



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département d'Hydraulique

Mémoire de magistère

Préparé par

MEZALI Farouk

Mémoire préparé partiellement à l'INRIA au sein de l'équipe COSIVIE

Sujet

Ecoulement diphasique avec prise en compte des aspects biochimiques pour la simulation numérique sous HPCN du traitement de l'eutrophisation par aération mécanique

Jury

Président :	A. KETTAB	Professeur ENP
Directeur(s) du mémoire :	S. BENMAMAR	Chargée cours ENP
	F. EL DABAGHI	Chargé de recherche INRIA
Examineurs :	C. LARBES	Maître de conférence
	S. BENZIADA	Chargé cours ENP
	A. BILEK	Chargé cours ENP

Juin 2006

الموضوع: الجريان ذو الوجهين مع الأخذ بعين الإعتبار الجانب البيوكيميائي للمحاكات الرقمية باستعمال الحساب الرقمي ذو الأداء العالي لمعالجة ظاهرة التزويد بالتهوية الميكانيكية.

ملخص: هذا العمل ينقسم إلى قسمين، القسم الأول مخصص لموازات محلل رقمي و القسم الثاني مخصص للنمذجة الكيميائية. المحلل الرقمي الذي قمنا بموازاته يحاكي جريان ذو وجهين - ماء/غاز - و ينمذج تطور التهوية الميكانيكية داخل بحيرة غنية بالأملاح المعدنية. نمذجة الجريان ذو الوجهين - ماء/غاز - تمت بإعتماد معادلات نافياي - ستوكس ذات البعدين و بصيغة سرعة - ضغط و هذا بالنسبة للوجه ماء. الوجه هواء عوض بشرط على الحدود على مستوى مضخ الهواء. المخطط الرقمي يعتمد على طريقة الخاصيات للحلول الزمنية، و على طريقة العناصر المنتهية ب1+فقاعة/ب1 للحلول الفضائية. المقاربة المعتمدة للموازات تتمثل في موازات الأقسام الأكثر إستهلاكاً لوقت الحساب للبرنامج. إختبارات رقمية بينت لنا أن قسم الحل، الذي يعتمد على خوارزمية التغيرات الموصولة أوزاوا، يستهلك أغلبية وقت الحساب. التغيرات الموصولة أوزاوا تحتاج إلى حل جمل مصفوفات بطريقة التغيرات الموصولة المكيفة، هذا الأخير يستهلك أكثر من 90% من وقت سي - بي - يو للبرنامج التسلسلي. علي هذا الأساس قمنا بالإقتصار على موازات طريقة التغيرات الموصولة المكيفة فقط. إختبارات رقمية للأداء على الأوقات سي - بي - يو و الكلية قد قدمت و هذا لإظهار نجاعة الطريقة المتبعة. الشق الثاني للأطروحة مخصص للنمذجة الكيميائية. عمل النمذجة المشروع فيه إنتهى بإقتراح نموذج كيميائي. هذا النموذج يعطي تطور تحلل و إنتشار غازي الأكسجين و النتروجين عبر الزمن.

كلمات رئيسية: التزويد، جريان ذو وجهين، معادلات نافياي - ستوكس، طريقة الخاصيات، عناصر منتهية ب1+فقاعة/ب1، تغيرات موصولة، حساب متوازي.

Sujet: Ecoulement diphasique avec prise en compte des aspects biochimiques pour la simulation numérique sous HPCN du traitement de l'eutrophisation par aération mécanique.

Résumé: ce travail est constitué de deux volets, le premier porte sur la parallélisation d'un solveur numérique et le deuxième porte sur la modélisation chimique.

Le solveur numérique parallélisé simule un écoulement diphasique eau-air et modélise le processus de l'aération mécanique dans un lac eutrophe. La modélisation de l'écoulement diphasique eau-air est faite par les équations de Navier-Stokes 2D en formulation vitesse pression pour la phase eau. La phase air est remplacée par une condition aux limites au niveau de l'injecteur. Le schéma numérique est basé sur la méthode des caractéristiques pour la résolution temporelle, et sur la méthode des éléments finis P1+bulle/P1, pour l'approximation spatiale. L'approche de parallélisation adoptée consiste à paralléliser les parties les plus consommatrices en temps de calcul du code. Des tests numériques nous ont montré que la partie résolution, basée sur un algorithme Gradient Conjugué Uzawa, est la partie qui consomme la majorité du temps de calcul. Ce gradient conjugué Uzawa nécessite la résolution des systèmes matriciels effectuée par gradient conjugué préconditionné, ce dernier consomme plus de 90% du temps CPU du code séquentiel. Par conséquent, on s'est focalisé uniquement sur le traitement parallèle de gradient conjugué préconditionné. Des tests numériques de performance sur les temps CPU et elapsed sont présentés afin de montrer l'efficacité de l'approche suivie. Le deuxième volet du mémoire est consacré à la modélisation chimique. Le travail entamé dans ce volet a abouti à la proposition d'un modèle chimique. Ce modèle donne le développement de la dissolution et de la diffusion du gaz de l'oxygène et du gaz de nitrogène à travers le temps.

Mots-clés: eutrophisation, écoulement diphasique, équations de Navier-Stokes, méthode des caractéristiques, éléments finis P1+bulle/P1, gradient conjugué, calcul parallèle.

Topic: Two phases flow with hold in account of the chemical aspects for the numeric simulation under HPCN of the treatment of the eutrophication by mechanical ventilation.

Abstracts: this work is constituted of two parts, the first is about the parallelization of a numeric solver and the second is about the chemical modelling. The parallelized numeric solver simulates a two phase flow water-air and modelling the process of the mechanical ventilation in a lake eutrophes. The modelling of the two phase flow water-air is made by the incompressible Navier-Stokes equations (2D) in formulation velocity-pressure for phase water. Phase air is replaced by a condition to the limits to the level of the injector. The numeric scheme is based on the characteristics method for the temporal resolution and on the mixed P1 + bubble/P1 finite element method for the spatial approximation. The approach of the adopted parallelization consists to parallelize the parts the more consumed in the time of the code calculation. Some numeric tests shows us that the part resolution, based on an algorithm Gradient Conjugated Uzawa, is the part that consumes the majority of the calculation time. This gradient conjugated Uzawa requires the resolution of the matrix systems done by gradient conjugated preconditioned, this last consumes more than 90% of the time CPU of the sequential code. Therefore, one focuses solely on the parallel treatment of gradient conjugated preconditioned. The numeric tests of performance on the elapsed times of calculation are presented in order to show the efficiency of the approach followed. The second part of the memory is dedicated to the chemical modelling. The work started in this part succeeded to the proposition of a chemical model. This model gives the development of the dissolution and the diffusion of the gas of oxygen and the gas of nitrogen through the time.

Keywords: eutrophication, two-phase flow, Navier-Stokes equations, characteristic method, P1+bubble/P1 finite element, conjugate gradient, parallel computing.

Dédicaces

Je dédié ce travail à :

Mes chers parents

Mes frères et à mes soeurs

Toute ma famille

Mes amis

Remerciements

Je tiens tout d'abord à remercier Mlle S. BENMAMAR qui a co-encadrée ce travail. Ses aides humaines et scientifiques ainsi que sa disponibilité et sa pédagogie ont été essentielles à la bonne conduite de ce mémoire.

Je remercie Mr. F. EL DABAGHI qui m'a accueilli au sein de son équipe (COSIVIE-INRIA). Je le remercie d'avoir co-encadré ce mémoire de magister. Je le remercie pour sa compétence de diriger ce travail, ses qualités d'encadrement et ses nombreuses discussions scientifiques qui était très bénéfique pour moi.

Ce stage n'aurait pu voir le jour sans la confiance et le soutien de Pr. M. M. K. BERRAH, Coordinateur du projet WADI pour la partie Algérienne. Je tiens à le remercier et lui exprimer ma reconnaissance.

Je remercie Pr. A. KETTAB qui m'a honoré d'accepter d'être le président de jury de ce mémoire.

Je remercie tous les membres de jury S. BENZIADA, A. BILEK, et C. LARBES qui ont bien voulu examiner ce travail.

Je remercie Mlle. D. ARAAR pour l'aide scientifique qu'il m'a donnée sur la partie modélisation chimique.

Je remercie Mr. M. ABDELOUAHED qui m'a orienté sur le choix de l'approche de parallélisation ; qu'il trouve ici ma reconnaissance pour son soutien.

Je remercie Mr. A. ELKACIMI, qui m'a fait profiter de ses connaissances théoriques et pratiques durant la période du stage.

Je remercie Mr. B. NAKHLE pour son suivi du près de mon travail sur la partie parallèle. Je le remercie beaucoup pour ce que j'ai appris avec lui, pour sa patience, son humour ainsi que pour sa gentillesse.

Je remercie particulièrement Mr. H. HENINE pour son accueil pendant mes premiers jours du stage, pour ses aides et ce au-delà de l'aspect purement scientifique.

Je remercie les thésards et les stagiaires de l'équipe COSIVIE (Mme. Guelmi, Chakib, Mounir et Xiomara) pour leurs collaboration et conseils, leur bonne humeur et leur sympathie.

Je remercie également toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail.

Enfin, je remercie mes parents pour leur sacrifice et soutien durant toutes ces années ; qu'ils trouvent ici l'expression de ma reconnaissance et de ma profonde gratitude.

Table des matières

Introduction	1
Chapitre 1	6
Modélisation physique	6
1.1 Introduction.....	6
1.2 Modèle physique instantané	7
1.2.1 Notations.....	7
1.2.2 Equations de conservation	9
1.2.3 Lois constitutives	10
1.2.4 Bilan interfacial.....	11
1.2.5 Bilan des variables et des équations.....	12
1.2.6 Commentaires sur le modèle instantané	12
1.3 Modèle à deux fluides	13
1.3.1 Définitions et notations	13
1.3.2 Equations de conservation	16
1.3.3 Lois constitutives	18
1.3.4 Conditions de transfert interfacial.....	22
1.3.5 Approches du modèle à deux fluides	23
1.3.6 Bilan des variables et des équations.....	27
1.3.7 Commentaires sur le modèle à deux fluides	27
1.4 Modèle de mélange	28
1.4.1 Variables du modèle	28
1.4.2 Equations de conservation	28
1.4.3 Lois constitutives	30
1.4.4 Commentaires sur le modèle de mélange	30
1.5 Approche lagrangienne	30
1.5.1 Forces exercées sur une bulle.....	31
1.5.2 Formulation du terme source	33
1.5.3 Commentaires sur l'approche lagrangienne	35
1.6 Modèles utilisés	35
1.6.1 Modèle monophasique	36
1.6.2 Modèle avec termes de correction diphasique.....	37
1.7 Conclusion.....	39
Chapitre 2.....	40
Modélisation biochimique de processus de l'aération	40
2.1 Introduction.....	40
2.2 Notations	40
2.3 Développement du model biochimique correspondant au modèle monophasique.....	41
2.3.1 Equations de conservation	41
2.3.3 Conditions initiales et aux limites	49

Table des matières

2.4 Développement du modelé biochimique correspondant au modèle monophasique corrigé	51
Chapitre 3.....	53
Approximation numérique du problème continu	53
3.1 Introduction	53
3.2 Notations	53
3.3 Position du problème	54
3.4 Discrétisation en temps (méthode des caractéristiques)	56
3.5 Problème continu	57
3.5.1 Formulation faible du problème de Stokes généralisé	57
3.5.2 Formulation variationnelle.....	59
3.6 Problème discret.....	60
3.6.1 L'élément P^1 +Bulle/ P^1	60
3.6.2 Discrétisation spatiale.....	62
3.7 Méthode de résolution matricielle	63
3.7.1 Algorithme d'Uzawa/Algorithme de gradient conjugué préconditionné.....	64
3.7.2 La méthode de gradient conjugué préconditionné (GCP)	67
3.8 Algorithme de résolution.....	68
3.9 Application.....	69
3.9.1 Simulations	71
3.10 Conclusion	72
Chapitre 4.....	73
Analyse de code NS incompressible et parallélisation avec MPI.....	73
4.1 Introduction	73
4.2 Position du problème	73
4.3 Présentation du calcul parallèle	74
4.3.1 Classification des ordinateurs	74
4.4 Problème à résoudre.....	77
4.5 Analyse du temps de code.....	79
4.5.1 Représentation graphique des pourcentages des temps dans le code principal et le solveur (STOBU1)	80
4.5.2 Représentation graphique des pourcentages des temps dans la subroutine RECGTB et PRODTB	80
4.5.3 Représentation graphique des pourcentages des temps dans la subroutine SOVRB et ICHR1R.....	81
4.6 Arborecence des appels qui consomme la majorité du temps.....	81
4.7 Parallélisation	83
4.7.1 Quelques bibliothèques de communications	83
4.7.2 MPI – Message Passing Interface	83
4.7.3 Modes de communication	88
4.7.4 Présentation de la HP 9000 V-Class	90
4.8 Analyse algorithmique et implémentation.....	93

Table des matières

4.8.1 Analyse de l'algorithme du gradient conjugué préconditionné	93
4.8.2 Gradient conjugué préconditionné en parallèle	96
4.8.3 Algorithme général.....	97
4.9 Présentation des cas test et Performance.....	98
4.9.1 Outils de performances	98
4.9.2 Présentation des cas tests	99
4.10 analyse des performances	104
4.11 Conclusion.....	105
Conclusions et perspectives	106
Bibliographie	108

Table des figures

Figure 1-1 : Coupe 2D du domaine d'étude au temps t	8
Figure 1-2 : Intervalle de temps $[\Delta t]$	14
Figure 1.3 : Volume de contrôle.....	33
Figure 2.1 : Dessin schématique des bulles d'air	46
Figure 2.2 : Le voisinage de l'injecteur.....	50
Figure 3-1 : Approximation de la dérivée totale	56
Figure 3.2 : Degrés de liberté pour l'élément P^1 +bulle/ P^1	61
Figure 3-3 : Position de la coupe 2D sur la bathymétrie.....	70
Figure 3-4 : Position de l'injecteur et directions des vecteurs vitesses.....	70
Figure 3-5 : Maillage de la coupe, 10791 nœuds et 20760 éléments.....	70
Figure 3-6 : Zooms au voisinage de l'injecteur	71
Figure 3-7 : Représentation des isovalues de la norme de la vitesse et des vecteurs vitesses aux instants $t=1s, 2s$ et $3s$	72
Figure 4-1 : Pourcentages des temps dans les code total et le solveur.....	80
Figure 4-2 : Pourcentages des temps dans les sousroutines RECGTB et PRODTB	80
Figure 4-3 : Pourcentages des temps dans les sousroutines SOVRB et ICHR1R	81
Figure 4-4 : Structure globale du code.....	81
Figure 4.5 : Algorithme du gradient conjugué d'Uzawa (les parties parallélisées sont montrées en gras italique).....	82
Figure 4-6 : Schématisation des communicateurs	85
Figure 4-7 : Architecture interne de la V-Class.....	91
Figure 4-8 : Diagramme de la configuration de l'ERAC.....	92
Figure 4-9 : Algorithme du gradient conjugué pour résoudre le système $AX=b$	94
Figure 4-10 : Produit matrice-vecteur $A.X$	95
Figure 4-11 : Schématisation du produit matrice -vecteur $A.X$	95
Figure 4-12 : Temps absolu de calcul (gradient conjugué)	100
Figure 4-13 : Temps absolu total d'exécution « Elapsed » (gradient conjugué)	100
Figure 4-14 : Speed-Up calcul (gradient conjugué)	101
Figure 4-15 : Speed-Up Elapsed (gradient conjugué).....	101
Figure 4-16 : Temps absolu de calcul (gradient conjugué)	102
Figure 4-17 : Temps absolu total d'exécution « Elapsed » (gradient conjugué)	103
Figure 4-18 : Speed-Up calcul (gradient conjugué)	103
Figure 4-19 : Speed-Up Elapsed (gradient conjugué).....	104
Figure 4-20 : Influence de l'overhead sur le temps d'exécution.....	104

Liste des tableaux

Tableau 1.1 : Données du problème	9
Tableau 1.2 : Variables du problème.....	9
Tableau 1.3 : Bilan des variables et des équations.....	12
Tableau 1.4 : Les différentes expressions de ψ_k	16
Tableau 1.5 : Variables principales	27
Tableau 1.6 : Variables secondaires.....	27
Tableau 1.7 : Variables principales du modèle de mélange	28
Tableau 1.8 : Variables secondaires du modèle de mélange.....	28
Tableau 2.1 : Solubilité des gaz dans l'eau (<i>F. Edeline</i> , 1996)	42
Tableau 2.2 : Coefficients de transferts et les constantes d'Henry.	44
Tableau 2.3 : Coefficients moléculaires de diffusion. (<i>F. Edeline</i> , 1996).....	44
Tableau 2.4 : Coefficient β	49
Tableau 4.1 : Maillages de la cavité unité (1)	99
Tableau 4.2 : Maillages de la cavité unité (2)	102

Introduction

Le problème de l'eutrophisation des eaux naturelles est l'un des conséquences de l'interaction croissante entre l'homme et son environnement. L'augmentation des sels minéraux (notamment nitrates et phosphates) provenant essentiellement des effluents urbains et agricoles provoque une prolifération massive de la végétation aquatique, en particulier les algues. Ces algues en excès conduisent, lorsqu'elles se décomposent, à une augmentation de la charge naturelle de l'écosystème en matières organiques biodégradables. Les algues mortes se déposent dans les profondeurs du lac, les bactéries aérobies qui s'en nourrissent prolifèrent à leur tour, consommant de plus en plus d'oxygène. Or, en l'absence de circulation suffisante des eaux, ce qui est le cas souvent des lac profond, le fond du lac est peu oxygéné et les bactéries finissent par épuiser l'oxygène des couches d'eau profondes. Elles ne peuvent plus dégrader toute la matière organique morte et celle-ci s'accumule dans les sédiments. Une telle situation, lorsqu'elle se produit, s'aggrave encore lorsqu'il fait chaud car la solubilité de l'oxygène dans l'eau (comme celle de tous les gaz) diminue lorsque la température augmente. Au fur et à mesure de la raréfaction de l'oxygène, les bactéries réalisent des fermentations anaérobies, qui produisent de l'hydrogène sulfuré, à l'odeur putride, et de l'ammoniac. A terme, toute vie animale disparaît. Dans les zones arides et semi arides, les variations climatiques et en particulier thermique durant l'année, engendrent une division du lac en trois zones différentes et superposées :

- épilimnion : couche superficielle peu épaisse d'environ 7m, brassée par les vents où la teneur en oxygène est proche de la saturation.
- Thermocline : zone intermédiaire soumise à une chute rapide de la température (du 27°C à 18°C). Elle peut s'étendre sur une profondeur allant de 7m à 12m, peu soumise à l'action des vents où la teneur en oxygène est moyenne ;
- Hypolimnion : couche profonde au-delà de 12m, avec des températures d'environ 14°C à 18°C. Dans les lac eutrophes, la teneur en oxygène est faible (si la concentration en oxygène est inférieure à 3mg/l dans le lac, celui-ci est considéré en phase d'eutrophisation) ; on y relève de fortes concentrations de gaz nocifs tels que hydrogène sulfuré, ammoniac, gaz carbonique.

Cette stratification thermique empêche les échanges entre les différentes couches ; c'est un peu comme si on obtenait une couche d'huile (eau chaude) qui flotterait sur une couche d'eau (eau froide). Il devient alors très difficile de brasser ces deux couches d'eau. Néanmoins, on peut avoir un brassage superficiel de l'épilimnion à l'aide du vent.

A part des répercussions écologiques catastrophiques, le phénomène de l'eutrophisation a des conséquences économiques non négligeable : dégradation des pêcheries, augmentation du coût du traitement de l'eau potable suite à la dégradation de la qualité d'eau, perte de l'aspect touristique des lac ... etc.

Il est donc impératif, face à ces résultats catastrophiques de l'eutrophisation, de lutter contre ce fléau. Il existe plusieurs techniques de restauration des lacs ou de prévention contre l'eutrophisation ; on peut citer (*S. Phelippot*, 1975):

- 1- Contrôle chimique : à l'aide des substances chimiques on empoisonne les algues et les plantes aquatiques trop envahissantes.
- 2- Contrôle biologique : ce type de lutte s'effectue en employant des organismes consommateurs d'algues et d'herbes aquatiques ou des parasites d'algues.
- 3- Siphonage et lavage : le siphonage des eaux profondes est destiné à éliminer ces eaux de très basse qualité dans des lacs eutrophiés. Le lavage (flushing) consiste à diluer les eaux d'un lac trop riches en substances nutritives à l'aide du débit puissant d'une rivière proche de bonne qualité que l'on dérive momentanément.
- 4- Aération mécanique : elle consiste à injecter de l'air sous pression à une certaine profondeur du lac afin d'aérer ces zones pauvres en oxygène soit par dissolution d'air ou en poussant l'eau vers le haut afin de l'oxygéner par contact des eaux superficielles ou par l'air atmosphérique. On peut aérer tout le lac s'il est peu profond (déstratification) et on peut aussi limiter l'aération à l'hypolimnion afin d'aérer que la couche profonde et de préserver la stratification.
 - a- déstratification : lors de l'aération, l'eau de la surface riche en algues remplace l'eau du fond et, au fur et mesure que le lac s'homogénéise, la distribution des algues devient uniforme. Ainsi, la densité moyenne des cellules par unités de volume diminue et, par conséquent, la clarté de l'eau augmente. D'autre part, quand on mélange l'eau en la déstratifiant, les algues restent moins longtemps dans la zone éclairée et se multiplient moins vite.
 - b- Aération de l'hypolimnion : si le lac est de grande profondeur, il importe de ne pas perturber son équilibre thermique, il faut augmenter fortement la concentration en oxygène de l'hypolimnion.

La technique qui nous intéresse dans ce mémoire c'est l'aération mécanique. On se propose d'étudier l'écoulement diphasique engendré par les bulles d'airs injectées dans le lac. La modélisation physique de cet écoulement, sans tenir compte de l'aspect chimique, se base sur un modèle diphasique constitué principalement par des équations de Navier – Stokes instantanées pour chaque phase, associées à des relations de type Rankine-Hugoniot représentant l'interaction entre phases. Des modèles simplifiés déduits de ce modèle général peuvent être utilisés pour décrire ce type d'écoulement.

La simulation numérique de tel écoulement pose d'énormes difficultés dues principalement à la complexité de la modélisation de l'écoulement diphasique (eau-bulles d'airs) et les coûts exorbitants des calculs 3D vu la taille du problème traité. Ces contraintes ont ramené, dans les travaux antérieurs, à entreprendre en 2D les deux approches suivantes (*M. Abdelwahed*, 2002 ; *E. Clement*, 1997) :

La première approche basée sur le fait que la phase air est négligée devant la phase eau. Elle consiste à traiter une seule phase, la phase eau, en tenant compte de l'effet

dynamique des bulles par le biais d'une condition aux limites au niveau de l'injecteur. Ce premier modèle - appelé modèle monophasique - est donnée par les équations de Navier-Stokes semi-compressible comme suit :

- Soit Ω un domaine plan et sa frontière Γ (on suppose qu'elles vérifient les conditions mathématique nécessaire) ; \mathbf{n} le vecteur unitaire normal à Γ ; $T \in \mathfrak{R}_+^*$, $Q_T = \Omega \times]0, T[$, $\Sigma_T = \Gamma \times]0, T[$ et $\Sigma(\mathbf{u}_d) = \{(\mathbf{x}, t) \in \Gamma \times]0, T[\mid \mathbf{u} \cdot \mathbf{n} < 0\}$.
- On se donne $\rho^0 : \Omega \rightarrow \mathfrak{R}$, $\mathbf{f} : Q_T \rightarrow \mathfrak{R}^2$, $\mathbf{u}^0 : \Omega \rightarrow \mathfrak{R}^2$, $\mathbf{u}_d : \Sigma_T \rightarrow \mathfrak{R}^2$.

