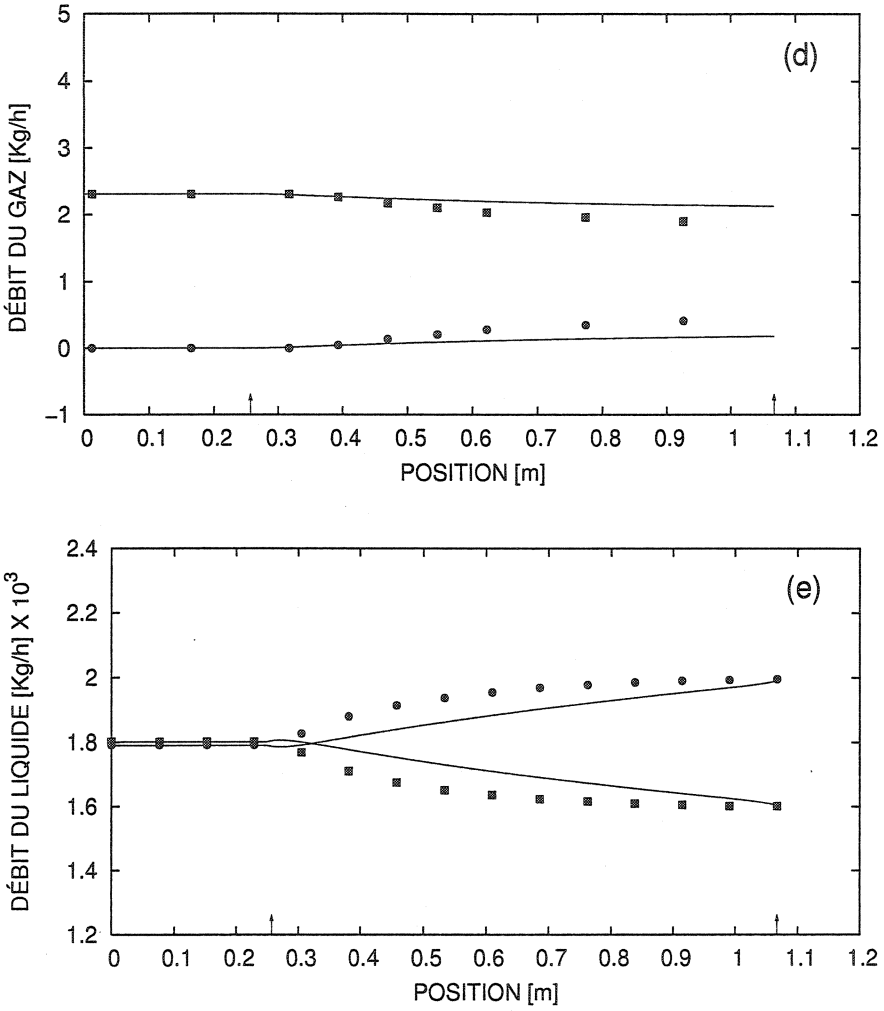


Figure 6.5(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 3

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.405	0
Débit du liquide [kg/m^2 s]	3127	3050
Débit du gaz [kg/m^2 s]	4.01	0

Résultats

$C_0 = 1.060$

$K_{tr} = 0.4882$

$D_\alpha = 0.3912\left(\frac{\alpha}{0.45}\right)^{6.795} \tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.0830\left(\frac{\alpha}{0.45}\right)^{4.894} \tilde{D}_h \ll J_T \gg$

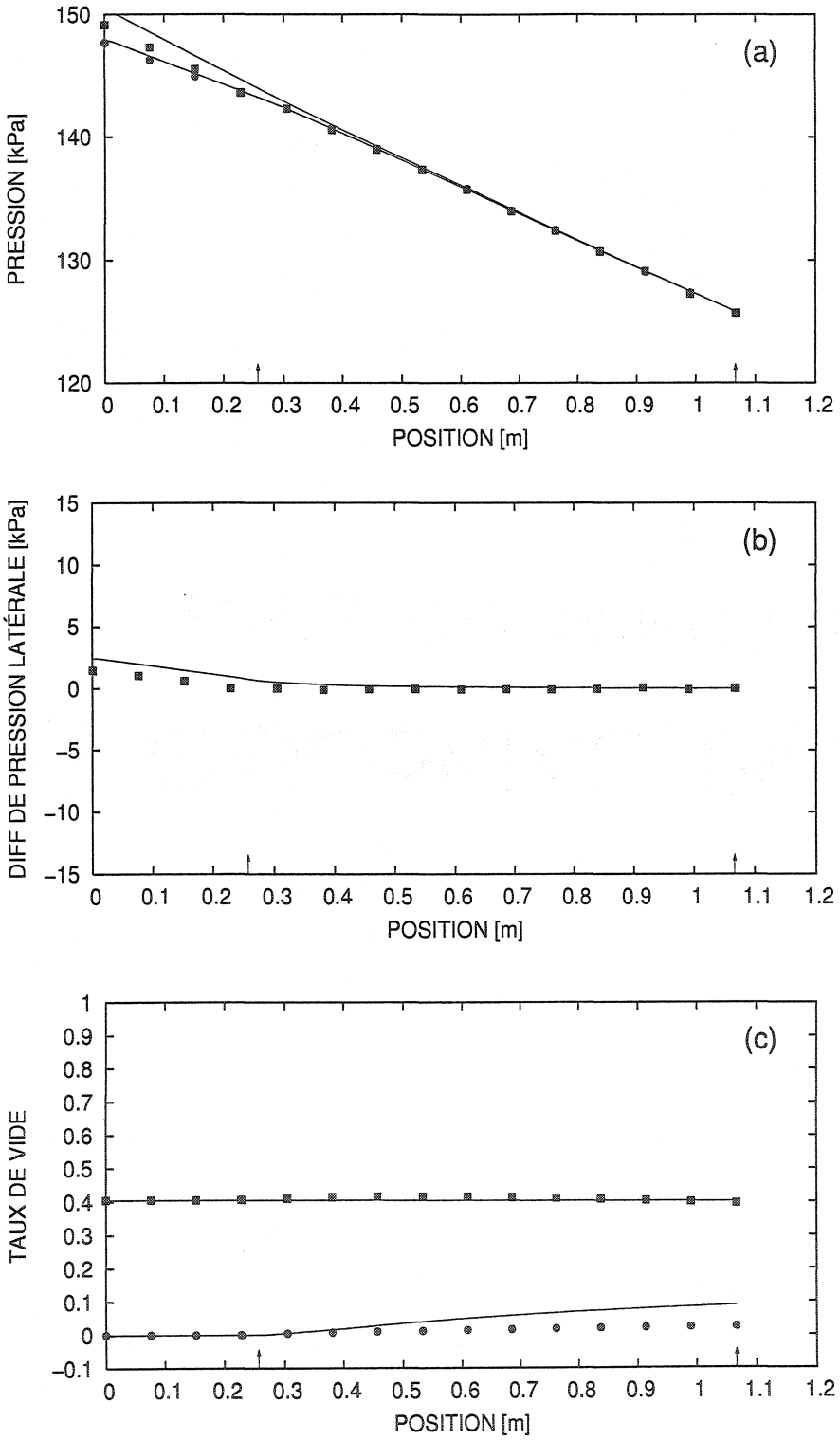


Figure 6.6(a)-(c): Comparaison des prédictions avec les données expérimentales. (fonction objectif : dissipation de l'énergie.)

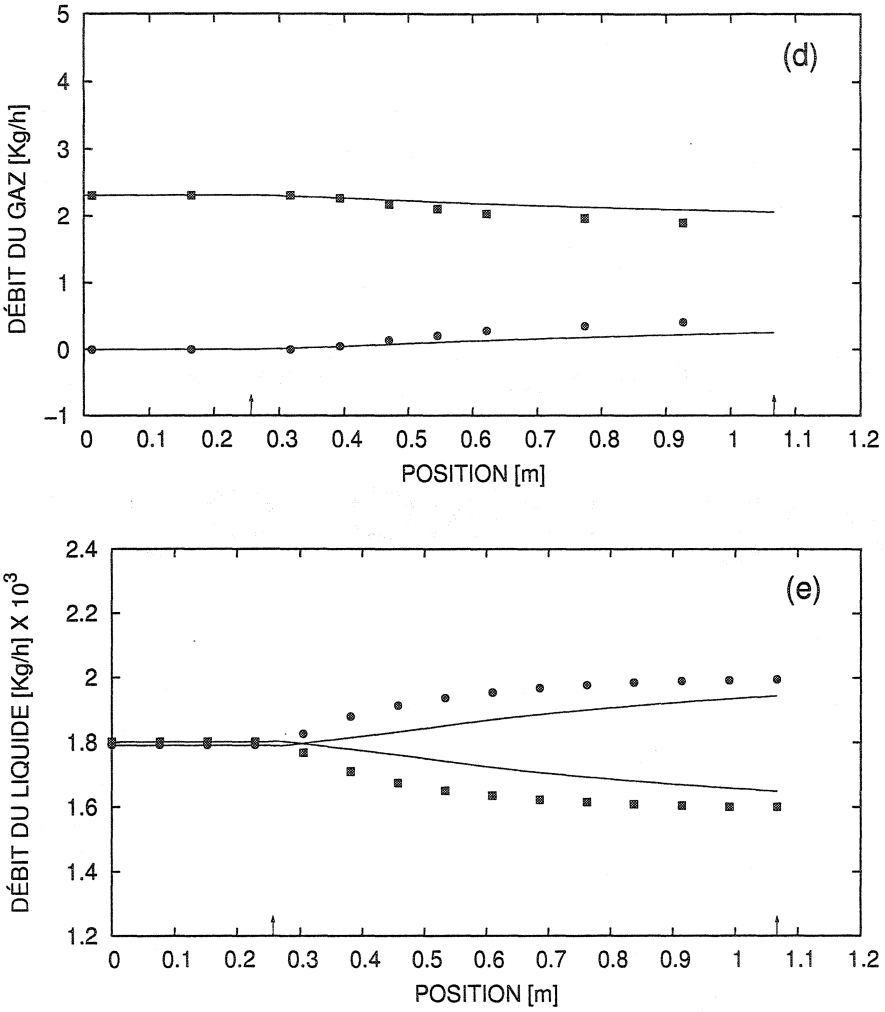


Figure 6.6(d)-(e) (Suite)
(fonction objectif : dissipation de l'énergie.)

EXPERIENCE 5

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.292	0
Débit du liquide [kg/m² s]	3139	3071
Débit du gaz [kg/m² s]	1.96	0

Résultats

$C_0 = 1.0$

$K_{tr} = 1.0$

$D_\alpha = 0.075\left(\frac{\alpha}{0.45}\right)^{6.0} \tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.075\left(\frac{\alpha}{0.45}\right)^{6.0} \tilde{D}_h \ll J_T \gg$

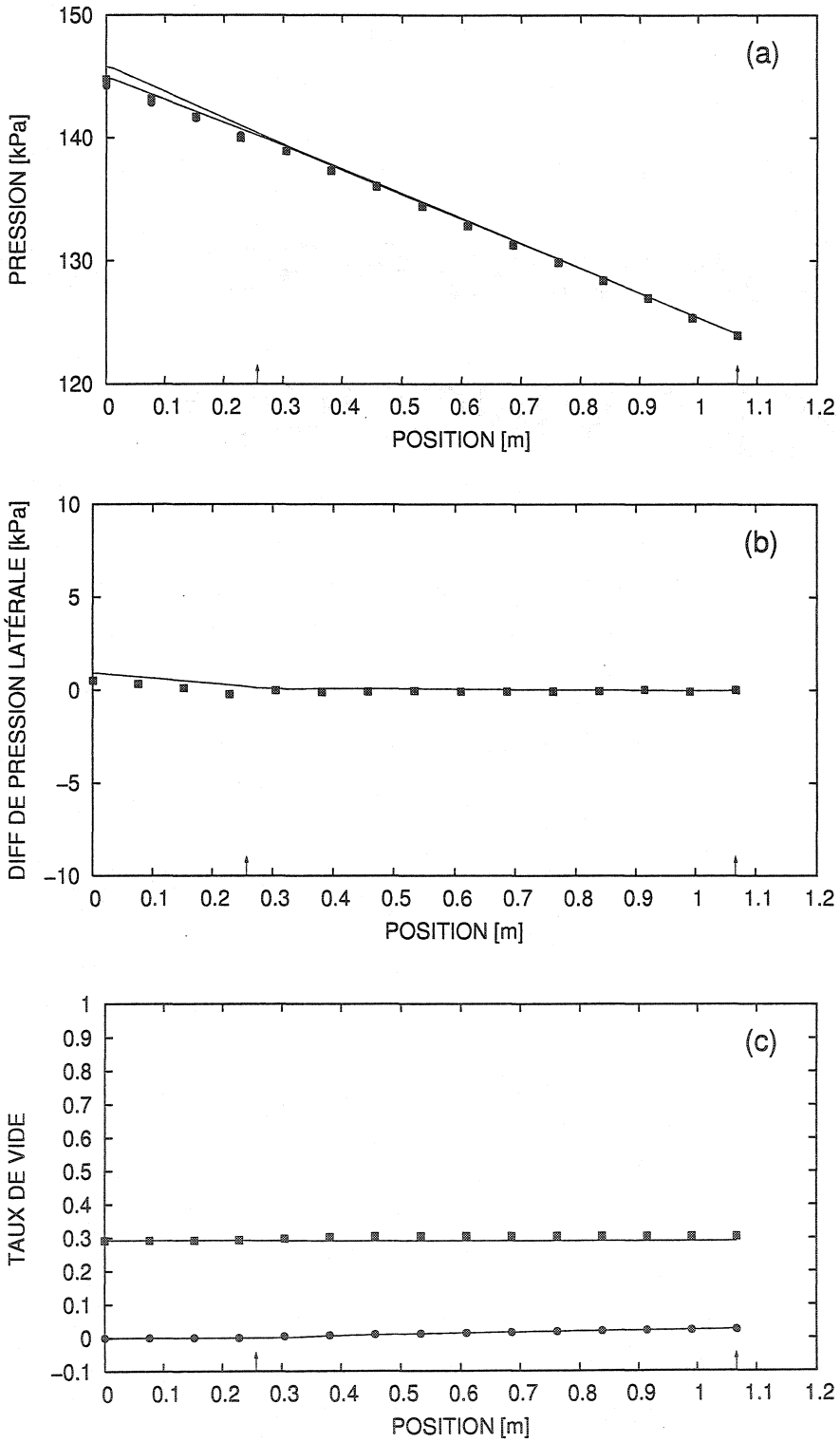


Figure 6.7(a)-(c): Comparaison des prédictions avec les données expérimentales. (fonction objectif : EQM.)

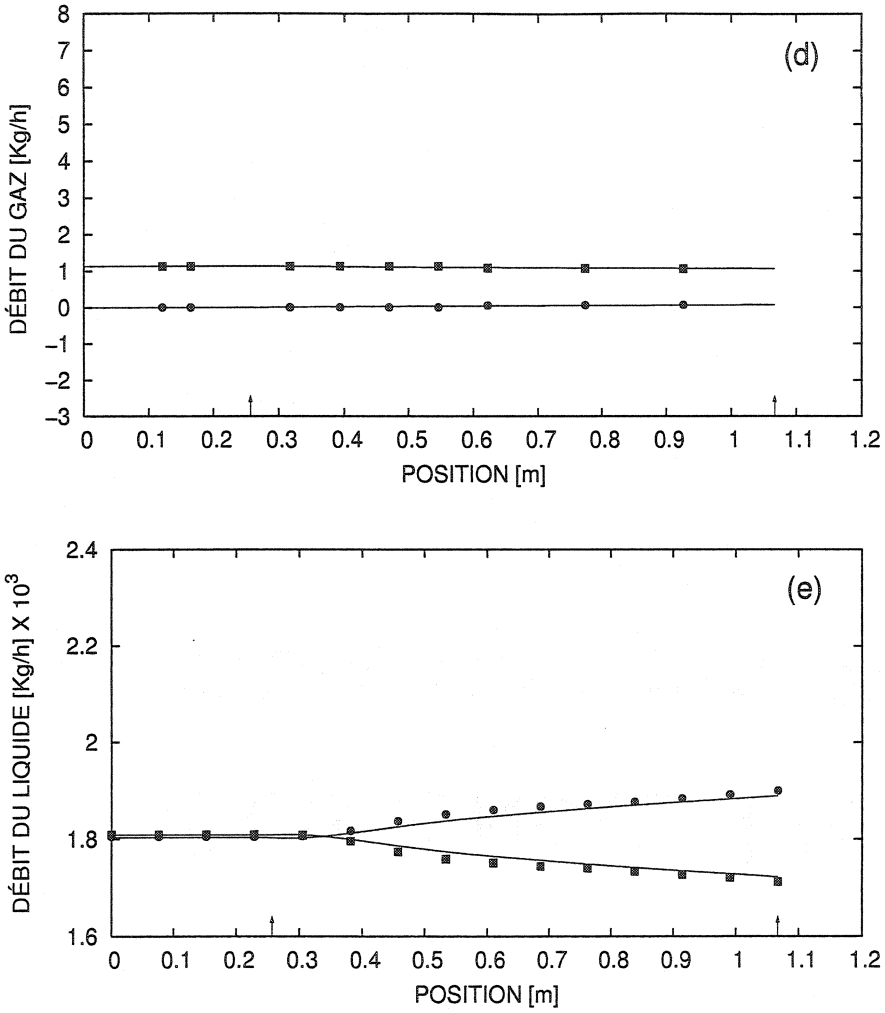


Figure 6.7(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 5

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.292	0
Débit du liquide [kg/m^2 s]	3139	3071
Débit du gaz [kg/m^2 s]	1.96	0

Résultats

$C_0 = 1.052$

$K_{tr} = 1.062$

$D_\alpha = 0.4915\left(\frac{\alpha}{0.45}\right)^{4.167} \tilde{D}_h \ll J_T \gg$

$D_{eq} = 0.9059\left(\frac{\alpha}{0.45}\right)^{5.874} \tilde{D}_h \ll J_T \gg$

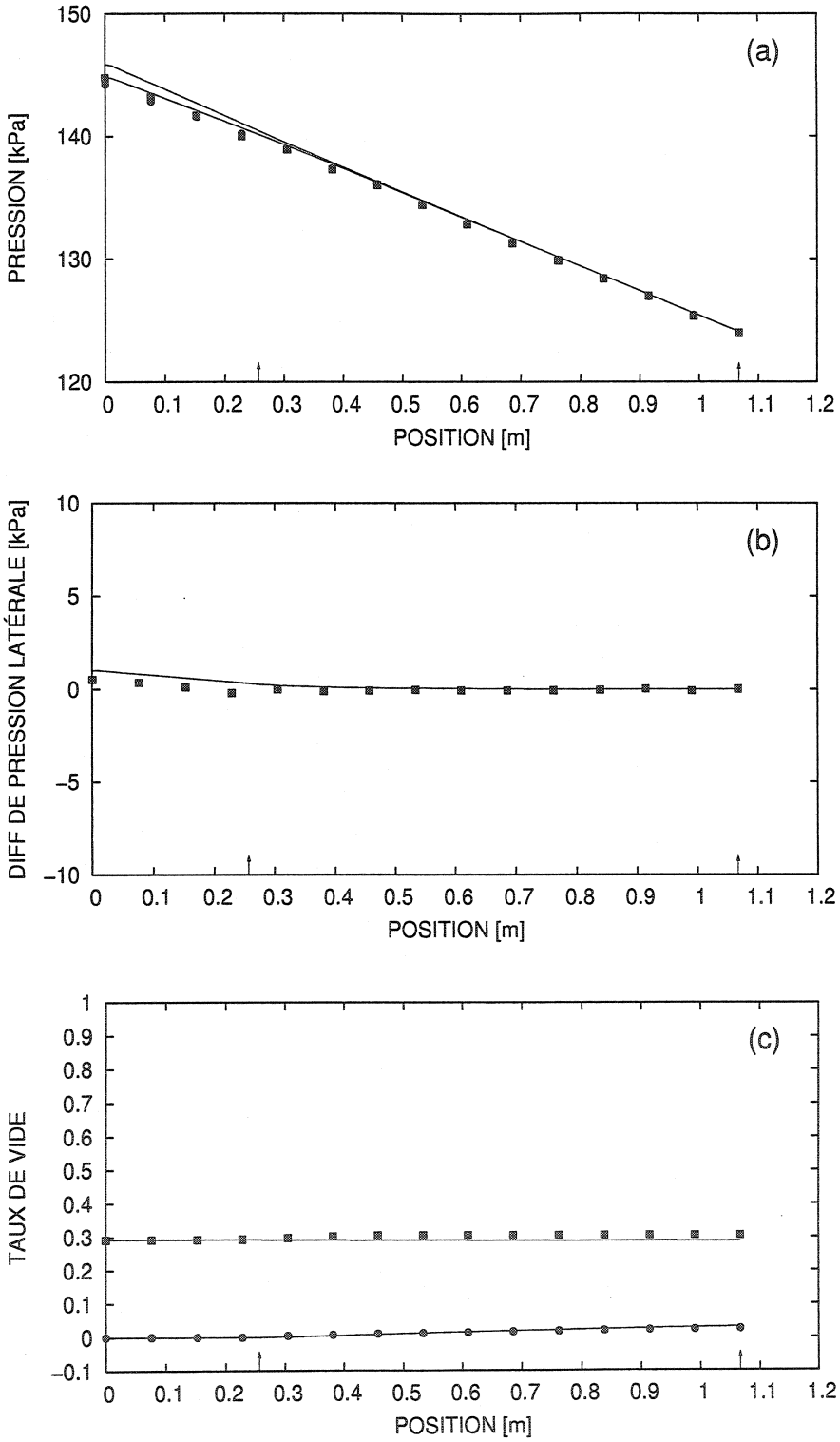


Figure 6.8(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : dissipation de l'énergie.)

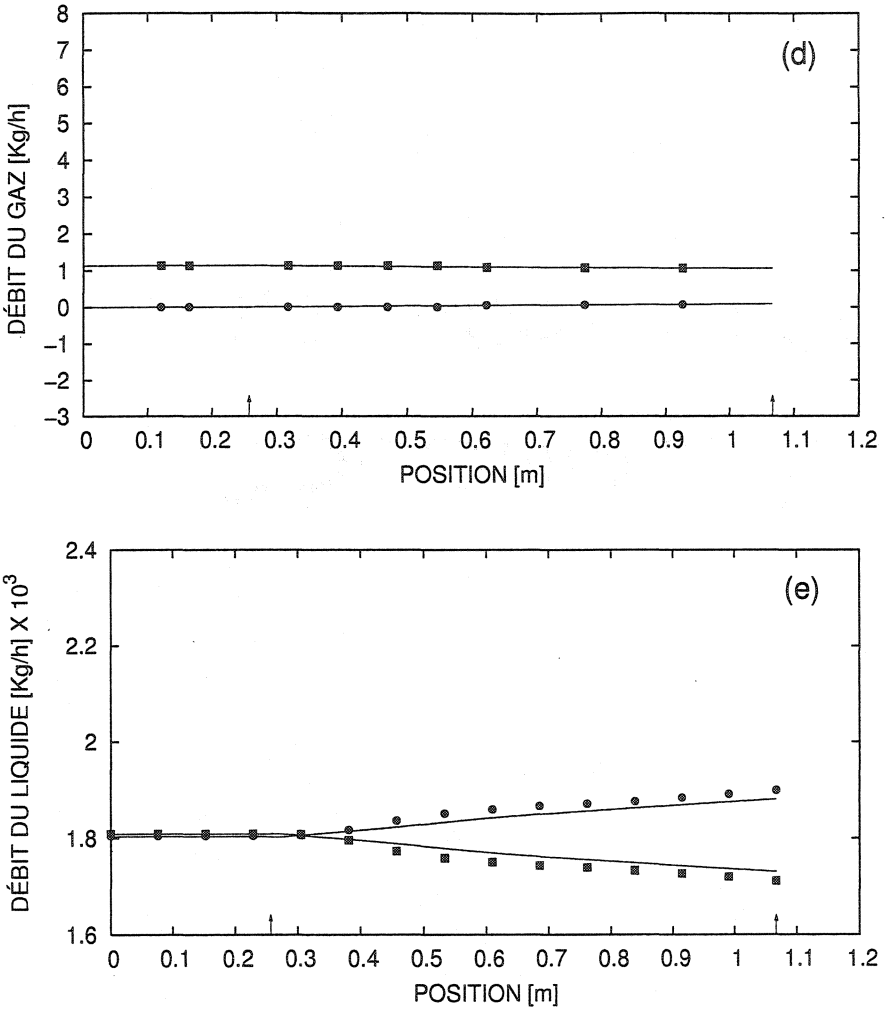


Figure 6.8(d)-(e) (Suite)
(fonction objectif : dissipation de l'énergie.)

EXPERIENCE 6

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.51	0
Débit du liquide [kg/m² s]	3125	3065
Débit du gaz [kg/m² s]	8.45	0

Résultats

$C_0 = 1.039$

$K_{tr} = 1.289$

$D_\alpha = 0.02311\left(\frac{\alpha}{0.45}\right)^{8.135} D_h \ll J \gg$

$D_{eq} = 0.08953\left(\frac{\alpha}{0.45}\right)^{4.919} D_h \ll J \gg$

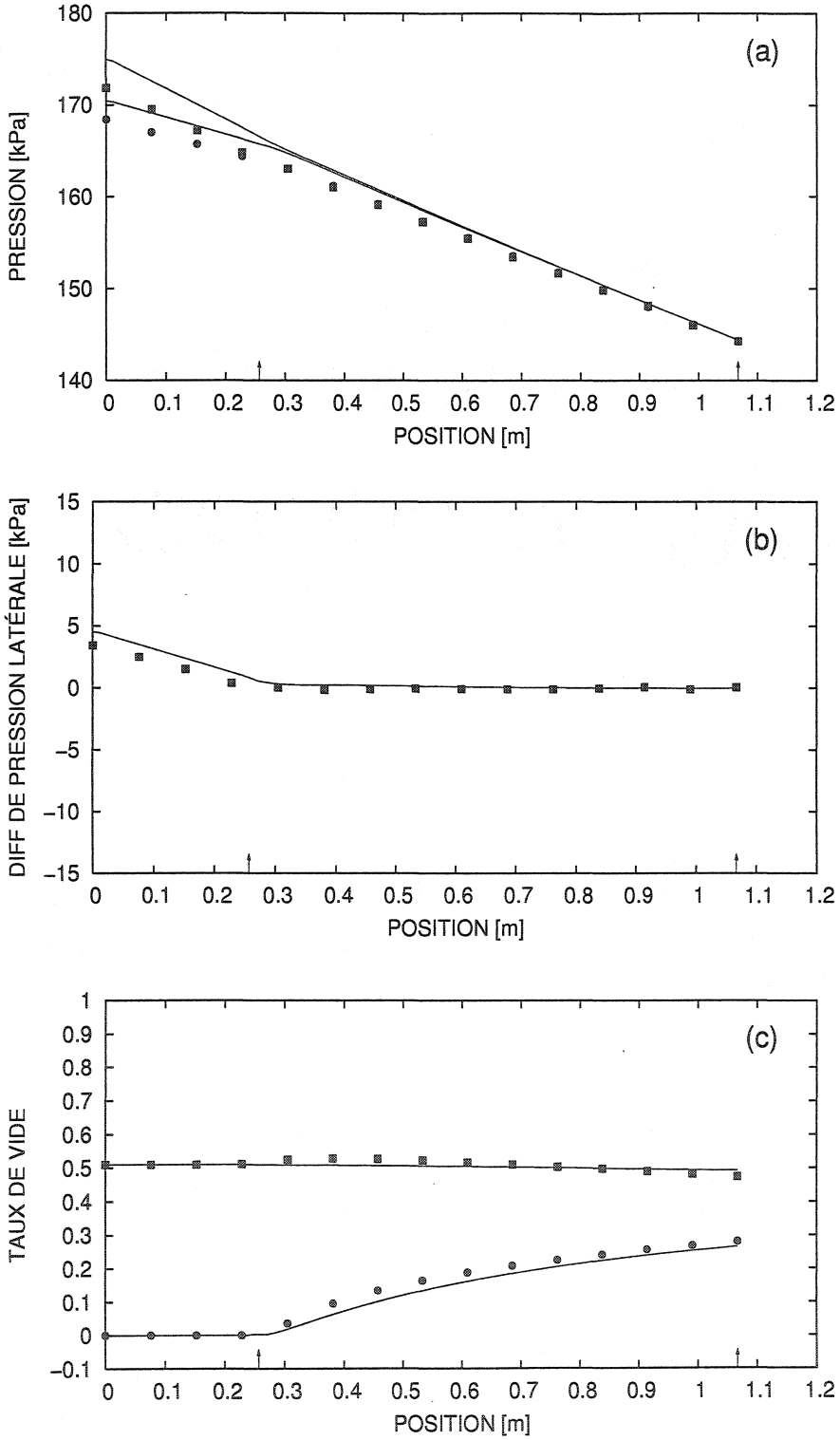


Figure 6.9(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : EQM.)

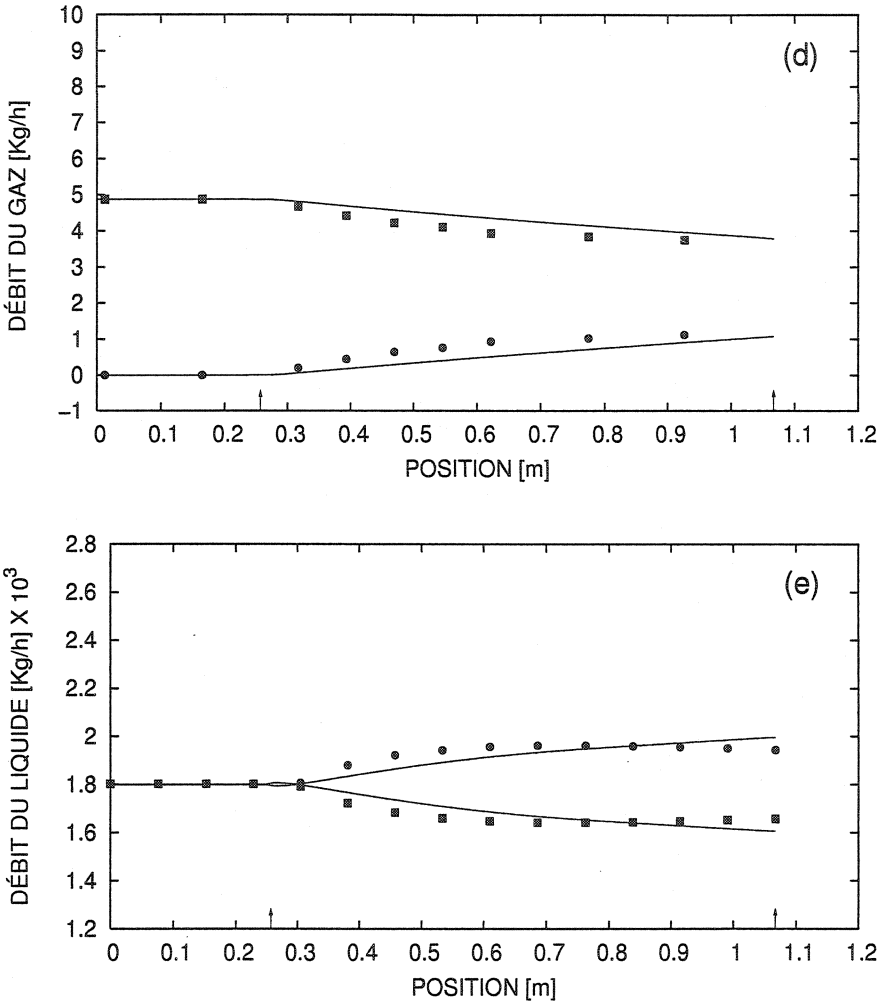


Figure 6.9(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 6

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.51	0
Débit du liquide [kg/m² s]	3125	3065
Débit du gaz [kg/m² s]	8.45	0

Résultats

$C_0 = 1.091$

$K_{tr} = 1.356$

$D_\alpha = 0.08214\left(\frac{\alpha}{0.45}\right)^{8.085} D_h \ll J \gg$

$D_{eq} = 0.09938\left(\frac{\alpha}{0.45}\right)^{4.777} D_h \ll J \gg$

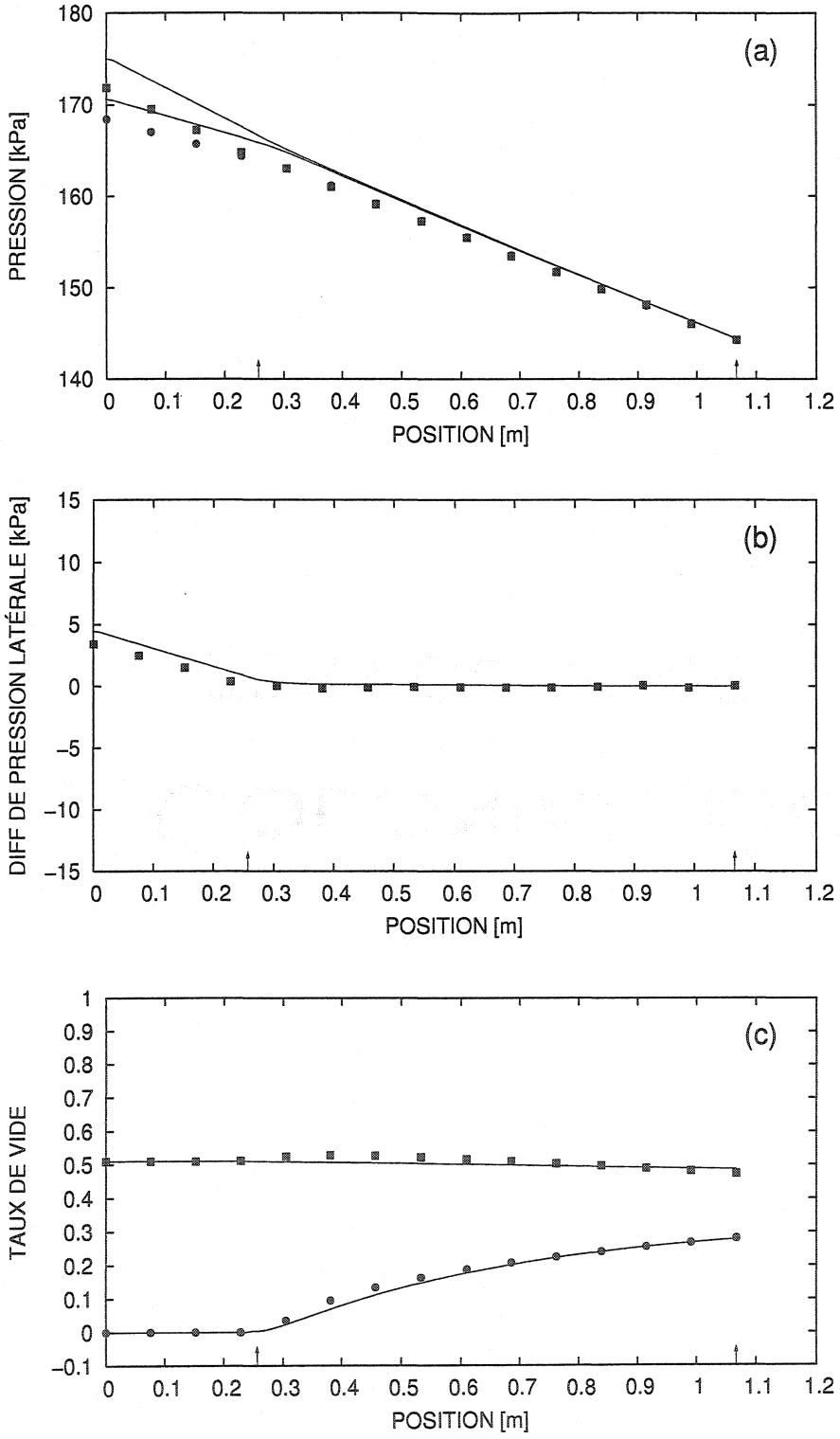


Figure 6.10(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : dissipation de l'énergie.)

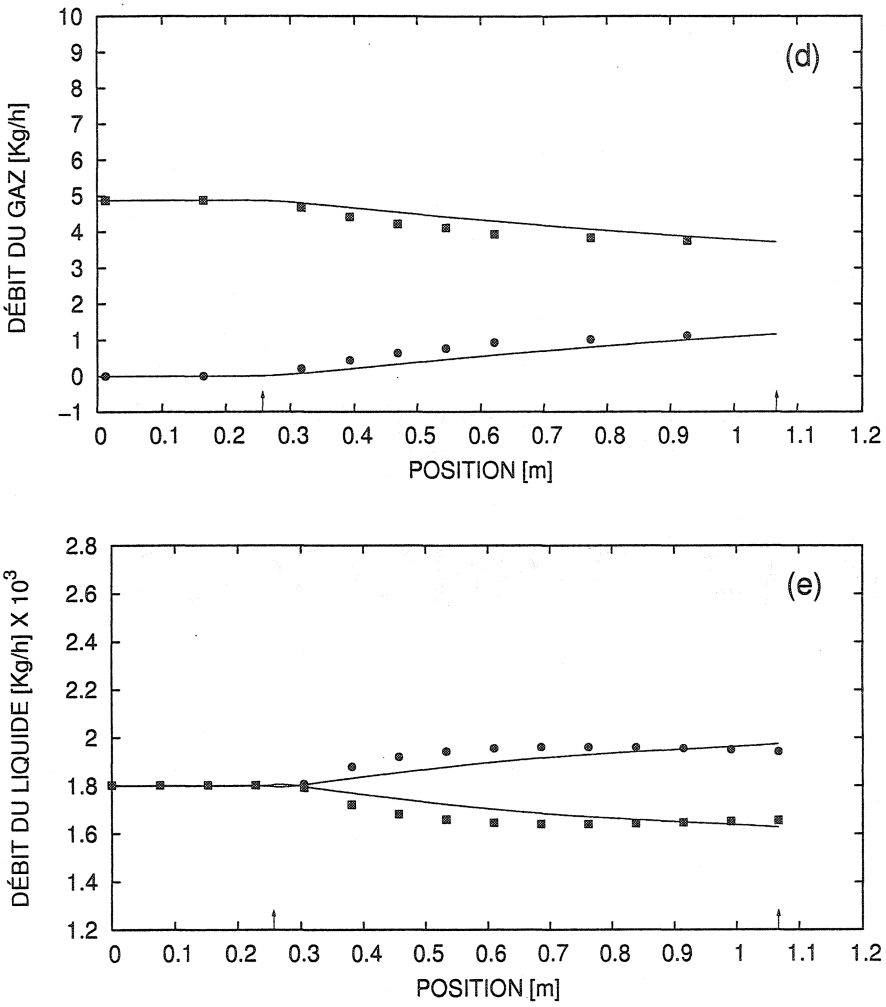


Figure 6.10(d)-(e): (Suite)
(fonction objectif : dissipation de l'énergie.)