On cherche $\rho : Q_T \rightarrow \mathfrak{R}$ la masse volumique de l'eau, $\mathbf{u} : Q_T \rightarrow \mathfrak{R}^2$ sa vitesse et $p : Q_T \rightarrow \mathfrak{R}$ sa pression telles que :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0 & \text{dans } D \\ \rho(., t = 0) = \rho^0(.) & \text{dans } \Omega(t) \\ \rho & \text{donnée sur } \Sigma(\mathbf{u}_d) \end{cases}$$

$$\begin{cases} \rho_L \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \text{div}(\mu_L (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \nabla p = \mathbf{f} & \text{dans } D \\ \text{div} \mathbf{u} = 0 & \text{dans } D \\ \mathbf{u}(., t = 0) = \mathbf{u}^0(.) & \text{dans } \Omega(t) \\ \mathbf{u} = \mathbf{u}_d & \text{dans } \Gamma_F(t) \cap \partial \Omega_G(t) \end{cases}$$

Où $\mathbf{f} = \rho \mathbf{g}$ avec \mathbf{g} la force de gravité.

La deuxième approche est plus complète que la précédente, elle est basée sur le modèle précédent corrigé par des termes diphasiques tenant compte de l'effet des bulles. Ce modèle s'écrit comme suit :

- Etant donné
 - $\alpha^0 : \Omega \rightarrow \mathfrak{R}$ taux de présence initial de la phase gaz (lorsque $\alpha^0 = 0$, il n'y a pas de bulles d'air),
 - $\mathbf{f} : Q_T \rightarrow \mathfrak{R}^2$, $\mathbf{u}^0 : \Omega \rightarrow \mathfrak{R}^2$, $\mathbf{u}_d : \Sigma_T \rightarrow \mathfrak{R}^2$,
 - ρ_G : la masse volumique de la phase gaz,
 - μ_L et μ_G les viscosités respectives des phases liquides et gaz.
- On cherche $\alpha : Q_T \rightarrow \mathfrak{R}$, $\rho_L : Q_T \rightarrow \mathfrak{R}$ la masse volumique de la phase liquide, $\mathbf{u} : Q_T \rightarrow \mathfrak{R}^2$ et $p : Q_T \rightarrow \mathfrak{R}$ telles que :

$$\left\{ \begin{array}{ll}
\frac{\partial \alpha}{\partial t} + \mathbf{u}_L \cdot \nabla \alpha = 0, \frac{\partial \rho_L}{\partial t} + \mathbf{u}_L \cdot \nabla \rho_L = 0 & \text{dans } Q_T \\
\rho_L(., t = 0) = \rho_L^0(.), \alpha(., t = 0) = \alpha^0(.) & \text{dans } \Omega \\
\alpha \text{ donnée}, \rho_L \text{ donnée sur } \Sigma(\mathbf{u}_d) & \\
((1 - \alpha)\rho_L + \alpha\rho_G) \frac{d\mathbf{u}_L}{dt} - \text{div}(\mu_{eff} (\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T)) + \nabla p_L = \rho_L \mathbf{g} + \alpha(\rho_G - \rho_L) \mathbf{g} & \text{dans } Q_T \\
\text{div} \mathbf{u}_L = 0 & \text{dans } Q_T \\
\mathbf{u}_L(., t = 0) = \mathbf{u}_L^0(.) & \text{dans } \Omega \\
\mathbf{u}_L = \mathbf{u}_d & \text{sur } \Sigma_T
\end{array} \right.$$

Ce mémoire est constitué de 4 chapitres, ils sont ordonnés comme suit :

- 1- modélisation physique
- 2- modélisation chimique du processus de l'aération
- 3- analyse numérique du problème continu
- 4- analyse de code NS incompressible et parallélisme avec MPI

Dans le premier chapitre, on décrit les différents modèles des écoulements diphasique. On commence par le modèle complet : le modèle physique instantané et après les différents modèles déduits de ce dernier. On termine le chapitre par la présentation des modèles monophasique et monophasique corrigé. Le deuxième chapitre aborde l'aspect chimique de processus de l'aération, un modèle de dissolution de l'air dans l'eau a été donné. L'air est considéré comme un gaz constitué de deux gaz principal : l'oxygène et le nitrogène ; les autres gaz sont négligés. Les conditions initiales et aux limites liées aux modèles chimiques seront traitées dans un paragraphe à part. On note que le travail fait dans ce mémoire concernant l'aspect chimique est un travail de modélisation. La résolution numérique de ces modèles n'est pas abordée à cause des délais.

Le troisième chapitre est consacré à la démarche prise pour discrétiser les équations de Navier – Stokes et un aperçu est donné sur les conditions qu'il faut vérifier pour assurer l'existence et l'unicité de la solution. La partie de transport de la vitesse \mathbf{u} :

$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}$ est discrétisée par la méthode des caractéristiques. L'équation de Stokes

généralisée déduite de l'étape précédente est discrétisée par la méthode des éléments finis mixte P1 continu +bulle (vitesse)/P1 continu (pression). Ensuite, le système matriciel obtenu est donnée par :

$$A\mathbf{u} + Bp = f$$

$$B^T \mathbf{u} = 0$$

Où $A \in \mathfrak{R}^{n,n}$ est une matrice symétrique définie positive, $B \in \mathfrak{R}^{n,m}$, $u \in \mathfrak{R}^n$, $p \in \mathfrak{R}^m$ et $f \in \mathfrak{R}^n$.

Après l'élimination de la vitesse du système précédent on obtient : $B^T A^{-1} B p = B^T A^{-1} f$ Ce système est résolu par une méthode de gradient conjugué préconditionné (algorithme d'Uzawa). Le préconditionnement est fondamentale, car ce système linéaire est mal conditionné : le choix adopté est celui de Chaouet-Chabard. A la fin de chapitre, des essais numériques sont présentés.

Dans le quatrième chapitre, on parle brièvement sur le calcul parallèle, la bibliothèque MPI ainsi sur les différentes machines parallèles. En fait, les problèmes traités réellement nécessitent un très grand temps de calcul et une place mémoire importante. Ce qui rend nécessaire la parallélisation du code. Le travail de parallélisation a été mené sur la partie solveur du code, cette partie consomme en général, plus de 90% de temps de calcul. En fin du chapitre, des tests de performance sur une machine parallèle sont présentés. Le travail de parallélisation constitue le premier volet principal de ce mémoire, le deuxième volet c'est la modélisation chimique de phénomène de l'aération.

Chapitre 1

Modélisation physique

1.1 Introduction

L'eutrophisation a des conséquences écologiques et économiques non négligeables, tel que la mort de la faune, la dégradation des pêcheries et l'augmentation du coût du traitement de l'eau potable. Pour ces raisons, la restauration des lacs et la lutte contre l'eutrophisation se révèlent indispensables. Parmi les solutions utilisées dans la pratique, nous avons :

- extraction des sédiments, curage, surtout pour les étangs ;
- recouvrement des sédiments, ajout de calcium pour limiter le recyclage du phosphore sédimentaire ;
- oxygénation des eaux profondes sans déstratification ;
- évacuation des eaux profondes vers l'aval pour soutirer le phosphore ;
- déstratification par insufflation d'air.

Le procédé de la lutte contre l'eutrophisation qui nous intéresse c'est le dernier : l'insufflation de l'air comprimé à l'aide des tubes perforés placés à proximité du fond. Ce procédé permet l'aération du lac ainsi que sa déstratification thermique. Les bulles d'airs ont un diamètre environ dix fois supérieur à celui du trou dont elles sortent. On donne les mesures suivantes (*F. Edeline*, 1996) :

Ø orifice (mm)	Diamètre moyen des bulles (mm)	Ø orifice (mm)	Diamètre moyen des bulles (mm)
0.4	4.7	1.6	7.7
0.8	6.6	3.2	7.2

Dans le lac, nous avons en fait deux fluides en contact : l'air (ou bien bulles d'air) et l'eau. Ces deux fluides sont en mouvement. Ils échangent entre eux de l'énergie et il va s'établir un transfert de gaz entre les deux phases.

Dans ce chapitre, Nous nous intéressons à l'écoulement diphasique établi dans un lac. Sa modélisation pose d'énormes difficultés dues principalement à la présence de l'interface entre la phase liquide et la phase gaz.

Les différents modèles existants et les difficultés inhérentes à chaque modèle sont présentés. Nous terminerons ce chapitre par la présentation des modèles proposés.

Nous commençons par présenter le modèle diphasique instantané basé sur les équations de Navier – Stokes compressible pour chaque phase, avec des conditions de saut aux interfaces traduisant la relation entre les phases. Deux modèles dérivés du modèle instantané sont présentés : le premier, appelé modèle à deux fluides, qui consiste à appliquer une moyenne temporelle sur les équations instantanées pour obtenir un système pratiquement utilisable avec des grandeurs mesurables. Le deuxième modèle, appelé modèle de mélange, consiste à considérer les deux phases comme un seul fluide caractérisé par des variables de mélange.

Nous présentons, ensuite, l'approche Lagrangienne qui consiste à traiter une seule phase qui est la phase eau et à tenir compte de l'effet de la phase bulle par l'introduction de termes de correction.

A la fin du chapitre, une présentation d'un modèle simplifié à une phase avec termes de correction traduisant l'effet des bulles est donnée.

Deux approches sont étudiées. La première approche a pour objectif de montrer l'effet de l'injection des bulles en se limitant simplement à une condition aux limites aux injecteurs. La deuxième approche, plus élaborée, tient compte de l'effet des bulles par le biais de termes de correction déterminés à partir des modèles diphasiques.

1.2 Modèle physique instantané

Pour mieux comprendre la dynamique et les phénomènes qui régissent les écoulements diphasiques, on utilise souvent une formulation locale instantanée. Elle consiste à regarder un écoulement diphasique comme étant un ensemble de régions monophasique limitées par des interfaces en mouvement.

Localement, le comportement de chaque phase peut être décrit par des équations aux dérivées partielles traduisant les bilans de conservations de la masse, de la quantité de mouvement et de l'énergie. La relation entre les phases se traduit ensuite par des conditions de saut aux interfaces.

1.2.1 Notations

Nous désignons par $x = (x_1, x_2, x_3)$ les coordonnées d'un point de l'espace, t la variable de temps et T un temps maximal d'étude.

Nous notons k la variable de phase définie par $k = G$ pour le gaz et $k = L$ pour le liquide. La phase L occupe un domaine volumique variable en temps.

Nous notons par Ω (Figure 1.1) ce domaine, sa surface est une frontière libre variable au cours du temps. La frontière Γ se décompose en deux parties : le fond Γ_F défini comme étant la partie séparant la phase L de la terre, et sa surface Γ_S qui la sépare de l'atmosphère. Nous négligeons les variations de Γ_S au cours du temps et nous supposons que : Ω , Γ_F et Γ_S sont indépendantes du temps.

$\Omega_G(t)$ et $\Omega_L(t)$ représentent les parties de Ω occupées respectivement par la phase G et la phase L au temps t et $\Gamma_i(t) = \partial\Omega_G(t) \cap \partial\Omega_L(t)$ représente l'interface au temps t entre les deux phases.

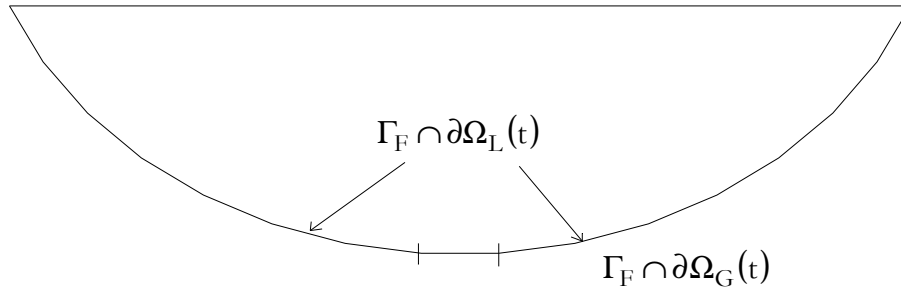


Figure 1-1 : Coupe 2D du domaine d'étude au temps t

On définit les ensembles suivants :

$$D = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Omega\}$$

$$D_k = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Omega_k(t)\}, \quad k = G, L,$$

$$F = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Gamma\}$$

$$F_k = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Omega_k(t)\}, \quad k = G, L,$$

$$F_i = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Gamma_i(t)\}$$

$$F_S = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Gamma_S(t)\}$$

$$F_F = \{(x, t) \in \mathfrak{R}^3 \times]0, T[; x \in \Gamma_F\}.$$

Les données du problème sont présentées dans le tableau 1.1

Symboles	définitions	Type
\mathbf{F}_k	Forces volumiques	Fonction vectorielle
Q_k	Source de chaleur	Fonction scalaire
μ_k	Viscosité moléculaire	Constante scalaire
λ_k	Viscosité volumique	Constante scalaire
K_k	Conductivité thermique	Constante scalaire

Tableau 1.1 : Données du problème

Les variables du problème sont données dans le tableau 1.2.

Symboles	Variabes	Type
ρ_k	Masse volumique	Fonction scalaire
\mathbf{u}_k	Vitesse	Fonction vectorielle
p_k	Pression	Fonction scalaire
e_k	Energie interne	Fonction scalaire
s_k	Entropie	Fonction scalaire
T_k	Température	Fonction scalaire
$\boldsymbol{\pi}_k$	Contraintes visqueuses	Fonction tensorielle
\mathbf{j}_k	Flux de température	Fonction tensorielle

Tableau 1.2 : Variables du problème

1.2.2 Equations de conservation

Les équations décrivant l'écoulement de la phase k dans le domaine D_k sont déduites par les équations de conservation de la mécanique.

- **Equation de conservation de la masse**

$$\frac{\partial \rho_k}{\partial t} + \operatorname{div}(\rho_k \mathbf{u}_k) = 0 \text{ dans } D_k \quad (1.1)$$

- **Equation de conservation de la quantité de mouvement**

$$\frac{\partial \rho_k u_k}{\partial t} + \operatorname{div}(\rho_k u_k u_k) = -\nabla p_k + \operatorname{div} \pi_k + \rho_k F_k \quad \text{dans } D_k \quad (1.2)$$

- **Equation de conservation de l'énergie**

$$\frac{\partial}{\partial t} \left(\rho_k \left(e_k + \frac{u_k^2}{2} \right) \right) + \operatorname{div} \left(\rho_k \left(e_k + \frac{u_k^2}{2} \right) u_k \right) = \operatorname{div}(\pi_k u_k) - \operatorname{div} j_k - \operatorname{div}(\rho_k u_k) + \rho_k F_k \cdot u_k + Q_k \quad \text{dans } D_k \quad (1.3)$$

Le système composé des équations (1.1) ÷ (1.3) est ouvert.

1.2.3 Lois constitutives

Il est nécessaires d'ajouter des équations complémentaires pour fermer les système (1.1) ÷ (1.3), nous en distinguons trois type : équations d'état, équations constitutives mécaniques et équations constitutives énergétiques.

- **Equations fondamentales d'état**

La forme de l'équation fondamentale d'état permet d'exprimer l'énergie interne en fonction de l'entropie et de la masse volumique, elle s'écrit :

$$e_k = e_k(s_k, \rho_k) \quad (1.4)$$

L'entropie vérifie le second principe de la thermodynamique donné par :

$$\frac{\partial \rho_k s_k}{\partial t} + \operatorname{div}(\rho_k s_k u_k) + \operatorname{div} \left(\frac{j_k}{T_k} \right) - \frac{Q_k}{T_k} \geq 0 \quad \text{dans } D_k \quad (1.5)$$

Comme la température et la pression thermodynamique sont données par :

$$T_k = \frac{\partial e_k}{\partial s_k}, \quad -p_k = \frac{\partial e_k}{\partial \left(\frac{1}{\rho_k} \right)} \quad (1.6)$$

On a :

$$p_k = p_k(\rho_k, T_k), \quad e_k = e_k(\rho_k, T_k) \quad \text{dans } D_k \quad (1.7)$$

Dans le cas des fluides parfaits, les équations (1.7) s'écrivent :

$$p_k = R_k \rho_k T_k, \quad e_k = C_{vk} T_k \quad (1.8)$$

R_k et C_{vk} sont respectivement, la constante des gaz parfaits et la chaleur spécifique à masse volumique constante de la phase k .

- **Equation constitutive mécanique**

Elle permet d'exprimer le tenseur de contraintes pour un fluide Newtonien :

Fluide non visqueux

$$\pi_k = 0 \quad \text{dans } D_k \quad (1.9)$$

Fluide visqueux

$$\pi_k = \mu_k (\nabla u_k + \nabla u_k^T) - \left(\frac{2}{3} \mu_k - \lambda_k \right) (\text{div} u_k) \mathbb{I} \quad \text{dans } D_k \quad (1.10)$$

• **Equation constitutive énergétique**– Flux de température

Le transfert de chaleur par contact est exprimé par le vecteur flux de température j_k . Son équation constitutive spécifie la nature et le mécanisme du transfert d'énergie par contact. Plusieurs fluides obéissent à la loi de Fourier généralisée qui s'écrit :

$$j_k = -K_k (T_k) \nabla T_k \quad \text{dans } D_k \quad (1.11)$$

où, $K_k(\cdot)$ est la conductivité thermique supposée souvent constante.

– Source de chaleur

La source de chaleur Q_k résulte des sources d'énergie extérieures et des interactions réciproques. Elle est souvent supposée nulle.

1.2.4 Bilan interfacial

Le bilan interfacial est déterminé à partir des conditions de saut de Rankine – Hugoniot.

Equilibre interfacial de masse

L'équation de l'équilibre interfacial de masse est :

$$\sum_{k=L,G} \rho_k n_k \cdot (u_k - u_i) = 0 \quad (1.12)$$

avec, u_i la vitesse à l'interface entre les deux phases et n_k est la normale à la phase k .

Equilibre interfacial de quantité de mouvement

L'équation de l'équilibre interfacial de quantité de mouvement s'écrit :

$$\sum_{k=L,G} \{ \rho_k n_k \cdot (u_k - u_i) u_k - n_k \cdot (\pi_k - p_k \mathbb{I}) \} + (t_{\gamma a}^{\gamma \beta} \sigma)_{,\beta} = 0 \quad (1.13)$$

où, σ , $t_{\gamma a}^{\gamma \beta}$ et $(\cdot)_{,\beta}$ sont respectivement la tension surfacique, le tenseur hybride entre les coordonnées spatiales et surfaciques, le tenseur métrique de l'interface et la dérivée surfacique covariante (*M. Abdelwahed, 2002*).

Equilibre interfacial d'énergie

L'équation de l'équilibre interfacial d'énergie est :

$$\sum_{k=L,G} \left\{ \rho_k n_k \cdot (u_k - u_i) \left(e_k + \frac{u_k^2}{2} \right) + n_k \cdot \left((-\pi_k + p_k \mathbf{II}) \cdot u_k + j_k \right) \right\} + \left(t_{\gamma a} \gamma_{\beta} \sigma_{u_i} \right)_{,\beta} = \frac{\partial e_a}{\partial t} + \text{div}(e_a u_i) \quad (1.14)$$

Le terme à droite représente le taux de change d'énergie à l'interface et celui de gauche représente le transfert d'énergie de chaque côté et le travail effectué par l'interface.

1.2.5 Bilan des variables et des équations

Le bilan des variables et des équations est donné dans le tableau 1.3 ci-dessous.

Inconnues	Nombres	Equations	Nombre
ρ_k	1	Masse	1
\mathbf{u}_k	3	Mouvement	3
p_k	1	Loi d'état	1
e_k	1	Loi d'état	1
T_k	1	Energie	1
π_k	9	Constitutives mécanique	9
j_k	3	Constitutives énergétiques	3
Total	20	Total	20

Tableau 1.3 : Bilan des variables et des équations

Aux équations (1.1) ÷ (1.14), s'ajoutent des conditions initiales et des conditions aux limites.

1.2.6 Commentaires sur le modèle instantané

Une modélisation diphasique basée sur les variables locales instantanées et sur les variables aux interfaces en mouvement, amène à une situation avec des frontières multiples ayant des positions inconnues. D'autre part, les échelles de longueur attachée aux bulles sont très différentes des tailles des structures caractéristiques de l'écoulement de la phase continue. La résolution d'un tel problème soulève trop de difficultés mathématiques et numériques et n'est pas envisageable actuellement. Récemment, des tentatives ont été faites pour des simulations directes sur quelques bulles en dimension trois (3D) et une centaine de bulles en dimension deux (2D) dans un petit domaine. En simulation 2D, Esmaeli et Trygvason (**A. Esmaeli et**

al., 1996) ont présenté une simulation d'une centaine de bulles dans un petit carré utilisant un modèle simplifié.

Pour obtenir un système pratiquement utilisable avec des grandeurs que l'on peut effectivement mesurer, on est souvent amené à effectuer des opérations de moyenne. On aboutit au modèle à deux fluides présenté ci-dessous.

1.3 Modèle à deux fluides

Le modèle à deux fluides est formulé en considérant chaque phase séparément. Ainsi, le modèle comprend des équations de conservation de la masse, de la quantité de mouvement et de l'énergie de chaque phase avec des termes d'interaction à l'interface qui apparaissent dans ces équations. Ces équations sont obtenues en effectuant des opérations de moyenne de la formulation locale instantanée des équations (1.1) ÷ (1.11). Plusieurs types de moyennes sont utilisés :

- des moyennes temporelles (*Ishii*, 1975),
- des moyennes spatiales (*Enwald et al.*, 1996),
- des moyennes spatio-temporelles (*Drew et Lahey*, 1998).

Cependant, la forme résultante des équations ne diffère pas sensiblement ; ce qui change, c'est le sens physique des quantités moyennes et l'expression des termes d'échange à l'interface.

Dans la suite, on considère un système d'équations obtenu en effectuant des moyennes temporelles. Ainsi, en vertu de cette moyenne, les deux phases sont considérées comme présentes dans tout l'espace. Par conséquent, les équations modélisant l'écoulement diphasique doivent être résolues simultanément dans tout le domaine représentant l'écoulement.