EXPERIENCE 8

fonction objectif : Erreur quadratique moyenne

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.40	0.13
Débit du liquide [kg/m^2 s]	3119	3064
Débit du gaz [kg/m^2 s]	4.25	0.73

Résultats

$C_0 = 1.014$

$K_{tr} = 1.278$

$D_\alpha = 0.03516\left(\frac{\alpha}{0.45}\right)^{8.076} D_h \ll J \gg$

$D_{eq} = 0.9545\left(\frac{\alpha}{0.45}\right)^{8.809} D_h \ll J \gg$

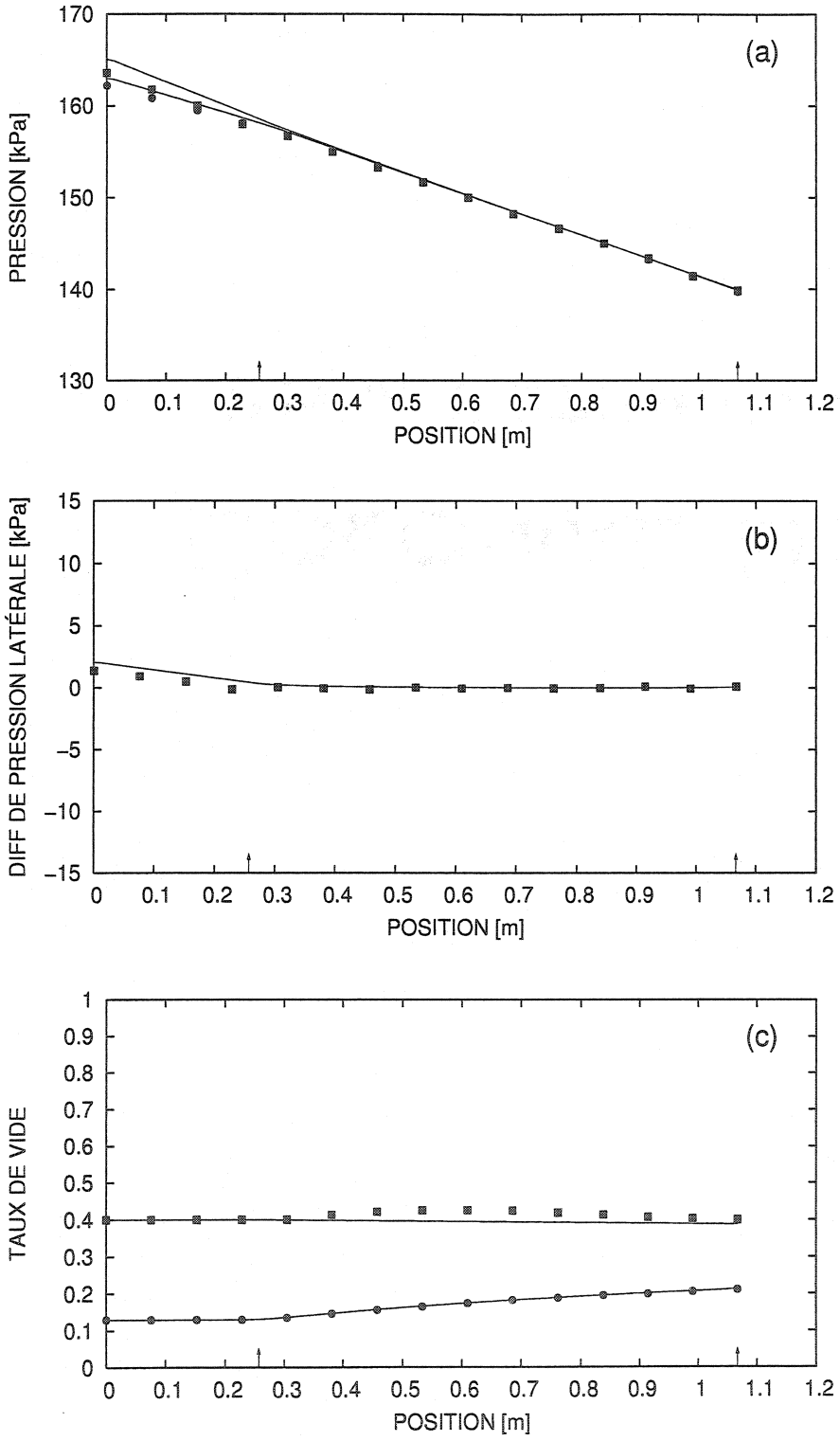


Figure 6.11(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : EQM.)

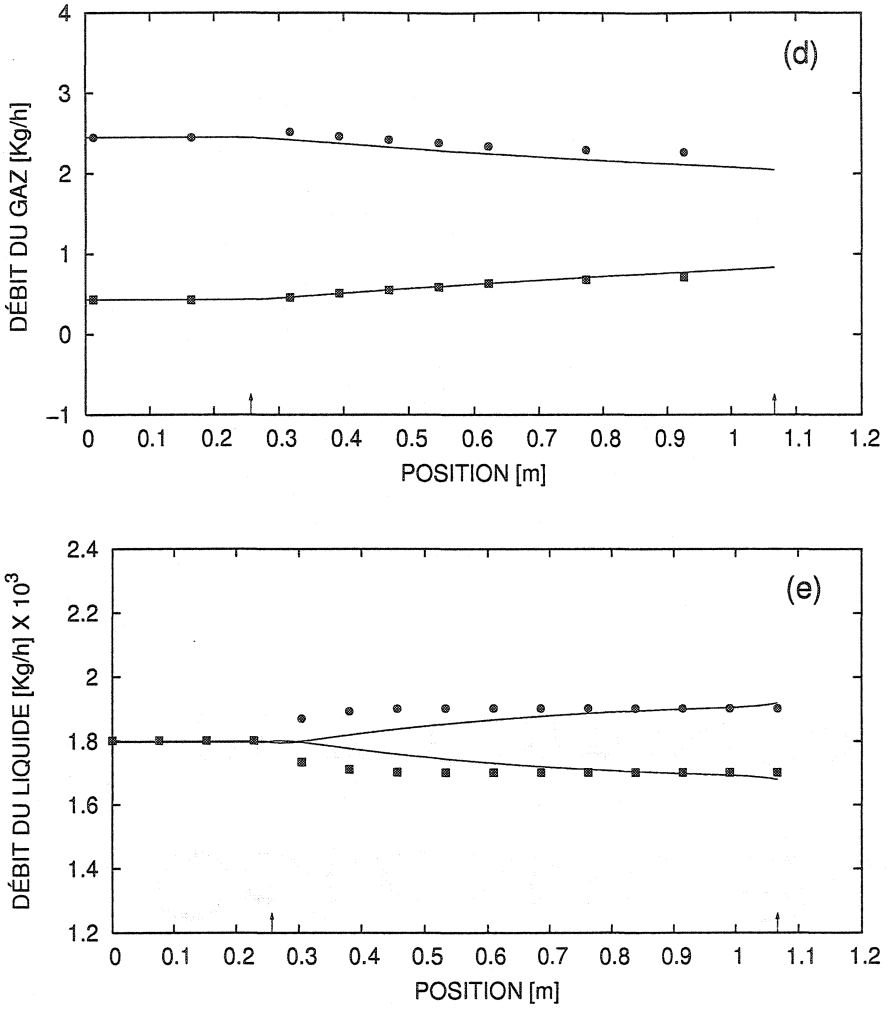


Figure 6.11(d)-(e) (Suite)
(fonction objectif : EQM.)

EXPERIENCE 8

fonction objectif : Dissipation de l'énergie

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.40	0.13
Débit du liquide [kg/m^2 s]	3119	3064
Débit du gaz [kg/m^2 s]	4.25	0.73

Résultats

$C_0 = 1.047$

$K_{tr} = 1.233$

$D_\alpha = 0.4554\left(\frac{\alpha}{0.45}\right)^{6.874} D_h \ll J \gg$

$D_{eq} = 0.1121\left(\frac{\alpha}{0.45}\right)^{5.567} D_h \ll J \gg$

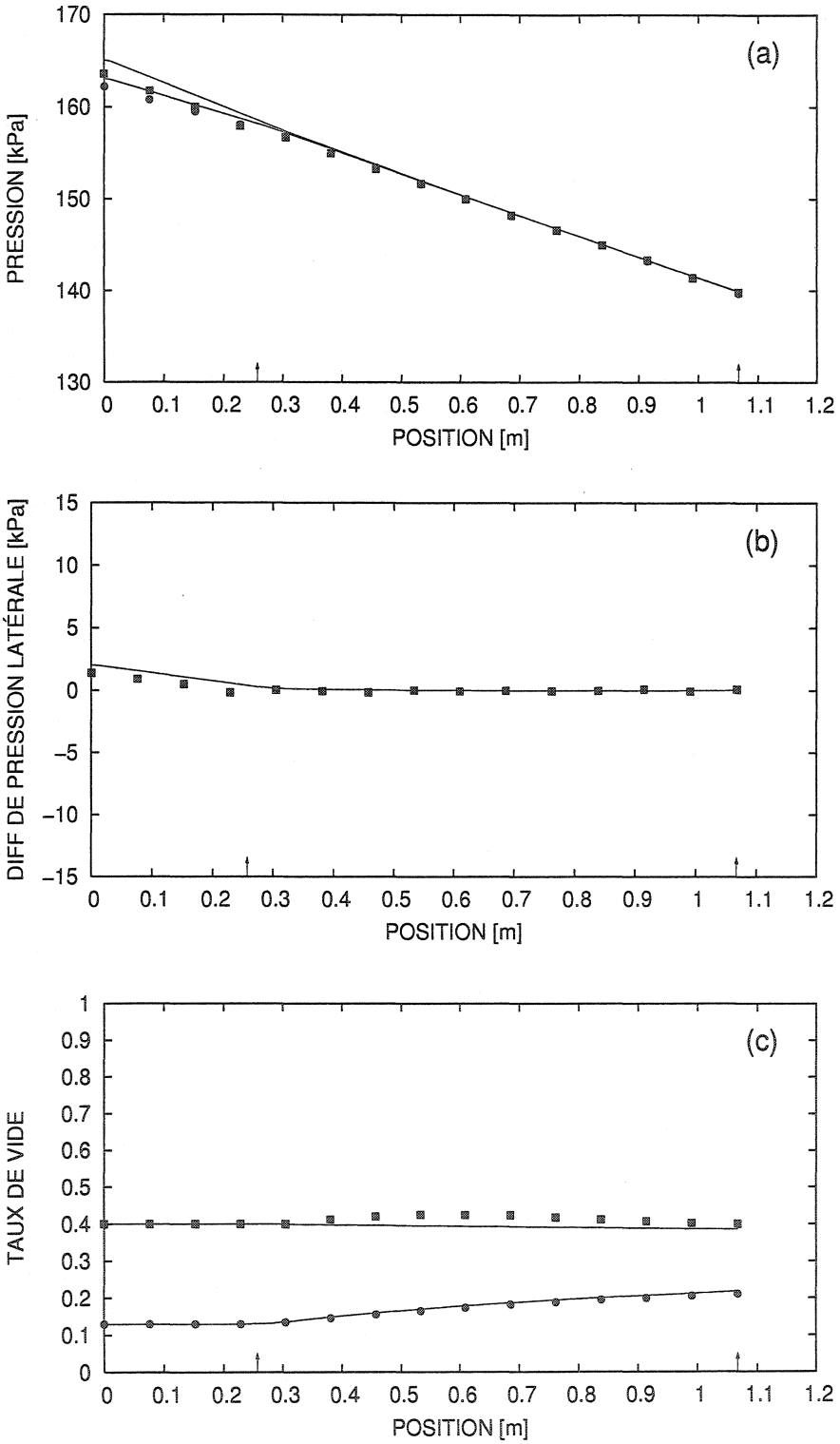


Figure 6.12(a)-(c): Comparaison des predictions avec les données expérimentales. (fonction objectif : dissipation de l'énergie.)

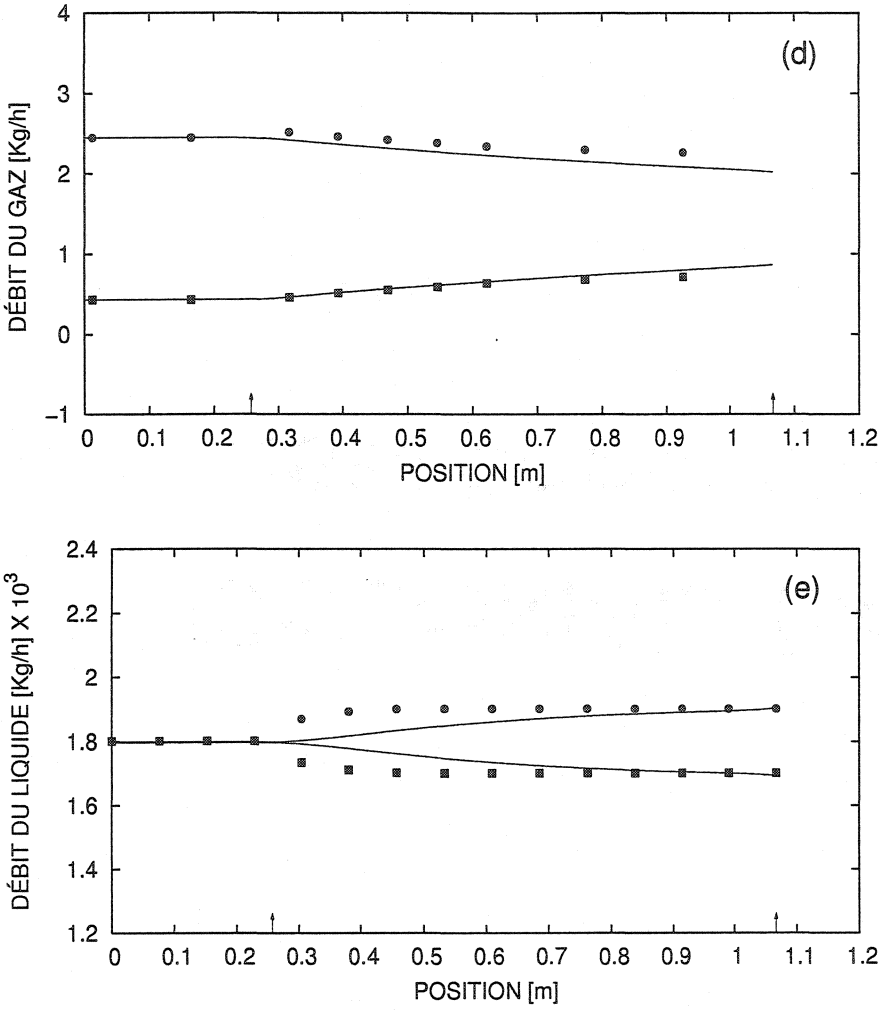


Figure 6.12(d)-(e)(Suite)
(fonction objectif : dissipation de l'énergie.)

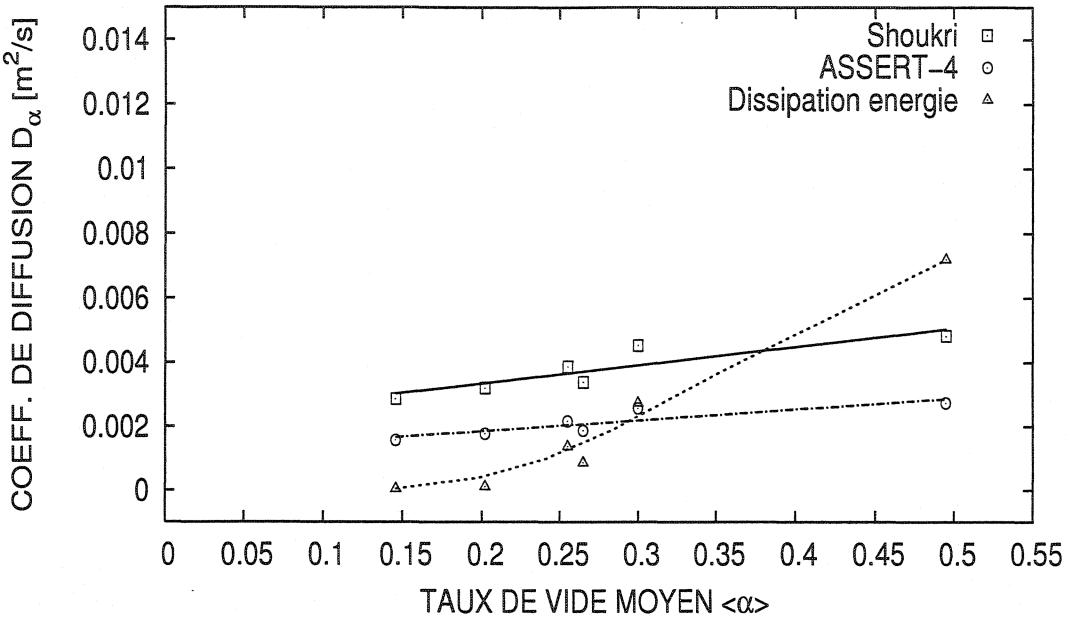


Figure 6.13: Coefficients de diffusion D_α.

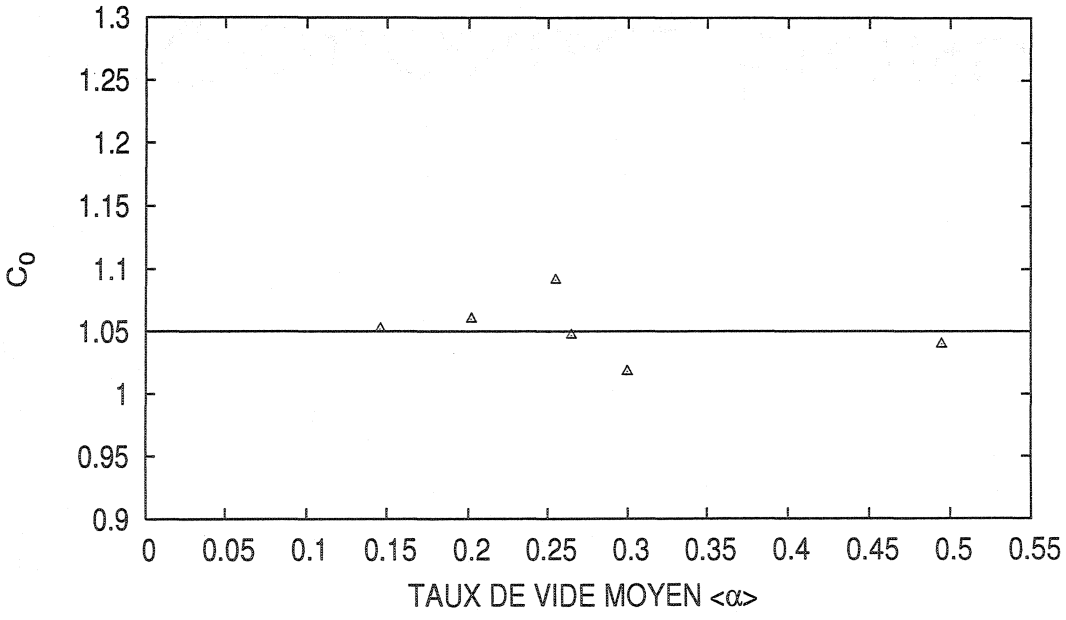


Figure 6.14: Coefficients de distribution C_0.

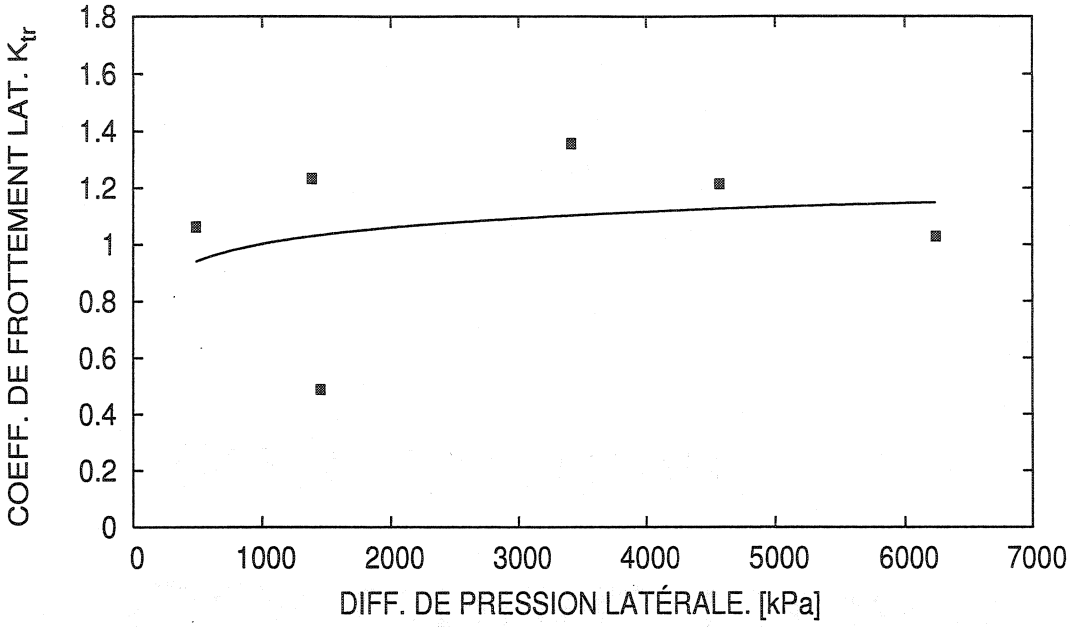


Figure 6.15: Coefficients de frottement latéral K_{tr}.

COMPARAISON DES PRÉDICTIONS

Données d'entrée

	Canal donneur ■	Canal Réservoir ●
Taux de vide	0.595	0.395
Débit du liquide [kg/m^2 s]	3125	3072
Débit du gaz [kg/m^2 s]	13.84	4.31

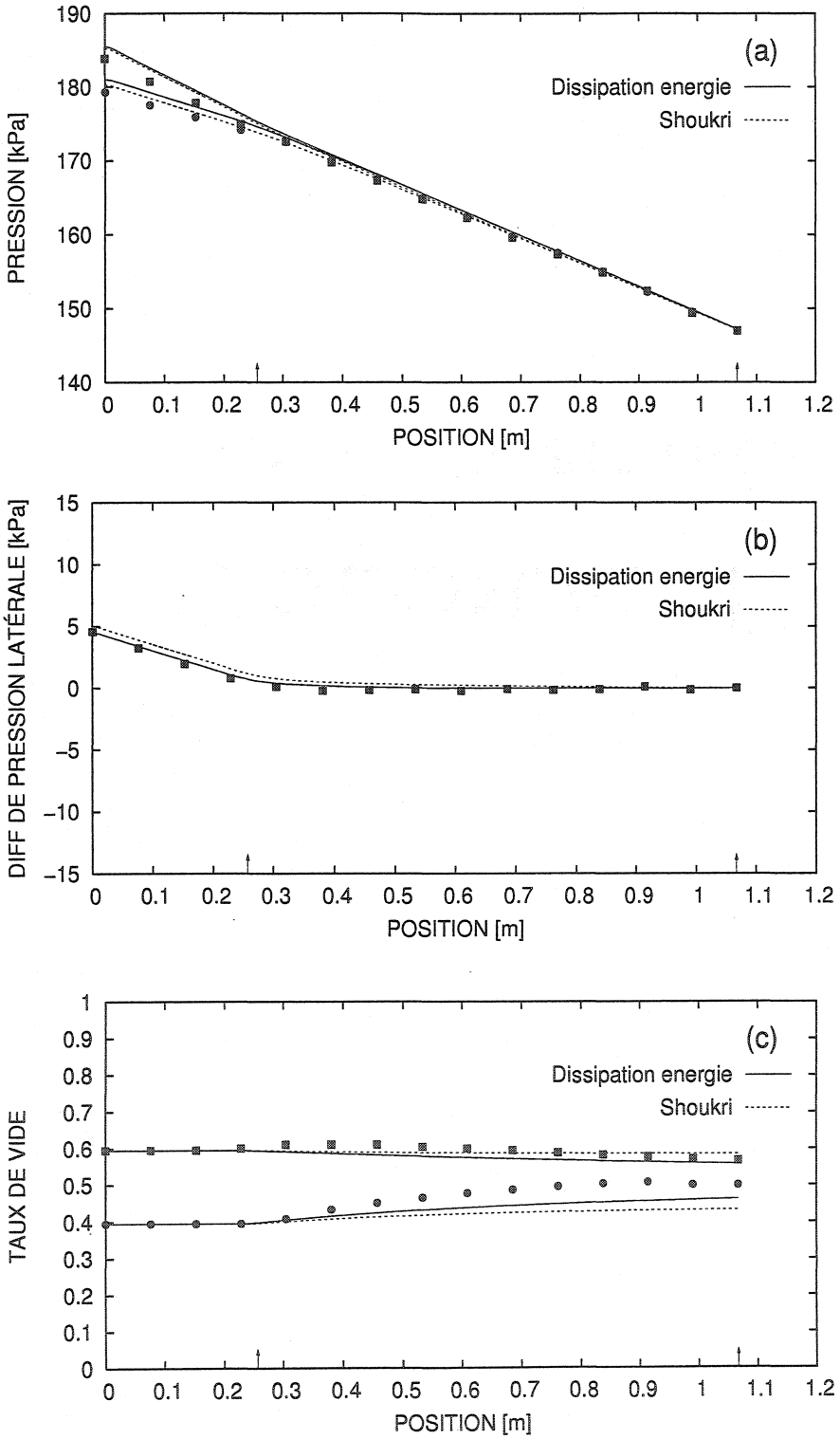
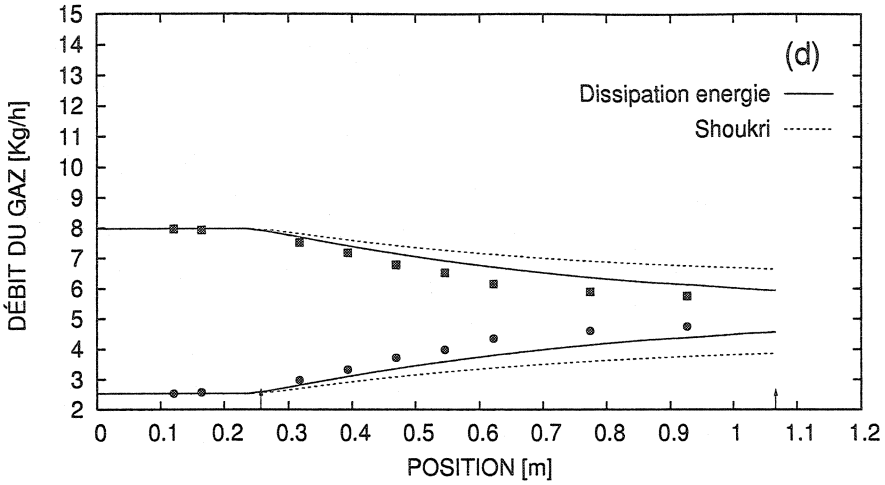


Figure 6.16(a)-(c): Comparaison de la corrélation de Shoukri et le modèle basé sur la dissipation de l'énergie.



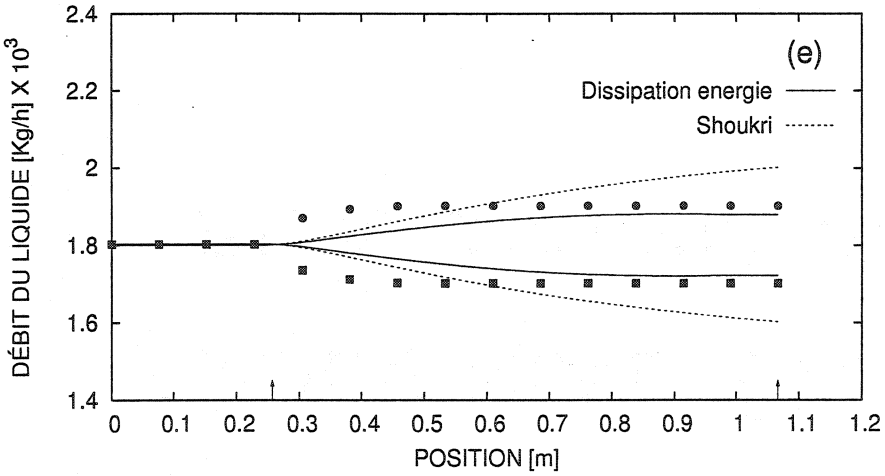


Figure 6.16(d)-(e) (Suite)

CHAPITRE 7

LES CONCLUSIONS ET LES RECOMMANDATIONS

Cette étude a permis d'approfondir les connaissances des phénomènes d'échanges latéraux entre les sous-canaux interconnectés. Comme on a discuté dans le Chapitre sur la revue bibliographique, on peut constater que malgré les efforts déployées par plusieurs chercheurs pour comprendre et modéliser ces mécanismes de mélange ils doivent être étudiés d'avantage. C'est particulièrement le cas du mélange turbulent et la dérive du vide vers une distribution à l'équilibre ("void drift").

L'ambition de ce projet était de comprendre la physique des mécanismes de mélange et de reconnaître les variables les plus importantes qui peuvent être utilisées pour optimiser les calculs d'un code d'analyse de sous-canal.

Ce dernier chapitre sera dédié à présenter les conclusions les plus remarquables et les recommandations qui peuvent aider à comprendre la physique de ces mécanismes.

7.1 Les conclusions

L'implantation d'un Algorithme Génétique avec un code de sous-canal a été effectué ayant comme but de chercher les meilleures valeurs des coefficients de diffusion D_α et D_{eq}, le coefficient de distribution du vide C_0 et le coefficient de frottement latéral K_{tr} que permettent d'optimiser les prédictions du code.

Le projet peut être divisé en deux parties. La première partie de ce projet est consacrée à l'analyse de sensibilité des différentes variables qu'interviennent dans les échanges latéraux. Cette analyse est réalisée en utilisant un AG pour minimiser la différence entre les valeurs expérimentales et les valeurs calculées.

L'analyse nous permet de conclure que la différence de pression entre les sous-canaux interconnectés est une variable très importante qui contrôle les mécanismes de mélange, cependant elle n'est pas complètement responsable des échanges laté-

raux.

La deuxième partie a été dédiée à l'analyse des différentes fonctions objectifs basées sur les principes de la minimisation de la création d'entropie et de sa dérivation : la minimisation de la dissipation de l'énergie.

Différents chercheurs ont utilisé ces principes pour comprendre les mécanismes de mélange. Shirai et Ninokata (2001) a utilisé le concept de la minimisation de l'énergie dissipé pour montrer qu'un écoulement à l'intérieur des sous-canaux interconnectés a tendance à minimiser le gradient de pression. Tahir et Carver (1982) ont utilisé le principe de la création minimale de l'entropie pour déterminer les coefficients de diffusion d'un modèle de mélange.

En se basant sur ces idées, nous avons élaboré un algorithme d'optimisation (AG) qui a permis d'utiliser ces différentes fonctions objectif. Le Chapitre V a été consacré à l'explication de ces fonctions.

Il faut mentionner que le couplage de deux codes permet la recherche des coefficients de diffusion sans l'intervention de l'utilisateur. L'utilisateur doit fournir seulement les intervalles de recherche, le code génétique cherchera parmi cet intervalle les meilleures qui permettent de minimiser l'énergie dissipée. Il semble que c'est le premier code de sous-canal capable de chercher ses propres valeurs des coefficients de diffusion.

Après avoir réalisé les simulations de l'ensemble des expériences, on peut conclure que :

a) Le couplage d'un algorithme génétique avec un code de sous canal a permis d'optimiser les prédictions du code. L'utilisation de l'algorithme génétique a besoin d'une fonction objectif, qui une fois sélectionnée le code sera capable de chercher ces coefficients sans l'intervention de l'utili sateur.

b) L'écoulement qui passe à travers d'un sous-canal tend vers l'équilibre hydrodynamique, en se redistribuant de telle manière à minimiser l'énergie

dissipée.

c) Le modèle de mélange utilisé dans ce projet est capable de reproduire les tendances expérimentales, sauf pour la région proche du commencement de l'interconnexion où les valeurs calculées sous estimant les valeurs expérimentales.

d) On confirme aussi qu'il n'existe pas une seule corrélation pour D_α et D_{eq} qui permettent de reproduire l'ensemble d'expériences analysées.

e) Les coefficients de diffusion la tendance à augmenter lorsque le taux de vide moyen augmente, cette tendance a déjà été observée par d'autres chercheurs.

f) Les coefficients de diffusion peuvent être corrélés en utilisant le nombre de Reynolds, en effet, on a montré que la corrélation utilisée par le modèle de Shoukri *et al.*, (1984) et la corrélation basée sur le nombre de Reynolds ont la même tendance.

À partir des simulations de l'ensemble des expériences, on a effectué une analyse plus approfondie pour déterminer le comportement des coefficients de mélange, C_0 et K_{tr}. Ainsi, on peut conclure :

a) Le coefficient C_0 pour un écoulement latéral reste proche de l'unité ($1 \leq C_0 \leq 1.1$) ce résultat permet de valider l'hypothèse considérée dans plusieurs codes consistant à utiliser la valeur de 1.0 pour ce coefficient.

b) La variation du coefficient K_{tr} n'affecte pas les résultats de la simulation, cependant on a eu une expérience où la valeur de ce coefficient a tombée à 0.5 tandis que pour les autres cas reste autour de 1.0. À partir de cette observation, on propose de faire des analyses plus détaillées pour vérifier dans quelles conditions la valeur de ce coefficient peut affecter la distribution de l'écoulement.

7.2 Les recommandations

Pendant l'élaboration de ce projet, on a remarqué plusieurs aspects qui pourraient

être améliorés et d'autres qui ont besoin d'être analysés plus en détail.

Dans le code d'analyse de sous-canal, il y a plusieurs coefficients qui ont été considérés comme zéro dû à l'hypothèse qu'ils n'affectent pas la distribution de l'écoulement, par exemple la valeur du λ. Il sera très intéressant de faire une analyse plus détaillé sur la variation de cette variable et de son impact sur les résultats.

Le facteur de quantité de mouvement F_m fixé égal à un pour toutes les simulations. Ce paramètre peut être aussi un sujet de recherche.

Le facteur de sous relaxation ϖ peut être optimisé, car la valeur utilisée pour l'ensemble de simulations a été de 0.5.

Il sera très intéressant d'élaborer des expériences qui permettent d'étudier toute la transition de l'écoulement de début jusqu'à l'équilibre. De cette façon, on pourra analyser le comportement complet des coefficients.

On recommande d'utiliser ce code pour analyser des expériences dans des sous-canaux en position horizontale et d'étudier les ou changements à apporter sur la fonction objectif.

Dans le processus d'analyse on a remarqué que la robustesse de l'AG est proportionnelle à la vitesse de convergence du code. En effet, ces méthodes sont très solides, cependant le temps de calcul est plus grand par rapport aux autres codes qui utilisent de l'information supplémentaire, comme la dérivée de la fonction objectif.

Les nouvelles techniques et les avances informatiques permettront à ces nouvelles méthodes d'être améliorées et de diminuer le temps de calcul. L'utilisation des nombres réels au lieu des nombres binaires permettra une convergence plus rapide. Le couplage avec des réseaux neuronales, l'utilisation de techniques de traitement en parallèle sont des exemples qui peuvent perfectionner la méthode proposée dans cette étude.

Bibliographie

ARMAND, A.A., 1946, "Resistance to two phase flow in horizontal tubes (en Russe), Izv, VTI, Vol 15, pp. 16-23. Traduction à l'anglais, NLL M882, Boston Spa, Yorks: National Lending Library.

BAEHR, H.D., 1961, "Mollier I-X Diagramme fr Feuchte Luft", Springer-Verlag.

BELLIL, A., 1998, "Modélisation des écoulements diphasiques horizontaux dans des sous-canaux interconnectés", Thèse de doctorat en génie nucléaire, École Polytechnique de Montréal.

BEJAN, A., 1996, "Method of entropy generation minimisation, or modeling and optimization based on combined heat transfer and thermodynamics," Rév Gén Them. 35: 418-419. 637-646.

BEJAN, A., 2003, "How nature takes shape". In Site d'American Society of Mechanical Engineers. [en ligne]. http://www.memagazine.org/backissues /october97/features/nature/nature.html (page consultée le 12 septembre 2003).

CARVER, M.B., JUDD, R.A., KITELEY, J.C. et TAHIR, A., 1987,"The drift flux model in ASSERT subchannel code," Nuclear Journal of Canada, Vol 1, No 2, pp. 153-165.

DIPAMA, J., 2003, "Développement d'un algorithme génétique pour l'optimisation de réseaux d'échangeurs de chaleur", Mémoire de maîtrise en génie énergétique, École Polytechnique de Montréal.

GEN M. et CHENG R., 1997, "Genetic Algorithms and engineering Design", A Willey-Interscience Publication.

GONZALES-SANTALO, J.M., 1971, "Two-phase flow mixing in rod bundle subchannels," Thèse de doctorat en génie mécanique, Massachussets Institute of Technology.

HAUPT R.L. et HAUPT S.E., 1998, "Practical Genetic Algorithms," A Wiley-Interscience Publication.

ISHII, M., 1975,"Thermo-Fluid dynamic theory of two-phase flow," Eyrolles, France.

KAWAHARA, A., SATADOMI M., TOMINO T., and SATO Y., 2000, "Prediction of Turbulent mixing rates of both gas and liquid phases between adjacent subchannels in a two phase slug-churn flow," Nuclear Engineering and Design, Vol 202, pp.27-38.

KOLEV, N.I., 2002, "Multiphase flow dynamics 2, thermal and mechanical interactions," Springer-Verlag, Berlin, Germany.

LAHEY, R.T et MOODY, F.J., 1993, "The thermal hydraulics of a boiling water nuclear reactor," Nuclear American Society, La grange Park, Ill, USA.

MELANIE, M., 1996,"An introduction to genetic algorithms," MIT Press, Cambridge, Mass., USA.

MAYER, C.A., McCLINTOCK, R.B., SILVESTRI, G.J., ET SPENCER, Jr. R.C., 1967, "Thermodynamic and transport Properties of steam," ASME steam tables.

MICHALEWICZ, Z, 1996, "Genetic Algorithms + Data Structures = Evolution Programs," Springer-Verlag, Berlin, Germany.

PATANKAR, S.V., 1980, "Numerical heat transfer and fluid flow," Hemisphere Publishing Corporation.

ROUHANI, Z. 1974, "Two phase flow with application to nuclear reactor design problems", 61101 Nykping, Sweden : Von karman institute for fluid dynamics. 34p.

SADATOMI, M., KAWAHARA A., et SATO Y., 1996, "Prediction of the single-phase turbulent mixing rate between two parallel subchannels using a subchannel geometry factor," Nuclear Engineering and Design, Vol 162, pp.245-256.

SADATOMI, M., KAWAHARA A.,KANO, K. et SUNI, Y., 2004, "Single- and

two-phase turbulent mixing rate between adjacent subchannels in a vertical 2×3 rod array channel,"International Journal of Multiphase Flow, Vol 30, pp. 481-498

SHIRAI, H. et NINOKATA, H., 2001, "Prediction of the equilibrium two-phase flow distributions in inter-connected subchannel systems," Journal of nuclear science and technology, Vol 38, No 6, pp.379-387.

SHOUKRI, M., TAWFIK, H., et CHAN, A.M., 1984, "Two-phase redistribution in horizontal subchannel flow -turbulent mixing and gravity separation," Int. J. Multiphase Flow, Vol.10, No. 3, pp.357-369.

STEWART, C.W., WHEELER, C.L., CENA, R.J., McMONAGLE, C.A., CUTA, J.M. et TRENT, D.S., 1977 "COBRA-IV: The model and the method," Batelle, Pacific Northwest Laboratories.

TAHIR, M. et CARVER B., 1982, "Numerical Prediction of void drift using minimun entropy production principles," Transaction of the ANS, Winter meeting, Washinton D.C., Nov 14-18, pp.784-785.

TAPUCU, A. et GENÇAY S., 1980, "Experimental investigation of mass exchanges between two laterally interconnected two-phase flows," "Part 1 : Experimental results on vertical flow," École Polytechnique de Montréal, Rapport IGN-354.

TAPUCU, A., TEYSSEDOU, A., GENÇAY S. et TROCHE, N., 1988, "Experimental Study of the diversion cross-flow caused by subchannel blockages," Vol 2, two-phase flow, Projet 1378-1, EPRI NP-3459.

TEYSSEDOU, A., 1987, "Thermo-hydraulic behavior of interconnected subchannels with blockages," Thèse de doctorat en génie nucléaire, École Polytechnique de Montréal.

TEYSSEDOU, A., TAPUCU, A., CAMARERO, R., 1992, "Blocked flow subchannel simulation comparison with single-phase flow data," Journal of fluids engineering, Vol 114, pp. 205-213.

TODREAS, N. et KAZIMI, M., 1990, "Nuclear systems II," Hemisphere Publishing Corporation.

TYE, P., 1991, "An Investigation of the model for mass, momentum and energy interchange between interconnected subchannels as applied in the ASSERT computer code," Mémoire de maîtrise en génie nucléaire, École Polytechnique de Montréal.

TYE, P., TEYSSEDOU, A. et HERNU, P., 2000, "A comparation of ASSERT prediction using different void drift models against equilibrium void distribution experiments," École Polytechnique de Montréal, Rapport IGE-221-R1

WALLIS, G.B., 1969, "One-dimensional two-phase flow," McGraw-Hill.

ZUBER, N., et FINDLAY, J.A., 1965,"Average volumetric concentration in two-phase Flow systems," A.S.M.E

ANNEXE A

Géométrie des sous-canaux et relations constitutives.

Géométrie du système

INTERCONNEXION

Longueur.....*874 mm*

Largeur (c).....*1.5 mm*

Épaisseur (h).....*3.2 mm*

Début de l'interconnexion.....*277.0 mm*

Fin de l'interconnexion.....*114.95 mm*

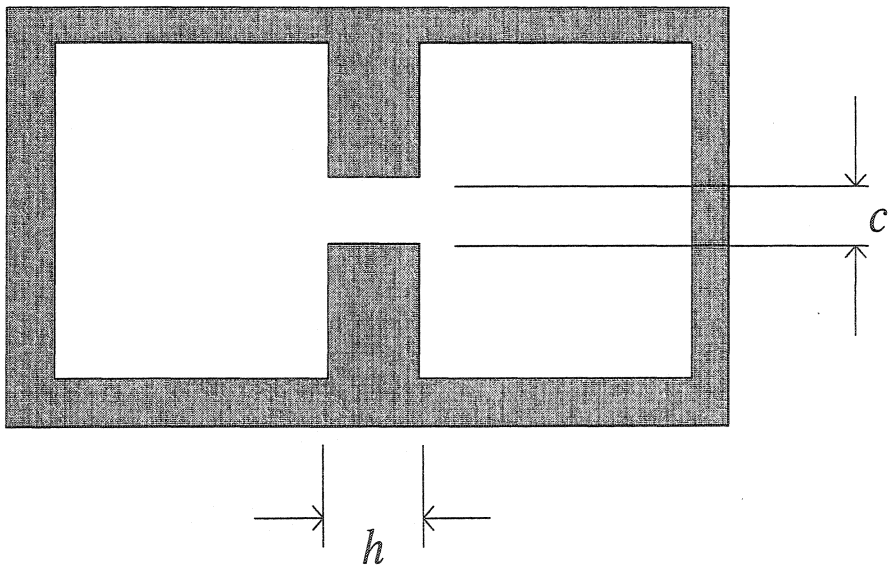


Figure a.1 : Représentation de la géométrie des sous-canaux

Relations constitutives

Pour fermer complètement le système d'équations, on a besoin d'introduire un certain nombres de relations constitutives qui vont nous permettre de résoudre le système de équations. Ces relations constitutives sont montrées dans cet annexe.

A.1 Corrélation du taux de vide en fonction du titre

Pour calculer le taux de vide en fonction du titre de l'écoulement on a utilisé des fonctions déterminées par la corrélation des données expérimentales (Tapucu *et al.,* 1988) Les fonctions utilisées ont la forme suivante :

Pour des titres inférieurs à 4×10^{-5} le taux de vide est pris égale à zéro. Pour des titres entre 4×10^{-5} et 0.001 la corrélation suivante a été utilisée :

$$\alpha(\hat{x}) = 2.241 \times 10^{8}\hat{x}^{3} - 6.3211 \times 10^{5}\hat{x}^{2} - 1.2781 \times 10^{-3}$$

Cependant, pour des titres supérieurs à cette intervalle, la fraction du taux de vide dépend du flux de masse, la fonction qui approxime les données expérimentales est :

$$\alpha(\hat{x}) = 2.54 \times \hat{x}^{0.2809}$$

A.2 Corrélation du taux de vide à l'équilibre

La différence du taux de vide à l'équilibre entre les sous-canaux adjacents doit être déterminée pour permettre d'évaluer le flux de masse du gaz $G_g^{'}$, donné par l'équation 3.24. La corrélation proposée; c'est le modèle de Lahey (Lahey et Moody, 1993) celle-ci est donnée par l'équation 3.25.

A.3 Facteur de frottement en simple phase

Le facteur de frottement en simple phase est obtenu à partir des essais effectués par Tapucu *et al.,* (1988), pour des écoulements en simple phase, en utilisant la même section d'essais. Les corrélations utilisées pour exprimer le facteur de frottement en simple phase dépend du type d'écoulement, de la géométrie des sous-canaux et du nombre de Reynolds.

Pour des nombres de Reynolds inférieurs à 10^3, la relation suivante est utilisée :

$$f = \frac{64}{Re}$$

Pour des nombres de Reynolds supérieurs à 10^3, la corrélation est donnée par:

$$f = 1.15 \times \left(0.0032 + \frac{0221}{Re^{0.0.237}}\right)$$

A.4 Multiplicateur diphasique

Le multiplicateur diphasique a été calculé en fonction du taux de vide. Le choix de la corrélation dépend du flux de masse de l'écoulement, de la position de sous-canaux et du type de la géométrie des sous-canaux. Pour les écoulements diphasiques verticaux, les fonctions qui permettent de calculer le multiplicateur des deux phases sont les suivantes :

Pour G=2000 kg/m^2s

$\Phi_l^2 = 1 + 0.9789\alpha + 16.094449\alpha^2 - 5163564\alpha^3 + 66.60886\alpha^4$

Pour G=2500 kg/m^2s

$\Phi_l^2 = 1 + 2.46102\alpha - 3.87197\alpha^2 + 14.96374\alpha^3$

Pour G=3000 kg/m^2s

$\Phi_l^2 = 1 + 2.5333421\alpha + 3.34927\alpha^2 - 11.13382\alpha^3$

Pour G=3500 kg/m^2s

$\Phi_l^2 = 1 + 2.80102\alpha + 4.53605\alpha^2 + 10.51165\alpha^3$

A.5 Propriétés de la phase liquide et gazeuse

A.5.1 Viscosité de la phase liquide

La viscosité de l'eau est calculée à partir de la corrélation donnée par Mayer *et al.*, (1967). Cette corrélation est exprimée comme suit :

$$\mu = (241.1X10^{[247.8(T-140)^{-1}]})\left[1 + \frac{(p-p_s)}{10^6}\Phi\right]$$

avec :

$\Phi =$ 1.0467(T-305),

$T =$ température $[K]$,

$P_s =$ pression de saturation $[bars]$,

$p =$ pression de l'eau $[bars]$.

Cette corrélation est valide à partir de la saturation jusqu'à la pression de 800 bars dans une plage de température variant entre 0°C et 300°C.

A.5.2 Densité de la phase liquide

Etant données que les expériences à simuler ont été obtenues pour une température proche de 20°C, la densité de l'eau est calculée par interpolation dans une table de densité. Cette table contient la densité de l'eau pour des températures qui varient entre 10° et 40°C.

A.5.3 Densité de la phase de gazeuse

Deux formules peuvent être utilisées pour calculer la densité du gaz en fonction de la température et de la pression. La première consiste à considérer l'air comme un gaz parfait, alors :

$$\rho_g = \frac{p}{RT}$$

où :

$R =$ constante des gazes parfaits, pour l'air 287 $[J/kgK]$,

$p =$ pression du gaz $[N/m^2]$,

$T =$ température du gaz $[K]$.

La deuxième formule est donné par Baehr (1961). Elle exprime la densité de l'air

humide en fonction de la température et de la pression comme suit :

$$\rho_g = \frac{(348.3X10^5 p - 0.1316 p_s)}{T}$$

Avec p_s la pression de saturation qui est approximée par :

$$p_s = 16.1446 - 0.7116t + 0.053126t^2$$

avec:

$p =$ pression de l'air $[Pa]$,

$T =$ température de l'air $[K]$,

$t =$ température de l'air $[°C]$.

ANNEXE B

ORGANISATION DU CODE

B.1 Description du code

Le programme est formé de deux parties principales, la première concerne l'analyse de sous-canaux, la deuxième l'algorithme génétique. En premier lieu on va à décrire l'organisation du code d'analyse de sous-canal.

La première partie du programme est utilisée pour lire les données d'entrées; le sous programme INPUT_DATA effectue cette tâche. La première partie concerne les caractéristiques géométriques des sous canaux, le deuxième groupe contient le nombre de noeuds de la grille, le facteur de sous relaxation et le nombre maximal des itérations. Le troisième groupe concerne les caractéristiques de l'écoulement. La dernière partie permet à l'utilisateur de choisir les corrélations ainsi que les différents coefficients, le facteur de quantité de mouvement, le coefficient de friction transversal, etc.

Une fois cette procédure finie le code appelle le sous-programme GRID, lequel discrétise les sous canaux, et aussi permet de localiser le début et la fin de l'interconnexion.

La procédure FLOW_INIT, manipule les conditions initiales du problème.

La procédure principale EQ_FLOW, permet la simulation de l'écoulement. L'écoulement latéral est calculé par le sous-programme TRANS_MOM_EQ; cette procédure est appelée deux fois. Le premier appel de ce sous-programme permet d'évaluer le vecteur contenant les flux de masse latéraux $[\tilde{G}']$ et les coefficients pour l'écoulement latéral nécessaire pour calculer les pressions et les flux de masse axiaux pour la prochaine itération.

Les autres sous-programmes utilisés sont :

VOID_EQ :

Retourne la différence du taux de vide à l'équilibre, le modèle de Lahey a été utilisé.

OUTPUT :

Sort le résultat de la simulation.

TRIDAG :

Résout le système de matrice tridiagonale.

Le module **PROPERTIES** contient les fonctions :

PHI :

Evalue le multiplicateur de deux phases.

VOID :

Calcule le taux de vide en fonction du titre de l'écoulement.

GDNSTY :

Calcule la densité de la phase gazeuse.

WDNSTY :

Calcule la viscosité de la phase liquide.

L'algorithme génétique est formé des sous-programmes suivants :

GA :

C'est le programme principal qui, contienne les caractéristiques de l'algorithme génétique, comme la taille du chromosome, nombre de générations, nombre de coefficients à optimiser.

Ce programme lit les données d'entrée de l'utilisateur, i.e., les probabilités de mutation et de croisement, le nombre des itérations, les intervalles de recherche des coefficients, le type de fonction objectif (EQM ou minimisation de la dissipation de l'énergie). Dans le cas où la méthode des EQM est choisie, l'utilisateur peut sélectionner une pondération pour chaque variable à simuler.

Ce programme appelle plusieurs sous-programmes comme :

PARENTS :

Génère la première génération.

DECODEF :

Ce sous-programme permet de décoder les chromosomes générés aux valeurs décimales pour pouvoir êtres évalués.

FUN_MIN :

Ce sous-programme appelle le code d'analyse de sous-canaux; il contient les coefficients de diffusion D_α et D_{eq} la valeur du coefficient de distribution C_o et le coefficient de frottement transversal K_{tr}.

Selon la fonction objectif sélectionnée par l'utilisateur, ce sous-programme appelle la fonction INTERPOLATION pour interpoler les valeurs expérimentales, ainsi que celles simulées et calculer par la suite l'EQM.

Si la fonction objectif basée sur la dissipation de l'énergie a été sélectionnée, le sous-programme DISSIPATION sera utilisée.

STATISTICS :

Ce sous-programme évalue les chromosomes et rejette les super chromosomes.

SELECTION_NAT :

Ce sous-programme permet de sélectionner les chromosomes qui vont passer à la génération suivante.

CROSSOVER :

Ce sous-programme effectue le croisement entre deux chromosomes, le rapport de croisement est déterminé par la probabilité de croisement P_c.

MUTATION :

Ce sous-programme permet de muter les chromosomes, selon la probabilité de mutation P_m.

GA_REPORT :

Ce sous-programme écrit un rapport contenant la solution finale de l'algorithme génétique.

ANNEXE C

CODE

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.--------------------------

!-------------------------- INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:    ARMANDO NAVA
!   DATE:       13-01-04
!



!   PROGRAMME GA.   Ce programme a comme but de optimiser les coefficients de mélange
!           dans des sous-canaux interconnectés et de géométrie carrée.
!           La méthode basée sur les Algorithmes génétiques a été utilisée
!           pour l'optimisation des paramètres du mélange.

!
! Sous programme CROSSOVER
!
! Ce sous programme a comme but de réaliser le croissement entre deux chromosomes, util
ise
! la méthode « sliccing crossover »

!-------------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-------------------------------------------------------------------------------------


! POP_SIZE      =   La taille de la population.
!                   Est une variable de type INTEGER.


! PROB_CROSS        =   La Probabilité de croissement.
!                       Est une variable de type DOUBLE PRECISION.

! RN1           =   Cette variable trouve un numéro aléatoire
!                   Est une variable dimensionnel de type INTEGER.

!-------------------------------------------------------------------------------------
! Les variables internes sont:
!-------------------------------------------------------------------------------------

! CROSSP        =   Ce variable est le point de coupure d'un chromosome.
! CROSSCHAIN        =   Ce variable détermine les deux chromosomes a croiser.
! CHROMOSOMEXY      =   Ce variable garde la population une fois croisée


!-------------------------------------------------------------------------------------
! Les variables de sorti sont:
!-------------------------------------------------------------------------------------
```

```
! POPULATION     =    La population.
!                      Est une variable dimensionnel de type INTEGER.


SUBROUTINE CROSSOVER
USE GA_VAR
USE VAR1
IMPLICIT NONE
INTEGER H,CRO,I,J
REAL::RN1(1000)


CROSSP=0
CROSSCHAIN=0

DO I=1,POP_SIZE


 CALL RANDOM1(RN1)
 CROSSP(I)                    =       CEILING((CHROM_SIZE-1)*RN1(I))

END DO


DO J=1,POP_SIZE

 CALL RANDOM1(RN1)
 CROSSCHAIN(J)                =       CEILING((POP_SIZE-1)*RN1(J))
  END DO


CHROMOSOMEXY = POPULATION

CRO=1

    IF (EVOL.EQ.GENERATION) CRO=POP_NUM+1
    CALL RANDOM1(RN1)
    DO WHILE(CRO.LE.POP_SIZE)
     IF (RN1(CRO).LE.PROB_CROSS) THEN
        DO H = CROSSP(CRO),CHROM_SIZE

         CHROMOSOMEXY(CRO,H)          =        POPULATION(CROSSCHAIN(CRO),H)
         CHROMOSOMEXY(CROSSCHAIN(CRO),H)     =        POPULATION(CRO,H)

         END DO

     END IF

        CRO=CRO+1

    END DO


POPULATION = CHROMOSOMEXY
END SUBROUTINE CROSSOVER
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------- INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!


!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnctés et de géometrie carée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.

!   Sous programme DECODEF

!   Ce sous programme a comme but:
!
!   Convertir les chaînes (chromosomes) binaires  vers ses valeurs decimals.

!-----------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-----------------------------------------------------------------------------------


! POP_SIZE       =   La taille de la population.
!                    Est une variable de type INTEGER.

! CO_COD         =   Cette variable est le chromosome que represent le paramètre Co (bin
aire)
! CO_INF
! CO_SUP         =   Les intervales de recherche pour le coefficient Co (COEFFICIENT DE
DISTRIBUTION À L'ÉQILIBRE)soit Co_INF le borne inferieure
!                    et CO_SUP soit le borne superieure.
!                    Sont variables de type DOUBLE PRECISION.


! KTR_COD        =   Cette variable est le chromosome que represent le paramètre Ktr (bi
naire)
! KTR_INF
! KTR_SUP        =   Les intervales de recherche pour le coefficient Ktr (COEFFICIENT DE
 FRICTION TRANSVERSAL)soit KTR_INF le borne inferieure
!                    et KTR_SUP soit le borne superieure.
!                    Sont variables de type DOUBLE PRECISION.

! DEQ_COD        =   Cette variable est le chromosome que represent le paramètre Deq (bi
naire)
! DEQ_INF
! DEQ_SUP        =   Les intervales de recherche pour le coefficient Deq (COEFFICIENT DE
 DIFFUSION À L'ÉQUILIBRE) soit DEQ_INF le borne inferieure
!                    et DEQ_SUP soit le borne superieure.
!                    Sont variables de type DOUBLE PRECISION.


! DALFA_COD      =   Cette variable est le chromosome que represent le paramètre Dalfa (
binaire)
! DALFA_INF
! DALFA_SUP      =   Les intervales de recherche pour le coefficient DALFA (COEFFICIENT
DE DIFFUSION) soit DALFA_INF le borne inferieure
!                    et DALFA_SUP soit le borne superieure.
!                    Sont variables de type DOUBLE PRECISION.


! LAMBDA_COD         =   Cette variable est le chromosome que represent le paramètre lam
bda (binaire)
! LAMBDA_INF
! LAMBDA_SUP     =   Les intervales de recherche pour le coefficient DALFA (COEFFICIENT
DE DIFFUSION) soit DALFA_INF le borne inferieure
```

```
!                   et DALFA_SUP soit le borne superieure.
!                   Sont variables de type DOUBLE PRECISION.

!-------------------------------------------------------------------------------
! Les variables de sorti sont:
!-------------------------------------------------------------------------------

! CO_DEC_VEC        =   Cette variable est le chromosome que represent le paramètre Co
(decimal)
!                       Est une variable de type DOUBLE PRECISION.


! DEQ_DEC_VEC       =   Cette variable est le chromosome que represent le paramètre Deq
 (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! DALFA_DEC_VEC     =   Cette variable est le chromosome que represent le paramètre Dal
fa (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! KTR_DEC_VEC       =   Cette variable est le chromosome que represent le paramètre Ktr
 (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! LAMBDA_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre lam
bda (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF1_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coe
ff1 (decimal)    Cette variable est le chromosome que represent le paramètre lambda (de
cimal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF2_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coe
ff2 (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF3_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coe
ff3 (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF3_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coe
ff4 (decimal)
!                       Est une variable de type DOUBLE PRECISION.

SUBROUTINE DECODEF
USE VAR1
USE GA_VAR
IMPLICIT NONE
INTEGER J,I


DO SELECT=1,9
 SELECT CASE(SELECT)
!-------------------------------------------------------------------------------
! DECODEF CO
!-------------------------------------------------------------------------------

  CASE (1)
  DO J=1,POP_SIZE
   SUMA=0.0
```

```
   DO I=1,COEFF_SIZE
    SUMA=SUMA+CO_COD(J,I)*2**(I-1)
   END DO

   CO_DEC_VEC(J)    =   CO_INF  +   (SUMA   *   ((CO_SUP-CO_INF)/((2**COEFF_SIZE)-1)))

  END DO

!-------------------------------------------------------------------------------
! DECODEF DEQ
!-------------------------------------------------------------------------------


CASE (2)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+DEQ_COD(J,I)*2**(I-1)
   END DO

    DEQ_DEC_VEC(J)  =   DEQ_INF +   (SUMA   *   ((DEQ_SUP-DEQ_INF)/((2**COEFF_SIZE)-1))
)

  END DO


!-------------------------------------------------------------------------------
! DECODEF DALFA
!-------------------------------------------------------------------------------


CASE (3)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+DALFA_COD(J,I)*2**(I-1)
   END DO

    DALFA_DEC_VEC(J)    =   DALFA_INF   +   (SUMA   *   ((DALFA_SUP-DALFA_INF)/((2**COE
FF_SIZE)-1)))

  END DO

!-------------------------------------------------------------------------------
! DECODEF KTR
!-------------------------------------------------------------------------------


CASE (4)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+KTR_COD(J,I)*2**(I-1)
   END DO

    KTR_DEC_VEC(J)  =   KTR_INF +   (SUMA   *   ((KTR_SUP-KTR_INF)/((2**COEFF_SIZE)-1))
)

  END DO
!-------------------------------------------------------------------------------
! DECODEF DVFEQ
!-------------------------------------------------------------------------------


 CASE (5)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
```

```
    SUMA=SUMA+LAMBDA_COD(J,I)*2**(I-1)
   END DO

    LAMBDA_DEC_VEC(J)   =   LAMBDA_INF  +   (SUMA   *   ((LAMBDA_SUP-LAMBDA_INF)/((2**C
OEFF_SIZE)-1)))

  END DO

CASE (6)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+COEFF1_COD(J,I)*2**(I-1)
   END DO

    COEFF1_DEC_VEC(J)   =   COEFF1_INF  +   (SUMA   *   ((COEFF1_SUP-COEFF1_INF)/((2**C
OEFF_SIZE)-1)))

  END DO

CASE (7)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+COEFF2_COD(J,I)*2**(I-1)
   END DO

    COEFF2_DEC_VEC(J)   =   COEFF2_INF  +   (SUMA   *   ((COEFF2_SUP-COEFF2_INF)/((2**C
OEFF_SIZE)-1)))

  END DO

CASE (8)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+COEFF3_COD(J,I)*2**(I-1)
   END DO

    COEFF3_DEC_VEC(J)   =   COEFF3_INF  +   (SUMA   *   ((COEFF3_SUP-COEFF3_INF)/((2**C
OEFF_SIZE)-1)))

  END DO

CASE (9)
  DO J=1,POP_SIZE
   SUMA=0.0
   DO I=1,COEFF_SIZE
    SUMA=SUMA+COEFF4_COD(J,I)*2**(I-1)
   END DO

    COEFF4_DEC_VEC(J)   =   COEFF4_INF  +   (SUMA   *   ((COEFF4_SUP-COEFF4_INF)/((2**C
OEFF_SIZE)-1)))

  END DO
  END SELECT
END DO
END SUBROUTINE DECODEF
```

```
!-------------------------- ECOLE POLYTECHNIQUE DE MONTREAL.-------------------------

!-------------------------- INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!
!                                      SOUS PROGRAMME
!**************************************DISSIPATION************************************
!
!   Ce sous programme est la fonction objectif principal, utilise les resultats de la si
mulation
!   du code d'analyse de sous-canal pour calculer l'énergie dissipée du système.
!

!KINETIC_X1L
!KINETIC_X2L
!KINETIC_X1G
!KINETIC_X2G
!POT_X1L
!POT_X2L
!POT_X1G
!POT_X2G
!PRESS_X1L
!PRESS_X2L
!PRESS_X1G
!PRESS_X2G




SUBROUTINE DISIPATION
USE GA_VAR
USE VAR1
IMPLICIT NONE
INTEGER::I,L,COUNT,CT,COMPT
DOUBLE PRECISION,DIMENSION(NPTS):: KINETIC_X1L,KINETIC_X2L,POT_X1L,POT_X2L,PRESS_X1L,PRE
SS_X2L,KINETIC_RAD
DOUBLE PRECISION,DIMENSION(NPTS):: KINETIC_X1G,KINETIC_X2G,POT_X1G,POT_X2G,PRESS_X1G,PRE
SS_X2G
DOUBLE PRECISION,DIMENSION(NPTS):: DELTA_KINETIC_X1L,DELTA_KINETIC_X2L,DELTA_KINETIC_X1G
,DELTA_KINETIC_X2G
DOUBLE PRECISION,DIMENSION(NPTS):: DELTA_POT_X1L,DELTA_POT_X2L,DELTA_POT_X1G,DELTA_POT_X
2G
DOUBLE PRECISION,DIMENSION(NPTS):: DELTA_PRESS_X1L, DELTA_PRESS_X2L, DELTA_PRESS_X1G, DE
LTA_PRESS_X2G

DOUBLE PRECISION:: POT_RAD,PRESS_RAD,LIQFLOW_CF_MOY,SUM_DP_RAD_BGN,RAD_ENERGY
DOUBLE PRECISION:: DLIQFLOW_DX,SUM_DISS_ENERGY


!-------------------------------------------------------------------------------------
-----------------
! TO AVOID LOCAL MINIMUMS OR TRIVIAL SOLUTION THE USE OF CONTRAINTS ARE USED
! THE CONVERGENCE IS ONE OF THE PARAMETERS
!-------------------------------------------------------------------------------------
-----------------

            IF(CONV_CRTRIA.GE. 1.0E-03) THEN
                 TOTAL_ERROR    =   -1
                 RETURN
            END IF
!-------------------------------------------------------------------------------------
```

```
------------------
! THE VOID FRACTION MUST BE BETWEEN 0 AND 1 OTHERWISE THE ANSWER IS PHYSICALLY UNREALLIS
TIC
!-------------------------------------------------------------------------------------
------------------


            DO I=1,N
                IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
                IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
            END DO
!-------------------------------------------------------------------------------------
------------------
! THE DIFF OF PRESSURES BEFORE THE CONNECTION SHOULD BE NOT EQUAL TO ZERO
! SUM_DP_BGN = THE SUMATION OF DIFFERENCE OF PRESSURES BEFORE INTERCONNECTION
!-------------------------------------------------------------------------------------
------------------

            SUM_DP_RAD_BGN= DABS(SUM(DP_RAD(1:I2))/1000)

            IF(SUM_DP_RAD_BGN.EQ.0.0) FLAG=.TRUE.

            IF(FLAG.EQ..TRUE.) THEN
                TOTAL_ERROR     =   -1
                RETURN
            END IF



            COUNT=0
            DO I=1,N

                IF (LIQFLOW(I,1).EQ.LIQFLOW(I,2) )THEN
                COUNT=COUNT+1

                    IF (COUNT.GT.2) THEN
                        TOTAL_ERROR =   -1
                        RETURN
                    END IF

                END IF

            END DO


 DELTA_KINETIC_X1L=0
 DELTA_KINETIC_X2L=0
 DELTA_KINETIC_X1G=0
 DELTA_KINETIC_X2G=0
 DELTA_POT_X1L=0
 DELTA_POT_X2L=0
 DELTA_POT_X1G=0
 DELTA_POT_X2G=0
 DELTA_PRESS_X1L=0
 DELTA_PRESS_X2L=0
 DELTA_PRESS_X1G=0
 DELTA_PRESS_X2G=0


KINETIC_X1L=0
KINETIC_X2L=0

POT_X1L=0
POT_X2L=0
```

```
PRESS_X1L=0
PRESS_X2L=0


KINETIC_X1G=0
KINETIC_X2G=0

POT_X1G=0
POT_X2G=0

PRESS_X1G=0
PRESS_X2G=0


KINETIC_RAD=0
POT_RAD=0
PRESS_RAD=0
RAD_ENERGY=0
SUM_DISS_ENERGY=0.0


DO I=NPTS-30,NPTS

!-------------------------------------------------------------------------------------------
-------------------------------------
!
!                                          **********Kinetic energy*********
!
!-------------------------------------------------------------------------------------------
-------------------------------------



IF(VOID_FRCTN(I,1).EQ.1.0 ) THEN
VOID_FRCTN(I,1)=0.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP T
HE PROGRAM DUE TO DIVISION BY ZERO
LIQFLOW(I,1)=0.0     ! THERE IS JUST AIR IN THE CHANNEL
END IF

IF(VOID_FRCTN(I,2).EQ.1.0 ) THEN
VOID_FRCTN(I,2)=0.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP T
HE PROGRAM DUE TO DIVISION BY ZERO
LIQFLOW(I,2)=0.0     ! THERE IS JUST AIR IN THE CHANNEL
END IF


IF(VOID_FRCTN(I,1).EQ.0.0 ) THEN
VOID_FRCTN(I,1)=1.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP T
HE PROGRAM DUE TO DIVISION BY ZERO
GASFLOW(I,1)=0.0     ! THERE IS JUST AIR IN THE CHANNEL
END IF

IF(VOID_FRCTN(I,2).EQ.0.0 ) THEN
VOID_FRCTN(I,2)=1.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP T
HE PROGRAM DUE TO DIVISION BY ZERO
GASFLOW(I,2)=0.0     ! THERE IS JUST AIR IN THE CHANNEL
END IF

IF(VOID_FRCTN_CF(I).EQ.1.0 ) THEN
VOID_FRCTN_CF(I)=0.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP
THE PROGRAM DUE TO DIVISION BY ZERO
LIQFLUX_CF(I)=0.0    ! THERE IS JUST AIR IN THE CHANNEL
END IF


IF(VOID_FRCTN_CF(I).EQ.0.0 ) THEN
```

```
VOID_FRCTN_CF(I)=1.0 ! BECAUSE (1-VOID_FRCTN(I,1)) IS IN THE DENOMINATOR AND COULD STOP
THE PROGRAM DUE TO DIVISION BY ZERO
GASFLUX_CF(I)=0.0    ! THERE IS JUST AIR IN THE CHANNEL
END IF



!-------------------------------------------------------------------------------------
-----------------------------------
! Liquid phase channel 1
!-------------------------------------------------------------------------------------
-----------------------------------

KINETIC_X1L(I)=0.5*(LIQFLOW(I,1) / 3600)     *    (    (LIQFLOW(I,1) / 3600)/((1-VOID_FRCTN
(I,1))*AREA(I,1)*LIQ_DNSTY) )**2&
&                            +    CT*((LIQFLUX_CF(I)*OMEGA(I) )     *    (0.5*    (LIQFLUX_CF(
I) / 3600)/((1-VOID_FRCTN_CF(I))*LIQ_DNSTY) )**2)


!-------------------------------------------------------------------------------------
-----------------------------------
! Liquid phase channel 2
!-------------------------------------------------------------------------------------
-----------------------------------

KINETIC_X2L(I)=0.5*(LIQFLOW(I,2) / 3600)     *    (    (LIQFLOW(I,2) / 3600)/((1-VOID_FRCTN
(I,2))*AREA(I,2)*LIQ_DNSTY) )**2&
&                            +  CT*((LIQFLUX_CF(I)*OMEGA(I) ) *    (0.5*    (LIQFLUX_CF(I) /
 3600)/((1-VOID_FRCTN_CF(I))*LIQ_DNSTY) )**2)




!-------------------------------------------------------------------------------------
-----------------------------------
! gas phase channel 1
!-------------------------------------------------------------------------------------
-----------------------------------
KINETIC_X1G(I)=0.5  *    (GASFLOW(I,1) / 3600)    *    (    (GASFLOW(I,1) / 3600)/((VOID_FRC
TN(I,1))*AREA(I,1)*GAS_DNSTY(I,1))   )**2&
&                            +  CT*((GASFLUX_CF(I)*OMEGA(I) ) *    (0.5*    (GASFLUX_CF(I) /
 3600)/((VOID_FRCTN_CF(I))*GAS_DNSTY_CF(I)) )**2)


!-------------------------------------------------------------------------------------
-----------------------------------
! gas phase channel 2
!-------------------------------------------------------------------------------------
-----------------------------------
KINETIC_X2G(I)= 0.5 *    (GASFLOW(I,2) / 3600)    *    (    (GASFLOW(I,2) / 3600)/((VOID_FRC
TN(I,2))*AREA(I,1)*GAS_DNSTY(I,2))   )**2&
&                            +  CT*((GASFLUX_CF(I)*OMEGA(I) ) *    (0.5*    (GASFLUX_CF(I) /
 3600)/((VOID_FRCTN_CF(I))*GAS_DNSTY_CF(I)) )**2)


!-------------------------------------------------------------------------------------
-----------------------------------
!
!                                        **********Potential energy*********
!
!-------------------------------------------------------------------------------------
-----------------------------------
```

```
!-------------------------------------------------------------------------------------------
-------------------------------------
! Liquid phase channel 1
!-------------------------------------------------------------------------------------------
-------------------------------------

POT_X1L(I)  =   (LIQFLOW(I,1) / 3600) * 9.8 * XPOSITION(I) + CT*(LIQFLUX_CF(I)*OMEGA(I)
) * 9.8 * XPOSITION(I)




!-------------------------------------------------------------------------------------------
-------------------------------------
! Liquid phase channel 2
!-------------------------------------------------------------------------------------------
-------------------------------------

POT_X2L(I)      =       (LIQFLOW(I,2) / 3600) * 9.8 * XPOSITION(I) + CT*(LIQFLUX_CF(I)*O
MEGA(I) ) * 9.8 * XPOSITION(I)



!-------------------------------------------------------------------------------------------
-------------------------------------
! gas phase channel 1
!-------------------------------------------------------------------------------------------
-------------------------------------

POT_X1G(I)  =   (GASFLOW(I,1) / 3600) * 9.8 * XPOSITION(I) + CT*(GASFLUX_CF(I)*OMEGA(I)
) * 9.8 * XPOSITION(I)




!-------------------------------------------------------------------------------------------
-------------------------------------
! gas phase channel 2
!-------------------------------------------------------------------------------------------
-------------------------------------

POT_X2G(I)      =   (GASFLOW(I,2) / 3600) * 9.8 * XPOSITION(I)+ CT*(GASFLUX_CF(I)*OMEGA(
I) ) * 9.8 * XPOSITION(I)


!-------------------------------------------------------------------------------------------
-------------------------------------
!
!                                       **********Energy by Pressure*********
!
!-------------------------------------------------------------------------------------------
-------------------------------------

!-------------------------------------------------------------------------------------------
-------------------------------------
! Liquid phase channel 1
!-------------------------------------------------------------------------------------------
-------------------------------------

PRESS_X1L(I)        =(LIQFLOW(I,1) / 3600)  *   PRSRE(I,1) / LIQ_DNSTY&
&                       +   CT*((LIQFLUX_CF(I)*OMEGA(I) ) * (DABS(PRSRE(I,1)-PRSRE(I
,2)) )  / LIQ_DNSTY)
```

```
!-------------------------------------------------------------------------------------------
-------------------------------------
! Liquid phase channel 2
!-------------------------------------------------------------------------------------------
-------------------------------------
PRESS_X2L(I)         =    (LIQFLOW(I,2) / 3600)    *    PRSRE(I,2) / LIQ_DNSTY&
&                           +    CT*((LIQFLUX_CF(I)*OMEGA(I) ) * (DABS(PRSRE(I,1)-PRSRE(I
,2)) )   / LIQ_DNSTY)


IF (GASFLUX_CF(I).GT.0.D0) THEN
    CT=1
    ELSE
    CT=-1
END IF

!-------------------------------------------------------------------------------------------
-------------------------------------
! gas phase channel 1
!-------------------------------------------------------------------------------------------
-------------------------------------
PRESS_X1G(I)         =    (GASFLOW(I,1) / 3600)    *    PRSRE(I,1) / GAS_DNSTY(I,1)&
&                           +    CT*((GASFLUX_CF(I)*OMEGA(I) ) * (DABS(PRSRE(I,1)-PRSRE(I
,2)) )   / GAS_DNSTY_CF(I))


!-------------------------------------------------------------------------------------------
-------------------------------------
! gas phase channel 2
!-------------------------------------------------------------------------------------------
-------------------------------------
PRESS_X2G(I)     =    (GASFLOW(I,2) / 3600)    *    PRSRE(I,2) / GAS_DNSTY(I,2)&
&                           +    CT*((GASFLUX_CF(I)*OMEGA(I) ) * (DABS(PRSRE(I,1)-PRSRE(I
,2)) )   / GAS_DNSTY_CF(I))

DELTA_KINETIC_X1L(I) = DABS(KINETIC_X1L(I)-KINETIC_X1L(I-1))
DELTA_KINETIC_X2L(I) = DABS(KINETIC_X2L(I)-KINETIC_X2L(I-1))
DELTA_KINETIC_X1G(I) = DABS(KINETIC_X1G(I)-KINETIC_X1G(I-1))
DELTA_KINETIC_X2G(I) = DABS(KINETIC_X2G(I)-KINETIC_X2G(I-1))
DELTA_POT_X1L(I)     = DABS(POT_X1L(I)-POT_X1L(I-1))
DELTA_POT_X2L(I)     = DABS(POT_X2L(I)-POT_X2L(I-1))
DELTA_POT_X1G(I)     = DABS(POT_X1G(I)-POT_X1G(I-1))
DELTA_POT_X2G(I)     = DABS(POT_X2G(I)-POT_X2G(I-1))
DELTA_PRESS_X1L(I)   = DABS(PRESS_X1L(I)-PRESS_X1L(I-1))
DELTA_PRESS_X2L(I)   = DABS(PRESS_X2L(I)-PRESS_X2L(I-1))
DELTA_PRESS_X1G(I)   = DABS(PRESS_X1G(I)-PRESS_X1G(I-1))
DELTA_PRESS_X2G(I)   = DABS(PRESS_X2G(I)-PRESS_X2G(I-1))


END DO




    DO I=i2,NPTS-1


        DELTA_KINETIC_X1L(I) = DABS(KINETIC_X1L(I)-KINETIC_X1L(I-1))
        DELTA_KINETIC_X2L(I) = DABS(KINETIC_X2L(I)-KINETIC_X2L(I-1))
```

```
        DELTA_KINETIC_X1G(I) = DABS(KINETIC_X1G(I)-KINETIC_X1G(I-1))
        DELTA_KINETIC_X2G(I) = DABS(KINETIC_X2G(I)-KINETIC_X2G(I-1))
        DELTA_POT_X1L(I)     = DABS(POT_X1L(I)-POT_X1L(I-1))
        DELTA_POT_X2L(I)     = DABS(POT_X2L(I)-POT_X2L(I-1))
        DELTA_POT_X1G(I)     = DABS(POT_X1G(I)-POT_X1G(I-1))
        DELTA_POT_X2G(I)     = DABS(POT_X2G(I)-POT_X2G(I-1))
        DELTA_PRESS_X1L(I)   = DABS(PRESS_X1L(I)-PRESS_X1L(I-1))
        DELTA_PRESS_X2L(I)   = DABS(PRESS_X2L(I)-PRESS_X2L(I-1))
        DELTA_PRESS_X1G(I)   = DABS(PRESS_X1G(I)-PRESS_X1G(I-1))
        DELTA_PRESS_X2G(I)   = DABS(PRESS_X2G(I)-PRESS_X2G(I-1))



    END DO

DO I=I2+1,NPTS

    SUM_DISS_ENERGY =  SUM_DISS_ENERGY +  (DELTA_KINETIC_X1L(I)+DELTA_KINETIC_X2L(I)+DEL
TA_KINETIC_X1G(I)+DELTA_KINETIC_X2G(I)+  &
       &                                   DELTA_POT_X1L(I)+DELTA_POT_X2L(I)+DELTA_POT_X
1G(I)+DELTA_POT_X2G(I)+ &
       &                                   DELTA_PRESS_X1L(I)+  DELTA_PRESS_X2L(I)+DELTA
_PRESS_X1G(I)+DELTA_PRESS_X2G(I))


END DO







DO I=NPTS-30,NPTS

 RAD_ENERGY =   RAD_ENERGY + DABS(DP_RAD(I))


END DO



!-------------------------------------------------------------------------------------
-------------------
!   AXIAL ENERGY BALANCE, KINEMATIC, POTENTIAL AND PRESSURE
!-------------------------------------------------------------------------------------
-------------------

        DISSIPATION_ENERGY(POP_NUM) = SUM_DISS_ENERGY
!-------------------------------------------------------------------------------------
-------------------
!   TRANSVERSAL ENERGY BALANCE 'PRESSURE'
!-------------------------------------------------------------------------------------
-------------------

        DISSIPATION_ENERGY_TRNSVS(POP_NUM)= RAD_ENERGY

        TOTAL_ERROR = DISSIPATION_ENERGY(POP_NUM)  + RAD_ENERGY

END SUBROUTINE DISIPATION
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.