1.3.1 Définitions et notations

Notations

Les écoulements diphasiques de deux fluides non miscibles sont caractérisés par la présence d'une interface entre les deux phases. On note :

- δ l'épaisseur de l'interface,
- u_i la vitesse à l'interface,
- n_j la normale à l'interface j ,
- $u_{ni} = u_i \cdot n_j$ la composante normale de la vitesse u_i à l'interface j ,
- l'intervalle de temps associé à chaque interface j (Figure 1-2) est donné

$$\text{par : } 2 \varepsilon_j = \frac{\delta}{u_{ni}}$$

On définit (Figure 1-2) :

- Le temps t_j correspondant à l'interface entre les deux phases,
- L'ensemble d'intervalles de temps associés aux interfaces par :

$$[\Delta t]_j = [t_j - \varepsilon_j, t_j + \varepsilon_j] \quad j = 1, \dots, n \quad , \quad n \text{ nombre d'interfaces}$$

- $[\Delta t]_L$ l'ensemble des intervalles de temps représentant la phase L (Figure 1-2).
- $[\Delta t]_G$ l'ensemble des intervalles de temps représentant la phase G (Figure 1-2).
- $[\Delta t]_T = [\Delta t]_L + [\Delta t]_G$

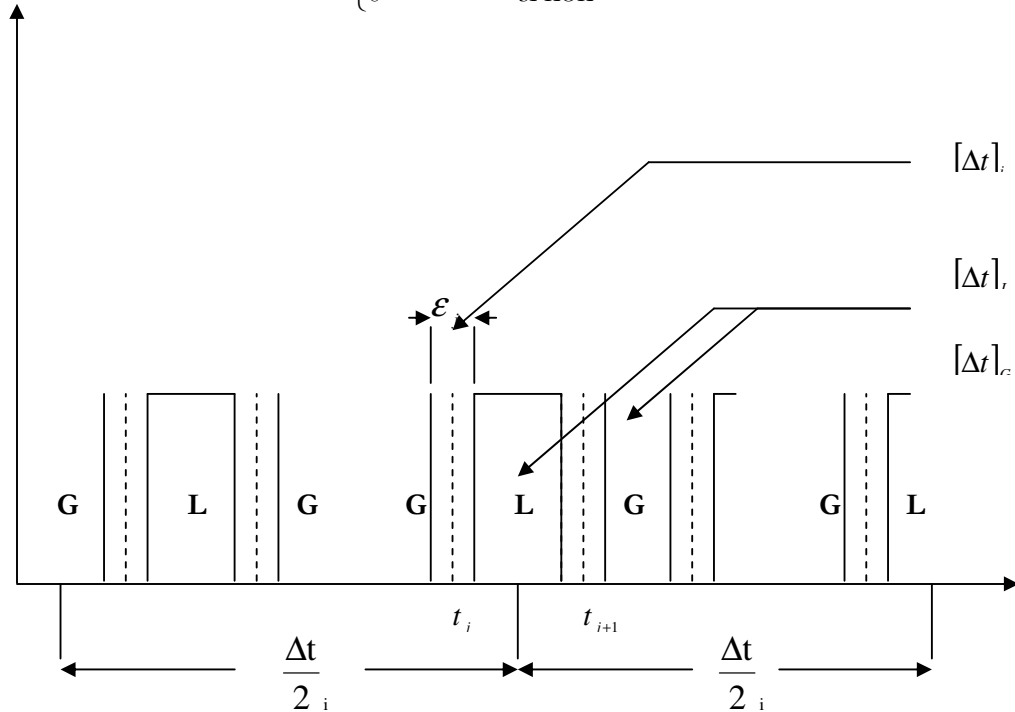
- $[\Delta t] = [\Delta t]_T + \sum_{j=1}^n [\Delta t]_j \quad , \quad t \in \left[t_0 - \frac{\Delta t}{2}, t_0 + \frac{\Delta t}{2} \right]$

- La fonction d'état $X_k(x, t)$ par :

$$X_k(x, t) = \begin{cases} 1 & \text{si } x \text{ appartient à la phase } k \text{ au temps } t \\ 0 & \text{si non} \end{cases} \quad (1.15)$$

- La fonction de la phase F_k , en un point x_0 par :

$$F_k(x_0, t) = X_k(x_0, t)F = \begin{cases} F(x_0, t) & \text{si } t \in [\Delta t]_k \\ 0 & \text{si non} \end{cases} \quad (1.16)$$

Figure 1-2 : Intervalle de temps $[\Delta t]$

Moyenne temporelle

La moyenne temporelle eulérienne d'une fonction F est définie par :

$$\bar{F}(x_0, t_0) = \lim_{\delta \rightarrow 0} \frac{1}{\Delta t} \int_{[\Delta t]_T} F(x_0, t) dt \quad (1.17)$$

On introduit la variable de phase α_k en un point (x_0, t_0) comme étant la moyenne temporelle de la fonction d'état définie par (1.15) :

$$\alpha_k(x_0, t_0) = \bar{X}_k(x_0, t_0) = \lim_{\delta \rightarrow 0} \frac{1}{\Delta t} \int_{[\Delta t]_T} X(x_0, t) dt \quad (1.18)$$

Elle vérifie :

- $\alpha_L(S, t) = 1$ en tout point S dans un volume occupé par la phase L au temps t ,
- $\alpha_G(S, t) = 1$ en tout point S dans un volume occupé par la phase G au temps t ,
- α_k vérifie $\alpha_L(S, t) + \alpha_G(S, t) = 1$ en tout point S du domaine $\Omega(t)$.

Propriétés de la moyenne temporelle

Soit F_k une fonction qui peut être un scalaire, un vecteur ou un tenseur. On a les propriétés suivantes :

$$\frac{\partial \bar{F}_k}{\partial t} = \frac{\partial F_k}{\partial t} - \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} (F_k n_k \cdot u_i) \quad (1.19)$$

$$\bar{\nabla F}_k = \nabla F_k + \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} (n_k F_k) \quad (1.20)$$

$$\overline{\text{div} F_k} = \text{div} F_k + \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} (n_k F_k) \quad (1.21)$$

La sommation est réalisée sur toutes les interfaces j qui passent pendant l'intervalle de temps de moyenne Δt au point x .

Ceci implique :

$$\begin{aligned} \frac{\partial \alpha_k}{\partial t} &= \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} (n_k \cdot u_i) \\ \nabla \alpha_k &= \frac{1}{\Delta t} \sum_j \frac{n_k}{u_{ni}} \end{aligned} \quad (1.22)$$

Moyenne de phase

On définit la moyenne de phase k d'une fonction F_k qu'on note $\bar{\bar{F}}_k$ par :

$$\overline{\overline{F_k}} = \frac{\overline{X_k F_k}}{\overline{X_k}} \quad (1.23)$$

En utilisant les équations (1.16) ÷ (1.18), on a :

$$\overline{\overline{F_k}} = \frac{\overline{F_k}}{\alpha} = \frac{1}{\Delta t_k} \int_{[\Delta t]_k} F_k dt \quad (1.24)$$

Moyenne pondérée par la masse

On définit la moyenne pondérée par la masse $\overline{\overline{\Psi}}$ d'une variable Ψ par :

$$\overline{\overline{\Psi}} = \frac{\overline{\rho \Psi}}{\rho} \quad (1.25)$$

où, ρ est la masse volumique locale instantanée du fluide.

De même pour chaque phase k :

$$\tilde{\Psi}_k = \frac{\overline{\rho_k \Psi_k}}{\rho_k} = \frac{\overline{\overline{\rho_k \Psi_k}}}{\rho_k} \quad (1.26)$$

La fluctuation Ψ'_k est donnée par :

$$\Psi'_k = \Psi_k - \tilde{\Psi}_k \quad (1.27)$$

1.3.2 Equations de conservation

Les équations de conservation (1.1)-(1.3) s'écrivent sous la forme générale suivante :

$$\frac{\partial \rho_k \Psi_k}{\partial t} + \text{div}(\rho_k \Psi_k u_k) + \text{div} \gamma_k - \rho_k \Phi_k = 0 \quad (1.28)$$

Les différentes expressions de Ψ_k sont données dans le tableau 1.4.

Conservation	Ψ_k	γ_k	Φ_k
Masse	1	0	0
Moment	u_k	$\rho_k \Pi - \pi_k$	F_k
Energie	$e_k + \frac{u_k^2}{2}$	$j_k + (\rho_k \Pi - \pi_k) \cdot u_k$	$F_k \cdot u_k$

Tableau 1.4 : Les différentes expressions de Ψ_k

On utilise les décompositions suivantes :

$$\Psi_k = \tilde{\Psi}_k + \Psi'_k \quad \text{et} \quad u_k = \tilde{u}_k + u'_k \quad (1.29)$$

et les propriétés :

$$\overline{\overline{\rho_k \psi_k'}} = 0 \quad \text{et} \quad \overline{\overline{\rho_k u_k'}} = 0 \quad (1.30)$$

Le modèle à deux fluides basé sur les trois lois de conservation pour chaque phase, est obtenu en utilisant les décompositions (1.29) et en appliquant la moyenne temporelle sur l'équation (1.28). En utilisant les équations (1.19) ÷ (1.22) et (1.27) ainsi que les propriétés de la moyenne de phase et la moyenne pondéré par la masse, on obtient :

$$\begin{aligned} \frac{\partial \alpha_k \overline{\overline{\rho_k \tilde{\psi}_k}}}{\partial t} + \text{div} \left(\alpha_k \overline{\overline{\rho_k \tilde{\psi}_k \tilde{u}_k}} \right) + \text{div} \left[\alpha_k \left(\overline{\overline{\gamma_k}} + \gamma_k^t \right) \right] - \alpha_k \overline{\overline{\rho_k \tilde{\Phi}_k}} \\ + \frac{1}{\Delta t} \sum_j \left\{ \frac{1}{u_i \cdot n_j} \left[n_k \cdot \rho_k (u_k - u_i) \psi_k + n_k \cdot \gamma_k \right] \right\} = 0 \end{aligned} \quad (1.31)$$

avec,

$$\gamma_k^t = -\overline{\overline{\rho_k \psi_k' u_k'}} \quad (1.32)$$

Ce qui donne les équations de conservation suivante :

- **Equation de conservation de la masse**

$$\frac{\partial \alpha_k \overline{\overline{\rho_k}}}{\partial t} + \text{div} \left(\alpha_k \overline{\overline{\rho_k \tilde{u}_k}} \right) = \Gamma_k \quad \text{dans } D \quad (1.33)$$

où, Γ_k est la source de masse aux interfaces due au changement de phase donnée par :

$$\Gamma_k = -\frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} (n_k \cdot \rho_k (u_k - u_i)) = -\frac{1}{\Delta t} \sum_j \frac{\dot{m}_k}{u_{ni}} \quad (1.34)$$

avec, \sum_j une sommation sur les interfaces passant par \mathbf{x} à l'instant Δt et \dot{m}_k le taux de perte de masse par la phase k par unité de temps.

- **Equation de conservation de la quantité de mouvement**

$$\begin{aligned} \frac{\partial \alpha_k \overline{\overline{\rho_k \tilde{u}_k}}}{\partial t} + \text{div} \left(\alpha_k \overline{\overline{\rho_k \tilde{u}_k \tilde{u}_k}} \right) = -\nabla \left(\alpha_k \overline{\overline{p_k}} \right) + \text{div} \left[\alpha_k \left(\overline{\overline{\pi_k}} + \pi_k^t \right) \right] + \alpha_k \overline{\overline{\rho_k \tilde{F}_k}} \\ + M_k \quad \text{dans } D \end{aligned} \quad (1.35)$$

où,

π_k^t est le tenseur des contraintes turbulentes (contraintes de Reynolds) donné par :

$$\pi_k^t = -\overline{\rho_k u_k' u_k'} \quad (1.36)$$

M_k est la source de moment de la phase k , due au transfert interfacial, donnée

$$\text{par : } M_k = -\frac{1}{\Delta t} \sum_j \left\{ \frac{1}{u_{ni}} n_k \cdot [\rho_k (u_k - u_i) u_k - p_k \mathbf{II} + \pi_k] \right\} \quad (1.37)$$

- **Equation de conservation de l'énergie**

$$\begin{aligned} \frac{\partial}{\partial t} \left(\alpha_k \overline{\rho_k \left(\tilde{\epsilon}_k + \frac{\tilde{u}_k^2}{2} \right)} \right) + \text{div} \left(\alpha_k \overline{\rho_k \left(\tilde{\epsilon}_k + \frac{\tilde{u}_k^2}{2} \right) u_k} \right) &= \text{div} \left[\alpha_k \overline{\pi_k \cdot \tilde{u}_k} \right] \\ - \text{div} \left(\alpha_k \overline{j_k + j_k^t} \right) - \text{div} \left(\alpha_k \overline{p_k \tilde{u}_k} \right) + \alpha_k \overline{\tilde{F}_k \tilde{u}_k} + E_k &\text{ dans } D \end{aligned} \quad (1.38)$$

✦ $\tilde{\epsilon}_k$ tient compte de l'énergie interne et de l'énergie cinétique turbulente. Elle est donnée par :

$$\tilde{\epsilon}_k = e_k + \frac{\overline{(u_k')^2}}{2} \quad (1.39)$$

✦ $\overline{j_k}$ est le flux moyen de température,

✦ j_k^t est le flux de chaleur turbulent donné par :

$$j_k^t = \overline{\rho_k \left(e_k + \frac{u_k^2}{2} \right) u_k'} - \overline{(\pi_k - p_k \mathbf{II}) u_k'} \quad (1.40)$$

✦ E_k est l'apport d'énergie totale des interfaces à la phase k donné par :

$$E_k = -\frac{1}{\Delta t} \sum_j \left\{ \frac{1}{u_{ni}} n_k \cdot \left[\rho_k (u_k - u_i) \left(e_k + \frac{u_k^2}{2} \right) + (p_k \mathbf{II} - \pi_k) \cdot u_k + j_k \right] \right\} \quad (1.41)$$

1.3.3 Lois constitutives

Ces équations forment un système ouvert, la fermeture se fait par des informations additionnelles.

- **Axiome de continuité**

$$\alpha_G + \alpha_L = 1 \quad (1.42)$$

- **Equation d'état**

$$\tilde{c}_k = \tilde{c}_k \left(\overline{\overline{T}_k}, \overline{\overline{\rho}_k} \right) \quad (1.43)$$

$$\tilde{p}_k = \tilde{p}_k \left(\overline{\overline{T}_k}, \overline{\overline{\rho}_k} \right) \quad (1.44)$$

- **Tenseur des contraintes visqueuses**

$$\boldsymbol{\pi}_k = 2\mu_k (D_{kb} + D_{ki}) \quad (1.45)$$

$$\text{où, } D_{kb} = \frac{1}{2} [\nabla \tilde{u}_k + (\nabla \tilde{u}_k)^T] \quad (1.46)$$

Représente le tenseur de déformation et

$$D_{ki} = \frac{1}{2\alpha_k} \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} \{n_k (u_k - \tilde{u}_k) + (u_k - \tilde{u}_k)n_k\} \quad (1.47)$$

est le tenseur d'extra-déformation interfaciale qui représente l'effet des mouvements de l'interface et le transfert de masse sur la déformation moyenne.

Dans le cas d'un écoulement à bulles, et si le mouvement de l'interface est régulier avec un faible effet de changement de phase, on a (*Ishii*, 1975) :

$$D_{Gi} = 0$$

$$D_{Li} = -\frac{1}{2\alpha_L} \{((\nabla \alpha_L)(\tilde{u}_G - \tilde{u}_L) + (\tilde{u}_G - \tilde{u}_L)(\nabla \alpha_L))\}$$

- **Flux de chaleur**

Le flux de chaleur moyenné pour un fluide obéit à la loi de Fourier et s'écrit :

$$\overline{\overline{j}_k} = -K_k \left\{ \nabla \overline{\overline{T}_k} + \frac{\nabla \alpha_k}{\alpha_k} (\overline{\overline{T}_i} - \overline{\overline{T}_k}) \right\} \quad (1.48)$$

où, $\overline{\overline{T}_i}$ est la température moyenne à l'interface.

- **Termes turbulents**

Pour fermer ce modèle, il est nécessaire d'évaluer ou de modéliser le tenseur des contraintes turbulentes $\boldsymbol{\pi}_k^t$, et le flux de chaleur turbulent j_k^t .

- **Tenseur de contraintes turbulentes $\boldsymbol{\pi}_k^t$**

Plusieurs études se sont intéressées à la modélisation du tenseur de contraintes turbulentes $\boldsymbol{\pi}_k^t$ qui reste l'une des difficultés majeures dans l'utilisation du modèle à deux fluides.

Parmi les approches les plus utilisées, nous pouvons citer le modèle $k-\varepsilon$, où k représente l'énergie cinétique turbulente et ε le taux de dissipation d'énergie

turbulente. Dans ce modèle, les tensions de Reynolds sont obtenues à l'aide des lois de comportement.

Dans le cas d'écoulements diphasiques compressibles, ce modèle devient plus complexe et tient compte du transfert de moment turbulent interfacial (**S. E. Elghobashi, T. W. Abou-Arab**, 1983).

Chahed et al. (**J. Chahed et al.**, 1992) ont développé ce modèle dans le cas d'un écoulement diphasique stationnaire incompressible sans transfert de masse. Ce modèle est basé sur une fermeture de second ordre du tenseur de contraintes turbulentes. Leur technique consiste à partager ce dernier en deux parties : une partie turbulente produite par le gradient de la vitesse et une partie pseudo-turbulente induite par le déplacement des bulles.

Boisson et al. (**N. Boisson et al.**, 1996) ont supposé que la turbulence est une propriété de l'écoulement la phase continue (liquide), donc, la viscosité cinématique turbulente de chaque phase est calculée en multipliant μ^t fonction de k et ε par la masse volumique phasique. k et ε vérifient le système d'équations suivant :

$$\begin{aligned} \frac{\partial(\alpha_L \overline{\rho_L k})}{\partial t} + \operatorname{div}(\alpha_L \overline{\rho_L \tilde{u}_L k}) &= \operatorname{div} \left[\overline{\rho_L} \left(\mathbf{v}_L + \frac{C_\mu k^2}{\sigma_k \varepsilon} \right) \nabla k \right] + \overline{\rho_L} \alpha_L (P_k - \varepsilon) \\ &+ \operatorname{div} \left[\overline{\rho_L} \frac{C_\mu k^2}{\sigma_\alpha \varepsilon} k \nabla \alpha_L \right] \alpha P_b \end{aligned} \quad (1.49)$$

$$\begin{aligned} \frac{\partial(\alpha_L \overline{\rho_L \varepsilon})}{\partial t} + \operatorname{div}(\alpha_L \overline{\rho_L \tilde{u}_L \varepsilon}) &= \operatorname{div} \left[\overline{\rho_L} \left(\mathbf{v}_L + \frac{C_\mu k^2}{\sigma_k \varepsilon} \right) \nabla \varepsilon \right] + \alpha_L c_{1s} P_b \frac{\varepsilon}{k} \\ &+ \left(\alpha_L \overline{\rho_L} \frac{\varepsilon}{k} \right) (C_{1\varepsilon} P_1 - C_{2\varepsilon} \varepsilon) + \operatorname{div} \left[\overline{\rho_L} \frac{C_\mu k^2}{\sigma_\alpha \varepsilon} \varepsilon \nabla \alpha_L \right] \end{aligned}$$

où, P_k est le taux de production volumétrique de k par les forces de cisaillement

$$P_k = \frac{\mu^t}{\rho_k} \left[(\nabla \tilde{u}_L + (\nabla \tilde{u}_L)^T) \right] : \nabla \tilde{u}_L \quad (1.50)$$

et P_b est le taux de production de k dû au travail de traînée que la bulle fait lorsqu'elle se déplace dans le liquide :

$$P_b = \frac{3}{8R} C_b C_D \alpha_G \rho_L |\tilde{u}_G - \tilde{u}_L| \tilde{u}_L \tilde{u}_L \quad (1.51)$$

avec, C_b une constante empirique.

$$* \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.314, \quad C_\mu = 0.09, \quad C_{1\varepsilon} = 1.44 \quad \text{et} \quad C_{2\varepsilon} = 1.92$$

Un autre modèle a été présenté par Drew dans le cas d'un écoulement diphasique eau-bulles d'air et s'écrit pour la phase liquide (**D. A. Drew**, 1983):

$$\pi_L^t = 2\mu_L^t D_{Lb} + a_2 \mathbf{II} + b_2 (\tilde{u}_L - \tilde{u}_G) (\tilde{u}_L - \tilde{u}_G) \quad (1.52)$$

où, μ_L^t est la viscosité turbulente de la phase eau, a_2 est une pression tourbillonnaire induite et b_2 est un coefficient associé au passage de bulles. Ces termes dépendent de α_G , du rayon moyen des bulles, de la vitesse relative $|\tilde{u}_L - \tilde{u}_G|$ et de la norme euclidienne de D_{Lb} .

En raison de l'incertitude entourant la modélisation du tenseur de contrainte turbulentes en écoulement diphasique, Davidson (**M. R. Davidson**, 1990) et Schwarz (**M. P. Schwarz**, 1996) ont présenté un modèle à zéro équation qui consiste à exprimer le tenseur des contraintes turbulentes en fonction d'une viscosité turbulente ; μ_k^t étant exprimée en fonction de la viscosité cinématique μ_k . Cette expression est basée sur des études phénoménologiques.

– **Flux de chaleur turbulent** j_k^t

Le flux de chaleur turbulent est défini par l'équation (1.40) et peut être partagé en trois parties ; les transferts turbulents de l'énergie interne, de l'énergie cinétique et le travail donné par la turbulence. Pour plusieurs cas d'écoulement diphasique, les deux derniers effets ont un rôle inférieur au premier, comme en écoulement monophasique.

Dans le cas où on ne tient compte que de l'effet du transport d'énergie thermique, c'est-à-dire le premier et le dernier terme de l'équation (1.40), on a l'expression suivante (**M. Ishii**, 1975) :

$$j_k^t = -K_k^t \left\{ \nabla T_k - \frac{\nabla \alpha_k}{\alpha_k} (T_i - T_k) \right\} \quad (1.53)$$

où, K_1^t est un coefficient de transport d'énergie turbulente difficile à exprimer. Ainsi j_k^t est souvent négligé dans les analyses des écoulements diphasiques (**D. A. Drew et al.**, 1988).

– **Relation entre les pressions**

Nous présentons des relations utilisées dans le cas d'un écoulement diphasique à bulle.

Dans le cas où on s'intéresse aux contractions/expressions des bulles, on a les relations suivantes:

$$\overline{p_{Gi}} - \overline{p_G} = 0 \quad (1.54)$$

$$\overline{p_{Li}} - \overline{p_L} = -\xi \overline{\rho_L} \left| \overline{u_G} - \overline{u_L} \right|^2 \quad (1.55)$$

$$\overline{p_{Gi}} - \overline{p_{Li}} = 2\overline{H_{LG}} \overline{\sigma} \quad (1.56)$$

où, ξ est un coefficient de pression empirique donné par : $\xi = \frac{1}{4}(1 - \alpha_G)$ et $\overline{H_{LG}}$ est la courbure moyenne des interfaces.