!   Sous programme DP_RADIAL

!   Ce sous programme a comme but:
!
!   Calculer la somme de la différence de pression latéral.

!-----------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-----------------------------------------------------------------------------------

!DP_RAD          = C'est la différence de pressions latérales

!-----------------------------------------------------------------------------------
! Les variables d'sortie sont:
!-----------------------------------------------------------------------------------

!TOTAL_ERROR     = C'est la somme de la différence des pressions latérales.


SUBROUTINE DP_RADIAL
USE VAR1
USE GA_VAR
IMPLICIT NONE


INTEGER I,J,K
TOTAL_ERROR=0.0

    IF(CONV_CRTRIA.GE.1.0) THEN
     TOTAL_ERROR    =   -1
     RETURN
    END IF



        DO I=1,N
            IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
            IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
           END DO

    IF(FLAG.EQ..TRUE.) THEN
        TOTAL_ERROR     =   -1
        RETURN
    ELSE
            DO J=2,NPTS
            TOTAL_ERROR=TOTAL_ERROR+DABS(DP_RAD(J)/1000)
            END DO
    END IF


END SUBROUTINE DP_RADIAL
```

```
!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME FLOW_INIT.    Ce sous-programme a comme but d'initialiser le code.


!GASFLUX         =   le flux de la phase gazeuse.
!LIQFLUX         =   le flux de la phase liquide.
!GASFLUX_CF      =   le flux de la phase gazeuse latéral.
!LIQFLUX_CF      =   le flux de la phase liquide latéral
!TTLFLUX_CF      =   le flux total latéral.
!PRSRE           =   la pression
!PC              =   la pression corrigée.
!LIQFLOW         =   le débit du liquide axial
!GASFLOW         =   le débit axial
!GASFLUX_INP1    =   le débit/surface axial à l'entrée du canal 1
!GASFLUX_INP2    =   le débit/surface axial à l'entrée du canal 2
!TTLFLUX         =   le débit/surface total = gaz + liquide
!LIQFLUX         =   le débit/surface du liquide, axial
!GASFLUX         =   le débit/surface du gaz, axial
!QUALITY         =   la qualité
!VOID_FRCTN      =   le taux de vide
!PRSSRE_OUT1     =   la pression de sortie du canal 1
!PRSSRE_OUT2     =   la pression de sortie du canal 2
!GASFLUX_INP1    =   l'entrée du gaz du canal 1
!GASFLUX_INP2    =   l'entrée du gaz du canal 2

!-------------------------------------------------------------------------------------



SUBROUTINE FLOW_INIT
USE VAR1
USE PROPERTIES
LIQDNSTY=WDNSTY(TEMP)

GASFLUX      =0.0
LIQFLUX      =0.0
GASFLUX_CF   =0.0
LIQFLUX_CF   =0.0
TTLFLUX_CF   =0.0
PRSRE        =0.0
PC           =0.0
LIQFLOW      =0.0
GASFLOW      =0.0

DO I=1,NPTS+3
DWL(I)=0.0
DWG(I)=0.0



GASFLUX(I,1)=GASFLUX_INP1
LIQFLUX(I,1)=LIQFLUX_INP1
TTLFLUX(I,1)=LIQFLUX(I,1)+GASFLUX(I,1)
QUALITY(I,1)=GASFLUX(I,1)/TTLFLUX(I,1)
VOID_FRCTN(I,1)=VOID(QUALITY(I,1))

IF(I.EQ.1) THEN
CORRECTION_S1=DABS(VOID_FRCTN(1,1)-VOIDFRC_INP1)
END IF
```

```
GASFLUX_CF(I)=0.0
LIQFLUX_CF(I)=0.0
TTLFLUX_CF(I)=1.0D-6


!   CHANNEL 2

GASFLUX(I,2)=GASFLUX_INP2
LIQFLUX(I,2)=LIQFLUX_INP2
TTLFLUX(I,2)=LIQFLUX(I,2)+GASFLUX(I,2)
QUALITY(I,2)=GASFLUX(I,2)/TTLFLUX(I,2)
VOID_FRCTN(I,2)=VOID(QUALITY(I,2))

IF(I.EQ.1) THEN
CORRECTION_S2=DABS(VOID_FRCTN(1,2)-VOIDFRC_INP2)
END IF


PRSRE(I,1)=PRSSRE_OUT1
PRSRE(I,2)=PRSSRE_OUT2
PC(I,1)=0.0
PC(I,2)=0.0


VG_CF(I)=0.0
D_CF(I)=0.0
GAP_GL(I)=0.0
GAP_GG(I)=0.0
DIV_CROSS=0.0
TURB_VOID_DIFF=0.0
VOID_DRIFT=0.0
END DO
VOID_FRCTN(1:NPTS+3,1)   =    VOID_FRCTN(1:NPTS+3,1)+CORRECTION_S1
VOID_FRCTN(1:NPTS+3,2)   =    VOID_FRCTN(1:NPTS+3,2)-CORRECTION_S2

END SUBROUTINE FLOW_INIT
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME SUBCHANNEL.   Ce programme a comme but de faire l'analyse de
!                           sous canal et l'optimiser en utilisant des
!                           algorithmes génétiques.
!
!
!   Sous programme :     FLOW EQUATION.

!   Ce sous programme est la partie principale du code d'analyse de sous-canal, c'est-à-
dire; les système d'équations
!    de base du modèle sont résolues ici.
!
!   Les sous-programmes appelés sont:
!   FLOW_INIT        =   Ce sous programme initialise toutes les variables à zéro.
!   EQVOID           =   Ce sous programme calcul la valeur de la différence de taux de v
ide à l'équilibre.
!   TRIDAG           =   Utilisé pour résoudre une matrice tri diagonal
!   TRANS_MOM_EQ     =   Ce sous-programme résoudre l'équation de conservation de la quan
tité de mouvement transversale
!
!!



SUBROUTINE FLOW_EQ
USE VAR1
USE PROPERTIES
USE TRIDIAGONAL
IMPLICIT NONE
INTEGER ::I,J,L,K
DOUBLE PRECISION::TEST(1000),VAR_LIQFLOW


FLAG=.FALSE.
EP=1.0E-6
GRVTY=9.8
NN1=I2+1

LIQ_DNSTY=WDNSTY(TEMP)
VISCOCITY=VISCTY(TEMP)



CALL FLOW_INIT
FLUX_BG=(TTLFLUX(2,1)*AREA(2,1)+TTLFLUX(2,2)*AREA(2,2))/(AREA(2,1)+AREA(2,2))


   DO I=I2,N1
    GAP_GL(I)=0.5*(LIQFLUX(I,1)+LIQFLUX(I,2))
    GAP_GG(I)=0.5*(GASFLUX(I,1)+GASFLUX(I,2))
   END DO


!-------------------------------------------------------------------------------
!
! Début du cycle principal, le code a convergé si le critère de convergence est atteint
! ou le nombre des itérations maximal a ....
!
! CONV_CRTRIA       = critère de convergence
!                     Variable de double presicion
```

```
!
! EPSILON           = Convergence
!                     Variable de double presicion
!
! NUM_ITER          = Nombre de itération
!                     Variable INTEGER
!
! NIT_MAX           = Nombre maximal des itérations
!                     Variable INTEGER
!-------------------------------------------------------------------------------
NUM_ITER=0
CONV_CRTRIA=1.0D0
DO WHILE((CONV_CRTRIA.LT.EPSILON).OR.(NUM_ITER.LT.NIT_MAX))


!-------------------------------------------------------------------------------!
!
!****************************** Bilan de masse ************************************
!
!
!GASFLUX(i,j)=          Débit du gaz dans le noeud i et sous-canal j
!                       Variable type DOUBLE PRECISION
!
!LIQFLUX(i,j)=          Débit du liquide dans le noeud i et sous-canal j
!                       Variable type DOUBLE PRECISION
!
!TTLFLUX(i,j)=          Débit total dans le noeud i et sous-canal j
!                       Variable type DOUBLE PRECISION
!
!AREA(I,J)=             Section de passage axiale, I représente le noeud et J le sous-ca
nal
!                       Variable type DOUBLE PRECISION
!
!
!OMEGA(I+1)=            Section de passage transversale, I représente le noeud
!                       Variable type DOUBLE PRECISION
!
!MIX_DNSTY(I,J)=        Densité du mélange, I représente le noeud et J le sous-canal
!                       Variable type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------

NUM_ITER=1 + NUM_ITER

  DO J=1,2
    DO I=2,N1

     SELECT CASE (J)

     CASE (1)
     GASFLUX(I+1,J)=(AREA(I,J)*GASFLUX(I,J)-OMEGA(I+1)*GASFLUX_CF(I+1))/AREA(I+1,J)
     CASE (2)
     GASFLUX(I+1,J)=(AREA(I,J)*GASFLUX(I,J)+OMEGA(I+1)*GASFLUX_CF(I+1))/AREA(I+1,J)
     END SELECT

     LIQFLUX(I+1,J) = TTLFLUX(I+1,J) - GASFLUX(I+1,J)

    END DO
  END DO

   DO J=1,2
    GAS_DNSTY(2,J)=GDNSTY(PRSRE(2,J),TEMP,IN_EQ_DNSTY)
    MIX_DNSTY(2,J)=(VOID_FRCTN(2,J)*GAS_DNSTY(2,J)) + ((1.0-VOID_FRCTN(2,J))*LIQ_DNSTY)

     DO I=3,N2
      PRSRE_AVE=0.5*(PRSRE(I,J)+PRSRE(I+1,J))
      GAS_DNSTY(I,J)=GDNSTY(PRSRE_AVE,TEMP,IN_EQ_DNSTY)
```

```
      =DMIN1(DABS(GASFLUX(I,J)/TTLFLUX(I,J)),1.D0)
      VOID_FRCTN(I,J)=VOID(QUALITY(I,J))

      IF (J.EQ.1) VOID_FRCTN(I,J)=VOID_FRCTN(I,J)+CORRECTION_S1
      IF (J.EQ.2) VOID_FRCTN(I,J)=VOID_FRCTN(I,J)-CORRECTION_S2

      MIX_DNSTY(I,J)=(VOID_FRCTN(I,J)*GAS_DNSTY(I,J)) + (1.0-VOID_FRCTN(I,J))*LIQ_DNSTY

     END DO
    END DO

!-------------------------------------------------------------------------------------!
!                          Début du modèle de mélange
!                       "Basé sur le modèle du "drift flux"
!
!QUALITY_AVE     =   La moyenne de la qualité
!                    Variable de type DOUBLE PRECISION
!
!PRSRE_CF        =   Pression moyenne, représente la pression à l'interconnexion
!                    Variable de type DOUBLE PRECISION
!
!GAS_DNSTY_CF    =   Densité du gaz à l'interconnexion
!                    Variable de type DOUBLE PRECISION
!
!TTLFLUX_CF      =   le débit latéral total, (cross-flow)
!                    Variable de type DOUBLE PRECISION
!-------------------------------------------------------------------------------------

    QUALITY_AVE=(GASFLUX(NN1,1)+GASFLUX(NN1,2))/(TTLFLUX(NN1,1)+TTLFLUX(NN1,2))


    DO I=I2,N2
      PRSRE_CF=0.5*(PRSRE(I,1)+PRSRE(I,2))
      GAS_DNSTY_CF(I)=GDNSTY(PRSRE_CF,TEMP,IN_EQ_DNSTY)
    END DO

!-------------------------------------------------------------------------------------!
!
!  Ce boucle détermine qui est le canal donneur et qui le recevoir
!
!
!-------------------------------------------------------------------------------------


DO I=I2,N2
   IF (TTLFLUX_CF(I).GT.0.D0) THEN
    L=1
    CHNL_WDTH=CHNL_WDTH2
    ELSE
    L=2
    CHNL_WDTH=CHNL_WDTH1
   END IF

!-------------------------------------------------------------------------------------!
!VOID_FRCTN1     =   La fraction du vide du canal 1
!                =   Variable de type DOUBLE PRECISION
!
!VOID_FRCTN2     =   La fraction du vide du canal 2
!                =   Variable de type DOUBLE PRECISION
!
!VOID_FRCTN_AVE  =   La fraction du vide moyenne entre les deux canaux
!                =   Variable de type DOUBLE PRECISION
!
!VOID_FRCTN_CF   =   La fraction du vide à l'interconnexion
!                =   Variable de type DOUBLE PRECISION
!-------------------------------------------------------------------------------------

 VOID_FRCTN1=0.5*(VOID_FRCTN(I,1)+VOID_FRCTN(I-1,1))
```

```
 VOID_FRCTN2=0.5*(VOID_FRCTN(I,2)+VOID_FRCTN(I-1,2))
 VOID_FRCTN_AVE=0.5*(VOID_FRCTN1+VOID_FRCTN2)
 QUALITY_AVE=0.5*(QUALITY(I-1,L)+QUALITY(I,L))

 VOID_FRCTN_CF(I)=VOID(QUALITY_AVE)+(CORRECTION_S1+CORRECTION_S2)/2
!-------------------------------------------------------------------------------!
!SPFCL_VLCTY1       =   La vitesse superficiel du canal 1
!                       Variable de type DOUBLE PRECISION
!
!SPFCL_VLCTY2       =   La vitesse superficiel du canal 2
!                       Variable de type DOUBLE PRECISION
!
!
!MIX_SPFCL_VLCTY    =   La vitesse superficiel du mélange
!                       Variable de type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------

SPFCL_VLCTY1=GASFLUX(I,1)/GAS_DNSTY(I,1) + GASFLUX(I-1,1)/GAS_DNSTY(I-1,1) + (LIQFLUX(I,
1)+LIQFLUX(I-1,1))/LIQ_DNSTY
SPFCL_VLCTY2=GASFLUX(I,2)/GAS_DNSTY(I,2) + GASFLUX(I-1,2)/GAS_DNSTY(I-1,2) + (LIQFLUX(I,
2)+LIQFLUX(I-1,2))/LIQ_DNSTY
MIX_SPFCL_VLCTY=(SPFCL_VLCTY1+SPFCL_VLCTY2)/4.0

!-------------------------------------------------------------------------------!
!HYDRL_DIAM_AVE     =   La moyenne du diamètre hydrolique entre les deux canaux
!                       Variable de type DOUBLE PRECISION
!
!TURB_VOID_DIF      =   Coefficient de mèlange D_ALPHA
!                       Variable de type DOUBLE PRECISION
!
!CALL EQVOID        =   SOUS PROGRAMME EQVOID:  Calcule le taux de vide à l'équilibre se
lon
!                       le modèle de Lahey
!-------------------------------------------------------------------------------
HYDRL_DIAM_AVE=0.5*(HYDRLC_DIAM(I,1)+HYDRLC_DIAM(I,2))

TURB_VOID_DIF= 0.075*(VOID_FRCTN_AVE/0.45)**6 * MIX_SPFCL_VLCTY*HYDRL_DIAM_AVE

CALL EQVOID(IN_EQ_VOIDF,I,DVFEQ)
!-------------------------------------------------------------------------------!
!CO                 =   Coefficient de distribution
!                       Variable de type DOUBLE PRECISION
!-------------------------------------------------------------------------------
CO=1.0
CX=CO
EP=1.0D-6

!-------------------------------------------------------------------------------!
!
!VOID_FRCTN_12  =       La différence de taux de vide latéral entre les deux canaux
!                       Variable de type DOUBLE PRECISION

!DELTAY_AVE     =       La longueur de mélange
!                       Variable de type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------
VOID_FRCTN_12=VOID_FRCTN2-VOID_FRCTN1
DELTAY_AVE=0.5*(DELTAY(I)+DELTAY(I-1))


!-------------------------------------------------------------------------------!
!DIV_CROSS(I)       =   Mécanisme de mélange causé par la différence de presion laterale
!                       Variable de type DOUBLE PRECISION
!
!TURB_VOID_DIFF(I)  =   Mécanisme de mélange causé pour les fluctuations aléatoires du f
luide
!                       Variable de type DOUBLE PRECISION
```

```
!
!VOID_DRIFT(I)       =   Mécanisme de mélange causé par la tendance de l'écoulement à l'é
tat de l'équilibre
!                        Variable de type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------

    COEF=GAS_DNSTY_CF(I)*VOID_FRCTN_CF(I)/(1.0-CO*VOID_FRCTN_CF(I))
    DIV_CROSS(I)=COEF*CO*TTLFLUX_CF(I)/LIQ_DNSTY
    COEF=COEF/(VOID_FRCTN_AVE+EP)
    TURB_VOID_DIFF(I)=-COEF*TURB_VOID_DIF*VOID_FRCTN_12/DELTAY_AVE
    VOID_DRIFT(I)=COEF*TURB_VOID_DIF*DVFEQ/DELTAY_AVE

!-------------------------------------------------------------------------------
!
!GA                   =   Si GA=1 le code n'est pas optimise, le coefficients de mélange
!                         D_alpha et D_eq sont égaux et sont basés sur le modèle donne par
 l'équation
!                          (Teyssedou)
!
!
!                         Si GA=2 le code sera optimisé et l'algorithme génétique sera uti
lisé
!
!
!                         Variable de type INTEGER
!
!
!
!
!
!-------------------------------------------------------------------------------

IF (GA.EQ.2) THEN

    HYDRL_DIAM_CF    =       2*OMEGA(I) / (GAP_LNGTH+GAP_WDTH)
    RE_FLOWEQ(I)     =       HYDRL_DIAM_CF*DABS(TTLFLUX_CF(I))/VISCTY(TEMP)


    CO               =       CO_DEC_FLOWEQ
    KTR              =       KTR_DEC_FLOWEQ
    DEQ              =       DEQ_DEC_FLOWEQ*(VOID_FRCTN_AVE/0.45)**COEFF2_DEC_FLOWEQ * MI
X_SPFCL_VLCTY*HYDRL_DIAM_AVE
    DALFA            =       DALFA_DEC_FLOWEQ*(VOID_FRCTN_AVE/0.45)**COEFF3_DEC_FLOWEQ *
MIX_SPFCL_VLCTY*HYDRL_DIAM_AVE
!   X_LAMBDA         =       LAMBDA_DEC_FLOWEQ
    DALFA            =       DEQ


    COEF=GAS_DNSTY_CF(I)*VOID_FRCTN_CF(I)/(1.0-CO*VOID_FRCTN_CF(I))
    DIV_CROSS(I)=COEF*CO*TTLFLUX_CF(I)/LIQ_DNSTY
    COEF=COEF/(VOID_FRCTN_AVE+EP)
    TURB_VOID_DIFF(I)=-COEF*DALFA*VOID_FRCTN_12/DELTAY_AVE
    VOID_DRIFT(I)=COEF*DEQ*DVFEQ/DELTAY_AVE


END IF

!-------------------------------------------------------------------------------!
!
!GASFLUX_CF(I)        =   Débit du gaz transversal (du canal donneur à recevoir)
!                         Représente la somme des trois mécanismes de mélange
!                         Variable de type DOUBLE PRECISION
!
!LIQFLUX_CF(I)        =   Débit du liquide transversale (basé sur l'équation de la
!                         conservation de la masse)
!                         Variable de type DOUBLE PRECISION
!
```

```
!
!TTLFLUX_CF(I)       =   Débit du mélange transversale (cross-flow)
!                        Variable de type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------


GASFLUX_CF(I)=DIV_CROSS(I)+ TURB_VOID_DIFF(I)+VOID_DRIFT(I)

LIQFLUX_CF(I)=TTLFLUX_CF(I)-GASFLUX_CF(I)

MIX_DNSTY_CF = VOID_FRCTN_CF(I)*GAS_DNSTY_CF(I) + ((1.0-VOID_FRCTN_CF(I))*LIQ_DNSTY)
DR=TTLFLUX_CF(I)*DABS(TTLFLUX_CF(I))*(1.0-(GAP_WDTH/CHNL_WDTH)**2)/4.0/MIX_DNSTY_CF
FR(I,1)=-DMAX1(0.D0,DR)
FR(I+1,2)=-FR(I,1)
FR(I,2)=-DMAX1(0.0D0,-DR)
FR(I+1,1)=-FR(I,2)
END DO

!-------------------------------------------------------------------------------!
!               Équation de la conservation de la quantité de mouvement axiale
!
!   N_ITER_MOM      =   Nombre des itérations
!                       Variable de type INTEGER
!
!
!   CONT_ITER_MOM   =   Cette variable sert a compter le nombre des itérations
!                       réalisées
!                       Variable de type INTEGER
!
!
!-------------------------------------------------------------------------------
CONT_ITER_MOM =0
N_ITER_MOM=2
DO WHILE(N_ITER_MOM.GT. CONT_ITER_MOM)
CONT_ITER_MOM=CONT_ITER_MOM+1

DO J=1,2
  DO I=2,N2
   IF (DABS(GASFLUX(I,J)).LE. EP) THEN
   BGAS=0.0
   ELSE
   BGAS=QUALITY(I,J)/VOID_FRCTN(I,J)/GAS_DNSTY(I,J)
   END IF


BGAS_INI=QUALITY(I,1)/(VOID_FRCTN(I,1)+EP)/GAS_DNSTY(I,1)
BLIQ=(1.0-QUALITY(I,J))/(1.0-VOID_FRCTN(I,J))/LIQ_DNSTY
BLIQ_INI=(1.0-QUALITY(I,1))/(1.0-VOID_FRCTN(I,1))/LIQ_DNSTY


U(I)=0.5*AREA(I,J)*(BLIQ*LIQFLUX(I,J)+BGAS*GASFLUX(I,J))


IF(I.GE.NN1) THEN
 IF (TTLFLUX_CF(I).LT.0) THEN
  BLIQ=BLIQ_INI
  BGAS=BGAS_INI
 END IF

 SELECT CASE(J)
 CASE(1)
 V(I)=0.5*GAP_WDTH*DELTAX_CV(I)*(BLIQ*(LIQFLUX_CF(I)+LIQFLUX_CF(I+1))+BGAS*(GASFLUX_CF(I
)+GASFLUX_CF(I+1)))*F_MOM
```

```
 CASE(2)
 V(I)=-0.5*GAP_WDTH*DELTAX_CV(I)*(BLIQ*(LIQFLUX_CF(I)+LIQFLUX_CF(I+1))+BGAS*(GASFLUX_CF(
I)+GASFLUX_CF(I+1)))*F_MOM
 END SELECT


 D(I)=2.0*X_LAMBDA*GAP_WDTH*DELTAX_CV(I)*U(I)/AREA(I,J)

 ELSE
 V(I)=0.0
 D(I)=0.0
 END IF
 END DO

 U(N3)=U(N2)
!-------------------------------------------------------------------------------------!
!
!                          CONDITIONS LIMITÉS DE PRESSION
!
!
!FO                  =    Facteur de friction,
!                         Variable de type DOUBLE PRECISION
!
!PHI2                =    Multiplicateur à deux phases
!                         Variable de type DOUBLE PRECISION
!
!DP_GRVTY            =    Perte de pression par gravité
!                         Variable de type DOUBLE PRECISION
!
!DP_FRCTN            =    Perte de pression par friction
!                         Variable de type DOUBLE PRECISION
!
!
!-------------------------------------------------------------------------------------
 IF (NUM_ITER.LE.1) THEN
 FO=FCTION(IN_FRIC_CORR,TTLFLUX(2,J),VISCOCITY,GAP_LNGTH,HYDRLC_DIAM(2,J))
 PHI2=PHI(QUALITY(2,J),LIQFLUX(2,J))
 DP_GRVTY=MIX_DNSTY(2,J)*GRVTY*AREA(2,J)*DELTAX_CV(2)
 DP_FRCTN=FO*PHI2*TTLFLUX(2,J)*AREA(2,J)*DELTAX_CV(2)/ LIQ_DNSTY/HYDRLC_DIAM(2,J)



 AA(2,J)=2.0*U(2)
 B(J)=U(2)+DP_FRCTN
 C(J)=-AREA(2,J)**2
 DD(2,J)=-DP_GRVTY+0.5*DP_FRCTN*TTLFLUX(2,J)
 DP_INI(J)=(DP_GRVTY+0.5*DP_FRCTN*TTLFLUX(2,J))/AREA(2,J)

 END IF

 BB(2,J)=B(J)+U(3)
 VG(2)=(AA(2,J)*TTLFLUX(2,J)+DD(2,J))/BB(2,J)
 CP(2,J)=C(J)/BB(2,J)


 DO I=3,N2
      U1=U(I-1)+U(I)
      U2=U(I)+U(I+1)

      FO=FCTION(IN_FRIC_CORR,TTLFLUX(I,J),VISCOCITY,GAP_LNGTH,HYDRLC_DIAM(I,J))
      PHI2=PHI(QUALITY(I,J),LIQFLUX(I,J))
      DP_GRVTY=GRVTY*MIX_DNSTY(I,J)*AREA(I,J)*DELTAX_CV(I)
      DP_FRCTN=FO*PHI2*AREA(I,J)*DELTAX_CV(I)/LIQ_DNSTY/HYDRLC_DIAM(I,J)


      AA(I,J)=DMAX1(U1,0.D0)
      CC(I,J)=DMAX1(-U2,0.D0)
```

```
      V12=D(I)+DMAX1(V(I),0.D0)
      BB(I,J)=AA(I,J)+CC(I,J)+U2-U1+DP_FRCTN*DABS(TTLFLUX(I,J))+V12
      DD(I,J)=-DP_GRVTY+0.5*DP_FRCTN*TTLFLUX(I,J)**2+(V12-V(I))*TTLFLUX(I,3-J)
      DD(I,J)=DD(I,J)+AREA(I,J)
      VG(I)=(AA(I,J)*TTLFLUX(I-1,J)+CC(I,J)*TTLFLUX(I+1,J)+DD(I,J))/BB(I,J)
      CP(I,J)=-(AREA(I,J)*AREA(I,J))/BB(I,J)
      AP(I,J)=CP(I-1,J)
      BP1(I,J)=-AP(I,J)-CP(I,J)
      BP(I,J)=BP1(I,J)+OMEGA(I)*D_CF(I)

        SELECT CASE (J)
         CASE(1)
           DP(I,J)=(AREA(I-1,J)*VG(I-1)-AREA(I,J)*VG(I))+OMEGA(I)*(D_CF(I)*PRSRE(I,J)-VG
_CF(I))
         CASE(2)
           DP(I,J)=(AREA(I-1,J)*VG(I-1)-AREA(I,J)*VG(I))+OMEGA(I)*(D_CF(I)*PRSRE(I,J)+VG
_CF(I))
        END SELECT


END DO
BP(3,J)=BP(3,J)+AP(3,J)
DP(3,J)=DP(3,J)-AP(3,J)*DP_INI(J)
DP(N2,J)=DP(N2,J)-CP(N2,J)*PRSRE(N3,J)


CALL TRIDAG(3,N2,AP,BP,CP,DP,PRSRE,J)
PRSRE(2,J)=PRSRE(3,J)+DP_INI(J)
PRSRE(1,J)=PRSRE(2,J)


!-------------------------------------------------------------------------------!
!
!                            DÉBIT AXIAL
!
!
!-------------------------------------------------------------------------------


 DO I=3,N2
  AA(I,J)=-RELAX*AA(I,J)
  CC(I,J)=-RELAX*CC(I,J)
  DD(I,J)=(1.0-RELAX)*BB(I,J)*TTLFLUX(I,J)+RELAX*(DD(I,J)+(AREA(I,J)+F(I,J))* (PRSRE(I,J
)-PRSRE(I+1,J)))
 END DO
  DD(3,J)=DD(3,J)-AA(3,J)*TTLFLUX(2,J)


  CALL TRIDAG(3,N2,AA,BB,CC,DD,TTLFLUX,J)
  TTLFLUX(N3,J)=TTLFLUX(N2,J)



 END DO
END DO


!-------------------------------------------------------------------------------!
!
!                              ÉQUATION DE LA CONSERVATION DE LA
!                              QUANTITÉ DE MOUVEMENT TRANSVERSALE

! TRANSV_MOM_EQ      =    Sous programme pour résoudre l'équation
!                         de la conservation de la quantité de mouvement
!                         transversale
!                         Variable de type DOUBLE PRECISION
```

```
!
!
! NAUX              =   Nombre des itérations
!                       Variable de type INTEGER
!
!-------------------------------------------------------------------------------------



   ICOEFF=0


   CALL TRANSV_MOM_EQ


   NAUX=0
   DO WHILE (NAUX.LT.30)
   NAUX= NAUX+1


    DO J=1,2
     DO I=3,N2
     BP(I,J)=BP1(I,J)+OMEGA(I)*D_CF(I)
     SELECT CASE(J)
     CASE(1)
     DP(I,J)=AREA(I-1,J)*TTLFLUX(I-1,J)-AREA(I,J)*TTLFLUX(I,J)-OMEGA(I)*(TTLFLUX_CF(I)-P
C(I,3-J)*D_CF(I))
     CASE(2)
     DP(I,J)=AREA(I-1,J)*TTLFLUX(I-1,J)-AREA(I,J)*TTLFLUX(I,J)-OMEGA(I)*(-1*TTLFLUX_CF(I
)-PC(I,3-J)*D_CF(I))
     END SELECT


    END DO

     BP(3,J)=BP(3,J)+AP(3,J)
     CALL TRIDAG(3,N2,AP,BP,CP,DP,PC,J)

   END DO


  DO J=1,2
     DO I=3,N2
      DELG=(AREA(I,J)+F(I,J))*(PC(I,J)-PC(I+1,J))/BB(I,J)
      TTLFLUX(I,J)=TTLFLUX(I,J)+DELG
      LIQFLUX(I,J)=TTLFLUX(I,J)-GASFLUX(I,J)
     END DO
    LIQFLUX(N3,J)=LIQFLUX(N2,J)
    TTLFLUX(N3,J)=TTLFLUX(N2,J)
  END DO



  DO I=NN1,N2
   TTLFLUX_CF(I)=TTLFLUX_CF(I)+D_CF(I)*(PC(I,1)-PC(I,2) )

   LIQFLUX_CF(I)=TTLFLUX_CF(I)-GASFLUX_CF(I)

  END DO


END DO
!-------------------------------------------------------------------------------------!
!
!
!BALANCE    =           Convergence du code basé sur la conservation de la masse
```

```
!                       Variable de type DOUBLE PRECISION
!
!CONV_CRTRIA=           Critère de convergence
!                       Variable de type DOUBLE PRECISION
!
!-------------------------------------------------------------------------------

BALANCE=0.0
 DO J=1,2
  DO I=3,N2
   SELECT CASE(J)
     CASE(1)
  DMASS=TTLFLUX(I,J)*AREA(I,J)-TTLFLUX(I-1,J)*AREA(I-1,J)+OMEGA(I)*TTLFLUX_CF(I)
     CASE(2)
  DMASS=TTLFLUX(I,J)*AREA(I,J)-TTLFLUX(I-1,J)*AREA(I-1,J)-OMEGA(I)*TTLFLUX_CF(I)
     END SELECT


  BALANCE=BALANCE+DMASS**2
  END DO
 END DO



            IF(BALANCE.GT.1D15) THEN
                BALANCE=0.0
                RETURN

            END IF

            IF(BALANCE.lt.0.0) THEN
                BALANCE=0.0
                RETURN
            END IF


    IF ((BALANCE.GE.0.0).AND.(BALANCE.LT.1D150)) THEN
        CONV_CRTRIA=DSQRT(BALANCE)
    ELSE
        CONV_CRTRIA=1.0
    END IF
!-------------------------------------------------------------------------------!
!
!   Mise à jour de l'équation de la conservation de la quantité de mouvement
!   transversale avec les valeurs corrigées de l'écoulement latéral
!
!-------------------------------------------------------------------------------!

        IF(CONV_CRTRIA.GT.EPSILON) THEN
            ICOEFF=1
            CALL TRANSV_MOM_EQ
        END IF


  IF(CONV_CRTRIA.LE.EPSILON) THEN
  EXIT
  END IF

  END DO


          DO I=1,N
           IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
           IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) FLAG=.TRUE.
          END DO

           IF (CONV_CRTRIA.GE.1.0) FLAG=.TRUE.
```

```
!-------------------------------------------------------------------------------!
!   Conversion des dimensions
!   Kg/m2/s ------> kG/h/s
!
!
!
!-------------------------------------------------------------------------------

  DO J=1,2
   DO I=2,N2
    CONV_FACT=3600*AREA(I,J)
    LIQFLOW(I,J)=LIQFLUX(I,J)*CONV_FACT
    GASFLOW(I,J)=GASFLUX(I,J)*CONV_FACT
   END DO
  END DO
!-------------------------------------------------------------------------------!
!   Intégral du transfert de la masse latéral
!-------------------------------------------------------------------------------


SUM_LIQ=0.0
SUM_GAS=0.0

DO I=3,N2
 SUM_LIQ=SUM_LIQ+LIQFLUX_CF(I)*OMEGA(I)*3600
 SUM_GAS=SUM_GAS+GASFLUX_CF(I)*OMEGA(I)*3600
 DWL(I)=SUM_LIQ
 DWG(I)=SUM_GAS
END DO

!-------------------------------------------------------------------------------!
!   Perte de pression et la pression latérale
!
!
!
!-------------------------------------------------------------------------------


SW=0.0
IF((GASFLUX(2,1).EQ.0.0).AND.(GASFLUX(2,2).EQ.0.0)) THEN
SW=1.0
END IF

DO J=1,2
 DO I=2,N
  DPFT(I,J)=PRSRE(I,J)-SW*GRVTY*MIX_DNSTY(I,J)*(TTL_LNGTH-XPOSITION(I)+0.5*DELTAX(I))-PR
SRE(N2,J)
  IF (I.EQ.N) THEN
    DPFT(I,J)=PRSRE(N2,J)

  END IF

 END DO
END DO

DO I=2,N2
 DP_RAD(I)=PRSRE(I,1)-PRSRE(I,2)
END DO

I4=I2+5
DO WHILE (I4.LT.NPTS)

        VAR_LIQFLOW= DABS(LIQFLOW(I4+1,1)-LIQFLOW(I4,1))
    I4=I4+1
    IF (VAR_LIQFLOW.LE.0.0001) THEN
        EXIT
```

```
    END IF
END DO

DO I=2,NPTS
    IF (DABS((LIQFLOW(I,1)+LIQFLOW(I,2)+GASFLOW(I,1)+GASFLOW(I,2))-(LIQFLOW(2,1)+LIQFLOW
(2,2)+GASFLOW(2,1)+GASFLOW(2,2))).GT.5.0) THEN
        FLAG=.TRUE.
        EXIT
    END IF
END DO


13 FORMAT (5(G14.7))
END SUBROUTINE FLOW_EQ
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.

!   Sous programme FUN_MIN

!   Ce sous programme a comme but:
!
!   Appeler le sous programme contenant la fonction objective sélectionnée par
!   l'utilisateur.

!   Ce sous programme est l'interphase entre l'algorithme génétique et le code
!   d'analyse de sous canal

!-----------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-----------------------------------------------------------------------------------
!FUNC_OBJ       = Cette variable détermine la fonction objective
!               si c'est = 1; la fonction objective est basée sur les moindres carrées
!               si c'est = 2; la fonction objective est basée sur la minimisation de la
dissipation de la
!               énergie.


!-----------------------------------------------------------------------------------
! Les variables de sortie sont:
!-----------------------------------------------------------------------------------

!   CO_DEC_FLOWEQ           =       La valeur de Co en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   KTR_DEC_FLOWEQ          =       La valeur de Ktr en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   DEQ_DEC_FLOWEQ          =       La valeur de Deq en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   DALFA_DEC_FLOWEQ        =       La valeur de Dalfa en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   LAMBDA_DEC_FLOWEQ       =       La valeur de Lambda en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   COEFF1_DEC_FLOWEQ       =       La valeur de Coeff1 en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   COEFF2_DEC_FLOWEQ       =       La valeur de Coeff2 en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   COEFF3_DEC_FLOWEQ       =       La valeur de Coeff3 en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

!   COEFF4_DEC_FLOWEQ       =       La valeur de Coeff4 en système décimal.
!                                   Cette variable est de type DOUBLE PRECISION.

! Ce sous programme appel les sous programmes suivantes
!
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:    ARMANDO NAVA
!   DATE:           13-01-04
!

!   PROGRAMME GA.   Ce programme a comme but de optimiser les coefficients de mélange
!                   dans des sous-canaux interconnectés et de géométrie carrée.
!                   La méthode basée sur les Algorithmes génétiques a été utilisée
!                   pour l'optimisation des paramètres du mélange.
!

! Sous programme GENETIC_ALGORITHM
!
! Ce sous programme a comme but:
!


! Lecture de fichier de données d'entrée.
! Générer la première génération.
! Variables de la taille de la population, la taille du coefficient à optimiser
! et la taille du chromosome.


! Les variables d'entrée sont:

! POP_SIZE      =   La taille de la population.
!                   Est une variable de type INTEGER.
!
! COEFF_SIZE    =   Le nombre de bits contenant un coefficient.
!                   Est une variable de type INTEGER.
!
! CHROM_SIZE    =   La taille du chromosome le nombre de coefficients fois la taille du
 coefficient
!                   Est une variable de type INTEGER.

! PROB_CROSS    =   La Probabilité de croisement.
!                   Est une variable de type DOUBLE PRECISION.


! GENERATION    =   Le nombre de générations à réaliser




! W1    =   Le facteur de pondération pour le transfert du liquide dans le sous canal 1
!           Est une variable de type DOUBLE PRECISION.

! W2    =   Le facteur de pondération pour le transfert du liquide dans le sous canal 2
!           Est une variable de type DOUBLE PRECISION.

! W3    =   Le facteur de pondération pour la pression absolue dans le sous canal 1
!           Est une variable de type DOUBLE PRECISION.

! W4    =   Le facteur de pondération pour la pression absolue dans le sous canal 2
!           Est une variable de type DOUBLE PRECISION.

! W5    =   Le facteur de pondération pour la perte de pression dans le sous canal 1
!           Est une variable de type DOUBLE PRECISION.

! W6    =   Le facteur de pondération pour la perte de pression dans le sous canal 2
!           Est une variable de type DOUBLE PRECISION.

! W7    =   Le facteur de pondération pour le taux de vide dans le sous canal 1
```

```
!           Est une variable de type DOUBLE PRECISION.

! W8     =   Le facteur de pondération pour le taux de vide dans le sous canal 2
!           Est une variable de type DOUBLE PRECISION.

! W9     =   Le facteur de pondération pour le transfert du gaz dans le sous canal 1
!           Est une variable de type DOUBLE PRECISION.

! W10    =   Le facteur de pondération pour le transfert du gaz dans le sous canal 2
!           Est une variable de type DOUBLE PRECISION.

! W11    =   Le facteur de pondération pour la différence de pression lateral
!           Est une variable de type DOUBLE PRECISION.



!FUNC_OBJ = Sélection du type de fonction Objective.




!-------------------------------------------------------------------------------------
! Le sous-programme appel les sous programmes suivants:
!-------------------------------------------------------------------------------------
!   PARENTS          =   Sous-programme pour générer la première population.
!   STATISTICS       =   Sous programme pour calculer les statistics de la population.
!   GA_REPORT        =   Sous-programme élabore un rapport.
!   SELECTION_NAT    =   Sous-programme sélectionne les meilleurs chromosomes.
!   CROSSOVER        =   Ce sous-programme  croisement.
!   MUTATION         =   Ce sous-programme a comme but de réaliser la mutation.




SUBROUTINE GENETIC_ALGORITHM
USE VAR1
USE GA_VAR
IMPLICIT NONE
INTEGER J,I
DOUBLE PRECISION::DP_DZ_1,WG_1,WL_1,DP_DZ_2,WG_2,WL_2
 POP_SIZE = 45
 COEFF_SIZE=10
 CHROM_SIZE=9*COEFF_SIZE



 !-------------------------------------------------------------------------------------
 CALL ALLOCATE_GA (POP_SIZE,CHROM_SIZE,COEFF_SIZE)

    DO J=1,POP_SIZE
     CALL FIRST_GENE
     POPULATION(J,1:CHROM_SIZE)     =   OFFSPRING_VEC(1:CHROM_SIZE)
    END DO



IF(OPENFILE.EQ.1) THEN
OPEN (UNIT=5, FILE=GA_WEIGHT_INPUT_DAT, ACTION='READ', STATUS='OLD')
```

```
READ(5,*)
READ(5,*)
READ(5,*) OPTIONMUT,PM
READ(5,*)
READ(5,*) PROB_CROSS
READ(5,*)
READ(5,*) CO_INF,CO_SUP
READ(5,*)
READ(5,*) KTR_INF,KTR_SUP
READ(5,*)
READ(5,*) DEQ_INF,DEQ_SUP
READ(5,*)
READ(5,*) DALFA_INF,DALFA_SUP
READ(5,*)
READ(5,*) LAMBDA_INF,LAMBDA_SUP
READ(5,*)
READ(5,*) COEFF1_INF,COEFF1_SUP
READ(5,*)
READ(5,*) COEFF2_INF,COEFF2_SUP
READ(5,*)
READ(5,*) COEFF3_INF,COEFF3_SUP
READ(5,*)
READ(5,*) COEFF4_INF,COEFF4_SUP
READ(5,*)
READ(5,*) GENERATION
READ(5,*)
READ(5,*)
READ(5,*) W1    !LIQUID FLOW SUBCHANNEL 1
READ(5,*)
READ(5,*) W2    !LIQUID FLOW SUBCHANNEL 2
READ(5,*)
READ(5,*) W3    !ABS PRESSURE SUBCHANNEL 1
READ(5,*)
READ(5,*) W4    !ABS PRESSURE SUBCHANNEL 2
READ(5,*)
READ(5,*) W5    !LOSS OF CHARGE SUBCHANNEL 1
READ(5,*)
READ(5,*) W6    !LOSS OF CHARGE SUBCHANNEL 2
READ(5,*)
READ(5,*) W7    !VOID FRACTION SUBCHANNEL 1
READ(5,*)
READ(5,*) W8    !VOID FRACTION SUBCHANNEL 2
READ(5,*)
READ(5,*) W9    !GAS FLOW SUBCHANNEL 1
READ(5,*)
READ(5,*) W10   !GAS FLOW SUBCHANNEL 2
READ(5,*)
READ(5,*) W11   !RADIAL PRESSURE DROP
READ(5,*)
READ(5,*) FUNC_OBJ



CLOSE (UNIT=5)
OPENFILE=2
ELSE
END IF



DO EVOL=1,GENERATION

    CALL PARENTS
    CALL STATISTICS
    CALL GA_REPORT
```

```
    CALL SELECTION_NAT
    CALL CROSSOVER
    CALL MUTATION

    WRITE(*,'(I3)') EVOL
END DO

1000 FORMAT(30(1X, I1))
END SUBROUTINE GENETIC_ALGORITHM
```

```
!------------------------- ECOLE POLYTECHNIQUE DE MONTREAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnctés et de géometrie carée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.

!   Sous programme GA_REPORT

!   Ce sous programme a comme but:
!
!   Faire un rapport contenant les résultats de l'optimisation

!-----------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-----------------------------------------------------------------------------------


! POP_SIZE      =   La taille de la population.
!                   Est une variable de type INTEGER.


!-----------------------------------------------------------------------------------
! Les variables de sorti sont:
!-----------------------------------------------------------------------------------

! CO_DEC_VEC        =   Cette variable est le chromosome que represent le paramètre Co (
decimal)
!                       Est une variable de type DOUBLE PRECISION.


! DEQ_DEC_VEC       =   Cette variable est le chromosome que represent le paramètre Deq
(decimal)
!                       Est une variable de type DOUBLE PRECISION.


! DALFA_DEC_VEC     =   Cette variable est le chromosome que represent le paramètre Dalf
a (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! KTR_DEC_VEC       =   Cette variable est le chromosome que represent le paramètre Ktr
(decimal)
!                       Est une variable de type DOUBLE PRECISION.


! LAMBDA_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre lamb
da (decimal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF1_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coef
f1 (decimal)    Cette variable est le chromosome que represent le paramètre lambda (deci
mal)
!                       Est une variable de type DOUBLE PRECISION.


! COEFF2_DEC_VEC    =   Cette variable est le chromosome que represent le paramètre coef
f2 (decimal)
!                       Est une variable de type DOUBLE PRECISION.
```

```
! COEFF3_DEC_VEC     =   Cette variable est le chromosome que represent le paramètre coef
f3 (decimal)
!                        Est une variable de type DOUBLE PRECISION.


! COEFF3_DEC_VEC     =   Cette variable est le chromosome que represent le paramètre coef
f4 (decimal)
!                        Est une variable de type DOUBLE PRECISION.

! CO                 =   Cette variable est le résultat final de l'optimisation, represen
te
!                        La valeur de Co
!                        Est une variable de type DOUBLE PRECISION.

! DEQ                =   Cette variable est le résultat final de l'optimisation, represen
te
!                        La valeur de Deq (coefficient de diffusion à l'équilibre)
!                        Est une variable de type DOUBLE PRECISION.

! DALFA              =   Cette variable est le résultat final de l'optimisation, represen
te
!                        La valeur de Dalfa (coefficient de diffusion )
!                        Est une variable de type DOUBLE PRECISION.

! KTR                =   Cette variable est le résultat final de l'optimisation, represen
te
!                        La valeur de Ktr (coefficient de fricction transversal )
!                        Est une variable de type DOUBLE PRECISION.

! XP                 =   Cette variable est le point de coupure du chromosome
!                        Est une variable de type INTEGER.

! XC                 =   Cette variable represent le chromosome a croisser
!                        Est une variable de type INTEGER.

! ACTUAL COUNT       =   Cette variable represent le numero du chromosome.
!                        Est une variable de type INTEGER.

! CONV_FLOW          =   Cette variable represent la convergence du code d'analyse de sou
s canal.
!                        Est une variable de type DOUBLE PRECISION.

! TOTAL_ERROR        =   Cette variable represent la difference entre les valeurs simulée
s et les valeurs experimentaux.
!                        En utilisant la methode de moindres carrées
!                        Est une variable de type DOUBLE PRECISION.

! COEFF1             =   Cette variable est
!                        Est une variable de type DOUBLE PRECISION.

! COEFF2             =   Cette variable est
!                        Est une variable de type DOUBLE PRECISION.

! COEFF3             =   Cette variable est
!                        Est une variable de type DOUBLE PRECISION.

! COEFF4             =   Cette variable est
!                        Est une variable de type DOUBLE PRECISION.

! CONV_FLOW_EQ       =   Cette variable est la convergence du code d'analyse de sous-cana
l

! TOTAL_ERROR_VEC    =   Cette variable est l'erreur entre les valeur mesurées au laborat
oire et les simulées

! LIN_F              =   Cette variable tient compte de la selection des chromosomes

! DISSIPATION_ENERGY=   Cette variable represente l'énergie dissipée par l'écoulement
```

```
! DISSIPATION_ENERGY_TRSNV = Cette variable represente l'energie de l'écoulement latéral

! TOTAL_ERROR_LOC   =   Cette variable est l'erreur entre les valeur mesurées au laborat
oire et les simulées

! EVOL              =   Cette variable represente le nombre des générationés pasées

! MAX               =   Cette variable represente la valeur maximale trove dans une popu
lation

! MIN               =   Cette variable represente la valeur minimale trove dans une popu
lation

! AVER              =   Cette variable represente l'average des tout le résultat d'une p
opulation

! SUM_TOTAL_ERROR   =   Cette variable est la somme de tous les erreurs


SUBROUTINE GA_REPORT
USE VAR1
USE GA_VAR
IMPLICIT NONE
INTEGER I,J
!-------------------------------------------------------------------------------------
---
! THIS SUBROUTINE CREATES A REPORT OF THE SEARCH
!-------------------------------------------------------------------------------------
---
 OPEN(UNIT=77, FILE=GA_REPORT_DAT, STATUS='UNKNOWN')
 WRITE(77,5) 'POPULATION', 'CO','D_EQ','KTR','D_ALFA','K DEQ','XP','XC','ACTUAL_COUNT','
CONV_FLOW','TOTAL ERROR'
! WRITE(77,10) ((POPULATION(J,I),I=1,CHROM_SIZE), CO_DEC_VEC(J),DEQ_DEC_VEC(J),KTR_DEC_V
EC(J), &
! &DALFA_DEC_VEC(J),K_DVFEQ_DEC_VEC(J),CROSSP(J),CROSSCHAIN(J),J,CONV_FLOW_EQ(J),TOTAL_E
RROR_VEC(J),LIN_F(J),Total_Energy(J),J=1,POP_SIZE)

 WRITE(77,15) (CO_DEC_VEC(J),KTR_DEC_VEC(J),DEQ_DEC_VEC(J),DALFA_DEC_VEC(J),LAMBDA_DEC_V
EC(J),COEFF1_DEC_VEC(J),COEFF2_DEC_VEC(J),COEFF3_DEC_VEC(J),COEFF4_DEC_VEC(J)&
 & ,J,CONV_FLOW_EQ(J),TOTAL_ERROR_VEC(J),LIN_F(J),DISSIPATION_ENERGY(J)&
 &,DISSIPATION_ENERGY_TRNSVS(J),TOTAL_ERROR_LOC(J),J=1,POP_SIZE)



 WRITE(77,20) 'NOTE: GENERATION', EVOL,'STATISTICS MAX:',MAX,'MIN:',MIN,'AVERAGE:',&
 &AVERAGE,'SUM:',SUM_TOTAL_ERROR


    IF (EVOL.EQ.GENERATION) THEN
        WRITE(77,'(A,2X,F9.7)') 'BESTFIT VALUE',BESTFIT_VAL
        WRITE(77,'(A,2X,I4)') 'BEST GENERATION',BESTFIT_LOC
    END IF



 5  FORMAT (T35,A,T107,A,T117,A,T128,A,T138,A,T149,A,T160,A,T165,A,T175,A,T189,A,T202,A)
 10 FORMAT (75(1X, I1), 3X, F8.6, 3X, F8.6, 3X, F8.6, 3X, F8.6,  3X, F8.6,  3X, I2, 3X, &
      & I2, 3X, I2, 3X, G11.4,3X,G11.4,3X,F7.4,3X,G17.10)

 15 FORMAT (3X, G11.4, 3X, G11.4, 3X, G11.4, 3X, G11.4, 3X, G11.4, &
      & 3X, G11.4, 3X, G11.4, 3X, G11.4,3X, G11.4,3X, G11.4,3X, G11.4,3X, G11.4, 3X, F7.
4, 3X, G17.10, &
      & 3X, G17.10,3X, G17.10)
```

```
 20 FORMAT (A, 2X, I3, 3X, A, 3X, G17.10, 3X, A, 3X, G17.10, 3X, A, 3X, &
     & G17.10, 3X, A, 3X, G17.10)
END SUBROUTINE GA_REPORT
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME SUBCHANNEL.   Ce programme a comme but de faire l'analyse de
!                           sous canal et l'optimiser en utilisant des
!                           algorithmes génétiques.

!
!   Ce module contienne toutes les variables utilisées pour le code génétique.Parmi les
plus importantes sont:
!   POPULATION
!CO_COD              =   La valeur de Co codée en système binaire
!DEQ_COD             =   La valeur de Deq codée en système binaire
!DALFA_COD           =   La valeur de Dalfa codée en système binaire
!KTR_COD             =   La valeur de Ktr codée en système binaire
!LAMBDA_COD          =   La valeur de Lambda codée en système binaire

!CO_DEC_VEC          =   La valeur de Co decodée en système decimal
!DEQ_DEC_VEC         =   La valeur de Deq decodée en système decimal
!DALFA_DEC_VEC       =   La valeur de Dalfa decodée en système decimal
!KTR_DEC_VEC         =   La valeur de Ktr decodée en système decimal
!LAMBDA_DEC_VEC      =   La valeur de Lambda decodée en système decimal

!POP_SIZE            =   La taille de la population
!CHROM_SIZE          =   La taille du chromosome
!COEFF_SIZE          =   La taille du coefficient
!FUNC_OBJ            =   La fonction objectif selectionée, les moindres carrés ou la diss
ipation de l'énergie.


MODULE GA_VAR


INTEGER ENTERO,MEC,CONST_BIN,SELECT,SELECTION,GENE_NUM,GENES,GENERATION,EVOL,POP_NUM,BES
TFIT_LOC,OPENFILE

INTEGER,ALLOCATABLE,DIMENSION(:,:)::CHROMOSOME,CHROMOSOMEXY,POPULATION
INTEGER,ALLOCATABLE,DIMENSION(:,:)::CO,DEQ,DALFA,EP
INTEGER,ALLOCATABLE,DIMENSION(:)::CROSSCHAIN
INTEGER,ALLOCATABLE,DIMENSION(:)::CROSSP
INTEGER,ALLOCATABLE,DIMENSION(:)::ACTUAL_COUNT
INTEGER,ALLOCATABLE,DIMENSION(:,:)::CO_COD,DEQ_COD,DALFA_COD,KTR_COD
INTEGER,ALLOCATABLE,DIMENSION(:,:)::LAMBDA_COD,COEFF1_COD,COEFF2_COD,,
INTEGER,ALLOCATABLE,DIMENSION(:)::OFFSPRING_VEC
DOUBLE PRECISION,ALLOCATABLE, DIMENSION(:)::TOTAL_ERROR_VEC,PERCENT,EXPECTED_COUNT,DISSI
PATION_ENERGY_TRNSVS
DOUBLE PRECISION,ALLOCATABLE, DIMENSION(:)::CO_DEC_VEC,DEQ_DEC_VEC,DALFA_DEC_VEC,KTR_DEC
_VEC
DOUBLE PRECISION,ALLOCATABLE, DIMENSION(:)::LAMBDA_DEC_VEC,COEFF1_DEC_VEC,COEFF2_DEC_VEC
,COEFF3_DEC_VEC,COEFF4_DEC_VEC
DOUBLE PRECISION,ALLOCATABLE, DIMENSION(:)::LIN_F,DISSIPATION_ENERGY
DOUBLE PRECISION,ALLOCATABLE, DIMENSION(:)::GEN_ENTROPY,TOTAL_ERROR_LOC
DOUBLE PRECISION::SUM_TOTAL_ERROR,TOTAL_ERROR,SUMA,AVERAGE,MAX,MIN,TOTAL_ERROR_ENTROPY
DOUBLE PRECISION::CO_SUP,DEQ_SUP,DALFA_SUP,KTR_SUP,CO_INF,DEQ_INF,DALFA_INF,KTR_INF
DOUBLE PRECISION::LAMBDA_SUP,LAMBDA_INF,COEFF1_SUP,COEFF1_INF,COEFF2_SUP,COEFF2_INF,COEF
F3_SUP,COEFF3_INF,COEFF4_SUP,COEFF4_INF
DOUBLE PRECISION::K_DVFEQ_INF, K_DVFEQ_SUP
DOUBLE PRECISION::OLD_MAX_VALUE,NEW_MAX_VALUE,MIN_ERROR,GENERATION_NUMBER
DOUBLE PRECISION::BESTFIT_VAL,PROB_CROSS,PM
DOUBLE PRECISION::W1,W2,W3,W4,W5,W6,W7,W8,W9,W10,W11
CHARACTER*45 GA_WEIGHT_INPUT_DAT,GA_REPORT_DAT,GRAPH_DATA_RES,INTER2_TXT
INTEGER::POP_SIZE,CHROM_SIZE, COEFF_SIZE,,OPTIONMUT
CONTAINS
```

```
SUBROUTINE ALLOCATE_GA(R,C,Z)

INTEGER R,C,Z
ALLOCATE (CO_COD(R,C),DEQ_COD(R,C),DALFA_COD(R,C),KTR_COD(R,C))
ALLOCATE (LAMBDA_COD(R,C),COEFF1_COD(R,C),COEFF2_COD(R,C),COEFF3_COD(R,C),COEFF4_COD(R,C
))
ALLOCATE (CHROMOSOME(R,C),CHROMOSOMEXY(R,C),POPULATION(R,C))
ALLOCATE (CO(R,C),DEQ(R,C),DALFA(R,C),EP(R,C))
ALLOCATE (CROSSCHAIN(R),DISSIPATION_ENERGY_TRNSVS(R))
ALLOCATE (CROSSP(R))
ALLOCATE (ACTUAL_COUNT(R))
ALLOCATE (TOTAL_ERROR_VEC(R),PERCENT(R),EXPECTED_COUNT(R),DISSIPATION_ENERGY(R),GEN_ENTR
OPY(R),TOTAL_ERROR_LOC(R))
ALLOCATE (CO_DEC_VEC(R),DEQ_DEC_VEC(R),DALFA_DEC_VEC(R),KTR_DEC_VEC(R))
ALLOCATE (LAMBDA_DEC_VEC(R),COEFF1_DEC_VEC(R),COEFF2_DEC_VEC(R),COEFF3_DEC_VEC(R),COEFF4
_DEC_VEC(R))
ALLOCATE (LIN_F(R))
ALLOCATE (OFFSPRING_VEC(C))

END SUBROUTINE ALLOCATE_GA


END MODULE GA_VAR
```

```
!------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.
!

!   Sous programme FIRST_GENE

!   Ce sous programme a comme but:
!
!   Faire la première génération, cette génération est aleatoirée.

!-----------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-----------------------------------------------------------------------------------


! POP_SIZE       =   La taille de la population.
!                    Est une variable de type INTEGER.

! NUM            =   Est un numéro aléatoire, si < 0.5 = 0 sinon = 1
!                    Est une vecteur de type INTEGER.


SUBROUTINE FIRST_GENE
USE GA_VAR
IMPLICIT NONE
INTEGER, DIMENSION(1)   :: Seed
REAL, DIMENSION(CHROM_SIZE) :: NUM
INTEGER I,J,COUNT,RET
REAL*8 RND,ELAPSED_TIME


  DO I=1,CHROM_SIZE

    CALL RANDOM_SEED(PUT=SEED)
    CALL RANDOM_NUMBER(NUM(I))
    CALL RANDOM_SEED(GET=SEED)

  IF (NUM(I)>0.5) OFFSPRING_VEC(I)=1
  IF (NUM(I)<0.5) OFFSPRING_VEC(I)=0

  END DO


    CALL RANDOM_SEED(GET=SEED)


END SUBROUTINE FIRST_GENE
```

```
!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04

!

!   PROGRAMME SUBCHANNEL.   Ce programme a comme but de faire l'analyse de
!                           sous canal et l'optimiser en utilisant des
!                           algorithmes génétiques.


!   SOUS-PROGRAMME INIT_VAR.    Ce sous-programme a comme but d'initialiser toutes les v
ariables à zero.



SUBROUTINE INIT_VAR
USE VAR1
USE GA_VAR

IMPLICIT NONE
AREA=0.0
GASFLUX=0.0
LIQFLUX=0.0
TTLFLUX=0.0
PRSRE=0.0
VOID_FRCTN=0.0
WET_PRMTER=0.0
HYDRLC_DIAM=0.0
QUALITY=0.0
LIQFLOW=0.0
GASFLOW=0.0
GAS_DNSTY=0.0
PC=0.0
GASFLOW_CF1=0.0
GASFLOW_CF2=0.0
GASFLOW_CF3=0.0
FR=0.0
F=0.0
MIX_DNSTY=0.0
AA=0.0
BB=0.0
CC=0.0
DD=0.0
AP=0.0
BP=0.0
CP=0.0
DP=0.0
BP1=0.0
DPFT=0.0
PC_AUX=0.0


! ONE DIMENSION ARRAY

GAP_GL=0.0
GAP_GG=0.0
VG_CF=0.0
D_CF=0.0
TTLFLUX_CF=0.0
LIQFLUX_CF=0.0
GASFLUX_CF=0.0
XPOSITION=0.0
DELTAX=0.0
DELTAY=0.0
DELTAX_CV=0.0
```

```
X_PRSRE=0.0
OMEGA=0.0
GAS_DNSTY_CF=0.0
VOID_FRCTN_CF=0.0
VG=0.0
DWL=0.0
DWG=0.0
V=0.0
D=0.0
DP_RAD=0.0
U=0.0
XKTR=0.0
DIV_CROSS=0.0
TURB_VOID_DIFF=0.0
VOID_DRIFT=0.0
BLIQ_TME=0.0
BGAS_TME=0.0
XA=0.0
YA=0.0
LIQFLOW1_INTERPOLATED=0.0
LIQFLOW2_INTERPOLATED=0.0
DPRADIAL_INTERPOLATED=0.0
PRSRE1_INTERPOLATED=0.0
PRSRE2_INTERPOLATED=0.0
VOID_FRCTN1_INTERPOLATED=0.0
VOID_FRCTN2_INTERPOLATED=0.0
DPFT1_INTERPOLATED=0.0
DPFT2_INTERPOLATED=0.0
GASFLOW1_INTERPOLATED=0.0
GASFLOW2_INTERPOLATED=0.0
XPOS_EXP=0.0
LIQFLOW_EXP=0.0
LIQFLOW_EXP2=0.0
PRSRE_EXP1=0.0
PRSRE_EXP2=0.0
DPFT_EXP1=0.0
DPFT_EXP2=0.0
XPOSITION_GAS=0.0
AIRFLOW_EXP1=0.0
AIRFLOW_EXP2=0.0
DPRADIAL=0.0
VOID_FRCTN_EXP1=0.0
VOID_FRCTN_EXP2=0.0
XA_2=0.0
YA_2=0.0


COEF_VF=0.0
COEF_PHI=0.0
K_IRRV=0.0
B=0.0
C=0.0
DP_INI=0.0
SUM_PC=0.0
SUM_PC_AUX=0.0
CO_COD=0.0
DEQ_COD=0.0
DALFA_COD=0.0
KTR_COD=0.0

GAP_WDTH=0.0
GAP_LNGTH=0.0
CHNL_WDTH1=0.0
CHNL_WDTH2=0.0
CHNL_LNGTH1=0.0
CHNL_LNGTH2=0.0
CHNL_WDTH=0.0
LNGTH=0.0
```

```
TTL_LNGTH=0.0
STRT_INP=0.0
DDX=0.0
GASFLUX_INP1=0.0
LIQFLUX_INP1=0.0
GASFLUX_INP2=0.0
LIQFLUX_INP2=0.0
PRSSRE_OUT1=0.0
PRSSRE_OUT2=0.0
VOIDFRC_INP1=0.0
VOIDFRC_INP2=0.0
TEMP=0.0
VOID_FRCTN_12=0.0
VOID_FRCTN2=0.0
VOID_FRCTN1=0.0
NIT_CF=0.0
NIT_MAS=0.0
NIT_PRS =0.0
LIQ_DNSTY=0.0
F_MOM=0.0
X_K_INPUT=0.0
X_LAMBDA=0.0
ROL=0.0
VISCOCITY=0.0
QUALITY_AVE=0.0
CONV_CRTRIA=0.0
RELAX=0.0
EPSILON=0.0
DP_GRVTY=0.0
DP_FRCTN=0.0
GRVTY =0.0
BGAS=0.0
BLIQ=0.0
BLIQ_INI=0.0
BGAS_INI=0.0
U1=0.0
U2=0.0
BALANCE=0.0
MIX_DNSTY_CF=0.0
DR=0.0
PRSRE_AVE=0.0
PRSRE_CF=0.0
FLUX_Y=0.0
DMASS=0.0
FLUX_BG=0.0
AREA_C=0.0
SIGMA=0.0
VOID_FRCTN_AVE=0.0
SPFCL_VLCTY1=0.0
SPFCL_VLCTY2=0.0
MIX_SPFCL_VLCTY=0.0
HYDRL_DIAM_AVE=0.0
TURB_VOID_DIF=0.0
COEF=0.0
CX=0.0
DELTAY_AVE=0.0
FO=0.0
PHI2=0.0
V12=0.0
DELG=0.0
SUM_LIQ=0.0
SUM_GAS=0.0
SW=0.0
DVFEQ=0.0
GAPARA=0.0
GAPER=0.0
DP_RAD=0.0
Y=0.0
```

```
CONV_FACT=0.0
ERROR_PC1=0.0
ERROR_PC2=0.0
ERROR_TTL_PC=0.0
DALFA_DEC=0.0
CO_DEC=0.0
KTR_DEC=0.0
DEQ_DEC=0.0


N=0
OPTION=0
TYPE_GRID=0
BLCKGE_TYPE=0
NPTS=0
N1=0
N2=0
N3=0
I1=0
I2=0
NN1=0
N_ITER=0
CONT_ITER=0
N_ITER_MOM=0
IN_FRIC_CORR=0
IN_KTRNSL=0
IIR_PRSRE=0
IN_EQ_VOIDF=0
IN_EQ_DNSTY =0
ICOEFF=0
NUM_ITER=0
NAUX=0
NIT_MAX=0
CONT_ITER_MOM=0
GA=0
AUX=0
AUX2=0
END SUBROUTINE INIT_VAR
```

```
 SUBROUTINE INPUT_DATA
 USE VAR1
 USE GA_VAR
 INTEGER GA_COUNT
 GA=1
 GA_COUNT=1
 OPENFILE=1
!---------------------------------------------------------------------
!
!  OPENING A FILE WITH THE EXPERIENCE NUMBER OR THE NUMERIC VALUES
!  0 MEANS USER VALUES
!  NOT EQUAL 0 THE EXPERIMENT NUMBER
!---------------------------------------------------------------------

!      OPEN(UNIT=1,FILE='C:\DATA.RES',STATUS='UNKNOWN')
!     READ(1,*) TYPE$

WRITE(*,*) 'NUMERO DE EXPERIENCIA '

READ(*,*) TYPE$


  IF (TYPE$.NE.0) THEN


      CHNL_WDTH1=0.01265D0
      CHNL_LNGTH1=0.01265D0
      CHNL_WDTH2=0.01277D0
      CHNL_LNGTH2=0.01277D0
      GAP_WDTH=0.0015D0
      GAP_LNGTH=0.003D0
      STRT_INP=0.257D0
      LNGTH=0.8098D0
      TYPE_GRID=1
      NPTS=90
      NIT_MAX=100
      EPSILON=1.0D-15
      RELAX=0.55


     SELECT CASE (TYPE$)

        CASE(1)
      PRSSRE_OUT1=140086.63D0
      LIQFLUX_INP1=3175.0D0
      GASFLUX_INP1=13.47D0
      VOIDFRC_INP1=0.6D0
      PRSSRE_OUT2=140228.31D0
      LIQFLUX_INP2=3089.0D0
      GASFLUX_INP2=0.0D0
      VOIDFRC_INP2=0.0D0

        CASE(2)
      PRSSRE_OUT1=146970.19D0
      LIQFLUX_INP1=3125.0D0
      GASFLUX_INP1=13.84D0
      VOIDFRC_INP1=0.595D0
      PRSSRE_OUT2=146966.75D0
      LIQFLUX_INP2=3072.0D0
      GASFLUX_INP2=4.31D0
      VOIDFRC_INP2=0.395D0


        CASE(3)
      PRSSRE_OUT1=125724.19D0
      LIQFLUX_INP1=3127.0D0
```

```
  GASFLUX_INP1=4.01D0
  VOIDFRC_INP1=0.405D0
  PRSSRE_OUT2=125699.50D0
  LIQFLUX_INP2=3050.0D0
  GASFLUX_INP2=0.0D0
  VOIDFRC_INP2=0.0D0
    CASE(5)
  PRSSRE_OUT1=123961.88D0
  LIQFLUX_INP1=3139.0D0
  GASFLUX_INP1=1.96D0
  VOIDFRC_INP1=0.292D0
  PRSSRE_OUT2=123944.25D0
  LIQFLUX_INP2=3071.0D0
  GASFLUX_INP2=0.0D0
  VOIDFRC_INP2=0.0D0
    CASE(6)
  PRSSRE_OUT1=144285.38D0
  LIQFLUX_INP1=3125.0D0
  GASFLUX_INP1=8.45D0
  VOIDFRC_INP1=0.51D0
  PRSSRE_OUT2=144230.88D0
  LIQFLUX_INP2=3065.0D0
  GASFLUX_INP2=0.0D0
  VOIDFRC_INP2=0.0D0
    CASE(8)
  PRSSRE_OUT1=139829.25D0
  LIQFLUX_INP1=3119.0D0
  GASFLUX_INP1=4.25D0
  VOIDFRC_INP1=0.40D0
  PRSSRE_OUT2=139725.06D0
  LIQFLUX_INP2=3064.0D0
  GASFLUX_INP2=0.73D0
  VOIDFRC_INP2=0.13D0
    CASE(9)
  PRSSRE_OUT1=20524.2
  LIQFLUX_INP1=1745.0
  GASFLUX_INP1=1.70
  VOIDFRC_INP1=0.36D0
  PRSSRE_OUT2=20504.6D0
  LIQFLUX_INP2=1680.0D0
  GASFLUX_INP2=2.43D0
  VOIDFRC_INP2=0.27D0
    CASE(10)

  PRSSRE_OUT1=165470.19D0
  LIQFLUX_INP1=3113.0D0
  GASFLUX_INP1=14.86D0
  VOIDFRC_INP1=0.0D0
  PRSSRE_OUT2=165438.44D0
  LIQFLUX_INP2=3075.0D0
  GASFLUX_INP2=1.20D0
  VOIDFRC_INP2=0.0D0


END SELECT


TEMP=20.0
K_IRRV(1)=0.0
K_IRRV(2)=0.0
F_MOM=1.0D0
X_K_INPUT=1.0D0
X_LAMBDA=0.0D0

IN_FRIC_CORR=1
IN_KTRNSL=1
IIR_PRSRE=0
```

```
    IN_EQ_VOIDF=2
    IN_EQ_DNSTY=1

!  DO I=1,33
!   READ(1,*)
!  END DO

!  READ(1,10) GRAPHFILE
!  READ(1,10) GRAPHFILE


!10 FORMAT (A10)


!-------------------------------------------------------------------------------------------
--------
!Opening the file with the name of all the files and locations
!-------------------------------------------------------------------------------------------
--------
OPEN(UNIT=45, FILE= 'D:\INPUT_FILES_NAMES.DAT', ACTION='READ',STATUS='UNKNOWN')

READ(45,*)
READ(45,46) GA_WEIGHT_INPUT_DAT
READ(45,*)
READ(45,46) GA_REPORT_DAT
READ(45,*)
READ(45,46) GRAPH_DATA_RES
READ(45,*)
READ(45,46) INTER2_TXT
READ(45,*)
READ(45,46) GRAPHFILE
READ(45,*)
READ(45,46) DATAFILE

CLOSE (UNIT=45)

 ELSE IF (TYPE$.EQ.0) THEN
READ(1,*) CHNL_WDTH1
READ(1,*) CHNL_WDTH2
READ(1,*) CHNL_LNGTH1
READ(1,*) CHNL_LNGTH2
READ(1,*) GAP_WDTH
READ(1,*) GAP_LNGTH
READ(1,*) LNGTH
READ(1,*) TYPE_GRID
READ(1,*) NPTS
READ(1,*) NIT_MAX
READ(1,*) EPSILON
READ(1,*) RELAX
READ(1,*) PRSSRE_OUT1
READ(1,*) PRSSRE_OUT2
READ(1,*) LIQFLUX_INP1
READ(1,*) LIQFLUX_INP2
READ(1,*) GASFLUX_INP1
READ(1,*) GASFLUX_INP2
READ(1,*) VOIDFRC_INP1
READ(1,*) VOIDFRC_INP2
READ(1,*)
READ(1,*) STRT_INP
READ(1,*) TEMP
READ(1,*) K_IRRV(1)
READ(1,*) K_IRRV(2)
READ(1,*) F_MOM
READ(1,*) X_K_INPUT
READ(1,*) X_LAMBDA
READ(1,*) IN_FRIC_CORR
READ(1,*) IN_KTRNSL
READ(1,*) IIR_PRSRE
```

```
READ(1,*) IN_EQ_VOIDF
READ(1,*) IN_EQ_DNSTY

END IF

OPTION=TYPE$


N=NPTS+3


GENES=30


IF (OPTION.EQ.1) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#1.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.2) THEN
OPEN (UNIT=7,FILE='D:\EXPERIENCE#2.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.3) THEN
OPEN (UNIT=7,FILE='D:\EXPERIENCE#3.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.5) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#5.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.6) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#6.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.8) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#8.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.9) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#9.TXT',STATUS='unknown')
END IF


    AUX2=0
    DO WHILE (.NOT.EOF(7))

    READ(7,'(t86,A3)') X
    IF(X.EQ.'' ) EXIT
    AUX2=AUX2+1
    END DO
    REWIND(UNIT=7)

    AUX=0
    DO WHILE (.NOT.EOF(7))
    AUX=AUX+1
    READ(7,'(f2.1)') XPOS_EXP
    END DO

    REWIND(UNIT=7)

IF (GA_COUNT.EQ.1) THEN
CALL ALLOCATION (N,AUX,AUX2)
GA_COUNT=2
END IF



I=1
 DO WHILE (I.LE.AUX)

      READ(7,800) XPOS_EXP(I),LIQFLOW_EXP(I),LIQFLOW_EXP2(I),PRSRE_EXP1(I),PRSRE_EXP2(I)
,DPFT_EXP1(I),DPFT_EXP2(I),&
      & VOID_FRCTN_EXP1(I),VOID_FRCTN_EXP2(I),DPRAD_EXP(I)
      I=I+1
  END DO
  REWIND(UNIT=7)
```

```
I=1
 DO WHILE (I.LE.AUX2)

      READ(7,900) XPOSITION_GAS(I),AIRFLOW_EXP1(I),AIRFLOW_EXP2(I)
      I=I+1
 END DO
REWIND(UNIT=7)


  46 FORMAT(A)
 800 FORMAT (F7.2, T8, F7.2, T16, F7.2, T24, F9.2, T34, F9.2, T44, &
     & F7.2, T52, F7.2, T70, F7.5, T78, F7.5, T60, E9.2)
 900 FORMAT(T86,F6.4,T93,F6.4,T100,F6.4)
 END SUBROUTINE INPUT_DATA
```