Dans la pratique, on suppose souvent que la pression est égale à la pression à l'interface en négligeant les effets des tensions superficielles σ .

$$\overline{p_k} = \overline{p_{ki}} \quad (1.57)$$

1.3.4 Conditions de transfert interfacial

- **Conservation de la masse à l'interface**

$$\sum_{k=L,G} \Gamma_k = 0 \quad \text{dans } F_i \quad (1.58)$$

où, Γ_k est défini par l'équation (1.34).

- **Conservation de la quantité de mouvement à l'interface**

$$\sum_{k=L,G} M_k - M_m = 0 \quad \text{dans } F_i \quad (1.59)$$

Pour définir M_m , on introduit les moyennes de surface par analogie avec la moyenne de phase (1.23) et la moyenne pondérée par la masse (1.24) :

$$\overline{F_{ki}^i} = \frac{\frac{1}{\Delta t} \sum_j \frac{F_{ki}}{u_{ni}}}{\frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}}}, \quad \tilde{\Psi}_{ki}^i = \frac{\frac{1}{\Delta t} \sum_j \frac{\dot{m}_k \Psi_k}{u_{ni}}}{\frac{1}{\Delta t} \sum_j \frac{\dot{m}_k}{u_{ni}}} \quad (1.60)$$

ainsi que la pression phasique à l'interface p_{ki} et la masse volumique phasique à l'interface ρ_{ki} .

En supposant que $\overline{\dot{m}_k^i} = \dot{m}_k$, $\overline{\sigma} = \sigma$ et $\overline{\rho_{ki}^i} = \rho_k$, on obtient :

$$M_k = M_k^\Gamma + \overline{\rho_{ki}^i} \nabla \alpha_k + M_k^n + M_k^t \quad (1.61)$$

où,

$$\begin{aligned}
M_k^\Gamma &= \Gamma_k \tilde{u}_{ki}^i \\
M_n^n &= \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} \left[(p_{ki}^i - p_k) n_k + \pi_{nk} \right] \\
M_k^t &= \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} \pi_{tk}
\end{aligned} \tag{1.62}$$

avec,

$$\pi_k \cdot n_k = \pi_{nk} + \pi_{tk}$$

Le premier terme dans l'équation (1.61) représente une source de moment de la phase k due au changement de phase, les trois derniers termes représentent une force moyenne interfaciale due aux forces de contrainte sur l'interface.

On pose souvent :

$$L_k^d = M_k^n + M_k^t \tag{1.63}$$

qui est traité comme une densité de forces interfaciales représentée par la force de traînée, de masse ajoutée, de portance, ainsi que d'autres forces (négligées souvent).

La source de moment du mélange s'écrit :

$$M_m = \overline{\overline{2H_{LG} \sigma \nabla \alpha_G}} + M_m^H \tag{1.64}$$

où, M_m^H est l'effet de changement de la courbure moyenne.

- **Conservation de l'énergie à l'interface**

$$\sum_{k=L,G} E_k - E_m = 0 \quad \text{dans } F_i \tag{1.65}$$

1.3.5 Approches du modèle à deux fluides

Nous avons présenté la formulation la plus générale du modèle à deux fluides (équations (1.33) ÷ (1.53)).

Nous présentons, par la suite, un modèle basé sur des simplifications physiques utilisées dans la majorité des travaux rencontrés. On suppose que :

- L'énergie cinétique turbulente ainsi que les termes qui y contribuent sont négligeables.
- L'effet de changement de courbure moyenne M_m^H est négligeable devant M_k^d .
- Pas de transfert de masse à l'interface $\Gamma_k = 0$.
- Le flux de chaleur turbulente est négligeable.
- L'effet des tensions superficielles est négligeable, ce qui donne $\overline{\overline{p_k}} = \overline{\overline{p_{ki}}}$.

- La phase dispersée est en équilibre thermique.
- La source d'énergie interfaciale E_m est négligeable.

Sous ses hypothèses, les équations de conservation (1.33) et (1.35) se ramènent à :

• **Equation de conservation de la masse**

$$\frac{\partial \alpha_k \bar{\rho}_k}{\partial t} + \text{div}(\alpha_k \bar{\rho}_k \tilde{u}_k) = 0 \quad \text{dans } D \quad (1.66)$$

• **Equations de conservation de la quantité de mouvement**

$$\frac{\partial \alpha_k \bar{\rho}_k \tilde{u}_k}{\partial t} + \text{div}(\alpha_k \bar{\rho}_k \tilde{u}_k \tilde{u}_k) = -\alpha_k \nabla(\bar{p}_k) + \text{div}[\alpha_k (\bar{\pi}_k + \pi_k^t)] + \alpha_k \bar{\rho}_k \tilde{F}_k + M_k^d \quad \text{dans } D \quad (1.67)$$

• **Equation de conservation de l'énergie**

L'équation (1.38) s'écrit en fonction de l'enthalpie sous la forme suivante :

$$\frac{\partial (\alpha_k \bar{\rho}_k \tilde{h})}{\partial t} + \text{div}(\alpha_k \bar{\rho}_k \tilde{h} \tilde{u}_k) = -\text{div}(\alpha_k \dot{j}_k) + \alpha_k \left(\frac{\partial \bar{p}_k}{\partial t} + \tilde{u}_k \cdot \nabla \bar{p}_k \right) + \alpha_k \bar{\pi}_k : \nabla \tilde{u}_k \quad \text{dans } D \quad (1.68)$$

où,

$$\alpha_k \bar{\pi}_k : \nabla \tilde{u}_k = \text{div}(\alpha_k \bar{\pi}_k \cdot \tilde{u}_k) - \tilde{u}_k \cdot \text{div}(\alpha_k \bar{\pi}_k) \quad \text{dans } D \quad (1.69)$$

En utilisant l'expression de l'enthalpie en fonction de la température :

$$\begin{aligned} dh_k &= c_{pk} dT_k + \frac{1}{\bar{\rho}_k} \left\{ 1 + \frac{T_k}{\bar{\rho}_k} \frac{\partial \bar{\rho}_k}{\partial T_k} \right\}_{p_k} \frac{d_k \bar{p}_k}{dt} d\bar{p}_k \\ &= c_{pk} dT_k + \frac{1}{\bar{\rho}_k} \left\{ 1 - \frac{d_k \bar{p}_k}{dt} \right\} d\bar{p}_k \end{aligned} \quad (1.70)$$

pour un fluide parfait

où, $c_{pk} = \frac{\partial \tilde{h}_k}{\partial T_k} \Big|_{p_k} = \frac{\partial \tilde{s}_k}{\partial T_k} \Big|_{p_k}$ est la chaleur spécifique à pression constante, on se

ramène à l'équation suivante :

$$c_{pk} \left(\frac{\partial (\overline{\alpha_k \rho_k T_k})}{\partial t} + \text{div} (\overline{\alpha_k \rho_k T_k \tilde{u}_k}) \right) = -\text{div} (\overline{\alpha_k j_k}) + \alpha_k \left(\frac{\partial \overline{p_k}}{\partial t} + \tilde{u}_k \cdot \nabla \overline{p_k} \right) + \alpha_k \overline{\pi_k} : \nabla \tilde{u}_k \quad (1.71)$$

- ***Condition de transfert de quantité de mouvement***

La modélisation du terme de transfert interfacial de quantité de mouvement est un point très important puisque ce dernier couple l'ensemble des deux phases. Il s'écrit :

$$\sum_k \left\{ \overline{p_{ki}} \cdot \nabla \alpha_k + M_k^d \right\} = 2H_{LG} \overline{\sigma} \nabla \alpha_G \quad (1.72)$$

En utilisant les équations (1.42) et (1.56), on obtient

$$M_L^d + M_G^d = 0 \quad (1.73)$$

Le terme de transfert de moment interfacial M_G^d contient les forces exercées sur les bulles ainsi que la distribution non équilibrée de la pression menant aux effets de masse ajoutée et de portance. Ce terme s'écrit sous la forme suivante :

$$M_G^d = M_{\tilde{u}} + M_{ma} + M_l \quad (1.74)$$

où,

- $M_{\tilde{u}}$ représente la force de traînée interfaciale donnée par :

$$M_{\tilde{u}} = \frac{3}{8R} C_D \alpha_G \rho_L |\tilde{u}_G - \tilde{u}_L| (\tilde{u}_L - \tilde{u}_G) \quad (1.75)$$

avec, C_D : le coefficient de traînée adimensionné et R est le rayon moyen des bulles.

- M_{ma} représente la force de masse ajoutée donnée par (**N. Boisson et al.**, 1996 ; **H. Enwald et al.**, 1996)

$$M_{ma} = C_m \alpha_G \rho_L \left\{ \left(\frac{\partial \tilde{u}_L}{\partial t} + \tilde{u}_L \cdot \nabla \tilde{u}_L \right) - \left(\frac{\partial \tilde{u}_G}{\partial t} + \tilde{u}_G \cdot \nabla \tilde{u}_G \right) \right\} \quad (1.76)$$

où, C_m est le coefficient de masse ajoutée. Il décrit le volume de fluide déplacé qui contribue à la masse effective de la phase dispersée. En général, ce paramètre dépend de α_G mais, il est toujours pris comme constant. A titre

d'exemple, il prend la valeur $\frac{1}{2}$ pour une bulle sphérique. Une autre forme a été utilisée par Boisson et al. (**N. Boisson et al.**, 1996) :

$$C_m = \frac{1}{2}(1 - 2.78 \text{Min}(0.2, \alpha_G))$$

- M_1 représente la force de portance ou de lift interfacial. La modélisation de cette force a été traitée par plusieurs travaux (*N. Boisson et al.*, 1996 ; *H. Enwald et al.*, 1996). Elle s'écrit :

$$M_1 = \alpha_G \rho_L C_1 (\tilde{u}_L - \tilde{u}_G) \times \nabla \times \tilde{u}_L \quad (1.77)$$

où C_1 est le coefficient de portance interfacial souvent supposé égale à C_m .

- **Condition de transfert de l'énergie**

L'équation de transfert d'énergie à l'interface s'écrit :

$$\sum_{k=L,G} E_k - E_m = 0 \quad (1.78)$$

où,

$$E_k = \Gamma_k \left(\tilde{h}_{ki} + \frac{\tilde{u}_k^2}{2} \right) + \frac{j_k''}{L_s} - p_{ki} \frac{\partial \alpha_k}{\partial t} + M_k^d \cdot \tilde{u}_k \quad (1.79)$$

avec,

✦ \tilde{h}_{ki} est l'enthalpie.

✦ $j_k'' = - \left\{ \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}} n_k \cdot j_k \right\} L_s$ représente le flux de chaleur superficielle.

✦ $\frac{1}{L_s} = \frac{1}{\Delta t} \sum_j \frac{1}{u_{ni}}$ est la concentration surfacique (aire de contact entre les deux phases).

Dans le cas d'un écoulement à vitesse relativement faible, on obtient :

$$\sum_k \left\{ \Gamma_k i_{li} + \frac{j_k''}{L_s} \right\} = 0 \quad (1.80)$$

1.3.6 Bilan des variables et des équations

Dans le tableau 1.5 sont données les variables principales.

Inconnues	Nombre	Equations	Nombre
α_k	2	Axiome de continuité (1.42)	1
$\overline{\overline{p}}_k$	2	Masse (1.66) + Relation pression (1.56)	3
\tilde{u}_k	6	Mouvement (1.67)	6
$\overline{\rho}_k$	2	Loi d'état (1.44)	2
$\overline{\overline{T}}_k$	2	Energie (1.71)	2
Total	14	Total	14

Tableau 1.5 : Variables principales

Dans le tableau 1.6 sont présentées les variables secondaires.

Inconnue	Nombre	Equations	Nombre
s			
$\overline{\overline{\pi}}_k$	9	Constitutives mécaniques (1.45)	9
π_k^t	9	Modèle de fermeture	9
\dot{j}_k	3	Constitutives énergétiques (1.48)	3
j_k^t	3	Modèle de fermeture (1.53)	3
M_k^d	3	Constitutives (1.73)	3
Total	27	Total	27

Tableau 1.6 : Variables secondaires

A ces équations, s'ajoutent les conditions initiales et aux limites correspondantes.

1.3.7 Commentaires sur le modèle à deux fluides

Le modèle à deux fluides est largement utilisé dans les applications industrielles modulo des simplifications le rendant plus abordable. Les avantages de cette formulation résident dans le fait que les études et les algorithmes de résolution faits sur les équations monophasiques sont transportables aux cas diphasiques, mais des problèmes se posent dans la modélisation des termes turbulents (tenseur de contrainte turbulente π_k^t , du flux de chaleur turbulent j_k^t) ainsi que les termes de transfert interfacial écrits sur chaque interface.

1.4 Modèle de mélange

Le modèle de mélange dérive du modèle à deux fluides. Le concept de base de ce modèle est de considérer le mélange comme étant un fluide équivalent et non un mélange de deux phases séparées. Par conséquent, le système d'équations de lois de conservation de masse, de quantité de mouvement et d'énergie, s'écrit avec des variables qui caractérisent le mélange. On y ajoute une équation de diffusion qui tient compte de l'échange de concentration.

1.4.1 Variables du modèle

Les variables du mélange sont définies à partir des variables du modèle à deux fluides (tableau 1.7 et 1.8).

Symboles	Expressions	Variables	Types
ρ_m	$\sum_{k=L,G} \alpha_k \overline{\rho}_k$	Masse volumique	Fonction scalaire
u_m	$\frac{1}{\rho_m} \sum_{k=L,G} \alpha_k \rho_k \tilde{u}_k$	Vitesse	Fonction vectorielle
p_m	$\sum_{k=L,G} \alpha_k \overline{p}_k$	Pression	Fonction scalaire
e_m	$\frac{1}{\rho_m} \sum_{k=L,G} \alpha_k \rho_k \overline{e}_k$	Energie interne	Fonction scalaire

Tableau 1.7 : Variables principales du modèle de mélange

Symbole	Expression	Variable	Type
π_m	$\sum_{k=L,G} \alpha_k \overline{\pi}_k$	Contraintes visqueuses	Fonction tensorielle
j_m	$\sum_{k=L,G} \alpha_k \dot{j}_k$	Flux de température	Fonction vectorielle
π_m^t	$\sum_{k=L,G} \alpha_k \pi_k^t$	Contraintes turbulentes	Fonction tensorielle
j_m^t	$\sum_{k=L,G} \alpha_k j_k^t$	Flux de température turbulent	Fonction vectorielle
F_m	$\frac{1}{\rho_m} \sum_{k=L,G} \alpha_k \rho_k \overline{F}_k$	Efforts extérieurs	Fonction vectorielle

Tableau 1.8 : Variables secondaires du modèle de mélange

1.4.2 Equations de conservation

Les équations décrivant l'écoulement du mélange sont données par :

- **Equation de conservation de la masse**

$$\frac{\partial \rho_m}{\partial t} + \text{div}(\rho_m u_m) = 0 \quad (1.81)$$

- **Equation de diffusion de la concentration**

$$\frac{\partial \alpha_k \bar{\rho}_k}{\partial t} + \text{div}(\alpha_k \bar{\rho}_k u_m) = \Gamma_k - \text{div}(\alpha_k \bar{\rho}_k (\tilde{u}_k - u_m)) \quad k = L \text{ ou } G \quad (1.82)$$

- **Equation de conservation de la quantité de mouvement**

$$\frac{\partial \rho_m u_m}{\partial t} + \text{div}(\rho_m u_m u_m) = -\text{div} p_m + \text{div}(\pi_m + \pi_m^t + \pi_m^D) + \rho_m F_m + M_m \quad (1.83)$$

où, π_m^D est le tenseur de contrainte de diffusion de mélange défini par :

$$\pi_m^D = - \sum_{k=L,G} \alpha_k \bar{\rho}_k (\tilde{u}_k - u_m)(\tilde{u}_k - u_m) \quad (1.84)$$

avec, M_m donnée par l'équation (1.64).

- **Equation de conservation d'énergie**

$$\begin{aligned} \frac{\partial}{\partial t} \left(\rho_m \left[\varepsilon_m + \left(\frac{u^2}{2} \right)_m \right] \right) + \text{div} \left(\rho_m \left[\varepsilon_m + \left(\frac{u^2}{2} \right)_m \right] u_m \right) &= -\text{div} (j_m + j_m^t + j_m^D) \\ &- \text{div}(\rho_m u_m) + \text{div}(\pi_m \cdot u_m) + \rho_m F_m \cdot u_m \\ &+ \sum_{k=L,G} \alpha_k \bar{\rho}_k F_k \cdot (\tilde{u}_k - u_m) + E_m \end{aligned} \quad (1.84)$$

Où, j_m^D est le flux de chaleur diffusif donné par :

$$j_m^D = \sum_{k=L,G} \alpha_k \left\{ \bar{\rho}_k \left(\bar{\varepsilon}_k + \frac{\tilde{u}_k^2}{2} \right) (\tilde{u}_k - u_m) - (\pi_k - p_k \Pi) (\tilde{u}_k - u_m) \right\} \quad (1.85)$$

$$\varepsilon_m = \frac{1}{\rho_m} \sum_{k=L,G} \alpha_k \bar{\rho}_k \varepsilon_k \quad (1.86)$$

avec, $\varepsilon_k = \tilde{\varepsilon}_k + \frac{(\tilde{u}_k')^2}{2}$ représente l'énergie interne et l'énergie cinétique turbulente

$$\text{et } \left(\frac{u^2}{2} \right)_m = \frac{1}{\rho_m} \sum_{k=L,G} \alpha_k \rho_k \frac{\tilde{u}_k^2}{2} \quad (1.87)$$

1.4.3 Lois constitutives

Pour compléter le modèle de mélange, il est nécessaire d'ajouter les lois constitutives suivantes :

- Axiome de continuité (1.42),
- Lois d'état sur l'énergie interne du mélange e_m et la pression du mélange p_m en utilisant (1.43) et (1.44),
- Equations constitutives mécaniques sur le tenseur de contraintes visqueuses du mélange π_m en utilisant (1.45) et le tenseur de contraintes turbulentes π_m^t en utilisant un des modèles de fermeture,
- Equations constitutives énergétiques sur le flux de chaleur j_m et le flux de chaleur turbulent j_m^t en utilisant (1.48) et (1.53),
- Equation constitutive pour la source de moment du mélange (1.64),
- Condition mécanique à l'interface (1.56) exprimant la relation entre les pressions des deux phases,
- Identité sur les vitesses de diffusion.

$$\sum_{k=L,G} \alpha_k \rho_k (\tilde{u}_k - u_m) = 0 \quad (1.88)$$

1.4.4 Commentaires sur le modèle de mélange

Le modèle de mélange présente moins d'équations et de variables que le modèle à deux fluides, mais il pose les mêmes difficultés au niveau du traitement des termes turbulents.

1.5 Approche lagrangienne

Cette approche propose une modélisation eulérienne de la phase continue (eau) et une modélisation lagrangienne de la phase dispersée (bulle). Pour tenir compte de l'influence des bulles sur la phase continue, des termes sources sont inclus dans l'équation de mouvement de celle-ci. Le traitement du mouvement des bulles par voie lagrangienne demande un suivi individuel de chaque bulle dans son mouvement, pour cela il faudra intégrer simultanément toutes les équations de la

trajectoire des bulles. Ces dernières se déplacent sous l'action des forces extérieures telles que la traînée, la flottabilité et les forces d'inertie. Pour déterminer la trajectoire d'une bulle dans l'écoulement, une intégration numérique est faite pour la relation fondamentale de la dynamique qui s'écrit :

$$\begin{cases} m_B \frac{du_G}{dt} = F \\ \frac{dx_G}{dt} = u_G \end{cases} \quad (1.89)$$

où, m_B est la masse de la bulle et F est la résultante des forces appliquées sur une bulle.

Cette approche a été utilisée par Hunt et *al.* (**J. C. R. Hunt et al.**, 1994) dans un écoulement à bulles pour plusieurs cas d'écoulement : non visqueux, visqueux et fluide stationnaire.

Sommerfeld et *al.* (**M. Sommerfeld and al.**, 1997) ont négligé les forces de masse ajoutée et de pression dans des simulations à deux dimensions d'écoulement dans une colonne à bulle. Dans cette étude, une extension de la méthode Euléro-Lagrangienne a été faite en résolvant une équation de transport ajoutée pour l'énergie cinétique turbulente et le taux de dissipation associé. L'effet des bulles est donné par des termes sources ajoutés dans l'équation de mouvement de la phase continue.

Clement (**E. Clement**, 1997) s'est intéressé au mouvement d'un nuage de bulles dans un liquide, il a présenté une étude complète sur les forces exercées sur une bulle ainsi que l'expression des termes sources ajoutés à l'équation de conservation de la quantité de mouvement du liquide.

1.5.1 Forces exercées sur une bulle

Les différentes forces qui s'exercent sur une bulle dépendent de beaucoup de paramètres telles que la taille de la bulle, sa forme, sa masse volumique, sa vitesse ainsi que sa viscosité et la vitesse du fluide qui l'entoure. Quelques-uns de ces facteurs dépendent aussi des propriétés physiques comme le coefficient de tension surfacique. La dérivation d'une expression générale des forces est donc compliquée. L'approche adoptée consiste à développer des expressions pour certains cas limites.

Sous les hypothèses suivantes :

- bulle sphérique indéformable de rayon R ,
- taille petite devant toutes les échelles de l'écoulement (pour permettre de négliger une force due à la non uniformité de l'écoulement à l'échelle de la bulle),
- absence d'interactions directes entre les bulles (pour permettre la résolution des équations de mouvement de chaque bulle de manière indépendante).

Les différentes forces qui s'exercent sur une bulle sont :

- ***Force de flottabilité*** (Archimède)

Sous l'effet de la pesanteur, une bulle en mouvement dans un fluide subit une force verticale, due à la différence de masse volumique, dirigée vers le haut si $\rho_G < \rho_L$ et vers le bas si $\rho_G > \rho_L$. C'est la force de flottabilité qui est généralement responsable de la présence d'une vitesse de glissement entre la particule et le fluide :

$$F_f = g(\rho_G - \rho_L)V_G \quad (1.90)$$

où, V_G est le volume de la bulle.

- ***Force de traînée*** (Drag)

Elle est due aux contraintes visqueuses que le fluide exerce sur la surface de la bulle. Elle prend la forme suivante pour une bulle sphérique de rayon R :

$$F_t = -C_D \frac{\pi R^2}{2} \rho_L |u_G - u_L| (u_G - u_L) \quad (1.91)$$

Le coefficient de traînée C_D dépend du nombre de Reynolds.

- ***Force de masse ajoutée*** (Added mass)

La force de masse ajoutée s'interprète physiquement comme étant la force qu'il faut exercer sur un fluide pour déplacer en mouvement accéléré un corps dans ce fluide ; en effet, le corps doit communiquer à un certain volume de fluide son accélération pour avancer et prendre sa place.

$$F_{ma} = \rho_L V_G C_m \left(\left(\frac{\partial u_L}{\partial t} + u_L \cdot \nabla u_L \right) - \left(\frac{\partial u_G}{\partial t} + u_G \cdot \nabla u_G \right) \right) \quad (1.92)$$

où, $C_m = \frac{1}{2}$ pour une bulle sphérique.

Lorsque le fluide est en mouvement accéléré, il convient de rajouter à l'équation (1.92) un terme $\rho_L V_G \frac{du_L}{dt}$ qui représente la force que le fluide extérieur exerce sur le volume de fluide correspondant à celui occupé par la bulle.