```
SUBROUTINE INPUT_DATA_GA
 USE VAR1
 USE GA_VAR
 INTEGER GA_COUNT
 OPENFILE=1
  IF (TYPE$.NE.0) THEN


      CHNL_WDTH1=0.01265D0
      CHNL_LNGTH1=0.01265D0
      CHNL_WDTH2=0.01277D0
      CHNL_LNGTH2=0.01277D0
      GAP_WDTH=0.0015D0
      GAP_LNGTH=0.003D0
      STRT_INP=0.257D0
      LNGTH=0.8098D0
      TYPE_GRID=1
      NPTS=90
      NIT_MAX=100
      EPSILON=1.0D-15
      RELAX=0.55


     SELECT CASE (TYPE$)

       CASE(1)
      PRSSRE_OUT1=140086.63D0
      LIQFLUX_INP1=3175.0D0
      GASFLUX_INP1=13.47D0
      VOIDFRC_INP1=0.6D0
      PRSSRE_OUT2=140228.31D0
      LIQFLUX_INP2=3089.0D0
      GASFLUX_INP2=0.0D0
      VOIDFRC_INP2=0.0D0
          CASE(2)
      PRSSRE_OUT1=146970.19D0
      LIQFLUX_INP1=3125.0D0
      GASFLUX_INP1=13.84D0
      VOIDFRC_INP1=0.595D0
      PRSSRE_OUT2=146966.75D0
      LIQFLUX_INP2=3072.0D0
      GASFLUX_INP2=4.31D0
      VOIDFRC_INP2=0.395D0
        CASE(3)
      PRSSRE_OUT1=125724.19D0
      LIQFLUX_INP1=3127.0D0
      GASFLUX_INP1=4.01D0
      VOIDFRC_INP1=0.405D0
      PRSSRE_OUT2=125699.50D0
      LIQFLUX_INP2=3050.0D0
      GASFLUX_INP2=0.0D0
      VOIDFRC_INP2=0.0D0
        CASE(5)
      PRSSRE_OUT1=123961.88D0
      LIQFLUX_INP1=3139.0D0
      GASFLUX_INP1=1.96D0
      VOIDFRC_INP1=0.292D0
      PRSSRE_OUT2=123944.25D0
      LIQFLUX_INP2=3071.0D0
      GASFLUX_INP2=0.0D0
      VOIDFRC_INP2=0.0D0
        CASE(6)
      PRSSRE_OUT1=144285.38D0
      LIQFLUX_INP1=3125.0D0
      GASFLUX_INP1=8.45D0
      VOIDFRC_INP1=0.51D0
      PRSSRE_OUT2=144230.88D0
```

```
      LIQFLUX_INP2=3065.0D0
      GASFLUX_INP2=0.0D0
      VOIDFRC_INP2=0.0D0
        CASE(8)
      PRSSRE_OUT1=139829.25D0
      LIQFLUX_INP1=3119.0D0
      GASFLUX_INP1=4.25D0
      VOIDFRC_INP1=0.40D0
      PRSSRE_OUT2=139725.06D0
      LIQFLUX_INP2=3064.0D0
      GASFLUX_INP2=0.73D0
      VOIDFRC_INP2=0.13D0
        CASE(9)
      PRSSRE_OUT1=165470.19D0
      LIQFLUX_INP1=3113.0D0
      GASFLUX_INP1=14.86D0
      VOIDFRC_INP1=0.595D0
      PRSSRE_OUT2=165438.44D0
      LIQFLUX_INP2=3075.0D0
      GASFLUX_INP2=1.20D0
      VOIDFRC_INP2=0.17D0
        CASE(10)

      PRSSRE_OUT1=165470.19D0
      LIQFLUX_INP1=3113.0D0
      GASFLUX_INP1=14.86D0
      VOIDFRC_INP1=0.0D0
      PRSSRE_OUT2=165438.44D0
      LIQFLUX_INP2=3075.0D0
      GASFLUX_INP2=1.20D0
      VOIDFRC_INP2=0.0D0


    END SELECT


    TEMP=20.0
    K_IRRV(1)=0.0
    K_IRRV(2)=0.0
    F_MOM=1.0D0
    X_K_INPUT=1.0D0
    X_LAMBDA=0.0

    IN_FRIC_CORR=1
    IN_KTRNSL=1
    IIR_PRSRE=0
    IN_EQ_VOIDF=2
    IN_EQ_DNSTY=1


 ELSE IF (TYPE$.EQ.0) THEN
READ(1,*) CHNL_WDTH1
READ(1,*) CHNL_WDTH2
READ(1,*) CHNL_LNGTH1
READ(1,*) CHNL_LNGTH2
READ(1,*) GAP_WDTH
READ(1,*) GAP_LNGTH
READ(1,*) LNGTH
READ(1,*) TYPE_GRID
READ(1,*) NPTS
READ(1,*) NIT_MAX
READ(1,*) EPSILON
READ(1,*) RELAX
READ(1,*) PRSSRE_OUT1
READ(1,*) PRSSRE_OUT2
READ(1,*) LIQFLUX_INP1
READ(1,*) LIQFLUX_INP2
```

```
READ(1,*) GASFLUX_INP1
READ(1,*) GASFLUX_INP2
READ(1,*) VOIDFRC_INP1
READ(1,*) VOIDFRC_INP2
READ(1,*)
READ(1,*) STRT_INP
READ(1,*) TEMP
READ(1,*) K_IRRV(1)
READ(1,*) K_IRRV(2)
READ(1,*) F_MOM
READ(1,*) X_K_INPUT
READ(1,*) X_LAMBDA
READ(1,*) IN_FRIC_CORR
READ(1,*) IN_KTRNSL
READ(1,*) IIR_PRSRE
READ(1,*) IN_EQ_VOIDF
READ(1,*) IN_EQ_DNSTY

END IF

OPTION=TYPE$


N=NPTS+3



GENES=30



IF (OPTION.EQ.1) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#1.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.2) THEN
OPEN (UNIT=7,FILE='D:\EXPERIENCE#2.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.3) THEN
OPEN (UNIT=7,FILE='D:\EXPERIENCE#3.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.5) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#5.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.6) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#6.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.8) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#8.TXT',STATUS='unknown')
ELSE IF(OPTION.EQ.9) THEN
OPEN (unit=7,FILE='D:\EXPERIENCE#9.TXT',STATUS='unknown')
END IF
    REWIND(UNIT=7)


    AUX2=0
    DO WHILE (.NOT.EOF(7))

    READ(7,'(t86,A3)') X
    IF(X.EQ.'' ) EXIT
    AUX2=AUX2+1
    END DO

    REWIND(UNIT=7)

    AUX=0
    DO WHILE (.NOT.EOF(7))
    AUX=AUX+1
    READ(7,'(f2.1)') X
    END DO

    REWIND(UNIT=7)
```

```
I=1
 DO WHILE (I.LE.AUX)

      READ(7,800) XPOS_EXP(I),LIQFLOW_EXP(I),LIQFLOW_EXP2(I),PRSRE_EXP1(I),PRSRE_EXP2(I)
,DPFT_EXP1(I),DPFT_EXP2(I),&
      & VOID_FRCTN_EXP1(I),VOID_FRCTN_EXP2(I),DPRAD_EXP(I)
      I=I+1
  END DO
  REWIND(UNIT=7)

I=1
 DO WHILE (I.LE.AUX2)

      READ(7,900) XPOSITION_GAS(I),AIRFLOW_EXP1(I),AIRFLOW_EXP2(I)
      I=I+1
 END DO
REWIND(UNIT=7)



 800 FORMAT (F7.2, T8, F7.2, T16, F7.2, T24, F9.2, T34, F9.2, T44, &
      & F7.2, T52, F7.2,T70,F7.5,T78,F7.5,T60,E9.2)
 900 FORMAT(T86,F6.4,T93,F6.4,T100,F6.4)
 END SUBROUTINE INPUT_DATA_GA
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.--------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.
!

! Sous programme LINEINTERPOLATION
!
! Ce sous programme a comme but:
!
! Réaliser une interpolation entre les valeurs expérimentales et les valeurs simulées,
! La méthode utilisée est linéal.
!

! Les variables d'entrée sont:


! xa     =    La valeur de x expérimental.
! ya     =    La valeur de y expérimental.
! n      =    Nombre de noeuds de la discretisation.
!

! Les variables de sortie sont:
! y      =    La valeur de y interpolée entre les valeurs expérimentales et simulées.

! Les variables internées sont:
!
! FLAG   =    Cette variable aide a déterminer les points ou se va à réaliser la interpola
tion.
! LOC    =    Cette variable aide a déterminer les points ou se va à réaliser la interpola
tion.


 SUBROUTINE LINEINTERPOLATION(xa,ya,n,x,y)

      INTEGER n
      DOUBLE PRECISION:: x,y,AUX
      DOUBLE PRECISION, DIMENSION(N)::XA,YA,FLAG
      DOUBLE PRECISION:: A,B,C

      DO I=1,N
      FLAG(I)=DABS(X-XA(I))
      END DO

      LOC=MINLOC(FLAG,DIM=1)


!---------------------------------------------------------------------
!
! CALCULATION THE LINEA INTERPOLATION
!---------------------------------------------------------------------
! IF THE FIRST POINT IS THE FIRST ONE



     IF (LOC.EQ.1) THEN
      A = XA(LOC)-XA(LOC+1)
      Y = YA(LOC)
```

```
    DO WHILE(A.NE.0.0)
      Y = (YA(LOC)-YA(LOC+1)) / (XA(LOC)-XA(LOC+1)) * (X - XA(LOC+1)) + YA(LOC+1)
      EXIT
    END DO

   END IF


 IF((LOC.GT.1).AND.(LOC.LT.N)) THEN
  B=XA(N)-XA(N-1)
  Y=YA(LOC)
   DO WHILE(B.NE.0.0)
     Y = (YA(LOC)-YA(LOC-1)) / (XA(LOC)-XA(LOC-1)) * (X - XA(LOC-1)) + YA(LOC-1)
     EXIT
   END DO
 END IF

IF (LOC.EQ.N) THEN
  C=XA(LOC)-XA(LOC-1)
  Y=YA(LOC)
  DO WHILE(C.NE.0.0)
   Y = (YA(N)-YA(N-1)) / (XA(N)-XA(N-1)) * (X - XA(N-1)) + YA(N-1)
   EXIT
  END DO
END IF
    return
    END
```

```
!------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.
!

! Sous programme INTERPOLATION2
!
! Ce sous programme a comme but:
!
! Calculer l'erreur relatif entre les valeurs interpolées et les valeur experimentales.
! Ce sous-programme utilise le sous-programme : LINEINTERPOLATION
!

! Les variables d'entrée sont:

! XPOS_EXP          =   Cette variable represente la position des mesurées.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)


! LIQFLOW_EXP       =   Cette variable represente la valeur experimental du transfer de
masse
!                       latéral du canal 1.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)


! LIQFLOW_EXP2      =   Cette variable represente la valeur experimental du transfer de
masse
!                       latéral du canal 2.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)

! PRSRE_EXP1        =   Cette variable represente la valeur experimental de la pression
absolue
!                       du canal 1.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)

! PRSRE_EXP2        =   Cette variable represente la valeur experimental de la pression
absolue
!                       du canal 2.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)

! DPFT_EXP1         =   Cette variable represente la valeur experimental de la perte de
pression absolue
!                       du canal 1.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)

! DPFT_EXP2         =   Cette variable represente la valeur experimental de la perte de
pression absolue
!                       du canal 2.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)


! VOID_FRCTN_EXP1   =   Cette variable represente la valeur experimental du taux de vide
!                       du canal 1.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)


! VOID_FRCTN_EXP2   =   Cette variable represente la valeur experimental du taux de vide
!                       du canal 2.
!                       Cette variable est de type DOUBLE PRECISION dimension (N)
```

```
! DPRAD_EXP          =   Cette variable represente la difference de pression lateral entr
e les
!                        deux canals
!                        Cette variable est de type DOUBLE PRECISION dimension (N)

! XA_2               =   Cette variable est intermediaire, sert à vider les variables d'e
ntrée vers
!                        une seule matriz en gardant les valeurs de X

! YA_2               =   Cette variable est intermediaire, sert à vider les variables d'e
ntrée vers
!                        une seule matriz en gardant les valeurs de Y


SUBROUTINE INTERPOLATION2

USE VAR1
USE GA_VAR
IMPLICIT NONE
INTEGER I,J,K

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   THIS SUBROUTINE INTERPOLE THE SIMULATION DATA TROUGHT THE PROGRAM
!   POLYINT.
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DOUBLE PRECISION:: INTER_ERROR1,INTER_ERROR2,INTER_ERROR3
DOUBLE PRECISION:: INTER_ERROR4,INTER_ERROR5,INTER_ERROR6,INTER_ERROR7,INTER_ERROR8
DOUBLE PRECISION:: INTER_ERROR9,INTER_ERROR10,INTER_ERROR11,SUM_DP_RAD_BGN
DOUBLE PRECISION, DIMENSION(N)::LIQFLOW1_INTERPOLATED_2,LIQFLOW2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::PRSRE1_INTERPOLATED_2,PRSRE2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::DPFT1_INTERPOLATED_2,DPFT2_INTERPOLATED_2,DPRAD1_INTERPO
LATED_2
DOUBLE PRECISION, DIMENSION(N)::GASFLOW1_INTERPOLATED_2,GASFLOW2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::VOID_FRCTN1_INTERPOLATED_2,VOID_FRCTN2_INTERPOLATED_2,DE
R_WEGTH

INTEGER H,LOC
!-------------------------------------------------------------------------------------
-
!
! OPENING A UNIT TO SAVE THE RESULTS
! THE RESULTS ARE SAVED IN THE FILE NAMED INTER.RES
!
!-------------------------------------------------------------------------------------
-

    OPEN(UNIT=8,FILE=INTER2_TXT,STATUS='UNKNOWN')

!-------------------------------------------------------------------------------------
-
!
!CONTRAINTS
!
!-------------------------------------------------------------------------------------
-

!-------------------------------------------------------------------------------------
-----------------
! TO AVOID LOCAL MINIMUMS OR TRIVIAL SOLUTION THE USE OF CONTRAINTS ARE USED
! THE CONVERGENCE IS ONE OF THE PARAMETERS
!-------------------------------------------------------------------------------------
-----------------

            IF(CONV_CRTRIA.GE. 1.0E-03) THEN
```

```
                TOTAL_ERROR     =   -1
                RETURN
            END IF
!-------------------------------------------------------------------------------------------
------------------
! THE VOID FRACTION MUST BE BETWEEN 0 AND 1 OTHERWISE THE ANSWER IS PHYSICALLY UNREALLIS
TIC
!-------------------------------------------------------------------------------------------
------------------


            DO I=1,N
                IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) )THEN
                    FLAG=.TRUE.
                    RETURN
                END IF


                IF ((VOID_FRCTN(I,1).LT.0.0).OR.(VOID_FRCTN(I,1).GT.1.0) ) THEN
                     FLAG=.TRUE.
                     RETURN
                END IF
            END DO
!-------------------------------------------------------------------------------------------
------------------
! THE DIFF OF PRESSURES BEFORE THE CONNECTION SHOULD BE NOT EQUAL TO ZERO
! SUM_DP_BGN = THE SUMATION OF DIFFERENCE OF PRESSURES BEFORE INTERCONNECTION
!-------------------------------------------------------------------------------------------
------------------

            SUM_DP_RAD_BGN= DABS(SUM(DP_RAD(1:I2))/1000)

            IF(SUM_DP_RAD_BGN.EQ.0.0) FLAG=.TRUE.

            IF(FLAG.EQ..TRUE.) THEN
                TOTAL_ERROR     =   -1
                RETURN
            END IF


 INTER_ERROR1=0.0D0
 INTER_ERROR2=0.0D0
 INTER_ERROR3=0.0D0
 INTER_ERROR4=0.0D0
 INTER_ERROR5=0.0D0
 INTER_ERROR6=0.0D0
 INTER_ERROR7=0.0D0
 INTER_ERROR8=0.0D0
 INTER_ERROR9=0.0
 INTER_ERROR10=0.0
 INTER_ERROR11=0.0


  DO H=1,10

 SELECT CASE (H)

 CASE (1)

  XA_2=XPOS_EXP          ! LETTING THE POSITION TO XA
  YA_2=LIQFLOW_EXP       ! LETTING THE  LIQUID FLOW OF THE FIRST CHANNEL TO YA

    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      LIQFLOW1_INTERPOLATED_2(I)=Y
      INTER_ERROR1  = INTER_ERROR1 + DSQRT(DABS((LIQFLOW1_INTERPOLATED_2(I)-LIQFLOW(I,1)
```

```
)/LIQFLOW1_INTERPOLATED_2(I))**2)


    END DO
       INTER_ERROR1 = INTER_ERROR1 / (N-1)

!-------------------------------------------------------------------------------------------
---
!
!      INTERPOLATING POSITION Vs QUALITY CHANNEL 2
!
!-------------------------------------------------------------------------------------------
---

  CASE(2)

  i=0
  XA_2=XPOS_EXP              ! LETTING THE POSITION TO XA
  YA_2=LIQFLOW_EXP2          ! LETTING THE QUALITY OF THE SECOND CHANNEL TO YA


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      LIQFLOW2_INTERPOLATED_2(I)=Y
      INTER_ERROR2  = INTER_ERROR2 + DSQRT(DABS((LIQFLOW2_INTERPOLATED_2(I)-LIQFLOW(I,2)
)/LIQFLOW2_INTERPOLATED_2(I))**2)

    END DO
      INTER_ERROR2  = INTER_ERROR2 / (N-1)
!-------------------------------------------------------------------------------------------
---
!       INTERPOLATING POSITION Vs PRESSURE CHANNEL 1
!
!
!-------------------------------------------------------------------------------------------
---


 CASE(3)
  i=0
  XA_2=XPOS_EXP              ! LETTING THE POSITION TO XA
  YA_2=PRSRE_EXP1           ! LETTING THE PRESSURE OF THE FIRST CHANNEL TO YA


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      PRSRE1_INTERPOLATED_2(I)=Y
      INTER_ERROR3  = INTER_ERROR3 + DSQRT(DABS((PRSRE1_INTERPOLATED_2(I)-PRSRE(I,1))/PR
SRE1_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR3    = INTER_ERROR3 / (N-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!      INTERPOLATING POSITION Vs PRESSURE CHANNEL 2
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


  CASE(4)
  i=0
  XA_2=XPOS_EXP              ! LETTING THE POSITION TO XA
  YA_2=PRSRE_EXP2            ! LETTING THE PRESSURE OF THE SECOND CHANNEL TO YA


    DO I=2,N-1
```

```
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      PRSRE2_INTERPOLATED_2(I)=Y
      INTER_ERROR4   = INTER_ERROR4 + DSQRT(DABS((PRSRE2_INTERPOLATED_2(I)-PRSRE(I,2))/PR
SRE2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR4     = INTER_ERROR4 / (N-1)


!-------------------------------------------------------------------------------------
---
!       INTERPOLATING POSITION Vs LOSS OF CHARGE CHANNEL 1
!
!-------------------------------------------------------------------------------------
---


CASE(5)

  XA_2=XPOS_EXP
  YA_2=DPFT_EXP1


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPFT1_INTERPOLATED_2(I)=Y
      INTER_ERROR5   = INTER_ERROR5 + DSQRT(DABS((DPFT1_INTERPOLATED_2(I)-DPFT(I,1))/DPFT
1_INTERPOLATED_2(I))**2)


    END DO

    INTER_ERROR5     = INTER_ERROR5 / (N-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!       INTERPOLATING POSITION Vs LOSS OF CHARGE CHANNEL 2
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


CASE(6)


  XA_2=XPOS_EXP
  YA_2=DPFT_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPFT2_INTERPOLATED_2(I)=Y
      INTER_ERROR6   = INTER_ERROR6 + DSQRT(DABS((DPFT2_INTERPOLATED_2(I)-DPFT(I,2))/DPFT
2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR6     = INTER_ERROR6 / (N-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!       INTERPOLATING POSITION Vs VOID FRACTION
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

CASE(7)
```

```
  XA_2=XPOS_EXP
  YA_2=VOID_FRCTN_EXP1


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      VOID_FRCTN1_INTERPOLATED_2(I)      =       Y

      IF(VOID_FRCTN1_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR7 = INTER_ERROR7 + DABS(VOID_FRCTN1_INTERPOLATED_2(I)-VOID_FRCTN(I,1)
)
        CYCLE
      END IF


      INTER_ERROR7  = INTER_ERROR7 + DSQRT(DABS((VOID_FRCTN1_INTERPOLATED_2(I)-VOID_FRCT
N(I,1))/VOID_FRCTN1_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR7    = INTER_ERROR7 / (N-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!       INTERPOLATING POSITION Vs VOID FRACTION 1
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


CASE(8)


  XA_2=XPOS_EXP
  YA_2=VOID_FRCTN_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      VOID_FRCTN2_INTERPOLATED_2(I)=Y

       IF(VOID_FRCTN2_INTERPOLATED_2(I).LT. 1.0E-5) THEN
            INTER_ERROR8 = INTER_ERROR8 + DABS(VOID_FRCTN2_INTERPOLATED_2(I)-VOID_FRCTN(
I,2))
        CYCLE
       END IF



      INTER_ERROR8  = INTER_ERROR8 + DSQRT(DABS((VOID_FRCTN2_INTERPOLATED_2(I)-VOID_FRCT
N(I,2))/VOID_FRCTN2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR8    = INTER_ERROR8 / (N-1)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!       INTERPOLATING POSITION Vs VOID FRACTION 1 GAS PHASE
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
  CASE(9)

    DO I=2,N-1
      CALL LINEINTERPOLATION(XPOSITION_GAS,AIRFLOW_EXP1,AUX2,XPOSITION(I),Y)
      GASFLOW1_INTERPOLATED_2(I)              =        Y

       IF(GASFLOW1_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR9 = INTER_ERROR9 + DABS(GASFLOW1_INTERPOLATED_2(I)-GASFLOW(I,1))
        CYCLE
       END IF

      INTER_ERROR9  = INTER_ERROR9 + DSQRT(DABS((GASFLOW1_INTERPOLATED_2(I)-GASFLOW(I,1)
)/GASFLOW1_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR9    = INTER_ERROR9 / (N-1)



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!       INTERPOLATING POSITION Vs VOID FRACTION 2 GAS PHASE
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 CASE(10)

    DO I=2,N-1
      CALL LINEINTERPOLATION(XPOSITION_GAS,AIRFLOW_EXP2,AUX2,XPOSITION(I),Y)
      GASFLOW2_INTERPOLATED_2(I)=Y
       IF(GASFLOW2_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR10 = INTER_ERROR10 + DABS(GASFLOW2_INTERPOLATED_2(I)-GASFLOW(I,2))
        CYCLE
       END IF

      INTER_ERROR10 = INTER_ERROR10 + DSQRT(DABS((GASFLOW2_INTERPOLATED_2(I)-GASFLOW(I,2
))/GASFLOW2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR10   = INTER_ERROR10 / (N-1)


CASE(11)

  XA_2=XPOS_EXP
  YA_2=DPRAD_EXP


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPRAD1_INTERPOLATED_2(I)=Y
      INTER_ERROR11 = INTER_ERROR11 + DSQRT(DABS((DPRAD1_INTERPOLATED_2(I)-DP_RAD(I))/DP
RAD1_INTERPOLATED_2(I))**2)


    END DO
```

```
    INTER_ERROR11    = INTER_ERROR11 / (N-1)
        END SELECT
    END DO




!-------------------------------------------------------------------------------------
-----------------------


   TOTAL_ERROR=(W1*INTER_ERROR1)+(W2*INTER_ERROR2)+(W3*INTER_ERROR3)+(W4*INTER_ERROR4)+(
W7*INTER_ERROR7)+(W8*INTER_ERROR8)+(W9*INTER_ERROR9)+(W10*INTER_ERROR10)+(W11*INTER_ERRO
R11)
!-------------------------------------------------------------------------------------
-----------------------

   IF(FLAG.EQ..TRUE.) THEN
        TOTAL_ERROR     =    -1
        RETURN

    ENDIF


IF (TOTAL_ERROR.LT.1E-7) THEN


    WRITE (8,*) 'GENERATION NUMBER ..............', EVOL
    WRITE (8,*) 'POPULATION NUMBER ..............', POP_NUM
    WRITE (8,*)
    WRITE (8,*) '-------------------------------------------------------------------------
----------------'



   WRITE(8,200) 'POSITION','LIQ FLOW 1','LIQ FLOW 1','LIQ FLOW 2','LIQ FLOW','PRESSURE 1
','PRESSURE1','PRESSURE 2','PRESSURE2','LOSS CHARGE1','LOSS CHARGE','LOSS CHARGE 2','LOS
S CHARGE',&
   &'VF 1','VF 1','VF2','VF2'
   WRITE(8,200)  'REAL','EXP','SIM','EXP','SIM','EXP','SIM','EXP','SIM','EXP','SIM','EXP
','SIM','EXP','SIM','EXP','SIM'
   WRITE(8,200) 'M','KG/HR','KG/HR','KG/HR','KG/HR','PA','PA','PA','PA','PA','PA','PA','
PA','ADIM','ADIM'


!  READING THE VALUES OF THE LIQUID
   rewind(unit=17)
   DO K=2,N-1
! WRITING THE INTERPOLATED VALUES AND THE EXPERIMENTAL VALUES FOR THE LIQUID
   WRITE (8,100) XPOSITION(K),LIQFLOW1_INTERPOLATED_2(K), LIQFLOW(K,1),&
   &LIQFLOW2_INTERPOLATED_2(K),LIQFLOW(K,2),&
   &PRSRE1_INTERPOLATED_2(K),PRSRE(K,1),&
   &PRSRE2_INTERPOLATED_2(K),PRSRE(K,2),    &
   &DPFT1_INTERPOLATED_2(K),DPFT(K,1),&
   &DPFT2_INTERPOLATED_2(K),DPFT(K,2),&
   &VOID_FRCTN1_INTERPOLATED_2(K),VOID_FRCTN(K,1),&
   &VOID_FRCTN2_INTERPOLATED_2(K),VOID_FRCTN(K,2)
   END DO


   WRITE (8,*)
   WRITE (8,*) 'GAS SIMULATION'
   WRITE (8,*) 'POSITION  GAS FLOW1  GAS FLOW1    GAS FLOW 2  GAS FLOW 2'
   WRITE (8,*) '  REAL      EXP        SIM          EXP        SIM'
```

```
   WRITE (8,*) '   M         PA          PA           PA          PA'

   DO K=2,N-1
    WRITE(8,500) XPOSITION(K),GASFLOW1_INTERPOLATED_2(K),GASFLOW(K,1),GASFLOW2_INTERPOLA
TED_2(K),GASFLOW(K,2)
   END DO

!-------------------------------------------------------------------------------------
-----------------------------------


   WRITE (8,600) 'DIFFERENCE BETWEEN LIQUID FLOW RATE SUBCHANNEL 1   ',INTER_ERROR1
   WRITE (8,600) 'DIFFERENCE BETWEEN LIQUID FLOW RATE SUBCHANNEL 2   ',INTER_ERROR2
   WRITE (8,600) 'DIFFERENCE BETWEEN PRESSURE SUBCHANNEL 1           ',INTER_ERROR3
   WRITE (8,600) 'DIFFERENCE BETWEEN PRESSURE SUBCHANNEL 2           ',INTER_ERROR4
   WRITE (8,600) 'DIFFERENCE BETWEEN AXIAL PRESSURE SUBCHANNEL 1     ',INTER_ERROR5
   WRITE (8,600) 'DIFFERENCE BETWEEN AXIAL PRESSURE SUBCHANNEL 2     ',INTER_ERROR6
   WRITE (8,600) 'DIFFERENCE BETWEEN VOID FRACTION SUBCHANNEL 1      ',INTER_ERROR7
   WRITE (8,600) 'DIFFERENCE BETWEEN VOID FRACTION SUBCHANNEL 2      ',INTER_ERROR8
   WRITE (8,600) 'DIFFERENCE BETWEEN GAS FLOW RATE SUBCHANNEL 1      ',INTER_ERROR9
   WRITE (8,600) 'DIFFERENCE BETWEEN GAS FLOW RATE SUBCHANNEL 2      ',INTER_ERROR10
   WRITE (8,600) 'DIFFERENCE BETWEEN DP RADIAL                       ',INTER_ERROR11
   WRITE (8,600) 'TOTAL ERROR..................................... ',TOTAL_ERROR

!-------------------------------------------------------------------------------------
------------------------------

    IF (EVOL.EQ.GENERATION ) THEN

        WRITE(8,'(A)') '----------------- EXPERIMENTAL VALUES LIQUID PHASE--------------
-----------'
        DO I=1,AUX
        WRITE(8,700) XPOS_EXP(I),LIQFLOW_EXP(I),LIQFLOW_EXP2(I),PRSRE_EXP1(I),PRSRE_EXP2
(I),DPFT_EXP1(I),DPFT_EXP2(I),&
        & VOID_FRCTN_EXP1(I),VOID_FRCTN_EXP2(I)
        END DO

        WRITE(8,'(A)') '----------------- EXPERIMENTAL VALUES GAS PHASE-----------------
--------'
        DO I=1,AUX2
        WRITE(8,800) XPOSITION_GAS(I),AIRFLOW_EXP1(I),AIRFLOW_EXP2(I)
        END DO
    END IF

!-------------------------------------------------------------------------------------
------------------------------


END IF


OPEN(UNIT=78,FILE='D:\GRAPH_DATA.RES',STATUS='UNKNOWN')
DO K=1,AUX2
WRITE (78,1) XPOSITION(K+1),LIQFLOW1_INTERPOLATED_2(K+1), LIQFLOW(K+1,1),LIQFLOW2_INTERP
OLATED_2(K+1),LIQFLOW(K+1,2),&
   &PRSRE1_INTERPOLATED_2(K+1)/1000,PRSRE(K+1,1)/1000,PRSRE2_INTERPOLATED_2(K+1)/1000,PR
SRE(K+1,2)/1000,&
   &DPFT1_INTERPOLATED_2(K+1)/1000,DPFT(K+1,1)/1000,DPFT2_INTERPOLATED_2(K+1)/1000,DPFT(
K+1,2)/1000,&
   &VOID_FRCTN1_INTERPOLATED_2(K+1),VOID_FRCTN(K+1,1),VOID_FRCTN2_INTERPOLATED_2(K+1),VO
ID_FRCTN(K+1,2),&
   &GASFLOW1_INTERPOLATED_2(K+1),GASFLOW(K+1,1),GASFLOW2_INTERPOLATED_2(K+1),GASFLOW(K+1
,2), DP_RAD (K+1)/1000&
```

```
! Printing the experiments values

   &,XPOS_EXP(K),LIQFLOW_EXP(K),LIQFLOW_EXP2(K),PRSRE_EXP1(K)/1000,PRSRE_EXP2(K)/1000,DP
FT_EXP1(K)/1000,DPFT_EXP2(K)/1000&
   &,VOID_FRCTN_EXP1(K),VOID_FRCTN_EXP2(K),DPRAD_EXP(K)/1000,XPOSITION_GAS(K),AIRFLOW_EX
P1(K),AIRFLOW_EXP2(K)
END DO



DO K=AUX2,AUX
WRITE (78,1) XPOSITION(K+1),LIQFLOW1_INTERPOLATED_2(K+1), LIQFLOW(K+1,1),LIQFLOW2_INTERP
OLATED_2(K+1),LIQFLOW(K+1,2),&
   &PRSRE1_INTERPOLATED_2(K+1)/1000,PRSRE(K+1,1)/1000,PRSRE2_INTERPOLATED_2(K+1)/1000,PR
SRE(K+1,2)/1000,&
   &DPFT1_INTERPOLATED_2(K+1)/1000,DPFT(K+1,1)/1000,DPFT2_INTERPOLATED_2(K+1)/1000,DPFT(
K+1,2)/1000,&
   &VOID_FRCTN1_INTERPOLATED_2(K+1),VOID_FRCTN(K+1,1),VOID_FRCTN2_INTERPOLATED_2(K+1),VO
ID_FRCTN(K+1,2),&
   &GASFLOW1_INTERPOLATED_2(K+1),GASFLOW(K+1,1),GASFLOW2_INTERPOLATED_2(K+1),GASFLOW(K+1
,2),DP_RAD(K+1)/1000&

! Printing the experiments values

   &,XPOS_EXP(K),LIQFLOW_EXP(K),LIQFLOW_EXP2(K),PRSRE_EXP1(K)/1000,PRSRE_EXP2(K)/1000,DP
FT_EXP1(K)/1000,DPFT_EXP2(K)/1000&
   &,VOID_FRCTN_EXP1(K),VOID_FRCTN_EXP2(K),DPRAD_EXP(K)/1000
END DO
DO K=AUX2,N-2
WRITE (78,1) XPOSITION(K+1),LIQFLOW1_INTERPOLATED_2(K+1), LIQFLOW(K+1,1),LIQFLOW2_INTERP
OLATED_2(K+1),LIQFLOW(K+1,2),&
   &PRSRE1_INTERPOLATED_2(K+1)/1000,PRSRE(K+1,1)/1000,PRSRE2_INTERPOLATED_2(K+1)/1000,PR
SRE(K+1,2)/1000,&
   &DPFT1_INTERPOLATED_2(K+1)/1000,DPFT(K+1,1)/1000,DPFT2_INTERPOLATED_2(K+1)/1000,DPFT(
K+1,2)/1000,&
   &VOID_FRCTN1_INTERPOLATED_2(K+1),VOID_FRCTN(K+1,1),VOID_FRCTN2_INTERPOLATED_2(K+1),VO
ID_FRCTN(K+1,2),&
   &GASFLOW1_INTERPOLATED_2(K+1),GASFLOW(K+1,1),GASFLOW2_INTERPOLATED_2(K+1),GASFLOW(K+1
,2),DP_RAD(K+1)/1000

END DO



CLOSE(UNIT=78)

1 FORMAT (22(1X, G14.7),  10(:,1X, G14.7), 3(:,1X, G14.7))
2 FORMAT (22(1X, G14.7),  10(:,1X, G14.7))
3 FORMAT (22(1X, G14.7))

100 FORMAT (F6.4, 12(4X, F9.2),4(4X,F7.5))
200 FORMAT (T2, A, T13, A, T26, A, T39, A, T52, A, T63, A, T77, A, &
      & T90, A, T103, A, T115, A, T129, A, T142, A, T157, A, T170, A,T180,A,T190,A,T200,
A)
500 FORMAT (F6.4, 6X, F6.4, 6X, F6.4, 6X, F6.4, 6X, F6.4)
600 FORMAT (/A,F19.5)
700 FORMAT (9(1X,G11.4))
800 FORMAT (3(1X,G11.4))
RETURN

END SUBROUTINE INTERPOLATION2
```

```
MODULE INTERPOLATION_VAR
IMPLICIT NONE
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XA,YA,LIQFLOW1_INTERPOLATED,LIQFLOW2_INTERP
OLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: PRSRE1_INTERPOLATED,PRSRE2_INTERPOLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: DPFT1_INTERPOLATED,DPFT2_INTERPOLATED,GASFL
OW1_INTERPOLATED,GASFLOW2_INTERPOLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XPOS_EXP,LIQFLOW_EXP,LIQFLOW_EXP2,PRSRE_EXP
1
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: PRSRE_EXP2,DPFT_EXP1,DPFT_EXP2
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XPOSITION_GAS,AIRFLOW_EXP1,AIRFLOW_EXP2

CONTAINS
SUBROUTINE ALLOCATION_VAR_INTERPOLATION(AUX,AUX2,N)
INTEGER,INTENT(IN)::AUX,AUX2,N
ALLOCATE (XA(AUX),YA(AUX),LIQFLOW1_INTERPOLATED(N-2),LIQFLOW2_INTERPOLATED(N-2))
ALLOCATE (PRSRE1_INTERPOLATED(N-2),PRSRE2_INTERPOLATED(N-2),DPFT1_INTERPOLATED(N-2),DPFT
2_INTERPOLATED(N-2))
ALLOCATE (GASFLOW1_INTERPOLATED(N-2),GASFLOW2_INTERPOLATED(N-2))
ALLOCATE (XPOS_EXP(AUX),LIQFLOW_EXP(AUX),LIQFLOW_EXP2(AUX),PRSRE_EXP1(AUX),PRSRE_EXP2(AU
X),DPFT_EXP1(AUX),DPFT_EXP2(AUX))
ALLOCATE (AIRFLOW_EXP1(AUX2),AIRFLOW_EXP2(AUX2),XPOSITION_GAS(AUX2))

END SUBROUTINE ALLOCATION_VAR_INTERPOLATION


SUBROUTINE DEALLOCATION_VAR_INTERPOLATION
    INTEGER A,B,C,D

    DEALLOCATE (XA,YA,LIQFLOW1_INTERPOLATED,LIQFLOW2_INTERPOLATED,STAT=A)
    DEALLOCATE (PRSRE1_INTERPOLATED,PRSRE2_INTERPOLATED,DPFT1_INTERPOLATED,DPFT2_INTERPO
LATED,STAT=B)
    DEALLOCATE (GASFLOW1_INTERPOLATED,GASFLOW2_INTERPOLATED,STAT=C)
    DEALLOCATE (XPOS_EXP,LIQFLOW_EXP,LIQFLOW_EXP2,PRSRE_EXP1,PRSRE_EXP2)
    DEALLOCATE (AIRFLOW_EXP1,STAT=D)
    DEALLOCATE (AIRFLOW_EXP2,STAT=D)
    DEALLOCATE (XPOSITION_GAS,STAT=D)
    DEALLOCATE (DPFT_EXP1,STAT=D)
    DEALLOCATE (DPFT_EXP2,STAT=D)




    RETURN
END SUBROUTINE DEALLOCATION_VAR_INTERPOLATION



END
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------- INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS PROGRAMME INTERPOLATION    Ce sous programme realise l'interpolation entre
!                                   les valeurs expérimentales et les valeurs simulées
!                                   Le but de ce sous-programme est de calculer l'erreur
!                                   entre ces valeurs.




SUBROUTINE INTERPOLATION
USE VAR1
USE GA_VAR

IMPLICIT NONE
INTEGER I,J,K
!-------------------------------------------------------------------------------
!
!-------------------------------------------------------------------------------
DOUBLE PRECISION:: INTER_ERROR1,INTER_ERROR2,INTER_ERROR3
DOUBLE PRECISION:: INTER_ERROR4,INTER_ERROR5,INTER_ERROR6,INTER_ERROR7,INTER_ERROR8
DOUBLE PRECISION:: INTER_ERROR9,INTER_ERROR10,INTER_ERROR11,SUM_DP_RAD_BGN
DOUBLE PRECISION, DIMENSION(N)::LIQFLOW1_INTERPOLATED_2,LIQFLOW2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::PRSRE1_INTERPOLATED_2,PRSRE2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::DPFT1_INTERPOLATED_2,DPFT2_INTERPOLATED_2,DPRAD1_INTERPO
LATED_2
DOUBLE PRECISION, DIMENSION(N)::GASFLOW1_INTERPOLATED_2,GASFLOW2_INTERPOLATED_2
DOUBLE PRECISION, DIMENSION(N)::VOID_FRCTN1_INTERPOLATED_2,VOID_FRCTN2_INTERPOLATED_2


INTEGER H



    REWIND (UNIT=7)


 INTER_ERROR1=0.0D0
 INTER_ERROR2=0.0D0
 INTER_ERROR3=0.0D0
 INTER_ERROR4=0.0D0
 INTER_ERROR5=0.0D0
 INTER_ERROR6=0.0D0
 INTER_ERROR7=0.0D0
 INTER_ERROR8=0.0D0
 INTER_ERROR9=0.0D0
 I=0


 DO H=1,11

 SELECT CASE (H)
```

```
 !-------------------------------------------------------------------------------------
----
!
!     Interpolation entre la position et qualité du canal 1
!       XA_2     =   Represente les valeurs de l'axe X
!                    Variable de type DOUBLE PRESICION
!
!       YA_2     =   Represente les valeurs de l'axe Y
!                    Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---


 CASE (1)

  XA_2=XPOS_EXP
  YA_2=LIQFLOW_EXP
    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      LIQFLOW1_INTERPOLATED_2(I)=Y
      INTER_ERROR1  = INTER_ERROR1 + DSQRT(DABS((LIQFLOW1_INTERPOLATED_2(I)-LIQFLOW(I,1)
)/LIQFLOW1_INTERPOLATED_2(I))**2)

    END DO
       INTER_ERROR1 = INTER_ERROR1 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et qualité du canal 2
!       XA_2     =   Represente les valeurs de l'axe X
!                    Variable de type DOUBLE PRESICION
!
!       YA_2     =   Represente les valeurs de l'axe Y
!                    Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---

  CASE(2)
  i=0
  XA_2=XPOS_EXP
  YA_2=LIQFLOW_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      LIQFLOW2_INTERPOLATED_2(I)=Y
      INTER_ERROR2  = INTER_ERROR2 + DSQRT(DABS((LIQFLOW2_INTERPOLATED_2(I)-LIQFLOW(I,2)
)/LIQFLOW2_INTERPOLATED_2(I))**2)

    END DO
      INTER_ERROR2  = INTER_ERROR2 / (N-1)
!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et pression du canal 1
!       XA_2     =   Represente les valeurs de l'axe X
!                    Variable de type DOUBLE PRESICION
!
!       YA_2     =   Represente les valeurs de l'axe Y
!                    Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---
```

```
 CASE(3)
  i=0
  XA_2=XPOS_EXP
  YA_2=PRSRE_EXP1


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      PRSRE1_INTERPOLATED_2(I)=Y
      INTER_ERROR3  = INTER_ERROR3 + DSQRT(DABS((PRSRE1_INTERPOLATED_2(I)-PRSRE(I,1))/PR
SRE1_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR3    = INTER_ERROR3 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et pression du canal 2
!       XA_2    =   Represente les valeurs de l'axe X
!                   Variable de type DOUBLE PRESICION
!
!       YA_2    =   Represente les valeurs de l'axe Y
!                   Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---


  CASE(4)
  i=0
  XA_2=XPOS_EXP
  YA_2=PRSRE_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      PRSRE2_INTERPOLATED_2(I)=Y
      INTER_ERROR4  = INTER_ERROR4 + DSQRT(DABS((PRSRE2_INTERPOLATED_2(I)-PRSRE(I,2))/PR
SRE2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR4    = INTER_ERROR4 / (N-1)


!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et la perte de pression du canal 1
!       XA_2    =   Represente les valeurs de l'axe X
!                   Variable de type DOUBLE PRESICION
!
!       YA_2    =   Represente les valeurs de l'axe Y
!                   Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---



CASE(5)

  XA_2=XPOS_EXP
  YA_2=DPFT_EXP1


    DO I=2,N-1
```

```
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPFT1_INTERPOLATED_2(I)=Y
      INTER_ERROR5  = INTER_ERROR5 + DSQRT(DABS((DPFT1_INTERPOLATED_2(I)-DPFT(I,1))/DPFT
1_INTERPOLATED_2(I))**2)


    END DO

    INTER_ERROR5    = INTER_ERROR5 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et la perte de pression du canal 2
!       XA_2    =   Represente les valeurs de l'axe X
!                   Variable de type DOUBLE PRESICION
!
!       YA_2    =   Represente les valeurs de l'axe Y
!                   Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---

CASE(6)


  XA_2=XPOS_EXP
  YA_2=DPFT_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPFT2_INTERPOLATED_2(I)=Y
      INTER_ERROR6  = INTER_ERROR6 + DSQRT(DABS((DPFT2_INTERPOLATED_2(I)-DPFT(I,2))/DPFT
2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR6    = INTER_ERROR6 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et la taux de vide du canal 1
!       XA_2    =   Represente les valeurs de l'axe X
!                   Variable de type DOUBLE PRESICION
!
!       YA_2    =   Represente les valeurs de l'axe Y
!                   Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---


CASE(7)


  XA_2=XPOS_EXP
  YA_2=VOID_FRCTN_EXP1


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      VOID_FRCTN1_INTERPOLATED_2(I)     =       Y

      IF(VOID_FRCTN1_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR7 = INTER_ERROR7 + DABS(VOID_FRCTN1_INTERPOLATED_2(I)-VOID_FRCTN(I,1)
)
```

```
        CYCLE
      END IF

      INTER_ERROR7  = INTER_ERROR7 + DSQRT(DABS((VOID_FRCTN1_INTERPOLATED_2(I)-VOID_FRCT
N(I,1))/VOID_FRCTN1_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR7    = INTER_ERROR7 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et la taux de vide du canal 2
!       XA_2    =   Represente les valeurs de l'axe X
!                   Variable de type DOUBLE PRESICION
!
!       YA_2    =   Represente les valeurs de l'axe Y
!                   Variable de type DOUBLE PRESICION
!
!-------------------------------------------------------------------------------------
---


CASE(8)

  XA_2=XPOS_EXP
  YA_2=VOID_FRCTN_EXP2


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      VOID_FRCTN2_INTERPOLATED_2(I)=Y

       IF(VOID_FRCTN2_INTERPOLATED_2(I).LT. 1.0E-5) THEN
            INTER_ERROR8 = INTER_ERROR8 + DABS(VOID_FRCTN2_INTERPOLATED_2(I)-VOID_FRCTN(
I,2))
        CYCLE
       END IF


      INTER_ERROR8  = INTER_ERROR8 + DSQRT(DABS((VOID_FRCTN2_INTERPOLATED_2(I)-VOID_FRCT
N(I,2))/VOID_FRCTN2_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR8    = INTER_ERROR8 / (N-1)

!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et le débit du gaz du canal 1
!
!-------------------------------------------------------------------------------------
---


  CASE(9)

    DO I=2,N-1
      CALL LINEINTERPOLATION(XPOSITION_GAS,AIRFLOW_EXP1,AUX2,XPOSITION(I),Y)
      GASFLOW1_INTERPOLATED_2(I)            =       Y
```

```
       IF(GASFLOW1_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR9 = INTER_ERROR9 + DABS(GASFLOW1_INTERPOLATED_2(I)-GASFLOW(I,1))
        CYCLE
       END IF


      INTER_ERROR9  = INTER_ERROR9 + DSQRT(DABS((GASFLOW1_INTERPOLATED_2(I)-GASFLOW(I,1)
)/GASFLOW1_INTERPOLATED_2(I))**2)

    END DO
    INTER_ERROR9    = INTER_ERROR9 / (N-1)


!-------------------------------------------------------------------------------------
---
!
!     Interpolation entre la position et le débit du gaz du canal 2
!
!-------------------------------------------------------------------------------------
---

 CASE(10)


    DO I=2,N-1
      CALL LINEINTERPOLATION(XPOSITION_GAS,AIRFLOW_EXP2,AUX2,XPOSITION(I),Y)
      GASFLOW2_INTERPOLATED_2(I)=Y
       IF(GASFLOW2_INTERPOLATED_2(I).LT. 1.0E-5) THEN
        INTER_ERROR10 = INTER_ERROR10 + DABS(GASFLOW2_INTERPOLATED_2(I)-GASFLOW(I,2))
        CYCLE
       END IF


      INTER_ERROR10 = INTER_ERROR10 + DSQRT(DABS((GASFLOW2_INTERPOLATED_2(I)-GASFLOW(I,2
))/GASFLOW2_INTERPOLATED_2(I))**2)


    END DO
    INTER_ERROR10   = INTER_ERROR10 / (N-1)


CASE(11)

  XA_2=XPOS_EXP
  YA_2=DPRAD_EXP


    DO I=2,N-1
      CALL LINEINTERPOLATION(XA_2,YA_2,AUX,XPOSITION(I),Y)
      DPRAD1_INTERPOLATED_2(I)=Y
      INTER_ERROR11 = INTER_ERROR11 + DSQRT(DABS((DPRAD1_INTERPOLATED_2(I)-DP_RAD(I))/DP
RAD1_INTERPOLATED_2(I))**2)


    END DO

    INTER_ERROR11   = INTER_ERROR11 / (N-1)
        END SELECT
    END DO

!-------------------------------------------------------------------------------------
----------------------
!   TOTAL_ERROR_LOC     =   La somme de tous les erreurs
!                           Variable de type DOUBLE PRECISION
!
!
!
```

```
   TOTAL_ERROR_LOC(POP_NUM)=(W1*INTER_ERROR1)+(W2*INTER_ERROR2) &  +(W3*INTER_ERROR3)+(W4
*INTER_ERROR4)+(W7*INTER_ERROR7)
   &  +(W8*INTER_ERROR8)+(W9*INTER_ERROR9)+(W10*INTER_ERROR10)+(W11*INTER_ERROR11)
!-------------------------------------------------------------------------------------
-----------------------


END SUBROUTINE INTERPOLATION
```

```
!------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:    ARMANDO NAVA
!   DATE:           13-01-04
!

!   PROGRAMME GA.   Ce programme a comme but de optimiser les coefficients de mélange
!                   dans des sous-canaux interconnectés et de géométrie carrée.
!                   La méthode basée sur les Algorithmes génétiques a été utilisée
!                   pour l'optimisation des paramètres du mélange.
!
! Sous programme MUTATION
!
! Ce sous programme a comme but:
!
! Réaliser la mutation d'un chromosome, Deux options sont disponibles:
! Si la option de mutation OPTIONMUT est égal à 1; la mutation sera uniforme durant
! toute l'évolution de la simulation.

! Si la option de mutation OPTIONMUT est égal à 2; la mutation sera non-uniforme dépenda
nte
! toute l'évolution de la simulation.
! utilise l'équation suivante.
!
!                           PM=1-(A**(1-EVOL/GENERATION)**B).
! A             =   C'est un nombre aléatoire entre 0 et 1
! B             =   C'est le degré de la non-uniformité

!

!-------------------------------------------------------------------------------------
! Les variables d'entrée sont:
!-------------------------------------------------------------------------------------


! POP_SIZE      =   La taille de la population.
!                   Est une variable de type INTEGER.


! OPTIONMUT     =   Sélectionne du type de mutation.
!                   Est une variable de type INTEGER.

! RN1           =   Cette variable trouve un numéro aléatoire.
!                   Est une variable dimensionnel de type INTEGER.

! GENERATION    =   C'est le nombre des générations a Réaliser.

! EVOL          =   C'est le nombre de générations passées.




!-------------------------------------------------------------------------------------
! Les variables internes sont:
!-------------------------------------------------------------------------------------

! CHANGE_NUM    =   C'est le nombre de gènes à changer.

! MUT_POINT     =   C'est le point de mutation


!-------------------------------------------------------------------------------------
! Les variables de sorti sont:
!-------------------------------------------------------------------------------------
```

```
! POPULATION     =   La population déjà mutée.
!                    Est une variable dimensionnel de type INTEGER.


SUBROUTINE MUTATION
USE VAR1
USE GA_VAR

IMPLICIT NONE
REAL:: RN1(100),MUT_PROB
INTEGER CHANGE_NUM
INTEGER::I,J,MUT_POINT,RN_MUT1,RN_MUT2

IF(OPTIONMUT.EQ.1) THEN

MUT_PROB=PM

IF (EVOL.EQ.GENERATION) RETURN


    IF(EVOL.EQ.INT(GENERATION*0.5)) THEN
        MUT_PROB=MUT_PROB*0.5
    ELSE IF((EVOL.GT.INT(GENERATION*0.5)).AND.(EVOL.LT.INT(GENERATION*0.85))) THEN
        MUT_PROB=MUT_PROB*0.75
    END IF




    IF (EVOL.GE.INT (GENERATION*0.75) ) THEN
     MUT_PROB=0.0
    END IF

IF (MUT_PROB.NE.0.0) THEN

CHANGE_NUM=CEILING(MUT_PROB*CHROM_SIZE/100)

    DO J=1,POP_SIZE
        DO I=1,CHANGE_NUM
            CALL RANDOM1(RN1)



            MUTATION_POINT: SELECT CASE(I)
            CASE (1)
              MUT_POINT=CEILING((I*CHROM_SIZE-1)*RN1(I))
            CASE (2)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (3)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (4)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (5)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (6)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (7)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (8)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))
            CASE (9)
              MUT_POINT=INT((I*COEFF_SIZE-1)+(RN1(1)*COEFF_SIZE))


            END SELECT MUTATION_POINT
```

```
            IF (MUT_POINT.EQ.0) MUT_POINT=7


            IF (MUT_POINT.GT.CHROM_SIZE) MUT_POINT=15
            IF (POPULATION(J,MUT_POINT).EQ.0) THEN
                POPULATION(J,MUT_POINT)=1
                    ELSE IF (POPULATION(J,MUT_POINT).EQ.1) THEN
                        POPULATION(J,MUT_POINT)=0
            END IF

        END DO
    END DO

END IF




    IF (EVOL.GE.INT (GENERATION*0.75) ) THEN
    CALL RANDOM1(RN1)
    RN_MUT1=CEILING(RN1(1)*POP_SIZE)    ! SEARCHING THE ROW TO MUTATE
    RN_MUT2=CEILING(RN1(7)*CHROM_SIZE)    ! SEARCHING THE COLUMN TO MUTATE
      IF (POPULATION(RN_MUT1,RN_MUT2) .EQ. 0) THEN
            POPULATION(RN_MUT1,RN_MUT2) = 1
        ELSE
            POPULATION(RN_MUT1,RN_MUT2) = 0
    END IF
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
! EVOLUTION DE LA MUTATION, UNE FOIS ARRIVE AU 90% DE LA EVOLUTION LA MUTATION SERA
! DE 1 BITE PAR GENERATION
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-

    END IF

END IF


IF (OPTIONMUT.EQ.2) THEN

    DO I=1,POP_SIZE
        DO J=1,CHROM_SIZE
            CALL RANDOM1(RN1)
             IF (RN1(1).LE.PM ) THEN
               IF(POPULATION(I,J).EQ.0) THEN
                POPULATION(I,J) = 1
                ELSE
                POPULATION(I,J) = 0
                END IF

            END IF

        END DO
    END DO

END IF

PM=1-(0.5**(1-EVOL/GENERATION)**3)

IF (EVOL.GE.INT(0.8*GENERATION)) PM=0.0



END SUBROUTINE MUTATION
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS-PROGRAMME OUTPUT_DATA.      Ce sous programme enregistre tous les paramètres
!                                    simulés, lorsque la simulation
!                                    totale est finie, soit optimisé ou non.
!
!                                    Le fichier GRAPFILE3.DAT peut-etre utilis. pour
!                                    traser les corbes avec Gnu-plot


      SUBROUTINE OUTPUT_DATA
      USE VAR1



      OPEN(UNIT=15,FILE=DATAFILE,STATUS='UNKNOWN')
      OPEN(UNIT=16,FILE='D:\GRAPFILE3.DAT',STATUS='UNKNOWN')
      WRITE(15,900)
      WRITE(15,1200)

      DO I=2,N2
       WRITE(15,1300) XPOSITION(I),QUALITY(I,1),QUALITY(I,2),LIQFLOW(I,1),LIQFLOW(I,2)&
       &,GASFLOW(I,1),GASFLOW(I,2),DWL(I),DWG(I)

      END DO

      WRITE(15,900)
      WRITE(15,1600)

      DO I=2,N2
         WRITE(15,1500) X_PRSRE(I),PRSRE(I,1),PRSRE(I,2),DPFT(I,1),DPFT(I,2),DP_RAD(I)
      END DO


     DO K=1,AUX2


    WRITE (16,1) XPOSITION(K+1), LIQFLOW(K+1,1),LIQFLOW(K+1,2),PRSRE(K+1,1)/1000,PRSRE(K
+1,2)/1000,&
   &DPFT(K+1,1)/1000,DPFT(K+1,2)/1000,VOID_FRCTN(K+1,1),VOID_FRCTN(K+1,2),GASFLOW(K+1,1)
,GASFLOW(K+1,2)&
   &, DP_RAD (K+1)/1000&

! Printing the experiments values

   &,XPOS_EXP(K),LIQFLOW_EXP(K),LIQFLOW_EXP2(K),PRSRE_EXP1(K)/1000,PRSRE_EXP2(K)/1000,DP
FT_EXP1(K)/1000,&
   &DPFT_EXP2(K)/1000&
   &,VOID_FRCTN_EXP1(K),VOID_FRCTN_EXP2(K),DPRAD_EXP(K)/1000,XPOSITION_GAS(K),AIRFLOW_EX
P1(K),AIRFLOW_EXP2(K)
    END DO



    DO K=AUX2,AUX
    WRITE (16,1) XPOSITION(K+1), LIQFLOW(K+1,1),LIQFLOW(K+1,2),PRSRE(K+1,1)/1000,PRSRE(K
+1,2)/1000,&
   &DPFT(K+1,1)/1000,DPFT(K+1,2)/1000,VOID_FRCTN(K+1,1),VOID_FRCTN(K+1,2),GASFLOW(K+1,1)
,&
   &GASFLOW(K+1,2),DP_RAD(K+1)/1000&
```

```
! Printing the experiments values

   &,XPOS_EXP(K),LIQFLOW_EXP(K),LIQFLOW_EXP2(K),PRSRE_EXP1(K)/1000,PRSRE_EXP2(K)/1000,DP
FT_EXP1(K)/1000,&
   &DPFT_EXP2(K)/1000,VOID_FRCTN_EXP1(K),VOID_FRCTN_EXP2(K),DPRAD_EXP(K)/1000
    END DO


    DO K=AUX2,N-2
    WRITE (16,1) XPOSITION(K+1), LIQFLOW(K+1,1),LIQFLOW(K+1,2),PRSRE(K+1,1)/1000,PRSRE(K
+1,2)/1000,&
    &DPFT(K+1,1)/1000,DPFT(K+1,2)/1000,VOID_FRCTN(K+1,1),VOID_FRCTN(K+1,2),GASFLOW(K+1,1
)&
    &,GASFLOW(K+1,2),DP_RAD(K+1)/1000
    END DO


      CLOSE (UNIT=15)
      CLOSE (UNIT=16)

  1   FORMAT (22(1X, G14.7),  10(:,1X, G14.7), 3(:,1X, G14.7))
  900 FORMAT(1H1,//,T10,3(4X,'* SIMULATION OUTPUT *'))
 1000 FORMAT(//,T40,'CHANNEL # ',I2,//,T10,'POSITION',T21,'VOID',T27,'LI&
     &Q.MASS FLUX',T43,'GAS MASS FLUX',T59,'TOTAL MASS FLUX',T77,'PRESSU&
     &RE',/,T12,' (M.)',T29,'(KG/M2/S)',T45,'(KG/M2/S)',T62,'(KG/M2/S)',&
     &T78,'(PA.)',/)
 1100 FORMAT(T12,F5.3,T20,F5.4,T30,F7.2,T45,F6.3,T63,F7.2,T77,F10.3)
 1200 FORMAT(//,T10,'POSITION',T26,'QUALITY',T42,' LIQUID  FLOW RATE ',T&
     &66,'GAS FLOW RATE',T83,'INTEG. MASS TRANSF.',/,T12,'(M.)',T22,'CH&
     &# 1',T31,'CH # 2',T44,'CH # 1',T53,'CH # 2',T65,'CH # 1',T74,'CH #&
     & 2',T84,'LIQUID',T97,'GAS',/,T48,'(KG/H)',T69,'(KG/H)',T90,'(KG/H)&
     &',/)
 1300 FORMAT(T11, F5.3, T21, F8.6, T31, F8.6, T44, F6.1, T52, F6.1, &
     & T66, F5.3, T75, F6.3, T83, F7.3, T94, F6.3, T104, F5.3, T112, &
     & T66, F5.3, T75, F6.3, T83, F7.3, T94, F6.3, T104, F5.3)
1400 FORMAT (//,T10,'POSITION',T28,'ABS.PRESSURE (PA.)',T56,'FRICT.+ ACC. PRESS. DROP (P
A.)'&
&,T94,'RADIAL PRESS. DIFF. (PA.)',/,T12,'(M.)&
     &',T26,'CH # 1',T43,'CH # 2',T60,'CH # 1',T76,'CH # 2',T101,'(P1 -&
     &P2)',/)
1500 FORMAT(T11,F5.3,T24,F9.1,T41,F9.1,T58,F8.1,T74,F8.1,T100,F8.1)

1600 FORMAT(22(1X,G11.4))

 1700 FORMAT (9(1X,G11.4))
 1800 FORMAT (3(1X,G11.4))
end
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME GA.    Ce programme a comme but de optimiser les coefficients de mélange
!                    dans des sous-canaux interconnectés et de géométrie carrée.
!                    La méthode basée sur les Algorithmes génétiques a été utilisée
!                    pour l'optimisation des paramètres du mélange.
!
! Sous programme PARENTS
!
! Ce sous programme a comme but:
!
! Realiser le croissement entre deux chromosomes, utilise la methode 'single cut point'

SUBROUTINE PARENTS
USE GA_VAR
USE VAR1
IMPLICIT NONE


    CO_COD      (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,0*COEFF_SIZE+1:1*CO
EFF_SIZE)
    DEQ_COD     (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,1*COEFF_SIZE+1:2*CO
EFF_SIZE)
    DALFA_COD   (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,2*COEFF_SIZE+1:3*CO
EFF_SIZE)
    KTR_COD     (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,3*COEFF_SIZE+1:4*CO
EFF_SIZE)
    LAMBDA_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,4*COEFF_SIZE+1:5*CO
EFF_SIZE)
    COEFF1_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,5*COEFF_SIZE+1:6*CO
EFF_SIZE)
    COEFF2_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,6*COEFF_SIZE+1:7*CO
EFF_SIZE)
    COEFF3_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,7*COEFF_SIZE+1:8*CO
EFF_SIZE)
    COEFF4_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,8*COEFF_SIZE+1:9*CO
EFF_SIZE)

    CALL DECODEF

    DO POP_NUM=1,POP_SIZE

        CALL INIT_VAR
        CO_DEC                  =   CO_DEC_VEC      (POP_NUM)
        KTR_DEC                 =   KTR_DEC_VEC     (POP_NUM)
        DEQ_DEC                 =   DEQ_DEC_VEC     (POP_NUM)
        DALFA_DEC               =   DALFA_DEC_VEC   (POP_NUM)
        LAMBDA_DEC              =   LAMBDA_DEC_VEC  (POP_NUM)
        COEFF1_DEC              =   COEFF1_DEC_VEC  (POP_NUM)
        COEFF2_DEC              =   COEFF2_DEC_VEC  (POP_NUM)
        COEFF3_DEC              =   COEFF3_DEC_VEC  (POP_NUM)
        COEFF4_DEC              =   COEFF4_DEC_VEC  (POP_NUM)


        CALL FUN_MIN
        CONV_FLOW_EQ    (POP_NUM)   =   CONV_CRTRIA
        TOTAL_ERROR_VEC (POP_NUM)   =   TOTAL_ERROR

    IF (TOTAL_ERROR_VEC (POP_NUM).LT.0.0) THEN
```

```
            DO WHILE((TOTAL_ERROR_VEC(POP_NUM).LE.0.0) )
             CALL FIRST_GENE
             POPULATION(POP_NUM,1:CHROM_SIZE)  =OFFSPRING_VEC(1:CHROM_SIZE)

               CO_COD      (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,0*COEFF
_SIZE+1:1*COEFF_SIZE)
               DEQ_COD     (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,1*COEFF
_SIZE+1:2*COEFF_SIZE)
               DALFA_COD   (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,2*COEFF
_SIZE+1:3*COEFF_SIZE)
               KTR_COD     (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,3*COEFF
_SIZE+1:4*COEFF_SIZE)
               LAMBDA_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,4*COEFF
_SIZE+1:5*COEFF_SIZE)
               COEFF1_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,5*COEFF
_SIZE+1:6*COEFF_SIZE)
               COEFF2_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,6*COEFF
_SIZE+1:7*COEFF_SIZE)
               COEFF3_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,7*COEFF
_SIZE+1:8*COEFF_SIZE)
               COEFF4_COD  (1:POP_SIZE,1:COEFF_SIZE)   =  POPULATION(1:POP_SIZE,8*COEFF
_SIZE+1:9*COEFF_SIZE)

               CALL DECODEF
               CALL INIT_VAR
               CO_DEC                      =   CO_DEC_VEC       (POP_NUM)
               KTR_DEC                     =   KTR_DEC_VEC      (POP_NUM)
               DEQ_DEC                     =   DEQ_DEC_VEC      (POP_NUM)
               DALFA_DEC                   =   DALFA_DEC_VEC    (POP_NUM)
               LAMBDA_DEC                  =   LAMBDA_DEC_VEC   (POP_NUM)
               COEFF1_DEC                  =   COEFF1_DEC_VEC   (POP_NUM)
               COEFF2_DEC                  =   COEFF2_DEC_VEC   (POP_NUM)
               COEFF3_DEC                  =   COEFF3_DEC_VEC   (POP_NUM)
               COEFF4_DEC                  =   COEFF4_DEC_VEC   (POP_NUM)

               CALL FUN_MIN
               CONV_FLOW_EQ    (POP_NUM)   =   CONV_CRTRIA
               TOTAL_ERROR_VEC (POP_NUM)   =   TOTAL_ERROR
            END DO

        END IF

        CALL INTERPOLATION

    END DO



END SUBROUTINE
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. -------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   MODULE PROPERTIES   Ce module contient les fonctions suivantes:
!                       PHI
!




MODULE PROPERTIES
IMPLICIT NONE
REAL,DIMENSION(8):: XVF=(/66.7D01,-41.7679D04,14.2300368D07,-25.471449346D09,22.62867161
534D11,-78.647874646389D12,1.092D0,0.09946D0/)
REAL,DIMENSION(8):: XPHI=(/1185.515,-100211.0,3373694.0,923.663,-67888.0,1774329.0,726.2
07,-37129.497/)
REAL,DIMENSION(4):: XROL=(/999.8,998.4026,995.8176,992.2604/)
REAL,DIMENSION(5):: XW=(/0.1746,1.082,-4.56,5.072,-1.757/)

CONTAINS

      FUNCTION PHI(QUALITY,TTLFLUX)
!     ************************************************
!     *                ---  P H I  ---                 *
!     *  TWO PHASE MULTIPLIER FROM EXP. DATA OF TAPUCU  *
!     ************************************************
      DOUBLE PRECISION:: QUALITY,TTLFLUX,PHI,PHI1,PHI2


      IF(TTLFLUX.LE.2080.) THEN
         PHI=QUALITY*(XPHI(1)+QUALITY*(XPHI(2)+QUALITY*XPHI(3)))+1.
      ELSE IF((TTLFLUX.GT.2080.).AND.(TTLFLUX.LT.3130)) THEN
        PHI1=QUALITY*(XPHI(1)+QUALITY*(XPHI(2)+QUALITY*XPHI(3)))+1.
        PHI2=QUALITY*(XPHI(4)+QUALITY*(XPHI(5)+QUALITY*XPHI(6)))+1.
        PHI=PHI1-(PHI2-PHI1)*(2080.-TTLFLUX)/(3130.-2080.)
      ELSE IF((TTLFLUX.GE.3130.).AND.(TTLFLUX.LT.3650.)) THEN
          PHI1=QUALITY*(XPHI(4)+QUALITY*(XPHI(5)+QUALITY*XPHI(6)))+1.
          PHI2=QUALITY*(XPHI(7)+XPHI(8)*QUALITY)+1.
          PHI=PHI1-(PHI2-PHI1)*(3130.-TTLFLUX)/(3650.-3130)
       ELSE
           PHI=QUALITY*(XPHI(7)+XPHI(8)*QUALITY)+1.
      END IF
      END FUNCTION PHI


      FUNCTION TUDIFF(GX,F0,VFX,RGX,RL,DL)
!     ************************************************
!     *             ---  T U D I F F  ---               *
!     * TURBULENT DIFFUSION COEFF. FOR TRANSVERSE MOMENTUM *
!     *      REF.: ROUHANI,                            *
!     ************************************************
!
!          GX   MASS FLUX OF THE CHANNEL
!          F0   FRICTION FACTOR
!         VFX   VOID FRACTION OF THE CHANNEL
!         RGX   GAS DENSITY
!          RL   LIQUID DENSITY
!          DL   CHARACTERISTIC LENGTH
!        ---------------------------------------
```

```
!
      DOUBLE PRECISION::GX,F0,VFX,RGX,RL,DL,TUDIFF

      TUDIFF=4.39D-03*DL*((F0/8.)**0.5)*GX/(VFX*RGX+(1.-VFX)*RL)

      END FUNCTION TUDIFF

      FUNCTION VOID(XX)
!     ***************************************************
!     *              ---  V O I D  ---                   *
!     *   VOID FRACTION AS A FUNCTION OF THE FLOW QUALITY  *
!     *   FROM THE EXP. DATA OF TAPUCU ET.AL               *
!     *   REF. TAPUCU,A.,                                  *
!     ***************************************************
!
      DOUBLE PRECISION::XX,X,V1,V2,VOID

      X=XX
      IF(X.LE.0.001) THEN


         VOID=2.241E8*X**3 - 6.3211E5*X**2 + 7.6836E2*X + 1.2781E-3


      ELSE

            VOID=2.54*X**(0.2809)

             END IF









      END FUNCTION VOID


      DOUBLE PRECISION FUNCTION WDNSTY(TEMPERATURE)
!     ***************************************************
!     *              ---  W D N S T Y  ---                *
!     * DENSITY OF THE LIQUID PHASE (10 DEG.c TO 40 DEG.c) *
!     ***************************************************
!
      DOUBLE PRECISION:: TEMPERATURE


      IF(TEMPERATURE.LE.10.) THEN
      WDNSTY=XROL(1)
        ELSE IF((TEMPERATURE.GE.10.).AND.(TEMPERATURE.LT.20.)) THEN
        WDNSTY=XROL(1)+(XROL(2)-XROL(1))*(TEMPERATURE-10.)/10.
          ELSE IF((TEMPERATURE.GE.20.).AND.(TEMPERATURE.LT.30.)) THEN
          WDNSTY=XROL(2)+(XROL(3)-XROL(2))*(TEMPERATURE-20.)/10.
            ELSE IF((TEMPERATURE.GE.30.).AND.(TEMPERATURE.LT.40.)) THEN
            WDNSTY=XROL(3)+(XROL(4)-XROL(3))*(TEMPERATURE-30.)/10.
              ELSE
              WDNSTY=XROL(4)
      END IF

      END FUNCTION WDNSTY
```

```
      DOUBLE PRECISION FUNCTION VISCTY(T)
!     ****************************************************
!     *             ---  V I S C T Y  ---                  *
!     *   LIQUID VISCOSITY AS A FUNCTION OF TEMPERATURE    *
!     * REF.: ASME STEAM TABLES;MEYER !.A.,R.B.MCCLINTOCK, *
!     *       G.J.SILVESTRI,AND R.!.SPENCER;APP.5,PP 72-75 *
!     *       (1967)                                       *
!     ****************************************************
!
      DOUBLE PRECISION T
      VISCTY=241.1D-07*10.**(247.8/(T+133.1))
      END FUNCTION VISCTY



      DOUBLE PRECISION FUNCTION FCTION(M,GX,XMU,GAP,HYD)
!     ****************************************************
!     *             ---  F C T I O N  ---                  *
!     *        SINGLE PHASE FRICTION COEFFICIENT          *
!     *           REF.: TAPUCU,A.                         *
!     *                 ROUHANI,                          *
!     ****************************************************
!
!            M   2 THE ROUHANI RELATIONSHIP IS USED
!                TO CALCULATE THE  FRICTION  COEFF.
!                AT THE GAP
!            M   1 THE TAPUCU CORRELATION  IS  USED
!                TO CALCULATE  THE  AXIAL  FRICTION
!                COEFFICIENT
!            ---------------------------------------
!
      INTEGER M
      DOUBLE PRECISION::GX,XMU,GAP,HYD,RE

      IF(M.EQ.2) THEN
      RE=2.*GAP*DABS(GX)/XMU
      FCTION=0.02*RE**(-0.2)
      RETURN
         ELSE IF(M.EQ.1) THEN
         RE=HYD*DABS(GX)/XMU
           IF(RE.LE.0.01) THEN
              FCTION=0.
              RETURN
           ELSE IF(RE.LE.1000.) THEN
              FCTION=64./RE
              RETURN
           ELSE
              FCTION=1.15*(0.0032+0.221/(RE**0.237))
              RETURN
           END IF
      ELSE
      FCTION=1.
      PRINT 1000
 1000 FORMAT(///,T10,'WARNING  WARNING  WARNING  WARNING  WARNING ',/,T1&
     &0,'-----------------------------------------',//,T12,'THE SELECT&
     &ED CORRELATION FOR THE FRICTION FACTOR IS NOT AVAILABLE  !!!!',//,&
     &T12,'THE FRICTION FACTOR IS TAKEN EQUAL TO ONE  !!!!',///)
      END IF
      RETURN
      END FUNCTION FCTION


      DOUBLE PRECISION FUNCTION GDNSTY(P,T,I)
!     ****************************************************
```

```
!      *              --- G D N S T Y ---                    *
!      *            DENSITY OF THE GAS PHASE                 *
!      *******************************************************
!
!             I = 0 AIR DENSITY AS IDEAL GAS
!             I = 1 WET AIR DENSITY FROM BAEHR'S CORRELATION
!      ------------------------------------------------------
       INTEGER I,R
       DOUBLE PRECISION::P,T
       DOUBLE PRECISION::A1,A2,A3,B1,B2,ZKELV,PSAT,TKELV

       PARAMETER(A1=16.1446,A2=-0.7116,A3=0.053126,B1=348.3D-5,B2=0.1316,R=287,ZKELV=273.
1)

       TKELV=ZKELV+T
       IF(I.EQ.0) THEN
       GDNSTY=P/R/TKELV
       RETURN
       ELSE IF(I.EQ.1) THEN
       PSAT=A1+T*(A2+T*A3)
       GDNSTY=(B1*P-B2*PSAT)/TKELV
       END IF
       RETURN
       END FUNCTION GDNSTY
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

       DOUBLE PRECISION FUNCTION TRPRCO(ITR,FLUX_YY,DL,XKINPT)
       USE VAR1
!      ******************************************
!      *         --- T R P R C O ---            *
!      *  TRANSVERSE PRESSURE LOSS COEFFICIENT  *
!      ******************************************
!
!          ITR = 2  A  !ONSTANT  VALUE  GIVEN  AS
!                   PROGRAMM INPUT IS RETURNED .
!          ITR = 1  A VARIABLE !OEFF. IS RETURNED
!
!      A CORRELATION OBTEINED FROM THE EXP.  DATA
!      OF TAPU!U ( ) AND FOR SINGLE PHASE FLOW IS
!      USED .
!      ------------------------------------------
       INTEGER ITR

       DOUBLE PRECISION CORR,DL,XKINPT,FLUX_YY
       REAL A0,A1,A2,B0,B1


       A1=0.568;A2=0.288;B0=0.372;B1=0.8619
       IF (ITR.EQ.2) THEN
         TRPRCO=XKINPT
         RETURN
        ELSE IF (ITR.EQ.1) THEN
           IF (FLUX_YY.LE.0.005) THEN
           TRPRCO=60.
           RETURN
             ELSE IF (GAP_WDTH.GE.0.00149) THEN
             CORR=1.
             ELSE IF (GAP_WDTH.LT.0.00149) THEN
             A0=-A1*FLUX_YY+A2
             CORR=(DL/GAP_WDTH)**A0
           END IF
```

```
      END IF
      TRPRCO=B0/(FLUX_YY**B1)*CORR
      RETURN
      END FUNCTION TRPRCO

      END MODULE PROPERTIES
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUT DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS PROGRAMME RANDOM   Sous programme utilisé pour générer nombres aleatoires


SUBROUTINE RANDOM1(RN1)
USE DFPORT
IMPLICIT NONE

INTEGER Count,RET,I
REAL,DIMENSION(100)::R,RN1
REAL,DIMENSION(100)::NUM
INTEGER, DIMENSION(1):: Seed
REAL*8 ELAPSED_TIME
RN1=0.0


DO I=1, 1E6
END DO



CALL SYSTEM_CLOCK( Count )
ELAPSED_TIME=TIMEF()
SEED=COUNT+ELAPSED_TIME+1358*R(1)**(2*(R(2)))
CALL RANDOM_SEED( PUT  = Seed )
CALL RANDOM_NUMBER( R )
NUM = R
RN1=NUM

CALL RANDOM_SEED( GET = Seed )

DO I=1, 1E6
END DO

END SUBROUTINE RANDOM1
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS-PROGRAMME RESULTS.     Ce sous programme garde dans le fichier INTER2_TXT les v
aleurs de la simulation
!