- ***Force de portance***

Elle désigne la force qui s'exerce sur la bulle perpendiculairement à sa vitesse de glissement

$$F_l = -C_l \rho_L V_G (u_G - u_L) \times \nabla \times u_L \quad (1.93)$$

où, C_l dépend du nombre de Reynolds.

1.5.2 Formulation du terme source

Pour la formulation du terme source, Clement (*E. Clement*, 1997) a montré que sous certaines hypothèses, l'équation de masse n'est pas modifiée par la présence des bulles. Pour l'équation de quantité de mouvement de fluide, on va effectuer un bilan de quantité de mouvement sur un volume de fluide contenant des bulles. Pour simplifier la démonstration qui suit, on va supposer qu'une seule bulle est présente à l'intérieur du volume de contrôle (la bulle délimitée par Σ_G , sur la figure 1.3 étant considérée comme extérieur à ce volume de contrôle).

La relation fondamentale de la dynamique intégrée sur le volume de fluide contenu dans le volume de contrôle s'écrit :

$$\int_{V_L} \rho_L \frac{du_L}{dt} d\Omega = \int_{V_L} \rho_L g d\Omega + \int_{\Sigma_e} \Sigma \cdot n d\Gamma + \int_{\Sigma_G} \Sigma \cdot n d\Gamma \quad (1.94)$$

où ρ_L et Σ sont respectivement la masse volumique et le tenseur de contraintes du fluide.

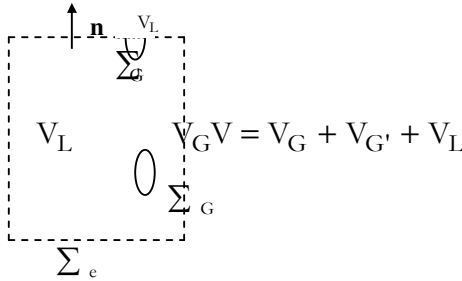


Figure 1.3 :Volume de contrôle

Le dernier terme de l'équation (1.94) représente l'action de la bulle sur le fluide qui l'entoure. Cette action est opposée à l'action du fluide sur la bulle. Or l'équation de la dynamique de celle-ci peut s'écrire

$$m_G \frac{du_G}{dt} = m_G g - \int_{\Sigma_G} \Sigma \cdot n d\Gamma \quad (1.95)$$

Comme, \mathbf{n} est dirigé vers l'intérieur de la bulle. On a donc :

$$\int_{\Sigma_G} \Sigma \cdot n d\Gamma = m_G g - m_G \frac{du_G}{dt} \quad (1.96)$$

où, $m_G = \frac{4}{3} \pi \rho_G R^3$ est la masse de la bulle et u_G sa vitesse.

Une partie du volume de contrôle V est occupée par la bulle. On a donc :

$$\int_{V_L} \rho_L g d\Omega = \int_V \rho_L g d\Omega - m_L g \quad (1.97)$$

où, $m_L = \frac{4}{3}\pi\rho_L R^3$ est la masse de fluide que contiendrait le volume occupé par la bulle.

De même, en supposant l'accélération du fluide uniforme sur le volume V , on a :

$$\int_{V_L} \rho_L \frac{du_L}{dt} d\Omega = \int_V \rho_L \frac{du_L}{dt} d\Omega - m_L \frac{du_L}{dt} \quad (1.98)$$

On peut enfin écrire le bilan de quantité de mouvement sur le volume V complet sous la forme :

$$\int_{V_L} \rho_L \frac{du_L}{dt} d\Omega = \int_V \rho_L g d\Omega + \int_{\Sigma_e} \Sigma_e n d\Gamma + m_L \left(\frac{du_L}{dt} - g \right) + m_G \left(g - \frac{du_G}{dt} \right) \quad (1.99)$$

A ce niveau, on voit que l'équation de quantité de mouvement peut être écrite sur des volumes arbitraires V à condition d'introduire deux termes complémentaires. Le dernier terme de l'équation (1.99) constitue l'échange de quantité de mouvement entre la bulle et le fluide qui l'entoure. Quant à l'avant dernier terme, il représente la diminution d'inertie du fluide causé par le « trou » que constitue le volume occupé par la bulle. Celle-ci est donc vue comme un point-source de quantité de mouvement. On peut écrire ces deux termes d'une manière différente en utilisant l'expression des forces hydrodynamiques :

$$\begin{aligned} \rho_G V_G \frac{du_G}{dt} &= (\rho_G - \rho_L) V_G g - C_D \frac{\pi R^2}{2} \rho_L |u_G - u_L| (u_G - u_L) \\ &+ \rho_L V_G C_m \left(\frac{du_L}{dt} - \frac{du_G}{dt} \right) + \rho_L V_G \frac{du_L}{dt} - C_I \rho_L V_G (u_G - u_L) \times \nabla \times u_L \end{aligned} \quad (1.100)$$

ce qui donne :

$$\begin{aligned} m_L \left(\frac{du_L}{dt} - g \right) + m_G \left(g - \frac{du_G}{dt} \right) &= - \left[-C_D \frac{\pi R^2}{2} \rho_L |u_G - u_L| (u_G - u_L) \right. \\ &\left. + \rho_L V_G C_m \left(\frac{du_L}{dt} - \frac{du_G}{dt} \right) - C_I \rho_L V_G (u_G - u_L) \times \nabla \times u_L \right] \end{aligned} \quad (1.101)$$

Si on souhaite écrire à partir de (1.99) des équations locales, on peut considérer que les différentes grandeurs sont uniformes à l'intérieur du volume V . On obtient ainsi les équations de Navier-Stokes avec un terme supplémentaire d'échange de quantité de mouvement t_s .

$$t_s = \lim_{V \rightarrow 0} \frac{V_G}{V} \left[\rho_L \left(\frac{du_L}{dt} - g \right) + \rho_G \left(g - \frac{du_G}{dt} \right) \right] \quad (1.102)$$

Dans le cas où plusieurs bulles sont présentes dans le volume V , le terme t_s devient évidemment la somme de leurs contributions respectives. On a les équations suivantes :

$$\begin{aligned} \rho_L \frac{du_L}{dt} - \text{div} \left\{ \mu_{\text{eff}} \left(\nabla u_L + \nabla u_L^T \right) \right\} + \nabla p_L &= \rho_L g + \Phi_s \\ \frac{\partial \rho_L}{\partial t} + \text{div}(\rho_L u_L) &= 0 \end{aligned} \quad (1.103)$$

Où,

- ✦ u_L est la vitesse non perturbée par la présence de bulles voisines (**E. Clement**, 1997).
- ✦ $\mu_{\text{eff}} = \mu_{\text{eff}}(x)$ dépend de la position de la maille par rapport aux bulles voisines.
- ✦ Φ_s est la densité volumique des actions de la phase dispersée sur la phase continue donnée par :

$$\Phi_s = \lim_{V \rightarrow 0} \frac{V_G}{V} \sum_{i=1}^{N_G} \left[\rho_L \left(\frac{du_L}{dt} - g \right) + \rho_G \left(g - \frac{du_G}{dt} \right) \right] \quad (1.104)$$

où, $N_G \frac{V_G}{V}$ représente le taux de présence local des bulles (N_G étant le nombre de bulles présentes dans le volume considéré).

Un autre terme peut être ajouté représentant la contribution locale des petites échelles associées à l'écoulement potentiel autour de chacune des bulles (**E. Clement**, 1997).

1.5.3 Commentaires sur l'approche lagrangienne

Les problèmes liés à ce modèle sont relatifs aux aspects numériques. En effet, les algorithmes utilisés pour la phase continue sont très différents du suivi lagrangien des bulles, ce qui nécessite un couplage de deux codes d'architectures différentes. De plus le suivi des trajectoires des bulles une à une nécessite des temps de calcul très importants et reste limité à des petits domaines avec un nombre limité de bulles.

1.6 Modèles utilisés

Nous avons présenté les différents types de modèles utilisés en écoulements diphasiques :

- le modèle à deux fluides dont les difficultés résident dans le nombre d'équations et les variables à traiter ainsi que la modélisation de la turbulence,

- le modèle de mélange qui permet certes de diminuer le nombre d'équations et de variables mais conserve toujours des difficultés similaires à celles du modèle à deux fluides,
- le modèle Lagrangien qui reste limité au cas d'un petit domaine et pour un nombre faible de bulles.

Dans notre problème, d'énormes challenges sont posés dans l'utilisation des modèles d'écoulement présentés, autant sur le plan physique que numérique. En effet, nous sommes face à des systèmes à grand nombre d'équations et de variables dans des cadres multi-échelles imposés par la modélisation. On rajoute un critère non négligeable dû au coût énorme au niveau numérique ; ceci est dû essentiellement à la nécessité d'un maillage relativement fin du domaine étudié afin de capter un spectre significatif des échelles en jeu. Ces raisons limitent donc l'application de l'approche directe des modèles diphasique et nous amène à chercher une approche plus réaliste qui simule convenablement le phénomène et qui soit basé sur les modèles présentés. L'idée est de représenter l'effet de la phase air sur la phase eau tout en traitant un modèle à une seule phase. Il s'agit de prendre en compte la dynamique des bulles : d'une part, par une condition aux limites liée à la vitesse d'injection de l'eau (ceci pour simuler le mouvement initial des bulles libérées dans l'eau) et d'autre part, par l'introduction de termes de correction diphasique. Deux approches sont dérivées. Une première approche ayant pour objectif de montrer l'effet de l'injection des bulles en se limitant simplement à une condition aux limites aux injecteurs. La deuxième approche, plus élaborée, tient compte en plus de la vitesse d'injection et de l'effet des bulles par le biais de termes de correction provenant des forces exercées par celles-ci. Les difficultés qui subsistent sont liées au calibrage et à la validation des modèles.

1.6.1 Modèle monophasique

Le modèle monophasique consiste à traiter une seule phase, la phase eau, en tenant compte de la dynamique des bulles d'air par le biais d'une condition aux limites portant sur la vitesse d'injection de l'eau au niveau de l'injecteur.

Dans notre cas, la masse volumique de la phase liquide varie faiblement, ce qui permet d'écrire la condition

$$\text{div}u_L = 0 \quad (1.105)$$

et donc

$$\frac{\partial \rho_L}{\partial t} + u_L \cdot \nabla \rho_L = 0 \quad (1.106)$$

Nous nous ramenons à la résolution des équations de mouvement de l'eau représentées par les équations de Navier – Stokes semi-compressibles (1.108) ainsi qu'à une équation de transport sur la masse volumique de l'eau (1.107).

$$\begin{cases} \frac{\partial \rho_L}{\partial t} + u_L \cdot \nabla \rho_L = 0 & \text{dans } D \\ \rho_L(., t = 0) = \rho_L^0(.) & \text{dans } \Omega(t) \end{cases} \quad (1.107)$$

$$\begin{cases} \rho_L \left(\frac{\partial u_L}{\partial t} + u_L \cdot \nabla u_L \right) - \operatorname{div}(\mu_L (\nabla u_L + \nabla u_L^T)) + \nabla p_L = \rho_L g & \text{dans } D \\ \operatorname{div} u_L = 0 & \text{dans } D \\ u_L(., t = 0) = u_L^0(.) & \text{dans } \Omega(t) \\ u_L = u_d & \text{dans } \Gamma_F(t) \cap \partial \Omega_G(t) \end{cases} \quad (1.108)$$

1.6.2 Modèle avec termes de correction diphasique

Cette deuxième approche consiste à corriger le modèle monophasique par l'introduction de termes de correction représentant l'effet des bulles sur la phase eau. Ces termes sont déduits de la modélisation lagrangienne. En effet, en admettant que la vitesse du liquide est égale à celle du gaz en chaque point du domaine, l'équation de conservation de quantité de mouvement (1.103) prend la forme suivante :

$$\begin{aligned} \rho_L \frac{du_L}{dt} - \operatorname{div}(\mu_{\text{eff}} (\nabla u_L + \nabla u_L^T)) + \nabla p_L = \rho_L g \\ + \alpha \left[\rho_L \left(\frac{du_L}{dt} - g \right) + \rho_G \left(g - \frac{du_L}{dt} \right) \right] \end{aligned} \quad (1.109)$$

et s'écrit :

$$\begin{aligned} ((1 - \alpha)\rho_L + \alpha\rho_G) \frac{du_L}{dt} - \operatorname{div}(\mu_{\text{eff}} (\nabla u_L + \nabla u_L^T)) + \nabla p_L = \rho_L g \\ + \alpha(\rho_G - \rho_L)g \end{aligned} \quad (1.110)$$

où, α est le taux de présence vérifiant : $0 \leq \alpha \leq 1$

La définition de μ_{eff} introduite par Clement (*E. Clement*, 1997) se base sur des considérations expérimentales. Nous supposons par la suite que μ_{eff} s'écrit sous la forme suivante (*M. Abdelwahed*, 2002) :

$$\mu_{\text{eff}} = (1 - \alpha)\mu_L + \alpha\mu_G \quad (1.111)$$

Pour connaître les points du domaine recevant l'effet de la phase air, il faut retrouver le chemin de chaque particule air pour chaque pas de temps. Ceci revient à chercher l'origine à un temps $t + \Delta t$ de chaque point du domaine d'étude au temps t en suivant les trajectoires inverses. Ceci nous ramène à la résolution de l'équation de transport sur le taux de présence α . Cette équation nous permet d'identifier les points recevant l'effet de la dynamique des bulles. Dans l'équation de transport sur le taux de présence, α , a été donnée par (*M. Abdelwahed*, 2002):

1.7 Conclusion

Nous avons présenté dans ce chapitre les différents modèles qui peuvent être utilisés dans le problème traité. Les difficultés liées à la complexité de la modélisation diphasique et la taille des problèmes induits nous ont amené à dégager deux modèles simplifiés :

- un premier modèle, donné par les équations (1.107) ÷ (1.108), baptisé modèle monophasique, et qui consiste à tenir compte de l'effet des bulles par une condition aux limites au niveau de l'injecteur,
- un deuxième modèle donné par les équations (1.110) ÷ (1.112), plus complet, appelé, modèle corrigé, et qui s'appuie sur la modélisation lagrangienne en tenant compte de l'effet des forces exercées par des bulles en plus de la vitesse d'injection.

Nous nous intéressons maintenant à la modélisation chimique du problème traité.

Chapitre 2

Modélisation biochimique de processus de l'aération

2.1 Introduction

Pour un emplacement optimal des injecteurs (pour mieux aérer le lac) et la bonne quantification de la consommation de l'énergie inhérente à l'injection de l'air, il est important de bien comprendre le phénomène de l'aération dans tous ses aspects.

Dans le chapitre 1, nous avons présenté une modélisation purement physique de processus de l'aération, ce qui fait, en réalité, une omission d'autres aspects qui sont aussi importants pour la bonne compréhension du phénomène. Dans ce chapitre, une expansion de modèle physique est faite pour inclure les aspects biochimiques de phénomène. En effet, pendant la montée des bulles, qui est due à la décroissance de la pression en montant, une dynamique d'échange gaz-eau va s'établir dans le lac. Le modèle biochimique développé est associé avec le modèle monophasique ou monophasique corrigé. Le type et la complexité de couplage sont fondamentalement différents, nous abordons cet aspect dans le dernier paragraphe.

Dans la littérature, la modélisation de processus de l'aération est largement étudiée, soient par des études expérimentales, soient théoriques (on peut se référer par exemple à : *A. Wuest and al.*, 1992, *E. E. Prepas and al.*, 1997 ; *F. Edeline*, 1996; *D. F. McGinnis and al.*, 1998).

Les références essentiellement utilisées dans ce chapitre sont : *A. Wuest and al.*, 1992 ; *F. Edeline*, 1996; *H. Chanson*, 1996.

2.2 Notations

Nous rappelons ci-après les principales notations introduites dans les premières pages de ce mémoire.

- c_o : Concentration de l'oxygène (O_2) dissous ;
- c_N : Concentration de l'azote (N_2) dissous ;
- m_o : Concentration du gaz d'oxygène (O_2) non dissous ;
- m_N : Concentration du gaz de l'azote (N_2) non dissous.

2.3 Développement du model biochimique correspondant au modèle monophasique

L'air est un mélange de plusieurs gaz dans les proportions sont presque constantes : 21% de l'oxygène, 78% de nitrogène et 1% de gaz rares. Le fait que la proportion des gaz rares est petite devant l'oxygène et le nitrogène, nous supposons donc, que l'échange gaz - eau ne concernent que l'oxygène et le nitrogène et que l'échange gaz -eau avec les autres gaz est négligeable. Toutefois, les autres gaz sont facilement inclus dans le modèle par l'addition des équations de conservations correspondantes. La dissolution de la phase gazeuse (air ou oxygène plus ou moins pur) fourni au milieu liquide l'oxygène. Le passage de la phase gazeuse à la phase liquide associe des phases de transfert par diffusion et par convection qu'il est essentiel de considérer en détail.

2.3.1 Equations de conservation

Si on introduit la variable générale φ , la forme de la conservation de toutes équations de l'écoulement, incluant les équations des quantités scalaires comme la température et la concentration des polluants et en supposant que la diffusion suit la loi de Fick, peut s'écrire sous la forme suivante :

$$\frac{\partial(\rho \varphi)}{\partial t} + \text{div}(\rho \varphi u) = \text{div}(\Gamma \text{grad } \varphi) + S_{\varphi} \quad (2.1)$$

Cela signifie, en un mot :

$$\left\{ \begin{array}{l} \text{Taux d'augmentation} \\ \text{de } \varphi \text{ dans l'élément} \\ \text{fluide} \end{array} \right\} + \left\{ \begin{array}{l} \text{taux net de} \\ \varphi \\ \text{sortie de} \\ \text{l'élément} \\ \text{fluide} \end{array} \right\} = \left\{ \begin{array}{l} \text{taux de la croissance} \\ \text{de} \\ \varphi \text{ du à la diffusion} \end{array} \right\} + \left\{ \begin{array}{l} \text{taux de la} \\ \text{croissance} \\ \text{de } \varphi \\ \text{du aux sources} \\ \text{ou aux puits} \end{array} \right\}$$

L'équation (2.1) est appelée « équation de transport de la propriété φ ». Elle met en évidence plusieurs processus de transport : le terme de taux de changement et le terme de convection à gauche, et le terme diffusif ($\Gamma =$ coefficient de diffusion) et le terme source à droite. Dans les paragraphes qui suivent, nous nous intéressons aux équations de transport de l'oxygène et de nitrogène en mettant en évidence le coefficient de diffusion et les termes source.

a. Equations de conservation de l'oxygène et de nitrogène dissous

Il y a sept variables qui doivent être calculées par sept équations qui décrivent la dynamique créée par les bulles ainsi que les transferts des gaz existant. Ces équations sont basées sur le principe de la conservation de la masse, et de moment.

Dans notre cas, nous avons deux gaz : l'oxygène (O_2) et le nitrogène (N_2), et deux phases pour chacun, dissoute et gazeuse.

Le transfert gazeux est un transfert de masse à travers une interface. Par analogie avec les réactions chimiques, on peut prévoir que ce transfert de masse M sera proportionnel au déficit D à combler (= « driving force », écart moteur, force de dissolution, (**F. Edeline**, 1996), soit à l'équilibre :

$$F = f(C_s - C) = k_L D = dM / A dt \quad (2.2)$$

où,

F : flux unitaire, masse passant par l'unité de surface A pendant l'unité de temps,

k_L : coefficient de transfert superficiel,

C_s : concentration de saturation de gaz dissous,

C : concentration réelle de gaz dissous.

Gaz	C_s à la température T (mg/l)		(°C)
	Concentrations de saturation des gaz (oxygène, azote, CO ₂) atmosphérique	Concentrations de saturation des gaz (oxygène, azote, CO ₂) pur	T
Oxygène	8,84	42,29	20
Azote	15,32	19,29	20
CO ₂	0,84	23,19	10
NH ₃	0	959,200	25

Tableau 2.1 : Solubilité des gaz dans l'eau (**F. Edeline**, 1996)

Dans un système instable, le gaz transféré s'accumule dans l'eau, la concentration augmente sans cesse et D n'est pas constant. Nous remplacerons alors F par $\frac{dM}{A dt}$.

$$\frac{dC}{dt} = \frac{dM}{V dt} = K_2 (C_s - C) = K_2 D = k_L \frac{A}{V} D \quad (2.3)$$

K_2 : coefficient global de transfert [T^{-1}] ($1/K_2$ est la résistance au transfert) ;

M : masse de gaz transférée ;

A/V : interface par unité de volume, parfois appelée a .

K_2 est un coefficient global, valable pour toute l'enceinte où se trouve le liquide.

Mais en fait le transfert n'a lieu qu'à travers l'interface eau/gaz, de surface A , et c'est par là que doit passer tout le gaz transféré pour se répartir dans le volume V .

Nous pouvons définir donc un coefficient de film liquide, ou coefficient de transfert superficiel k_L , tel que : $K_2 = k_L a$

Les dimensions de k_L sont alors $[LT^{-1}]$, ce qui est bien une unité de transfert de masse : masse de gaz dissoute par unité de temps dans une unité de volume, à travers une unité de surface, et sous l'influence d'un déficit unitaire.

Si on veut accélérer le transfert, on s'ingéniera à augmenter le rapport $\frac{A}{V}$. Ce qui indique que la surface A de l'interface doit être la plus grande possible vis-à-vis du volume aéré. Cela peut être obtenu par insufflation des bulles d'air nombreuses et fines dans l'eau.

Si dans le volume V , nous avons N bulles, la surface A vaut $4\pi r^2 \times N$ (r est le rayon moyen de la bulle) ce qui donne :

$$a = \frac{4N\pi r^2}{V} = \frac{6}{d_b} \frac{\frac{4}{3}N\pi r^3}{V} = \frac{6}{d_b} \frac{V_G}{V} = \frac{6\alpha}{d_b} \quad (2.4)$$

avec, α : Taux de présence et d_b : Diamètre moyen de la bulle.

Maintenant, si on veut s'intéresser au transfert des deux gaz : l'oxygène et le nitrogène et si on prend en compte les notations du paragraphe 2.2, le flux de gaz à travers la surface des bulles est donnée par : $k_{Li}(C_{si} - C_i)$ ($g\ m^{-2}\ s^{-1}$)

où, k_{Li} est le coefficient de transfert superficiel de gaz i (O_2 et N_2), c_i est la concentration du gaz dissous, et C_{si} est la concentration du saturation du gaz dissous. Elle est déterminée par la loi d'Henry : $C_{si} = K_i p_i$ ($g\ m^{-3}$)

avec, p_i est la pression partielle de la phase gazeuse i , elle est donnée par :

$$p_i = \frac{m_i}{m_O + m_N} p$$

où, p est la pression totale donnée par le système hydrodynamique. Le coefficient K_i est la constante de solubilité.

Le coefficient de transfert superficiel k_{Li} est fonction de rayon de la bulle r_i . Pour les deux gaz, ce coefficient est presque le même $k_{LO}(r) = k_{LN}(r)$ (A. *Wuest and al.*, 1992).

On note que dans la pratique, on considère (*H. Roques*, 1980) :

- les petites bulles ayant des diamètres de l'ordre de 0,5 mm et au-dessous,
- les moyennes bulles entre 0,5 et 2,5 mm,
- les grosses bulles au-dessus de 2,5 mm.

Coefficient(s)	Remarques
$k_{LO} = 0.6r \text{ (m s}^{-1}\text{)}$	$r < 6.67 \times 10^{-4} \text{ m}$
$k_{LO} = 4 \times 10^{-4} \text{ (m s}^{-1}\text{)}$	$r \geq 6.67 \times 10^{-4} \text{ m}$
$K_O = 2.125 \times 10^{-5} - 5.021 \times 10^{-7} T + 5.77 \times 10^{-9} T^2 \text{ (gm}^{-3}\text{Pa}^{-1}\text{)}$	T en (°C)
$K_N = 1.042 \times 10^{-5} - 2.450 \times 10^{-7} T + 3.171 \times 10^{-9} T^2 \text{ (gm}^{-3}\text{Pa}^{-1}\text{)}$	T en (°C)

Tableau 2.2 : Coefficients de transferts et les constantes d'Henry.

Maintenant, on s'intéresse aux termes de diffusion. Tout d'abord, la diffusion de la matière c'est un phénomène de transport de matière tendant à compenser respectivement des différences de concentration. A cause de ça, la partie de l'air dissoute va se diffuser dans l'eau.

La diffusion est gouvernée par l'équation fondamentale : « l'équation de Fick », qui définit la densité de flux de particules – notée j_n – qui est la quantité de matière traversant par seconde l'unité d'aire d'une surface normale à la direction de propagation des particules. D'après la loi de Fick, j_n est proportionnel au gradient de la concentration C des particules :

$$j_n = -D_m \nabla C \quad (2.5)$$

avec, D_m : le coefficient de diffusion moléculaire de la matière diffusante.

Dans notre cas, nous avons :

$$j_{ni} = -D_{mi} \nabla C_i \quad (2.6)$$

où, D_{mi} est le coefficient de diffusion moléculaire de gaz i (O_2 et N_2) dans l'eau, C_i est la concentration du gaz dissous.

Le coefficient de diffusion gaz -liquide de l'oxygène et de nitrogène dans l'eau est :

Température (°C)	$D_m(O_2)$ Oxygène m^2/s	$D_m(N_2)$ Nitrogène m^2/s
10	1.54E-9	1.29E-9
25	2.20E-9	2.01E-9
40	3.33E-9	2.83E-9
55	4.50E-9	3.80E-9

Tableau 2.3 : Coefficients moléculaires de diffusion. (**F. Edeline**, 1996)

Après avoir présenter les différents termes qui constituent l'équation de conservation, nous pouvons donc écrire les deux équations de transport des deux gaz dissous (oxygène et nitrogène dissous) :

$$\left\{ \begin{array}{l} \frac{\partial C_O}{\partial t} + u \nabla C_O = k_{LO} a (K_O p_O - C_O) + D_O \Delta C_O \\ \frac{\partial C_N}{\partial t} + u \nabla C_N = k_{LN} a (K_N p_N - C_N) + D_N \Delta C_N \\ p_O = \frac{m_O}{m_O + m_N} p \\ p_N = \frac{m_N}{m_O + m_N} p \\ a = \frac{6\alpha}{d_B} \end{array} \right. \quad (2.7)$$

En ce qui concerne la conservation des deux gaz non dissous, leurs équations se déduisent immédiatement. En effet, la phase liquide des deux gaz s'enrichit toujours par le terme source $k_{Li} (C_{si} - C_i)$, ce qui constitue un gain pour elle. Par contre, la phase gazeuse va perdre la même quantité :

$$\left\{ \begin{array}{l} \frac{\partial m_O}{\partial t} + u \nabla m_O = -k_{LO} a (K_O p_O - C_O) \\ \frac{\partial m_N}{\partial t} + u \nabla m_N = -k_{LN} a (K_N p_N - C_N) \end{array} \right. \quad (2.8)$$

Il nous reste à exprimer le taux de présence α pour fermer le système.

Le terme α constitue le taux de présence de l'air dans le liquide, c'est-à-dire, si on prend un volume de gaz (l'air) V_G contenue dans le volume de total (air + liquide) V_T , le taux de présence α est égal à :

$$\alpha = \frac{\sum V_{Gi}}{V_G + V_L} = \frac{V_G}{V_T} = \frac{V_O + V_N}{V_T} = \frac{M_O / \rho_O + M_N / \rho_N}{V_T}$$

ici, M_O et M_N constituent respectivement la masse de l'oxygène et la masse de nitrogène contenues dans le volume V_T .

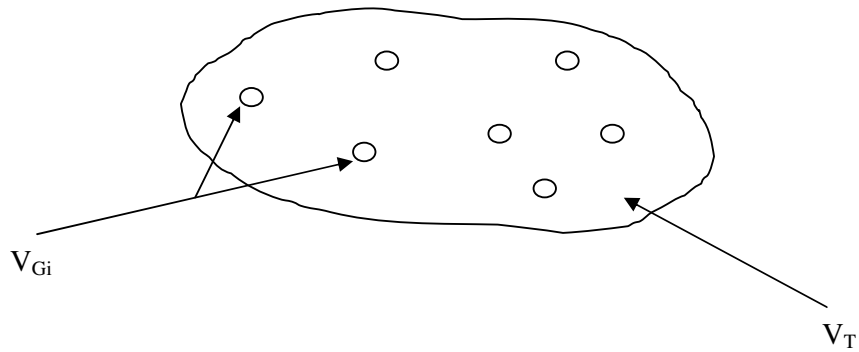


Figure 2.1 : Dessin schématique des bulles d'air

On a : $m_i = \frac{M_i}{V_T}$ ce qui implique :

$$\alpha = \frac{m_O}{\rho_O} + \frac{m_N}{\rho_N} \quad (2.9)$$

Le système final (correspondant au modèle hydrodynamique monophasique) est :

$$\left\{ \begin{array}{ll} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \mathbf{u} \right) - \mu \Delta \mathbf{u} + \nabla p = \rho \mathbf{g} & \text{dans } D \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } D \\ \mathbf{u} = \mathbf{u}_d & \text{sur } \Gamma_F(t) \cap \partial \Omega_G(t) \\ \mathbf{u}(\cdot, t = 0) = \mathbf{u}^0(\cdot) & \text{dans } \Omega(t) \\ \frac{d\rho}{dt} = 0 & \text{dans } D \\ \rho(\cdot, t = 0) = \rho^0(\cdot) & \text{dans } \Omega(t) \end{array} \right. \quad (2.10)$$

en ajoutant le modèle chimique :

$$\left\{ \begin{array}{l}
\frac{\partial C_O}{\partial t} + u \nabla C_O = k_{LO} a (K_{OP} P_O - C_O) + D_O \Delta C_O \\
\frac{\partial C_N}{\partial t} + u \nabla C_N = k_{LN} a (K_{NP} P_N - C_N) + D_N \Delta C_N \\
\frac{\partial m_O}{\partial t} + u \nabla m_O = -k_{LO} a (K_{OP} P_O - C_O) \\
\frac{\partial m_N}{\partial t} + u \nabla m_N = -k_{LN} a (K_{NP} P_N - C_N) \\
p_O = \frac{m_O}{m_O + m_N} P \\
p_N = \frac{m_N}{m_O + m_N} P \\
a = \frac{6 \alpha}{d_B} \\
\alpha = \frac{m_O}{\rho_O} + \frac{m_N}{\rho_N} \quad (8)
\end{array} \right. \quad (2.11)$$

En multipliant (2.3) par $\frac{1}{\rho_O}$ et (2.4) par $\frac{1}{\rho_N}$ et par sommation en utilisant (2.8), nous obtenons :

$$\begin{aligned}
\frac{\partial \alpha}{\partial t} + u \nabla \alpha &= - \left(k_{LO} a \frac{(K_{OP} P_O - C_O)}{\rho_O} + k_{LN} a \frac{(K_{NP} P_N - C_N)}{\rho_N} \right) \\
&= -k_{LO} a \left(\frac{(K_{OP} P_O - C_O)}{\rho_O} + \frac{(K_{NP} P_N - C_N)}{\rho_N} \right)
\end{aligned}$$

Le modèle chimique devient :

$$\left\{ \begin{array}{l}
 \frac{\partial C_O}{\partial t} + u \nabla C_O = k_{LO} a (K_O p_O - C_O) + D_O \Delta C_O \quad \text{dans D} \\
 \frac{\partial C_N}{\partial t} + u \nabla C_N = k_{LN} a (K_N p_N - C_N) + D_N \Delta C_N \quad \text{dans D} \\
 \frac{\partial \alpha}{\partial t} + u \nabla \alpha = -k_{LO} a \left(\frac{(K_O p_O - C_O)}{\rho_O} + \frac{(K_N p_N - C_N)}{\rho_N} \right) \quad \text{dans D} \\
 p_O = \frac{m_O}{m_O + m_N} p \\
 p_N = \frac{m_N}{m_O + m_N} p \\
 a = \frac{6\alpha}{d_B}
 \end{array} \right. \quad (2.12)$$

2.3.2 Influence de la qualité de l'eau sur les coefficients de transferts

Les mesures du coefficient de transfert ont montré que si tout reste égal en dehors de la qualité de l'eau, le coefficient global de transfert subit des variations notables (*F. Edeline*, 1996 ; *C. Gomella et al.*, 1982).

Influence de la température : Ce coefficient global varie avec la température. Lorsqu'on souhaite calculer K_2 à $\theta^\circ\text{C}$ à partir du K_2 mesuré à une température de référence, 10°C par exemple, on utilise des formules de correction empiriques du type :

$$K_2 \text{ à } \theta^\circ\text{C} = (K_2 \text{ à } 10^\circ\text{C}) 1,024^{\theta-10}$$

Influence de la pollution : le coefficient de transfert k_L de l'eau polluée est sensiblement plus faible que celui de l'eau pure. On désigne par β le rapport:

$$\beta = \frac{k_L \text{ eau usée}}{k_L \text{ eau pure}}$$

Les valeurs de k_L données dans le tableau 2.2 sont des valeurs mesurées de l'eau de robinet. Nous considérons que l'eau de robinet est une eau pure.

Les valeurs numérique de ce coefficient β varient de 0,2 à 0,7 pour les eaux d'égout domestiques en fonction de leur teneur en matière polluantes ; les détergents, les produits antimousses, les matières albuminoïdes ont un effet considérable dont quelques exemples sont donnés dans le tableau 2.4.

Polluant	Concentration (mg/l)	α
Peptone	25	0,5
Détergent	50	0,35

Tableau 2.4 : Coefficient β .

2.3.3 Conditions initiales et aux limites

De point de vu physique, le processus de l'aération - c'est-à-dire la résolution dans le temps du modèle chimique- dépend des conditions initiales et aux limites.

Mathématiquement, le modèle chimique a un sens si on lui ajoute les conditions aux limites et initiales.

a- Conditions aux limites

Si nous avons une frontière solide que l'on considère imperméable - ce qui empêche toute sorte de flux à travers cette surface - la densité de flux est égale à : $j_n = -D_m \nabla C$.

Le flux à travers la surface élémentaire ds et pendant le temps dt est égal à :

$$j_{ni} n ds dt = -D_{mi} \nabla C_i n ds dt = -D_{mi} \frac{\partial C_i}{\partial n} ds dt = 0 \Rightarrow \frac{\partial C_i}{\partial n} = 0 \quad (2.13)$$

Ce qui constitue une condition de Newman.

La même chose pour la phase gazeuse représentée par les concentrations m_i et α .

On impose une condition de Newman sur m_i et α :

$$\frac{\partial m_i}{\partial n} = 0 \text{ et } \frac{\partial \alpha}{\partial n} = 0. \quad (2.14)$$

où, n est le vecteur normal à la surface ds .

Nous imposons une condition de Dirichlet au niveau de l'injecteur. En effet, le flux des bulles d'air est connu et imposé, on le note g_b . Ce flux sort à travers les perforations avec la vitesse v_b .

Pendant un temps t , le fluide parcourt verticalement une distance L' , soit $t = \frac{L'}{v_b}$.

Dans le volume $L' \times L \times 1$, on trouvera tout le gaz émis pendant le temps t , soit $V_G = \frac{g_b L'}{v_b}$.

Nous supposons approximativement que la vitesse des bulles reste verticale et constante au voisinage de l'injecteur.

On déduit une valeur approximative de α à proximité de l'injecteur :

$$\alpha = \frac{V_G}{L' \times L \times 1} = \frac{g_b}{L v_b}$$

Sachant que l'air est composé approximativement de 79% de l'azote et 21% de l'oxygène, on peut déduire m_i à la sortie de l'injecteur.

Pour C_i , nous considérons que l'air n'a pas commencé sa dissolution, c'est-à-dire $C_i = 0$.

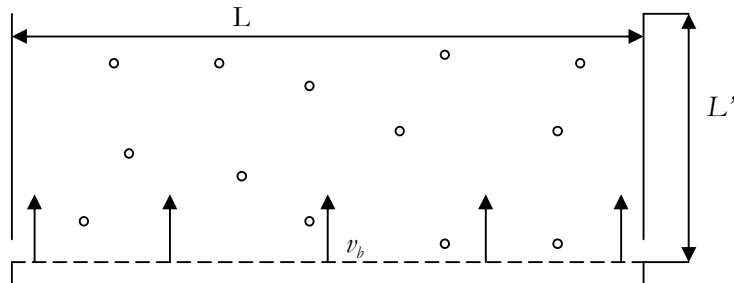


Figure 2.2 : Le voisinage de l'injecteur.

À la surface libre, l'eau et l'air sont en échange permanent et l'air apporte de l'oxygène et l'azote dissous à l'eau ; donc le flux des gaz dissous apportés à l'eau est égale au taux de dissolution de l'air de l'atmosphère dans l'eau :

$$j_{ni} n = -D_i \frac{\partial C_i}{\partial n} = k_{Li} (C_{si} - C_i)$$

ce qui donne :

$$\frac{\partial C_i}{\partial n} + \frac{k_{Li}}{D_i} (C_{si} - C_i) = 0. \quad (2.15)$$

b- Conditions initiales

Les conditions initiales dépendent de l'état du lac considéré et de la saison dans laquelle on fait l'aération. Toutefois, les conditions initiales peuvent être prises comme les solutions stationnaires des modèles chimiques (2.11-2.12). En effet, avant le commencement de l'aération, le lac est considéré immobile et en équilibre avec l'atmosphère de point de vue échange de gaz.

Ce qui reste comme processus de transformation dans le lac dans un tel cas, c'est la transformation par diffusion moléculaire, ce qui nous donne des Laplace à résoudre : $\Delta C_i = 0$ avec $C_i = C_{si}$ à la surface libre.

Pour la partie gazeuse, à $t = 0$, on prend $m_i = 0$ et par suite, $\alpha = 0$.

2.4 Développement du modèle biochimique correspondant au modèle monophasique corrigé

Le modèle chimique est le même pour le modèle monophasique et le monophasique corrigé. Mais le lien entre les équations qui représentent le modèle monophasique et monophasique corrigé avec le modèle chimique n'est pas le même.

Pour le modèle monophasique, le système d'équations hydrodynamique est découplé numériquement au système d'équations chimiques ; on résout chaque système sans avoir recours à l'autre.

Par contre, pour le modèle monophasique corrigé, les deux systèmes sont liés, ce qui engendre une difficulté supplémentaire dans la procédure de résolution numérique.

Nous donnons ci-après le modèle complet, hydrodynamique et chimique :

$$\left\{ \begin{array}{l} (\alpha \rho_G + (1 - \alpha) \rho_L) \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) - (\alpha \mu_G + (1 - \alpha) \mu_L) \Delta u + \nabla p = \rho_L g \\ \quad + \alpha (\rho_G - \rho_L) g \quad \text{dans } D \\ \operatorname{div} u = 0 \quad \text{dans } D \\ u = u_d \quad \text{sur } \Gamma_F(t) \cap \partial \Omega_G(t) \\ u(., t = 0) = u^0(.) \quad \text{dans } \Omega(t) \\ \frac{d\rho_L}{dt} = 0 \quad \text{dans } D \\ \rho_L(., t = 0) = \rho_L^0(.) \quad \text{dans } \Omega(t) \\ \text{avec les conditions aux limites citées dans le paragraphe précédents} \\ \alpha(., t = 0) = \alpha^0(.) \quad \text{dans } \Omega(t) \end{array} \right.$$

$$\left\{ \begin{array}{l}
 \frac{\partial C_O}{\partial t} + u \nabla C_O = k_{LO} a (K_{OP} - C_O) + D_O \Delta C_O \text{ dans } D \\
 \frac{\partial C_N}{\partial t} + u \nabla C_N = k_{LN} a (K_{NP} - C_N) + D_N \Delta C_N \text{ dans } D \\
 \frac{\partial \alpha}{\partial t} + u \nabla \alpha = -k_{LO} a \left(\frac{(K_{OP} - C_O)}{\rho_O} + \frac{(K_{NP} - C_N)}{\rho_N} \right) \text{ dans } D \\
 p_O = \frac{m_O}{m_O + m_N} p \\
 p_N = \frac{m_N}{m_O + m_N} p \\
 a = \frac{6\alpha}{d_B} \\
 \text{avec les conditions initiales et aux limites citées dans le paragraphes précédents.}
 \end{array} \right.$$

Chapitre 3

Approximation numérique du problème continu

3.1 Introduction

Dans ce chapitre, nous présentons les équations de Navier – Stokes incompressible, puis la méthode des caractéristiques qui est utilisée pour la discrétisation en temps. Dans le paragraphe 3.5 nous présentons la formulation variationnelle des équations de Navier – Stokes incompressible. Après, nous parlerons de la discrétisation en espace et la nécessité et l'avantage de l'utilisation de l'élément $P^1 + \text{Bulle} / P^1$. La résolution du système obtenu est faite par la méthode de gradient conjugué Uzawa.

Ensuite, l'algorithme de résolution utilisé dans le code sera présenté. Finalement, des cas test de code seront montrés sur le lac de Bouregreg.

3.2 Notations

Dans ce qui suit, le domaine d'étude est noté Ω , sa frontière notée Γ , on suppose de plus que Ω et Γ vérifient les hypothèses mathématiques qui seront nécessaires par la suite.

On adopte les notations suivantes :

- l'élément d'aire est symbolisé par $d\Omega$, l'élément surfacique par $d\Gamma$ et l'élément temporel par dt .
- ∂_i désigne la dérivée partielle d'ordre 1 par rapport à la variable i pour $i = 1, 2$.
- On note $\nabla_{\mathbf{v}} = \begin{pmatrix} \partial_1 \mathbf{v} \\ \partial_2 \mathbf{v} \end{pmatrix}$ pour tout $\mathbf{v} \in D'(\Omega)$.
- On note $\mathbf{u} = (u_1, u_2)$ et

$$\nabla \mathbf{u} = \begin{pmatrix} \partial_1 u_1 & \partial_2 u_1 \\ \partial_1 u_2 & \partial_2 u_2 \end{pmatrix}$$

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \nabla \mathbf{u} + (\nabla \mathbf{u})^t = (\boldsymbol{\varepsilon}_{i,j}(\mathbf{u}))_{1 \leq i,j \leq 2} = (\partial_i u_j + \partial_j u_i)_{1 \leq i,j \leq 2} = \begin{pmatrix} 2\partial_1 u_1 & \partial_2 u_1 + \partial_1 u_2 \\ \partial_1 u_2 + \partial_2 u_1 & 2\partial_2 u_2 \end{pmatrix}$$

$$\boldsymbol{\varepsilon}(\mathbf{u})\boldsymbol{\varepsilon}(\mathbf{v}) = \sum_{i,j=1}^2 \boldsymbol{\varepsilon}_{ij}(\mathbf{u})\boldsymbol{\varepsilon}_{ij}(\mathbf{v}) = 4(\partial_1 u_1 \partial_1 v_1 + \partial_2 u_2 \partial_2 v_2) + 2(\partial_1 u_2 + \partial_2 u_1)(\partial_1 v_2 + \partial_2 v_1)$$

- T désigne un temps fixe, $T \in \mathfrak{R}_+^*$.
- $Q_T = \Omega \times]0, T[$.
- $\Sigma_T = \Gamma \times]0, T[$.
- $\Sigma(\mathbf{u}) = \{(x, t) | \mathbf{u} \cdot \mathbf{n}(x) < 0\}$.
- $(T_h)_h$ est une famille régulière de décomposition de $\overline{\Omega}$ en triangles.

3.3 Position du problème

Les modèles physiques qui sont utilisées pour la modélisation de processus de l'aération sont les suivants :

- Modèle monophasique :

$$\begin{cases} \frac{\partial \rho_L}{\partial t} + \mathbf{u}_L \cdot \nabla \rho_L = 0 & \text{dans } Q_T \\ \rho_L(., t = 0) = \rho_L^0(.) & \text{dans } \Omega \\ \rho \text{ donnée} & \text{sur } \Sigma(\mathbf{u}_d) \cap \Sigma_T \end{cases} \quad (3.1)$$

$$\begin{cases} \rho_L \left(\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right) - \operatorname{div}(\boldsymbol{\mu}_L (\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T)) + \nabla p_L = \rho_L \mathbf{g} & \text{dans } Q_T \\ \operatorname{div} \mathbf{u}_L = 0 & \text{dans } Q_T \\ \mathbf{u}_L(., t = 0) = \mathbf{u}_L^0(.) & \text{dans } \Omega \\ \mathbf{u}_L = \mathbf{u}_d & \text{dans } \Sigma_T \end{cases} \quad (3.2)$$

- Modèle monophasique avec termes de correction diphasique :

$$\left\{ \begin{array}{ll}
\frac{\partial \alpha}{\partial t} + \mathbf{u}_L \cdot \nabla \alpha = 0 & \text{dans } Q_T \\
\alpha(., t = 0) = \alpha^0(.) & \text{dans } \Omega \\
\alpha \text{ donnée} & \text{sur } \Sigma(\mathbf{u}_d) \cap \Sigma_T \\
\frac{\partial \rho_L}{\partial t} + \mathbf{u}_L \cdot \nabla \rho_L = 0 & \text{dans } Q_T \\
\rho_L(., t = 0) = \rho_L^0(.) & \text{dans } \Omega \\
((1 - \alpha)\rho_L + \alpha\rho_G) \frac{d\mathbf{u}_L}{dt} - \text{div}(\boldsymbol{\mu}_{\text{eff}} (\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T)) + \nabla p_L = \rho_L \mathbf{g} & \\
+ \alpha(\rho_G - \rho_L) \mathbf{g} & \text{dans } Q_T \\
\text{div} \mathbf{u}_L = 0 & \text{dans } Q_T \\
\mathbf{u}_L(., t = 0) = \mathbf{u}_L^0(.) & \text{dans } \Omega \\
\mathbf{u}_L = \mathbf{u}_d & \text{sur } \Sigma_T
\end{array} \right. \quad (3.3)$$

Les codes développés sont basés sur ces deux modèles. Comme le but est de montrer la méthode de discrétisation utilisée, on se limitera à la discrétisation des équations de Navier – Stokes incompressible. En effet, de point de vue théorique, la procédure de discrétisation est pratiquement la même pour les trois modèles : le modèle monophasique, le modèle monophasique corrigé et le modèle de Navier – Stokes incompressible ; aussi, dans ce mémoire, le travail de parallélisation a été fait sur le modèle de Navier – Stokes incompressible. Le modèle qui nous intéresse est donc le suivant :

$$\left\{ \begin{array}{ll}
\rho_L \left(\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right) - \text{div}(\boldsymbol{\mu}_L (\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T)) + \nabla p_L = \rho_L \mathbf{g} & \text{dans } Q_T \\
\text{div} \mathbf{u}_L = 0 & \text{dans } Q_T \\
\mathbf{u}_L(., t = 0) = \mathbf{u}_L^0(.) & \text{dans } \Omega \\
\mathbf{u}_L = \mathbf{u}_d & \text{dans } \Sigma_T
\end{array} \right. \quad (3.4)$$

où, \mathbf{g} est la force de gravité, ρ la masse volumique de fluide, p sa pression et \mathbf{u} sa vitesse.