SUBROUTINE RESULTS
USE GA_VAR
USE VAR1
IMPLICIT NONE
INTEGER K,I

IF (TOTAL_ERROR.GE.0.0) THEN


IF (EVOL.EQ.GENERATION) THEN


OPEN(UNIT=8,FILE=INTER2_TXT,STATUS='UNKNOWN')

    WRITE (8,*) 'GENERATION NUMBER ..............', EVOL
    WRITE (8,*) 'POPULATION NUMBER ..............', POP_NUM
    WRITE (8,*)
    WRITE (8,*) '-------------------------------------------------------------------------
----------------'



   WRITE(8,200) 'POSITION','LIQ FLOW 1','LIQ FLOW','PRESSURE1','PRESSURE2','LOSS CHARGE'
,'LOSS CHARGE',&
   &'VF 1','VF2'
   WRITE(8,200)  'REAL','SIM','SIM','SIM','SIM','SIM','SIM','SIM','SIM'
   WRITE(8,200)  'M','KG/HR','KG/HR','PA','PA','PA','PA','ADIM','ADIM'



   rewind(unit=17)
   DO K=2,N-1

   WRITE (8,100) XPOSITION(K), LIQFLOW(K,1),&
   &LIQFLOW(K,2),&
   &PRSRE(K,1),&
   &PRSRE(K,2), &
   &DPFT(K,1),&
   &DPFT(K,2),&
   &VOID_FRCTN(K,1),&
   &VOID_FRCTN(K,2)
   END DO


   WRITE (8,*)
   WRITE (8,*) 'GAS SIMULATION'
   WRITE (8,*) 'POSITION  GAS FLOW1  GAS FLOW1    GAS FLOW 2  GAS FLOW 2'
   WRITE (8,*) '  REAL       EXP         SIM          EXP         SIM'
   WRITE (8,*) '   M         PA          PA           PA          PA'

   DO K=2,N-1
    WRITE(8,500) XPOSITION(K),GASFLOW(K,1),GASFLOW(K,2)
   END DO


!-------------------------------------------------------------------------------------
```

```
-----------------------------------


   WRITE (8,600) 'TOTAL ERROR..................................... ',TOTAL_ERROR

!-------------------------------------------------------------------------------------------
------------------------------


        WRITE(8,'(A)') '----------------- EXPERIMENTAL VALUES LIQUID PHASE--------------
-----------'
        DO I=1,AUX
        WRITE(8,700) XPOS_EXP(I),LIQFLOW_EXP(I),LIQFLOW_EXP2(I),PRSRE_EXP1(I),PRSRE_EXP2
(I),DPFT_EXP1(I),DPFT_EXP2(I),&
        & VOID_FRCTN_EXP1(I),VOID_FRCTN_EXP2(I)
        END DO

        WRITE(8,'(A)') '----------------- EXPERIMENTAL VALUES GAS PHASE-----------------
--------'
        DO I=1,AUX2
        WRITE(8,800) XPOSITION_GAS(I),AIRFLOW_EXP1(I),AIRFLOW_EXP2(I)
        END DO


!-------------------------------------------------------------------------------------------
------------------------------

END IF



OPEN(UNIT=78,FILE='D:\GRAPH_DATA.RES',STATUS='UNKNOWN')
DO K=1,AUX2
WRITE (78,1) XPOSITION(K+1),&
&LIQFLOW(K+1,1),&
&LIQFLOW(K+1,2),&
&PRSRE(K+1,1)/1000,&
&PRSRE(K+1,2)/1000,&
&DPFT(K+1,1)/1000,&
&DPFT(K+1,2)/1000,&
&VOID_FRCTN(K+1,1),&
&VOID_FRCTN(K+1,2),&
&GASFLOW(K+1,1),&
&GASFLOW(K+1,2),&
&DP_RAD (K+1)/1000&

! Printing the experiments values

&,XPOS_EXP(K),&
&LIQFLOW_EXP(K),&
&LIQFLOW_EXP2(K),&
&PRSRE_EXP1(K)/1000,&
&PRSRE_EXP2(K)/1000,&
&DPFT_EXP1(K)/1000,&
&DPFT_EXP2(K)/1000,&
&VOID_FRCTN_EXP1(K),&
&VOID_FRCTN_EXP2(K),&
&DPRAD_EXP(K)/1000,&
&XPOSITION_GAS(K),&
&AIRFLOW_EXP1(K),&
&AIRFLOW_EXP2(K)
END DO
```

```
DO K=AUX2,AUX
WRITE (78,1) XPOSITION(K+1), LIQFLOW(K+1,1),LIQFLOW(K+1,2),PRSRE(K+1,1)/1000,PRSRE(K+1,2
)/1000,&
   &DPFT(K+1,1)/1000,DPFT(K+1,2)/1000,VOID_FRCTN(K+1,1),VOID_FRCTN(K+1,2),GASFLOW(K+1,1)
,GASFLOW(K+1,2),DP_RAD(K+1)/1000&

! Printing the experiments values

   &,XPOS_EXP(K),LIQFLOW_EXP(K),LIQFLOW_EXP2(K),PRSRE_EXP1(K)/1000,PRSRE_EXP2(K)/1000,DP
FT_EXP1(K)/1000,DPFT_EXP2(K)/1000&
   &,VOID_FRCTN_EXP1(K),VOID_FRCTN_EXP2(K),DPRAD_EXP(K)/1000
END DO
DO K=AUX2,N-2
WRITE (78,1) XPOSITION(K+1), LIQFLOW(K+1,1),LIQFLOW(K+1,2),PRSRE(K+1,1)/1000,PRSRE(K+1,2
)/1000,&
   &DPFT(K+1,1)/1000,DPFT(K+1,2)/1000,VOID_FRCTN(K+1,1),VOID_FRCTN(K+1,2),GASFLOW(K+1,1)
,GASFLOW(K+1,2),DP_RAD(K+1)/1000

END DO


CLOSE(UNIT=78)

1 FORMAT (22(1X, G14.7),  10(:,1X, G14.7), 3(:,1X, G14.7))


100 FORMAT (F6.4, 6(4X, F9.2),2(4X,F7.5))
200 FORMAT (T2, A, T13, A, T26, A, T39, A, T52, A, T63, A, T77, A, &
      & T90, A, T103, A)
500 FORMAT (F6.4, 6X, F6.4, 6X, F6.4, 6X, F6.4, 6X, F6.4)
600 FORMAT (/A,F19.5)
700 FORMAT (9(1X,G11.4))
800 FORMAT (3(1X,G11.4))
RETURN
ENDIF

END SUBROUTINE RESULTS
```

```
SUBROUTINE SELECTIONAT(X,Y,ACTUAL,CHROMOSOME)
INTEGER AUXI,I,J,H,X,Y
INTEGER,DIMENSION(Y)::ACTUAL
INTEGER,DIMENSION(Y,X)::CHROMOSOME


AUXI=0
DO J= 1,Y

IF(ACTUAL(J)/=0) THEN

 DO I=1,ACTUAL(J)
   AUXI=AUXI+1

     DO H=1,X
     CHROMOSOME(AUXI,H)=CHROMOSOME(J,H)
     END DO

 END DO
END IF


 END DO

 END SUBROUTINE SELECTIONAT
```

```
SUBROUTINE SELECTION_NAT
USE VAR1
USE GA_VAR
IMPLICIT NONE
INTEGER,DIMENSION(CHROM_SIZE)::BESTPOP
INTEGER I,J,H

H=MINLOC(TOTAL_ERROR_VEC,DIM=1)
BESTPOP(1:CHROM_SIZE)=POPULATION(H,1:CHROM_SIZE)

 DO I=1,POP_SIZE
    IF (LIN_F(I).EQ.0.0) THEN
        POPULATION(I,1:CHROM_SIZE)       =    BESTPOP(1:CHROM_SIZE)
    END IF
END DO




END SUBROUTINE SELECTION_NAT
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS PROGRAMME STATISTICS   =   Ce sous programme determine les chromosomes
!                                   qui vont passer à la suivante génération
!                                   La methode utilisé est "WINDOWING"

!-------------------------------------------------------------------------------

!   MAX                 =   La valeur (erreur) maximale de la population
!                           Variable de type DOUBLE PRESICION
!
!   MIN                 =   La valeur (erreur) minimale de la population
!                           Variable de type DOUBLE PRESICION
!
!   LIN_F               =   Normalisation des valeurs de chaque chromosome
!                           Variable de type DOUBLE PRESICION
!
!   SUM_TOTAL_ERROR     =   La somme totale des érreurs de la population
!                           Variable de type DOUBLE PRESICION
!
!   PERCENT             =   Le percentage de chaque individu
!                           Variable de type DOUBLE PRESICION
!
!   AVERAGE             =   La moyenne de la population
!                           Variable de type DOUBLE PRESICION
!
!   BESTFIT_VAL         =   la meilleure valeur de la population
!                           Variable de type DOUBLE PRESICION
!
!   BESTFIT_LOC         =   les coordonnés du meilleur chromosome
!                           Variable de type DOUBLE PRESICION



SUBROUTINE STATISTICS
USE GA_VAR
USE VAR1

IMPLICIT NONE
INTEGER K
DOUBLE PRECISION,DIMENSION(GENERATION)::BESTFIT
DOUBLE PRECISION,DIMENSION(POP_SIZE)::LIN_DISSIPATION_ENERGY
DOUBLE PRECISION::F1,F2
REAL FILTER




   MAX                 =   MAXVAL(TOTAL_ERROR_VEC)
   MIN                 =   MINVAL(TOTAL_ERROR_VEC)

   WHERE (TOTAL_ERROR_VEC.GE.3*MIN)
   TOTAL_ERROR_VEC=3*MIN
   ENDWHERE

  DO K=1,POP_SIZE

   LIN_F(K)            =   (MAX-TOTAL_ERROR_VEC(K)+0.001) / (MAX-MIN+0.001)
   SUM_TOTAL_ERROR     =   (SUM(TOTAL_ERROR_VEC))
```

```
    PERCENT(K)          =   LIN_F(K)/SUM(LIN_F)
    AVERAGE             =   SUM(LIN_F)/POP_SIZE
    EXPECTED_COUNT(K)   =   TOTAL_ERROR_VEC(K)/AVERAGE
    ACTUAL_COUNT(K)     =   INT(PERCENT(K))


  FILTER=(0.1-0.001)/(GENERATION-1)* EVOL + (0.001 - ((0.1-0.001)/(GENERATION-1)) )

  END DO

       WHERE(LIN_F<AVERAGE)
       LIN_F               =   0.0
       ELSEWHERE
       LIN_F               =   1.0
       ENDWHERE


    BESTFIT(EVOL)=MIN


    IF (EVOL.EQ.GENERATION) THEN
        BESTFIT_VAL     =       MINVAL(BESTFIT)
        BESTFIT_LOC     =       MINLOC(BESTFIT,DIM=1)
    END IF

END SUBROUTINE STATISTICS
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME SUBCHANNEL.   Ce programme a comme but de faire l'analyse de
    !                       sous canal et l'optimiser en utilisant des
    !                       algorithmes génétiques.

!
!   Les sous-programmes utilisés sont:
!   INIT_VAR         =   Ce sous programme est utilise pour initialiser les variables.
!   INPUT_DATA   =   Ce sont les conditions initiales.
!   GRID             =   Sous programme pour réaliser la discretisation.
!   FLOW_EQ      =   Ce sous programme résoudre le système d'équations diff
!                    partiales.
!   OUTPUT_DATA =   Ce sous programme fais un rapport contenant les résultats
!                     de la simulation
!   GENETIC_ALGORITHM=  Ce sous programme optimise le code d'analyse de sous canal

PROGRAM SUBCHANNEL
USE VAR1
CALL INIT_VAR
CALL INPUT_DATA
CALL GRID
CALL FLOW_EQ
CALL OUTPUT_DATA
CALL GENETIC_ALGORITHM

END PROGRAM SUBCHANNEL
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   SOUS PROGRAMME TRANSV_MOM_EQ        =   Ce sous programme est utilisé pour résoudre
!                                           l'équation de la conservation de la quantité
!                                           de mouvement transversale.



SUBROUTINE TRANSV_MOM_EQ
USE VAR1
USE PROPERTIES

DOUBLE PRECISION,DIMENSION(N)::BLIQ_LOC,BGAS_LOC,AA_LOC,BB_LOC,CC_LOC,DD_LOC,U_LOC
DOUBLE PRECISION SUM,RESID,CRITERIA,DP_FRCTN_LOC
INTEGER NUM_ITER_TRSNV
AA_LOC=0.0
BB_LOC=0.0
CC_LOC=0.0
DD_LOC=0.0
U_LOC=0.0
sum=0.0
XKTR=0.0
NN1=I2+1


DO I=NN1,N2
IF(TTLFLUX_CF(I).GE.0.D0) THEN

  FLUX_Y=TTLFLUX_CF(I)/TTLFLUX(I,1)
  ELSE IF(TTLFLUX_CF(I).LT.0.0D0) THEN
  FLUX_Y=-TTLFLUX_CF(I)/TTLFLUX(I,2)
END IF

!-------------------------------------------------------------------------------------
-
!   XKTR        =       le coefficient de friction lateral,
!                       Si GA = 2 la valeur de XKTR sera égale à la valeur d'entrée
!                       du code.
!
!
!                       Si GA = 1 La valeur de XKTR sera égale à la valeur que l'AG
!                       envoayera
!
!   ICOEFF      =       Détermine le nombre des iterations à realiser, si ICOEFF est
!                       égal a 1 le code arretera si le critere de convergence est
!                       plus petite que 1E-6 ou si le nombre des iterations attenit
!                       les 150 iterations.
!                       Dans le cas où le critère de convergence ne soit pas achevé
!                       l'algorithme génétique n'utilisara pas ce chromosome


!-------------------------------------------------------------------------------------
-
               IF (GA.EQ.2) THEN
                   XKTR=KTR
                ELSE IF(GA.EQ.1) THEN
                   XKTR=IN_KTRNSL
               END IF
!-------------------------------------------------------------------------------------
-
```

```
END DO

TTLFLUX_CF(NN1)=0.D0
TTLFLUX_CF(N3)=0.D0
AA_LOC(NN1)=0.D0
CC_LOC(NN1)=0.D0
U_LOC(I2)=0.D0
U_LOC(N2)=0.D0

SELECT CASE (ICOEFF)
CASE(0)
SUM=0.D0
CRITERIA=0.0
NUM_ITER_TRSNV=0
RESID=1.0
DO WHILE(DABS(RESID).GE.1.0E-6)
NUM_ITER_TRSNV=NUM_ITER_TRSNV+1

    IF (NUM_ITER_TRSNV.GT.150) THEN
        FLAG=.TRUE.
        EXIT
    END IF

DO I=NN1,N2

   IF(DABS(TTLFLUX_CF(I)).LE.EP) THEN
    BLIQ_LOC(I)=0.D0
    BGAS_LOC(I)=0.D0
    ELSE IF (DABS(GASFLUX_CF(I)).LE.EP) THEN
    BGAS_LOC(I)=0.D0
    BLIQ_LOC(I)=LIQFLUX_CF(I)/(1.0-VOID_FRCTN_CF(I))/TTLFLUX_CF(I)/LIQ_DNSTY
    ELSE
    BGAS_LOC(I)=GASFLUX_CF(I)/GAS_DNSTY_CF(I)/VOID_FRCTN_CF(I)/TTLFLUX_CF(I)
    BLIQ_LOC(I)=LIQFLUX_CF(I)/(1.0-VOID_FRCTN_CF(I))/TTLFLUX_CF(I)/LIQ_DNSTY

   END IF
    V(I)=(TTLFLUX_CF(I)*BLIQ_LOC(I)+GASFLUX_CF(I)*BGAS_LOC(I))*OMEGA(I)

END DO

DO I=NN1,N1
 U_LOC(I)=0.5*GAP_WDTH*DELTAY(I)*(GAP_GL(I)*(BLIQ_LOC(I)+BLIQ_LOC(I+1))+GAP_GG(I)*(BGAS_
LOC(I)+BGAS_LOC(I+1)))
 AA_LOC(I+1)=DMAX1(U_LOC(I),0.D0)
 CC_LOC(I)=AA_LOC(I+1)-U_LOC(I)
END DO

U_LOC(N2)=U_LOC(N1)

DO I=NN1,N2
 MIX_DNSTY_CF=LIQ_DNSTY*(1.0-VOID_FRCTN_CF(I))+GAS_DNSTY_CF(I)*VOID_FRCTN_CF(I)
 DP_FRCTN_LOC=GAP_WDTH*DELTAX(I)*XKTR/MIX_DNSTY_CF
 V12=DMAX1(V(I),0.D0)
 DD_LOC(I)=(0.5*DP_FRCTN_LOC*TTLFLUX_CF(I)+V12-V(I))*TTLFLUX_CF(I)
 BB_LOC(I)=AA_LOC(I)+CC_LOC(I)+U_LOC(I)-U_LOC(I-1)+V12+DP_FRCTN_LOC*DABS(TTLFLUX_CF(I))+
EP
 D_CF(I)=OMEGA(I)/BB_LOC(I)
 VG_CF(I)=(AA_LOC(I)*TTLFLUX_CF(I-1)+CC_LOC(I)*TTLFLUX_CF(I+1)+DD_LOC(I))/BB_LOC(I)

            IF(BB_LOC(I)>1.0D10) THEN
                return
                pause
            END IF
                IF(VG_CF(I)>1.0D25) THEN
                return
```

```
            pause
          END IF

          IF(BB_LOC(I).EQ.0.0) THEN
            BB_LOC(I)=1E-4
            pause
          END IF


 RESID=BB_LOC(I)*(VG_CF(I)-TTLFLUX_CF(I))+OMEGA(I)*(PRSRE(I,1)-PRSRE(I,2))
 TTLFLUX_CF(I)=(1.0-RELAX)*TTLFLUX_CF(I)+RELAX*(VG_CF(I)+D_CF(I)*(PRSRE(I,1)-PRSRE(I,2))
)

 END DO


DO I=NN1,N2

 LIQFLUX_CF(I)=TTLFLUX_CF(I)-GASFLUX_CF(I)
END DO


END DO



CASE(1)
DO K=1,1
  SUM=0.D0

 DO I=NN1,N2

   IF(DABS(TTLFLUX_CF(I)).LE.EP) THEN
    BLIQ_LOC(I)=0.D0
    BGAS_LOC(I)=0.D0
    ELSE IF (DABS(GASFLUX_CF(I)).LT.EP) THEN
    BGAS_LOC(I)=0.D0
    BLIQ_LOC(I)=LIQFLUX_CF(I)/(1.0-VOID_FRCTN_CF(I))/TTLFLUX_CF(I)/LIQ_DNSTY
    ELSE
    BGAS_LOC(I)=GASFLUX_CF(I)/GAS_DNSTY_CF(I)/VOID_FRCTN_CF(I)/TTLFLUX_CF(I)
    BLIQ_LOC(I)=LIQFLUX_CF(I)/(1.0-VOID_FRCTN_CF(I))/TTLFLUX_CF(I)/LIQ_DNSTY
   END IF
    V(I)=(TTLFLUX_CF(I)*BLIQ_LOC(I)+GASFLUX_CF(I)*BGAS_LOC(I))*OMEGA(I)

END DO

DO I=NN1,N1
 U_LOC(I)=0.5*GAP_WDTH*DELTAY(I)*(GAP_GL(I)*(BLIQ_LOC(I)+BLIQ_LOC(I+1))+GAP_GG(I)*(BGAS_
LOC(I)+BGAS_LOC(I+1)))
 AA_LOC(I+1)=DMAX1(U_LOC(I),0.D0)
 CC_LOC(I)=AA_LOC(I+1)-U_LOC(I)
END DO

U_LOC(N2)=U_LOC(N1)

DO I=NN1,N2
 MIX_DNSTY_CF=LIQ_DNSTY*(1.0-VOID_FRCTN_CF(I))+GAS_DNSTY_CF(I)*VOID_FRCTN_CF(I)
 DP_FRCTN_LOC=GAP_WDTH*DELTAX(I)*XKTR/MIX_DNSTY_CF
 V12=DMAX1(V(I),0.D0)
 DD_LOC(I)=(0.5*DP_FRCTN_LOC*TTLFLUX_CF(I)+V12-V(I))*TTLFLUX_CF(I)
 BB_LOC(I)=AA_LOC(I)+CC_LOC(I)+U_LOC(I)-U_LOC(I-1)+V12+DP_FRCTN_LOC*DABS(TTLFLUX_CF(I))+
EP
 D_CF(I)=OMEGA(I)/BB_LOC(I)
 VG_CF(I)=(AA_LOC(I)*TTLFLUX_CF(I-1)+CC_LOC(I)*TTLFLUX_CF(I+1)+DD_LOC(I))/BB_LOC(I)
 END DO

DO I=NN1,N2
```

```
 LIQFLUX_CF(I)=TTLFLUX_CF(I)-GASFLUX_CF(I)
END DO

END DO
  END SELECT

  END SUBROUTINE TRANSV_MOM_EQ
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   MODULE TRIDIAGONAL      =   Ce module contient le sous programme
!                               TRIDAG pour la resolution d'un
!                               système d'équations donnant une
!                               matrix tridiagonelle




MODULE TRIDIAGONAL
          CONTAINS

      SUBROUTINE TRIDAG(KIN,N,A,B,C,D,F,J)
! ------------------------------------------------------------
!     *              ---  T R I D A G  ---                  *
!     *           TRIDIAGONAL MATRIX SOLUTION               *
! ------------------------------------------------------------
!
DOUBLE PRECISION,DIMENSION(N+1):: BETA,GAMMA
DOUBLE PRECISION,DIMENSION(N+1,2)::F,A,B,C,D
INTEGER::J
      BETA(KIN)=B(KIN,J)
      GAMMA(KIN)=D(KIN,J)/BETA(KIN)
      IFP1=KIN+1

      DO I=IFP1,N
        BETA(I)=B(I,J)-A(I,J)*C(I-1,J)/BETA(I-1)
        GAMMA(I)=(D(I,J)-A(I,J)*GAMMA(I-1))/BETA(I)
      END DO

       F(N,J)=GAMMA(N)
       L=N-KIN

      DO K=1,L
        I=N-K
        F(I,J)=GAMMA(I)-C(I,J)*F(I+1,J)/BETA(I)
      END DO

      END SUBROUTINE TRIDAG

   END MODULE TRIDIAGONAL
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!-------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   PROGRAMME SUBCHANNEL.   Ce programme a comme but de faire l'analyse de
!                           sous canal et l'optimiser en utilisant des
!                           algorithmes génétiques.
!
!
!   Sous programme :    GRID

!   Ce sous programme prepare la (grille) discretisation du canaux, c-a-d le nombre de n
oeuds, et leur dimmensions
!
!
!

SUBROUTINE GRID
USE VAR1
IMPLICIT NONE
INTEGER I,J

N1=NPTS+1
N2=N1+1
N3=N2+1
OMEGA=0.0

!-----------------------------------------------------------------------------------------
-----------------
!   TOTAL LENGTH     =   START OF INTERCONECTION + DISTANCE OF THE BEGINNING OF TH INTER
!                        TO THE END.
!                        Variable de type DOUBLE PRESICION
!
!-----------------------------------------------------------------------------------------
-----------------
TTL_LNGTH=STRT_INP+LNGTH



XPOSITION(1)=0.0
XPOSITION(2)=0.0


!-----------------------------------------------------------------------------------------
-----------------
!   DDX          =       La dimmension du control du volume (dx)
!                        Variable de type DOUBLE PRESICION
!
!
!
!
!-----------------------------------------------------------------------------------------
-----------------


          DDX=TTL_LNGTH/NPTS

             DO I=3,N1
              DELTAX(I)=DDX
              XPOSITION(I)=XPOSITION(I-1)+DDX
             END DO
```

```
    XPOSITION(N2)=TTL_LNGTH
    XPOSITION(N3)=XPOSITION(N2)
    DELTAX(2)=0.D0
    DELTAX(N2)=TTL_LNGTH-XPOSITION(N1)

!-------------------------------------------------------------------------------------------
-----------------
!   X_PRSRE(I)      =   Grille décalée
!                       Variable de type DOUBLE PRESICION
!
!   DELTAX_CV       =   Distance entre les noeuds de la grille décalée
!                       Variable de type DOUBLE PRESICION
!
!
!-------------------------------------------------------------------------------------------
-----------------


       DO I=2,N2
        X_PRSRE(I)=(XPOSITION(I)+XPOSITION(I-1))/2.0
        DELTAY(I)=0.D0
       END DO



        X_PRSRE(N3)=XPOSITION(N2)

        DO I=2,N2
        DELTAX_CV(I)=X_PRSRE(I+1)-X_PRSRE(I)
        END DO



!-------------------------------------------------------------------------------------------
-----------------
!
!   I2          =   Début de l'interconnexion
!                   Variable de type INTEGER
!
!
!-------------------------------------------------------------------------------------------
-----------------


DO I=2,NPTS

   IF (XPOSITION(I).GE.STRT_INP) THEN
   I2=I-1
   EXIT

   END IF

END DO



DO I=2, I2
   IF(XPOSITION(I).GE.STRT_INP-0.0251) THEN
   I1=I
   EXIT
   END IF
END DO
```

```
!-------------------------------------------------------------------------------------------
-----------------
!   AREA        =   Represent la section du passage axial
!                   Variable de type DOUBLE PRESICION
!
!   WET_PRMTER  =   Le perimetre mouillé
!                   Variable de type DOUBLE PRESICION
!
!   HYDRLC_DIAM =   Le diametre hydrolique
!                   Variable de type DOUBLE PRESICION
!
!
!-------------------------------------------------------------------------------------------
-----------------


AREA(1,1)=CHNL_WDTH1*CHNL_LNGTH1
AREA(1,2)=CHNL_WDTH2*CHNL_LNGTH2

WET_PRMTER(1,1)=2.0*(CHNL_WDTH1+CHNL_LNGTH1)
WET_PRMTER(1,2)=2.0*(CHNL_WDTH2+CHNL_LNGTH2)
HYDRLC_DIAM(1,1)=4.0*AREA(1,1) / WET_PRMTER(1,1)
HYDRLC_DIAM(1,2)=4.0*AREA(1,2) / WET_PRMTER(1,2)

DO J=1,2
 DO I=2,N3
  AREA(I,J)=AREA(1,J)
  WET_PRMTER(I,J)=WET_PRMTER(1,J)
  HYDRLC_DIAM(I,J)=4.0*AREA(I,J) / WET_PRMTER(I,J)
 END DO
END DO

!-------------------------------------------------------------------------------------------
-----------------
!
!   DELTAY  =       la longueur de mélange
!                   Variable de type DOUBLE PRESICION
!
!
!-------------------------------------------------------------------------------------------
-----------------


DO I=I1, N2
  DELTAY(I)=GAP_LNGTH+ (0.5*(CHNL_LNGTH1 + CHNL_LNGTH2))
  OMEGA(I)=DELTAX(I)*GAP_WDTH
END DO

AREA(1:NPTS,1)=CHNL_WDTH1*CHNL_LNGTH1
AREA(1:NPTS,2)=CHNL_WDTH2*CHNL_LNGTH2
WET_PRMTER(1:NPTS,1)=2.0*(CHNL_WDTH1+CHNL_LNGTH1)
WET_PRMTER(1:NPTS,2)=2.0*(CHNL_WDTH2+CHNL_LNGTH2)
HYDRLC_DIAM(1:NPTS,1)=4.0*AREA(1,1) / WET_PRMTER(1,1)
HYDRLC_DIAM(1:NPTS,2)=4.0*AREA(1,2) / WET_PRMTER(1,2)


END SUBROUTINE GRID
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL.-------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE. ------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:            13-01-04
!

!   MODULE VAR1      =   Module utilisé pour garder toutes les varibles et l'alocation du
 memoire

MODULE VAR1


! TWO DIMENSION ARRAYS

IMPLICIT NONE
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: AREA
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: GASFLUX,LIQFLUX,TTLFLUX
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: PRSRE,VOID_FRCTN,WET_PRMTER,HYDRLC_DIAM,Q
UALITY
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: LIQFLOW,GASFLOW,GAS_DNSTY
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: PC
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: GASFLOW_CF1,GASFLOW_CF2,GASFLOW_CF3
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: FR,F,MIX_DNSTY
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:):: AA,BB,CC,DD,AP,BP,CP,DP,BP1,DPFT,PC_AUX


! ONE DIMENSION ARRAY

DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: GAP_GL, GAP_GG, VG_CF, D_CF,TTLFLUX_CF,LIQF
LUX_CF,GASFLUX_CF
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XPOSITION, DELTAX,DELTAY,DELTAX_CV,X_PRSRE,
OMEGA
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: GAS_DNSTY_CF,VOID_FRCTN_CF,VG,DWL,DWG,V,D,D
P_RAD,U
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: DIV_CROSS,TURB_VOID_DIFF,VOID_DRIFT,BLIQ_TM
E,BGAS_TME
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XA,YA,LIQFLOW1_INTERPOLATED,LIQFLOW2_INTERP
OLATED,DPRADIAL_INTERPOLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: PRSRE1_INTERPOLATED,PRSRE2_INTERPOLATED,VOI
D_FRCTN1_INTERPOLATED,VOID_FRCTN2_INTERPOLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: DPFT1_INTERPOLATED,DPFT2_INTERPOLATED,GASFL
OW1_INTERPOLATED,GASFLOW2_INTERPOLATED
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XPOS_EXP,LIQFLOW_EXP,LIQFLOW_EXP2,PRSRE_EXP
1,PRSRE_EXP2,DPFT_EXP1,DPFT_EXP2,DPRAD_EXP
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XPOSITION_GAS,AIRFLOW_EXP1,AIRFLOW_EXP2,DPR
ADIAL,VOID_FRCTN_EXP1,VOID_FRCTN_EXP2
DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:):: XA_2,YA_2,CONV_FLOW_EQ,Total_Energy,Energy_
S1,Energy_S2,RE_FLOWEQ

DOUBLE PRECISION, DIMENSION(8):: COEF_VF,COEF_PHI
DOUBLE PRECISION, DIMENSION(2):: K_IRRV,B,C,DP_INI,SUM_PC,SUM_PC_AUX


DOUBLE PRECISION:: GAP_WDTH, GAP_LNGTH, CHNL_WDTH1, CHNL_WDTH2, CHNL_LNGTH1, CHNL_LNGTH2
,CHNL_WDTH
DOUBLE PRECISION:: LNGTH,TTL_LNGTH, STRT_INP,DDX
DOUBLE PRECISION:: GASFLUX_INP1, LIQFLUX_INP1, GASFLUX_INP2, LIQFLUX_INP2
DOUBLE PRECISION:: PRSSRE_OUT1,PRSSRE_OUT2, VOIDFRC_INP1, VOIDFRC_INP2, TEMP,VOID_FRCTN_
12,VOID_FRCTN2,VOID_FRCTN1
DOUBLE PRECISION:: NIT_CF, NIT_MAS, NIT_PRS
DOUBLE PRECISION:: LIQ_DNSTY
DOUBLE PRECISION:: F_MOM,X_K_INPUT,X_LAMBDA
DOUBLE PRECISION:: ROL,VISCOCITY ,QUALITY_AVE
DOUBLE PRECISION:: CONV_CRTRIA,RELAX, EPSILON,DP_GRVTY,DP_FRCTN,GRVTY
DOUBLE PRECISION:: BGAS,BLIQ,BLIQ_INI,BGAS_INI,U1,U2,BALANCE,MIX_DNSTY_CF,DR
DOUBLE PRECISION:: PRSRE_AVE,PRSRE_CF,FLUX_Y,DMASS
```

```
DOUBLE PRECISION:: FLUX_BG,AREA_C,SIGMA,VOID_FRCTN_AVE,SPFCL_VLCTY1,SPFCL_VLCTY2
DOUBLE PRECISION:: MIX_SPFCL_VLCTY, HYDRL_DIAM_AVE,TURB_VOID_DIF,COEF,HYDRL_DIAM_CF,DEQ
DOUBLE PRECISION:: CX,DELTAY_AVE,FO,PHI2,V12,DELG,SUM_LIQ,SUM_GAS,SW,DVFEQ
DOUBLE PRECISION:: GAPARA,GAPER,Y,CONV_FACT,ERROR_PC1,ERROR_PC2,ERROR_TTL_PC
DOUBLE PRECISION:: CO,DALFA_DEC,CO_DEC,KTR_DEC,DEQ_DEC,LAMBDA_DEC, COEFF1_DEC,COEFF2_DEC
,COEFF3_DEC,COEFF4_DEC,EP,DALFA,CORRECTION_S1,CORRECTION_S2
DOUBLE PRECISION:: CO_DEC_FLOWEQ,KTR_DEC_FLOWEQ,DEQ_DEC_FLOWEQ,DALFA_DEC_FLOWEQ,DVFEQ_DE
C_FLOWEQ,XKTR,KTR
DOUBLE PRECISION:: LAMBDA_DEC_FLOWEQ,COEFF1_DEC_FLOWEQ,COEFF2_DEC_FLOWEQ,COEFF3_DEC_FLOW
EQ,COEFF4_DEC_FLOWEQ


INTEGER:: N,OPTION,TYPE_GRID, BLCKGE_TYPE
INTEGER:: NPTS, N1, N2, N3, I1, I2, NN1
INTEGER:: N_ITER,CONT_ITER,N_ITER_MOM
INTEGER:: IN_FRIC_CORR, IN_KTRNSL, IIR_PRSRE, IN_EQ_VOIDF, IN_EQ_DNSTY
INTEGER:: ICOEFF,NUM_ITER,NAUX,NIT_MAX,CONT_ITER_MOM,GA,AUX,AUX2, TYPE$,I4
LOGICAL:: FLAG
CHARACTER*45 DATAFILE,GRAPHFILE

CONTAINS
SUBROUTINE ALLOCATION(N,AUX,AUX2)
INTEGER, INTENT(IN) :: N,AUX,AUX2

!MATRIX

ALLOCATE (AREA(N,2))
ALLOCATE (GASFLUX(N,2),LIQFLUX(N,2),TTLFLUX(N,2),TTLFLUX_CF(N))
ALLOCATE (PRSRE(N,2),VOID_FRCTN(N,2),WET_PRMTER(N,2),HYDRLC_DIAM(N,2),QUALITY(N,2))
ALLOCATE (LIQFLOW(N,2),GASFLOW(N,2),GAS_DNSTY(N,2))
ALLOCATE (PC(N,2))
ALLOCATE (GASFLOW_CF1(N,2),GASFLOW_CF2(N,2),GASFLOW_CF3(N,2) )
ALLOCATE (FR(N,2),F(N,2),MIX_DNSTY(N,2))
ALLOCATE (AA(N,2),BB(N,2),CC(N,2),DD(N,2),AP(N,2),BP(N,2),CP(N,2),DP(N,2))
ALLOCATE (BP1(N,2),DPFT(N,2),PC_AUX(N,2))
! VECTORS
ALLOCATE( GAP_GL(N), GAP_GG(N), VG_CF(N), D_CF(N),LIQFLUX_CF(N),GASFLUX_CF(N))
ALLOCATE( XPOSITION(N),DELTAX(N),DELTAY(N),DELTAX_CV(N),X_PRSRE(N),OMEGA(N))
ALLOCATE( GAS_DNSTY_CF(N),VOID_FRCTN_CF(N),VG(N),DWL(N),DWG(N),V(N),D(N),DP_RAD(N),U(N))
ALLOCATE( DIV_CROSS(N),TURB_VOID_DIFF(N),VOID_DRIFT(N),BLIQ_TME(N),BGAS_TME(N))
ALLOCATE( CONV_FLOW_EQ(N),Total_Energy(N),Energy_S1(N),Energy_S2(N),RE_FLOWEQ(N))


ALLOCATE(XPOS_EXP(AUX),LIQFLOW_EXP(AUX),LIQFLOW_EXP2(AUX),PRSRE_EXP1(AUX),PRSRE_EXP2(AUX
),DPFT_EXP1(AUX),DPFT_EXP2(AUX))
ALLOCATE(XPOSITION_GAS(AUX2),AIRFLOW_EXP1(AUX2),AIRFLOW_EXP2(AUX2),DPRADIAL(AUX),DPRAD_E
XP(AUX))
ALLOCATE (VOID_FRCTN_EXP1(AUX),VOID_FRCTN_EXP2(AUX))
    IF (GA.EQ.1) THEN
    ALLOCATE (XA(N-2),YA(N-2),LIQFLOW1_INTERPOLATED(AUX),LIQFLOW2_INTERPOLATED(AUX))
    ALLOCATE (PRSRE1_INTERPOLATED(AUX),PRSRE2_INTERPOLATED(AUX),DPFT1_INTERPOLATED(AUX),
DPFT2_INTERPOLATED(AUX))
    ALLOCATE (VOID_FRCTN1_INTERPOLATED(AUX),VOID_FRCTN2_INTERPOLATED(AUX))
    ALLOCATE (DPRADIAL_INTERPOLATED(AUX))
    ALLOCATE (GASFLOW1_INTERPOLATED(AUX2),GASFLOW2_INTERPOLATED(AUX2))
    ALLOCATE (XA_2(AUX),YA_2(AUX))
    END IF




END SUBROUTINE ALLOCATION

END MODULE VAR1
```

```
!-------------------------- ÉCOLE POLYTECHNIQUE DE MONTRÉAL. --------------------------

!--------------------------  INSTITUTE DE GÉNIE PHYSIQUE.  ---------------------------

!   ELABORÉ PAR:     ARMANDO NAVA
!   DATE:        13-01-04
!

!   SOUS PROGRAMME EQVOID       =   Sous programme utilisé pour calculer la
!                         différence de taux de vide à l'équilibre
!                         Basé sur le modèle de Lahey
!
!
!
!
!
SUBROUTINE EQVOID(M,I,DEQVF)
USE VAR1
REAL,DIMENSION(5):: XW=(/0.1746,1.082,-4.56,5.072,-1.757/)
INTEGER I,M,K
DOUBLE PRECISION DEQVF,VFBAR,SUM,TTLFLUXBAR


VFBAR=(VOID_FRCTN(I,1)+VOID_FRCTN(I,2))/2.
      IF(M.EQ.1) THEN
      IF(VFBAR.LE.0.15) THEN
      DEQVF=VOID_FRCTN(I,1)-VOID_FRCTN(I,2)
      RETURN
      ELSE IF(VFBAR.GT.0.75) THEN
        DEQVF=0.
        RETURN
         ELSE
          SUM=0.
          DO K=2,5
          SUM=SUM+XW(K)*VFBAR**(K-1)
         END DO
          DEQVF=SUM+XW(1)
      END IF
      RETURN
      ELSE IF(M.EQ.2) THEN
        TTLFLUXBAR=0.5*(TTLFLUX(I,1)+TTLFLUX(I,2))
        COEFF=VFBAR/TTLFLUXBAR
        DEQVF=COEFF*(TTLFLUX(I,1)-TTLFLUX(I,2))
        RETURN
      ELSE
      DEQVF=0.
      WRITE(*,1000)
 1000 FORMAT(///,T10,'WARNING  WARNING  WARNING  WARNING  WARNING     ',&
     &/,T10,'-------------------------------------------',//,T12,'THE SE&
     &LECTED CORRELATION FOR THE EQUILIBRIUM VOID IS NOT AVAILABLE !!!!'&
     &,/,T12,'THE "DEQVF" IS TAKEN EQUAL TO ZERO !!!',///)
      END IF
      RETURN
      END
```