Pour résoudre le problème précédent, nous avons choisi une méthode de Lagrange-Galerkin : on utilise la méthode des caractéristiques pour la discrétisation en temps et une méthode d'élément finis mixte pour la discrétisation en espace.

Ce type de schéma est très utilisé en mécanique des fluides numérique pour résoudre les problèmes de type Navier – Stokes (*Pares*, 1988 ; *G. Fourestey* 2002, *F. Hecht et al.* 1991).

3.4 Discrétisation en temps (méthode des caractéristiques)

L'idée de la méthode des caractéristiques vient du fait que la partie convective de la première équation (3.4) coïncide avec la dérivée totale de \mathbf{u} :

$$\left\{ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right\} (\mathbf{x}, t) = \frac{D\mathbf{u}}{Dt} (\mathbf{x}, t) = \frac{d}{d\tau} \mathbf{u}(X(\mathbf{u}, t; \boldsymbol{\tau}), \boldsymbol{\tau}) \Big|_{\tau=t} \quad (3.5)$$

où, $X(\mathbf{x}, t; \boldsymbol{\tau})$ est la caractéristique qui est au point \mathbf{x} au temps t , c'est-à-dire, la solution de l'équation différentiel ordinaire :

$$\frac{dX(\mathbf{x}, t; \boldsymbol{\tau})}{dt} = \mathbf{u}(X(\mathbf{x}, t; \boldsymbol{\tau}), \boldsymbol{\tau}); \quad X(\mathbf{x}, t; t) = \mathbf{x} \quad (3.6)$$

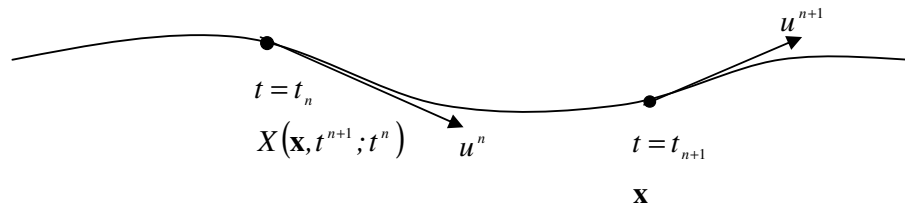


Figure 3-1 : Approximation de la dérivée totale

On fait alors l'approximation suivante :

$$\frac{\partial \mathbf{u}}{\partial t} (\mathbf{x}, t^{n+1}) + (\mathbf{u}(\mathbf{x}, t^{n+1}) \cdot \nabla) \mathbf{u}(\mathbf{x}, t^{n+1}) \cong \frac{\mathbf{u}^{n+1}(\mathbf{x}) - \mathbf{u}^n(\boldsymbol{\chi}^n(\mathbf{x}))}{\delta t} \quad (3.7)$$

où, δt est le pas de temps choisi, $t^n = n\delta t$ le nième pas de temps, \mathbf{u}^n la vitesse approchée au temps t^n et $\boldsymbol{\chi}^n$ est la fonction qui fait correspondre au point \mathbf{x} le point $X(\mathbf{x}, t^{n+1}; t^n)$, qui est la position au temps t^n de la particule qui se trouve au point \mathbf{x} au temps t^{n+1} .

On voit donc que la partie de convection non linéaire est cachée dans la dérivée totale de la première équation. Son calcul reviendra en fait à déterminer les courbes caractéristiques issues des points \mathbf{x} .

En introduisant l'approximation (3.7) dans le système de Navier – Stokes (3.4) et en utilisant un schéma d'Euler implicite pour approcher le problème en temps :

Etant calculé \mathbf{u}^n, p^n , trouver $\mathbf{u}^{n+1}, p^{n+1}$ tels que :

$$\begin{cases} \frac{\rho}{\Delta t} \mathbf{u}^{n+1} - \operatorname{div}(\boldsymbol{\mu}(\nabla \mathbf{u}^{n+1} + \nabla \mathbf{u}^{n+1 T})) + \nabla p^{n+1} = \rho \mathbf{g} + \frac{\rho}{\Delta t} \mathbf{u}^n \circ \boldsymbol{\chi}^n & \text{dans } \Omega \\ \operatorname{div} \mathbf{u}^{n+1} = 0 & \text{dans } \Omega \\ \mathbf{u}^{n+1} = \mathbf{u}_d(t^{n+1}) & \text{sur } \Gamma \end{cases} \quad (3.8)$$

A chaque étape de l'algorithme précédent, on a à résoudre un problème de Stokes généralisé, que l'on peut l'écrire de la façon suivante :

Trouver \mathbf{u} et p solution de :

$$\begin{cases} \frac{\rho}{\Delta t} \mathbf{u} - \operatorname{div}(\boldsymbol{\mu}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \nabla p = \mathbf{F} & \text{p.p. dans } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{p.p. dans } \Omega \\ \mathbf{u} = \mathbf{u}_d & \text{p.p. sur } \Gamma \end{cases} \quad (3.9)$$

3.5 Problème continu

On s'intéresse maintenant à la résolution numérique de problème de Stokes généralisé (3.9).

On pose : $\lambda = \frac{\rho}{\Delta t}$, le problème (3.9) peut s'écrire :

Trouver \mathbf{u} et p tels que :

$$\begin{cases} \lambda \mathbf{u} - \boldsymbol{\mu} \Delta \mathbf{u} + \nabla p = \mathbf{F} & \text{p.p. dans } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{p.p. dans } \Omega \\ \mathbf{u} = \mathbf{u}_d & \text{p.p. sur } \Gamma \end{cases} \quad (3.10)$$

avec, \mathbf{u}_d vérifie :

$$\int_{\Gamma} \mathbf{u}_d d\Gamma = 0. \quad (3.11)$$

3.5.1 Formulation faible du problème de Stokes généralisé

Après la discrétisation en temps par la méthode des caractéristiques, on passe à la résolution du problème de Stokes généralisé par élément finis. Cette résolution passe par une formulation variationnelle (ou formulation faible) du problème de Stokes généralisé.

On définit l'espace $L^2(\Omega)$ des fonction de carré intégrable au sens de Lebesgues sur Ω , par :

$$L^2(\Omega) = \left\{ \psi \mid \int_{\Omega} \psi^2 < \infty \right\}.$$

Il est doté d'une norme :

$$\|\psi\|_{L^2(\Omega)} = \left(\int_{\Omega} \psi^2 \right)^{\frac{1}{2}}.$$

Le symbole $L^2(\Omega)$ désigne l'espace $L^2(\Omega) \times L^2(\Omega)$.

$L_0^2(\Omega)$ est le sous-espace défini par :

$$L_0^2(\Omega) = \left\{ f \in L^2(\Omega) \mid \int_{\Omega} f d\Omega = 0 \right\}$$

Pour tout multi-indice $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ de dimensions d , on peut définir l'opérateur différentiel linéaire suivant :

$$D^{\alpha} = \frac{\partial^{\alpha_1 + \alpha_2 + \dots + \alpha_d}}{\partial_{x_1}^{\alpha_1} \partial_{x_2}^{\alpha_2} \dots \partial_{x_d}^{\alpha_d}}$$

On définit alors, pour tout entier s non négatif, l'espace de Sobolev $H^s(\Omega)$ défini par :

$$H^s(\Omega) = \left\{ \psi \in L^2(\Omega) \mid D^m \psi \in L^2(\Omega), m = 1, \dots, s \right\}.$$

On pose : $IH^s(\Omega) = H^s(\Omega) \times H^s(\Omega)$.

On notera que : $H^0(\Omega) = L^2(\Omega)$. On associe à cet espace la norme suivante :

$$\|\psi\|_{H^s(\Omega)} = \left(\|\psi\|_{L^2(\Omega)}^2 + \sum_{|\alpha|=1}^s \|D^{\alpha} \psi\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}.$$

On définit aussi, H_0^1 le sous espace de H^1 défini par :

$$H_0^1(\Omega) = \left\{ \psi \in H^1 \mid \psi = 0 \text{ sur } \partial\Omega \right\},$$

$\partial\Omega$: étant la frontière supposée assez régulière de Ω . Cet espace possède un espace dual, noté H^{-1} , associé à la norme :

$$\|\psi\|_{-1, \Omega} = \sup_{v \in H_0^1, v \neq 0} \frac{\langle \psi, v \rangle_{H^{-1}, H_0^1}}{\|v\|_{1, \Omega}}$$

avec ces définitions, on définit maintenant la formulation faible du problème de Stokes généralisé.

3.5.2 Formulation variationnelle

On pose :

$$X^d = \left\{ v \in \text{IH}^1(\Omega), v = u_d \text{ p.p. sur } \Gamma \right\} \quad (3.12)$$

On a $u_d \in \text{IH}^{\frac{1}{2}}(\Gamma)$; on peut montrer qu'il existe

$$\bar{u} \in X^d \text{ tel que } \nabla \cdot \bar{u} = 0 \text{ dans } \Omega \quad (3.13)$$

En utilisant la propriété (3.13), la formulation variationnelle du problème (3.10) s'écrit :

$$\begin{cases} \text{Trouver } u \text{ dans } \bar{u} + \text{IH}_0^1(\Omega) \text{ et } p \text{ dans } L_0^2(\Omega) \text{ telles que} \\ \lambda(u, v) + \mu(\nabla u, \nabla v) - (p, \nabla \cdot v) = F(v) \quad \forall v \in \text{IH}_0^1(\Omega) \\ -(q, \nabla \cdot u) = 0 \quad \forall q \in L_0^2(\Omega) \end{cases} \quad (3.14)$$

Où, $(u, v) = \int_{\Omega} u v d\Omega, \dots$

Le problème (3.14) n'est pas homogène, or la théorie de l'existence et l'unicité sur telles équations est faite aux cas homogènes. Pour homogénéiser le problème (3.14), on procède comme suit :

On définit sur $\text{IH}_0^1(\Omega)$ la forme linéaire continue

$$\tilde{F}(v) = F(v) - \lambda(\bar{u}, v) - \mu(\nabla \bar{u}, \nabla v) \quad \forall v \in \text{IH}_0^1(\Omega) \quad (3.15)$$

On considère donc, le problème homogène suivant :

$$\begin{cases} \text{Trouver } \tilde{u} \text{ dans } \text{IH}_0^1(\Omega) \text{ et } p \text{ dans } L_0^2(\Omega) \text{ telles que} \\ \lambda(\tilde{u}, v) + \mu(\nabla \tilde{u}, \nabla v) - (p, \nabla \cdot v) = (\tilde{F}, v) \quad \forall v \in \text{IH}_0^1(\Omega) \\ -(q, \nabla \cdot \tilde{u}) = 0 \quad \forall q \in L_0^2(\Omega) \end{cases} \quad (3.16)$$

Le problème (3.16) est équivalent à

$$\begin{cases} \text{Trouver } u = \tilde{u} + \bar{u} \text{ dans } X^d \text{ et } p \text{ dans } L_0^2(\Omega) \text{ telles que} \\ \lambda(u, v) + \mu(\nabla u, \nabla v) - (p, \nabla \cdot v) = (F, v) \quad \forall v \in \text{IH}_0^1(\Omega) \\ -(q, \nabla \cdot u) = 0 \quad \forall q \in L_0^2(\Omega) \end{cases} \quad (3.17)$$

On a donc équivalence entre les problèmes (3.16) et (3.17). D'autre part, le problème (3.17) est équivalent au problème (3.10).

Cette formulation peut s'écrire en terme de formes bilinéaires :

$$\begin{aligned} a(u, v) &= \int_{\Omega} \lambda u v d\Omega + \int_{\Omega} \mu \nabla u \nabla v d\Omega \\ b(v, q) &= - \int_{\Omega} q \nabla \cdot v d\Omega \end{aligned} \quad (3.18)$$

Ainsi le problème (3.16) s'écrit sous la forme suivante :

$$\begin{cases} \text{Trouver } (u, p) \in \mathbf{IH}_0^1(\Omega) \times L_0^2(\Omega) \text{ solution de} \\ a(u, v) + b(v, p) = \tilde{F}(v) & \forall v \in \mathbf{IH}_0^1(\Omega) \\ b(u, q) = 0 & \forall q \in L_0^2(\Omega) \end{cases} \quad (3.19)$$

L'existence et l'unicité de la solution du système de Stokes se fait à l'aide de théorème de Babuska-Brezzi. On montre l'existence et l'unicité du problème homogène (3.16) et à l'aide des équivalences entre les problèmes (3.16), (3.17) et le problème de Stokes (3.10), on déduit l'existence et l'unicité des problèmes (3.17) et (3.10) (pour plus de détail voir A. *Ern et al*, 2002 ; O. *Pironneau*, 1989 ; M. *Abdelwahed*, 2002).

3.6 Problème discret

On étudie dans ce paragraphe la discrétisation en espace des équations de Stokes généralisées. Après avoir écrit la formulation variationnelle (3.19) du système de Stokes généralisés (3.9), on s'intéresse maintenant à la discrétisation de cette formulation variationnelle. Il est bien connu que les équations de type Navier – Stokes ne peuvent être discrétisées dans n'importe quelles combinaison d'espaces (O. *Pironneau*, 1989).

Par exemple, le problème variationnelle (3.19) est mal posé pour une approximation d'ordre 1 en vitesse et en pression, en fait, même pour toute approximation polynomiale d'ordre identique pour la vitesse et la pression conduira aux mêmes problèmes (O. *Pironneau*, 1989 ; A. *Ern et al*, 2002 ; G. *Fourestey*, 2002). Cependant, il y a un critère, appelé « le critère inf-sup », s'il est vérifié, il nous permet de s'assurer de l'existence et de l'unicité des solutions du problème de Stokes généralisées. Ce critère est donné par (A. *Ern et al*, 2002) :

Soient X et M deux espaces fonctionnels des champs de vitesse et de pression u et p vérifient la condition suivante

$$\inf_{p \in M_h} \sup_{v \in X_h} \frac{(\nabla \cdot u, p)}{\|u\|_X \|p\|_M} \geq \gamma > 0$$

pour $Q_h = \{q \in M_h \mid (q, \nabla \cdot v) = 0 \quad \forall v \in X_h\}$ et γ constante indépendante de h . M_h et X_h sont des sous-espaces vectoriels qui approximent M et X respectivement. Dans notre cas, on va utiliser une méthode d'éléments finis mixte. L'idée est de considérer des sous-espaces vectoriels de dimension finie de $\mathbf{IH}_0^1(\Omega)$ et $L_0^2(\Omega)$, dont les éléments sont des fonctions polynomiales par morceaux.

3.6.1 L'élément $\mathbf{P}^1 + \text{Bulle} / \mathbf{P}^1$

Avant de présenter l'élément ' $\mathbf{P}^1 + \text{Bulle} / \mathbf{P}^1$ ' on donne quelques notations.

On note $(T_h)_h$ une famille régulière de décomposition de $\bar{\Omega}$ en triangles. Pour tout $h > 0$ et tout triangle K de T_h , on note h_K son diamètre et $|K|$ son aire et on définit :

$$h = \max_{K \in T_h} h_K \quad (3.20)$$

La méthode choisie est la méthode 'P¹+Bulle/P¹' où 'mini-élément', introduite par Arnold-Brezzi-Fortin (*A. Ern et al*, 2002 ; *O. Pironneau*, 1988). La raison de ce choix est d'ordre pratique : ce type d'élément comporte un nombre relativement bas de degrés de liberté tout en donnant une bonne approximation de la solution. Or, l'une des grandes difficultés de la résolution numérique du problème traité, est le coût de calcul, tant en place mémoire qu'en temps C.P.U. Il est donc important de choisir un type d'élément fini qui permettra de traiter raisonnablement ce genre de problème à un coût modéré.

Pour construire les espaces correspondants à cette méthode, on commence par définir les fonctions 'bulles' :

Etant donné un triangle $K \in T_h$ de sommets $a_i^K|_{i=1,3}$, on désigne par $\lambda_i^K|_{i=1,3}$ les coordonnées barycentriques par rapport aux sommets a_i^K de K définies par :

$$\lambda_i^K \in P_1(K), \quad \lambda_i^K(a_j^K) = \delta_{i,j}, \quad i, j = 1, 3 \quad (3.21)$$

où $P_1(K)$ est l'espace des polynômes de degré 1.

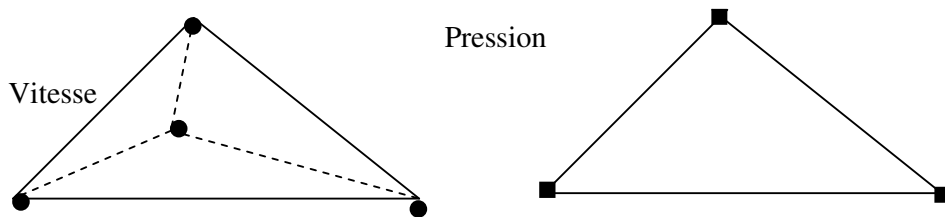


Figure 3.2 : Degrés de liberté pour l'élément P¹+bulle/P¹

La fonction 'bulle' b^K associée à un élément K est définie par

$$b^K = \prod_{i=1}^3 \lambda_i^K \quad (3.22)$$

Il s'agit d'un polynôme de degré 3, qui s'annule sur les arrêtes de K .

On pose :

$$IB_h = \{w_h \in C^0(\bar{\Omega}); \forall K \in T_h, w_h|_K = \zeta b^K, \zeta \in \mathfrak{R}^2\} \quad (3.23)$$

Par construction, on a : $IB_h \subset C^0(\bar{\Omega}) \cap IH_0^1(\Omega)$.

On définit par la suite les espaces de fonctions suivants :

$$\begin{aligned} Y_h &= \{y \in C(\bar{\Omega}); \forall K \in T_h, y|_K \in P^1\} \\ IW_h &= IY_h \cap IH_0^1(\Omega) \\ IX_h &= IW_h \oplus IB_h \\ M_h &= Y_h \cap L_0^2(\Omega) \end{aligned} \quad (3.24)$$

On a :

- * $\dim Y_h = NS$, NS nombre de sommets de T_h
- * $\dim IW_h = 2NSi$, NSi nombre de sommets internes de T_h
- * $\dim IB_h = 2NT$, NT nombre de triangles de T_h
- * $\dim IX_h = 2(NSi + NT)$

3.6.2 Discrétisation spatiale

Soit \bar{u}_h l'approximation de \bar{u} définie par (3.13). La discrétisation spatiale par la méthode des éléments finis du problème (3.17) s'écrit :

$$\begin{cases} \text{Trouver } u_h \text{ dans } \bar{u}_h + IX_h, p_h \text{ dans } M_h \text{ solutions de} \\ a_h(u_h, v_h) + b(v_h, p_h) = F_h(v_h) \quad \forall v_h \in IX_h \\ b(u_h, q_h) = 0 \quad \forall q_h \in M_h \end{cases} \quad (3.25)$$

- $F_h(\cdot)$ est donnée par :

$$F_h(v) = \int_{\Omega} (\rho_h g + f_h) v d\Omega$$

où f_h s'écrit au temps $t = t^{n+1}$

$$f_h^{n+1} = \lambda_h^{n+1} u_h^n \circ X_h^n$$

$X_h^n = X_h(x, t^{n+1}; t^n)$ et λ_h^{n+1} est l'approximation au temps t^{n+1} de λ_h définie par :

$$\lambda_h = \frac{\rho_h}{\Delta t} \quad (3.26)$$

avec ρ_h l'approximation P_1 conforme de ρ .

- $a_h(\cdot, \cdot)$ est définie par

$$a_h(u, v) = \int_{\Omega} \lambda_h u v d\Omega + \mu \int_{\Omega} \nabla \cdot u \nabla \cdot v d\Omega \quad (3.27)$$

- $b(.,.)$ est définie par

$$b(v, p) = - \int_{\Omega} p \operatorname{div} v d\Omega \quad (3.28)$$

Le problème (3.25) est équivalent, comme dans le cas continu au problème suivant :

$$\begin{cases} \text{Trouver } \mathbf{u}_h \text{ dans } IX_h, p_h \text{ dans } M_h \text{ solutions de} \\ a_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \tilde{\mathbf{F}}_h(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in IX_h \\ b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in M_h \end{cases} \quad (3.29)$$

où, $\tilde{\mathbf{F}}_h(\cdot)$ est donnée par

$$\tilde{\mathbf{F}}_h(v) = F_h(v) - \int \lambda_h u_h v d\Omega - \mu \int \nabla u_h \nabla v d\Omega$$

3.7 Méthode de résolution matricielle

Comme on a vu, à chaque pas de temps on a un système de Stokes à résoudre. Mais de point de vue pratique, pour pouvoir traiter de telles équations sur machines, il faut écrire ce système sous forme matricielle. Pour ce faire, soient $\{\phi_i\}_{i=1, \dots, I}$ et $\{\psi_j\}_{j=1, \dots, J}$ une base de X_h et M_h respectivement, avec $I = \dim[X_h]$ et $J = \dim[M_h]$. Alors,

$$\begin{aligned} u_h &= \sum_{i=1}^I u_i \phi_i \\ p_h &= \sum_{j=1}^J p_j \psi_j \end{aligned} \quad (3.30)$$

Dans notre cas, $I = 4$ et $J = 3$.

En injectant ces formulations dans (3.25) on aura sa écriture matricielle :

$$\begin{pmatrix} A_h & B_h \\ B_h^T & 0 \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} = \begin{pmatrix} f_h \\ 0 \end{pmatrix} \quad (3.31)$$

En considérant les définitions suivantes :

$$\begin{cases} A_{h,i,j} = \lambda(\phi_i, \phi_j) + \mu(\nabla \phi_i, \nabla \phi_j) \\ B_{h,i,j} = -(\nabla \cdot \phi_i, \psi_j) \\ f_{hi} = (F_h, \phi_i) \end{cases} \quad (3.32)$$

Le problème de Stokes généralisé discret se ramène donc à résoudre le système linéaire suivant :

Trouver $u = (u_i)_{1 \leq i \leq I}$ et $p = (p_i)_{1 \leq i \leq J}$ tels que

$$\begin{cases} A_h u_h + B_h p_h = f_h \\ B^T u_h = 0 \end{cases} \quad (3.33)$$

où, A_h matrice symétrique définie positive et B_h matrice associée aux gradients de pression.

3.7.1 Algorithme d'Uzawa/Algorithme de gradient conjugué préconditionné

On résout le système matriciel (3.33) par la méthode d'Uzawa. Une forme simple pour la présenter, elle consiste à passer des équations (3.33) à une autre équation équivalente où la seule inconnue est la pression. Pour ce faire, on écrit u_h en fonction de p_h à partir de la première équation de (3.33)

$$u_h = A_h^{-1}(f_h - B_h p_h) \quad (3.34) \text{ et on}$$

utilise cette égalité dans la deuxième équation de (3.33).

On obtient donc :

$$N_h p_h = B_h^T A_h^{-1} f_h \quad (3.35)$$

On démontre que la matrice $N_h = B_h^T A_h^{-1} B_h$ est symétrique et définie positive (*F. Hecht et al., 1991*).

La méthode choisie consiste donc à résoudre le système (3.35) par la méthode du gradient conjugué. Cette résolution présente deux inconvénients : la matrice N_h est pleine et mal conditionnée. Nous signalons d'abord que l'implémentation de la méthode du gradient conjugué peut être faite sans qu'il soit nécessaire de calculer et stocker explicitement la matrice N_h . En effet, la seule opération qu'il faut réaliser avec la matrice N_h pour mettre en œuvre ladite méthode est le produit par un vecteur. Or, étant donné un vecteur Q_h , ce produit peut être calculé sans connaître explicitement les coefficients de N_h .

Pour ce faire, nous procédons en trois étapes :

- 1- Calcul de $B_h Q_h$
- 2- Résolution du système $A_h V_h = B_h Q_h$
- 3- Calcul de $B_h^T V_h$

La seule matrice qui est calculée et stockée est donc A_h .

Afin d'éviter la deuxième difficulté (mauvais conditionnement de la matrice), il faut choisir un préconditionnement efficace du système. Grosso modo, cela revient à résoudre, plutôt que le système (3.35) un nouveau système.

$$C_h^{-1} N_h p_h = C_h^{-1} B_h^T A_h^{-1} \mathbf{f}_h \quad (3.36)$$

où, la matrice C_h doit être choisie de façon à ce que $C_h^{-1} N_h$ soit une matrice mieux conditionnée que M_h .

Intuitivement, l'idée est de choisir C_h « proche » de N_h , ce qui entraîne la résolution d'un système dont la matrice est « proche » de l'identité.

Un bon choix pour C_h a été présenté par Cahouet-Chabart (*M. Abdelwahed*, 2002 ; *F. Hecht et al.*, 1991). On peut présenter ce choix d'une façon simple à partir des opérateurs continus dans un problème de type Stokes généralisé :

$$\begin{cases} \lambda u - \mu \Delta u + \nabla p = F \\ \operatorname{div} u = 0 \end{cases} \quad (3.37)$$

où, λ et μ sont définies dans les paragraphes précédents.

En utilisant la première équation de (3.37), on a :

$$u = A^{-1}(F - \nabla p) \quad (3.38)$$

avec, $A = \lambda I - \mu \Delta$

où, I désigne l'opérateur identité.

En utilisant la deuxième équation de (3.37), on obtient :

$$Np = \nabla \cdot A^{-1} F \quad (3.39)$$

avec, $N = \nabla \cdot A^{-1} \nabla$

Si on considère le domaine infini \mathfrak{R}^2 , tous les opérateurs différentiels intervenant dans la définition pourraient être commutés, ce qui donne :

$$Np = \nabla \cdot [(\lambda I - \mu \Delta)^{-1} \nabla p] = [(\lambda I - \mu \Delta)]^{-1} \Delta p = [\lambda \Delta^{-1} - \mu I]^{-1} p \quad (3.40)$$

Donc :

$$N = [\lambda \Delta^{-1} - \mu I] \quad (3.41)$$

Maintenant, si on discrétise l'opérateur continu (3.41), on obtient :

$$N_h \cong C_h \quad (3.42)$$

où

$$C_h = [\lambda G_h^{-1} - \mu I_h^{-1}]^{-1} \quad (3.43)$$

G_h, I_h , étant respectivement les matrices associées aux produits scalaires

$$(\nabla p_h, \nabla q_h)_{0,\Omega} \quad \forall p_h, q_h \in M_h$$

et

$$(p_h, q_h)_{0,\Omega} \quad \forall p_h, q_h \in M_h$$

Finalement, nous rappelons l'algorithme du gradient conjugué pour la résolution du système (3.36), quand la matrice C_h donnée par (3.43) peut être construite.

- Initialisation

- Choisir $p_{h,0}$
- Calculer \mathbf{u}_{h0} , solution de

$$A_h \mathbf{u}_{h0} = F_h - B_h p_{h0}$$

- $\mathbf{r}_{0h} = B_h^T \mathbf{u}_{h0}$
- Calculer z_{h0} solution de

$$C_h z_{h0} = \mathbf{r}_{0h}$$

- $\mathbf{w}_{h0} = z_{h0}$

- Etape n+1 $\mathbf{u}_{h,n}, p_{h,n}, \mathbf{r}_{h,n}, \mathbf{w}_{h,n}$ étant calculé :

- Calculer $X_{h,n}$ solution de :

$$A_h X_{h,n} = -B_h \mathbf{w}_{h,n}$$

- $\beta_n = (z_{h,n}, \mathbf{r}_{h,n}) / (B_h^T X_{h,n}, z_{h,n})$

- $p_{h,n+1} = p_{h,n} + \beta_n x_{h,n}$

- $\mathbf{r}_{h,n+1} = \mathbf{r}_{h,n} - \beta_n B_h^T X_{h,n}$

- Calculer $z_{h,n+1}$ solution de $C_h z_{h,n+1} = \mathbf{r}_{h,n+1}$

- $\gamma_n = (\mathbf{r}_{h,n+1}, z_{h,n+1}) / (\mathbf{r}_{h,n}, z_{h,n})$

- $\mathbf{w}_{h,n+1} = z_{h,n+1} + \gamma_n \mathbf{w}_{h,n}$

où (\cdot, \cdot) désigne le produit scalaire.

Remarque. La résolution du système $C_h z_{h,n+1} = \mathbf{r}_{h,n+1}$ est faite en deux étapes :

- Résolution du système $G_g z_{h,n} = \mathbf{r}_{h,n}$ qui représente une discrétisation du système

$$-\Delta z_{h,n} = r_{h,n} \quad \text{dans } \Omega$$

$$\frac{\partial z_{h,n}}{\partial n} = 0 \quad \text{sur } \Gamma$$

- Calcul de $z_{h,n}$ par

$$z_{h,n} = \delta z_{h,n} + v \tilde{I}_h r_{h,n}$$

où \tilde{I}_h est la matrice diagonale issue du « mass lumping ».

3.7.2 La méthode de gradient conjugué préconditionné (GCP)

A chaque itération de l'algorithme d'Uzawa, il faut résoudre trois systèmes linéaires : les deux premiers, donnés par $A_h X_{h,n} = -B_h w_{h,n}$, correspondent à la discrétisation de l'opérateur elliptique de second ordre et le second correspond au préconditionneur donné par $C_h z_{h,n+1} = r_{h,n+1}$. Il sera à ce niveau là, les parties les plus coûteuses du code. Les deux types de systèmes linéaires seront résolus à leur tour par la méthode du gradient conjugué préconditionné. Rappelons brièvement l'algorithme GCP pour la résolution de système linéaire $Ax = b$ avec A une matrice carrée, de taille $n \times n$, est symétrique définie positive.

- Initialisation

$$x_0 \in \mathfrak{R}^n \text{ quelconque}$$

$$r_0 = b - Ax_0$$

$$z_0 = C^{-1} r_0$$

$$d_0 = z_0$$

Itérations : pour $k = 0, 1, \dots$, faire

a) approximation de la solution

$$\alpha_k = (r_k, z_k) / (d_k, Ad_k)$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$r_{k+1} = r_k - \alpha_k Ad_k$$

b) préconditionnement

$$z_{k+1} = C^{-1} r_{k+1}$$

c) calcul de la nouvelle direction

$$\beta_{k+1} = (z_{k+1}, r_{k+1}) / (z_k, r_k)$$

$$d_{k+1} = z_{k+1} + \beta_{k+1} d_k$$

Fin

où, C est la matrice de conditionnement.

On peut citer comme préconditionneur : le préconditionneur diagonale et le préconditionneur de Cholesky incomplet:

- Le préconditionneur diagonale c'est le plus simple à utiliser : $C = D$ où, D est la matrice diagonale de la matrice A ($D_{ij} = A_{ii} \delta_{ij}$).
- Le préconditionneur de Cholesky incomplet : $C = L^* (L^*)^T$, L^* factorisée de Cholesky incomplète de A (on garde la même structure creuse de A pour la matrice L).

3.8 Algorithme de résolution

Les étapes de l'algorithme du code **u - p** sont les suivants :

a. Initialisation

On commence par la lecture des données utilisées pour le calcul (maillage, paramètres liés au problème traité). Ensuite, on procède à la construction des tables géométriques (pointeurs ligne et colonne utilisés pour le stockage). Enfin, on calcule les matrices du système à résoudre (masse, rigidité, ...).

b. Transport

Transport de la vitesse par la méthode des caractéristiques.

c. Second membre

Calcul du second membre qui dépend des grandeurs transportées.

d. Conditions aux limites

Imposition des conditions aux limites sur la matrice et le second membre.

e. Résolution

On résout à chaque pas du temps un ensemble des systèmes linéaire par la méthode d'Uzawa.

f. calcul des normes de contrôle

on calcul les normes qui sont utilisées pour tester la convergence du code.

g. test d'arrêt

le programme s'arrête dès la convergence ou dès qu'il finit le nombre maximal d'itération autorisé.

h. post- traitement

stockage des solutions calculées dans des fichiers pour d'éventuelles applications graphiques.

3.9 Application

Des essais numériques ont été faits sur le code parallèle tourné sur un seul processeur. Ces essais numériques sont faits sur une coupe 2D du barrage de Bouregreg. Ce barrage appelé aussi barrage de Sidi Mohamed Ben Abdellah (SMBA), est situé dans la zone de Rabat (Maroc), il a été mis sous exploitation en 1974. Sa destination principale est l'approvisionnement en eau de la zone côtière entre les villes de Salé et Casablanca, ce qui représente environ 20% de la population totale du Maroc. Le volume de ce barrage se situe entre 480-1000 millions m^3 , la surface mouillée est estimée à 32 km^2 , la profondeur moyenne est de 17 m, la profondeur maximale est de 42 m et la capacité de production du barrage est de 9 m^3/s . Le lac est alimenté par un bassin versant de 9400 km^2 ayant un rapport d'écoulement moyen de 21,4 m^3/s . En fait, ce bassin est composé de trois (03) sous bassins, ayant chacun un fleuve ou une rivière principale.

Des analyses faites dans ce barrage ont montré une moyenne de 60 millimètres/l de phosphore et 54 à 60 millimètres/l de concentration maximale de la chlorophylle. Selon les normes de l'OCDE (Office de Coopération et de Développement Economique), le barrage est eutrophe. Depuis 1975, le laboratoire central de qualité de l'eau de l'ONEP (Office Nationale de l'Eau Potable, Maroc) a observé l'épuisement de l'oxygène dissous dans l'hypolimnion avec une stratification thermique pendant l'été. L'ONEP a testé plusieurs procédures pour combattre cette eutrophisation. La conclusion est que le procédé d'aération s'est avéré selon l'ONEP comme étant le plus efficace et économique. Cette solution adoptée à partir de 1992, consiste à injecter l'oxygène par une source d'air comprimé. La section 2D est définie à partir de la bathymétrie du lac dans la direction dominante du vent, passant par l'axe symétrique d'un injecteur de bulle d'air (Figure 3.3). Les dimensions de cette coupe sont : 512 m de longueur et 32 m de hauteur maximale, l'injecteur est de longueur 10 m, il est situé à 15 m de hauteur, il comporte 100 trous et la distance entre les trous est de 10 cm.

Les simulations numériques effectuées sont basées sur les hypothèses suivantes :

- Vitesse constante du vent (1 m/s),
- Condition d'adhérence sur le fond de la section,
- Vitesse d'injection constante (20 m/s),
- Position fixe de l'injecteur (Figure 3-4)

Le maillage considéré prend en compte la variabilité des paramètres au voisinage de l'injecteur (Figure 3-5). Il est constitué de 10791 nœuds et 20760 éléments ce qui

représente 32373 inconnues.

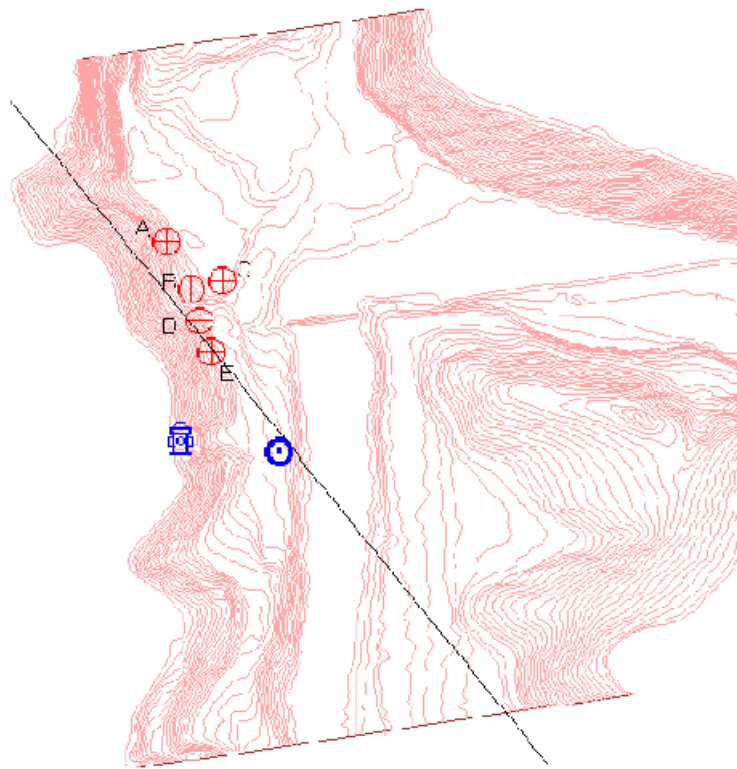


Figure 3-3 : Position de la coupe 2D sur la bathymétrie



Figure 3-4 : Position de l'injecteur et directions des vecteurs vitesses



Figure 3-5 : Maillage de la coupe, 10791 nœuds et 20760 éléments

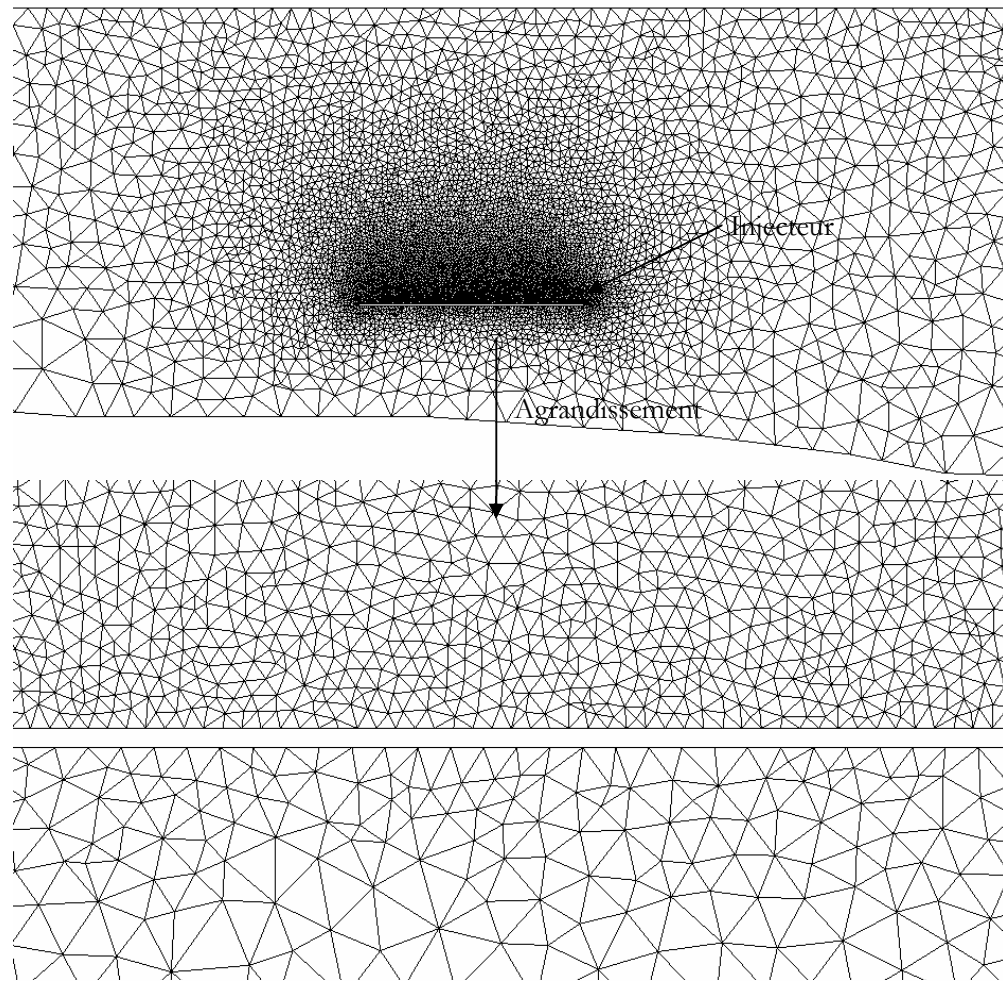


Figure 3-6 : Zooms au voisinage de l'injecteur

3.9.1 Simulations

Plusieurs tests numériques ont été conduits en utilisant le code parallèle sur un seul processeur. Au temps $t=0$, le lac est soumis seulement à l'action du vent.

Les figures (3-7) représentent la vitesse dans le lac aux instants $t=1s$, $180s$ et $378s$.

Elles montrent le développement de l'effet de l'injection des bulles d'air dans le lac.

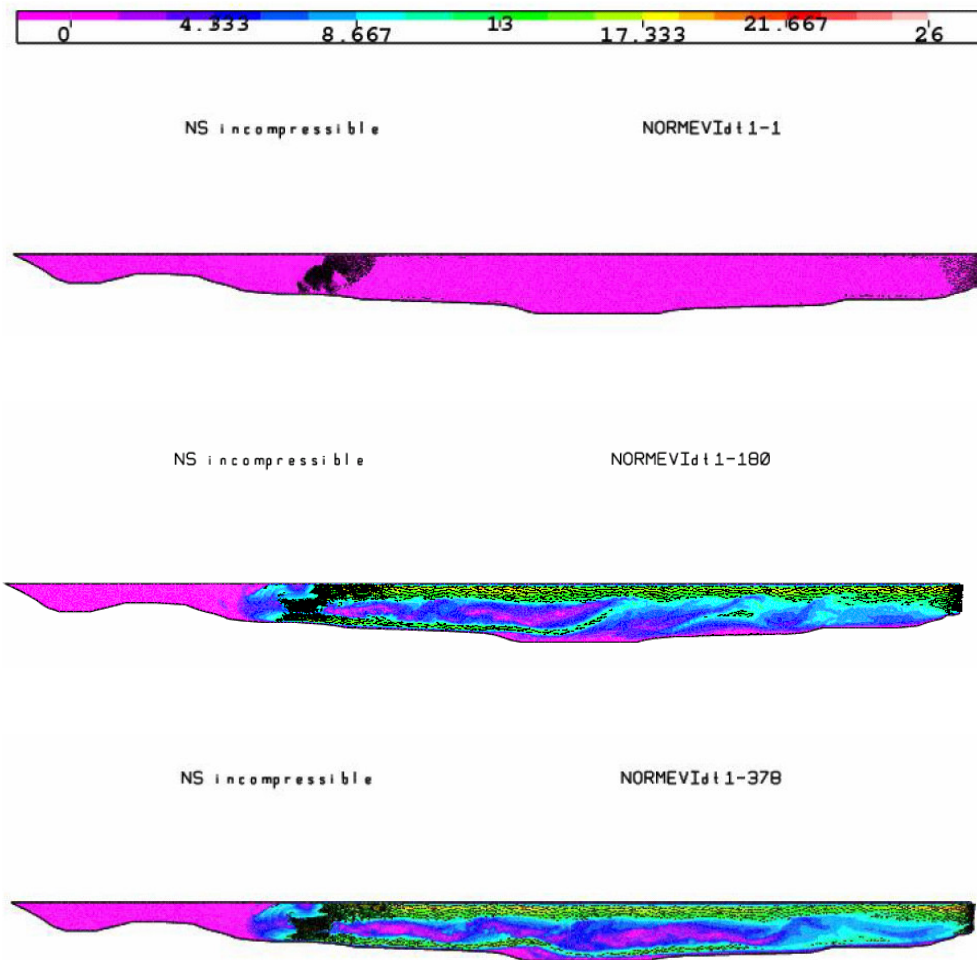


Figure 3-7 : Représentation des isovaleurs de la norme de la vitesse et des vecteurs vitesses aux instants $t=1s$, $2s$ et $3s$.

3.10 Conclusion

D'après les figures (3.7), on voit clairement les zones brassées et les zones qui restent immobile. Ainsi, le modèle monophasique incompressible est suffisant si on considère que le mouvement dans le lac -créé par l'injecteur- comme critère de comparaison (brassage). La simulation numérique peut nous donner donc, une idée sur l'emplacement adéquat de l'injecteur et aussi le nombre nécessaire pour balayer tout le lac.

Chapitre 4

Analyse de code NS incompressible et parallélisation avec MPI

4.1 Introduction

Les modèles présentés dans les chapitres précédents, qui concernent la modélisation du processus de l'aération des retenues d'eau, sont basés essentiellement sur les équations de Navier – Stokes semi compressible. Dans ce travail, le parallélisme a été restreint au modèle Navier – Stokes incompressible. En effet, la masse volumique de l'eau dans le lac varie mais faiblement : c'est-à-dire, si on adopte le modèle incompressible, on reste dans une approximation raisonnable de la dynamique d'eau.

La résolution théorique des deux modèles cités est presque la même ; mais l'implémentation sur machine requiert quelques différences dans les détails. Par contre, si on veut une parallélisation efficace des deux codes, l'approche est fondamentalement différente. Elle se complique dans le cas semi compressible, elle nécessite un remodelage total du code. Dans ce mémoire, la parallélisation est limitée au modèle incompressible 2D. Les dimensions des retenues d'eaux (de quelque centaine de mètres à quelque kilomètre), et la nature des équations Navier - Stokes, exigent un temps de simulation très grand et un espace mémoire très important. Cela rend nécessaire la parallélisation des codes, afin de pouvoir traiter des cas d'applications réels.

Une brève introduction à la bibliothèque MPI a été incluse dans ce chapitre ainsi qu'un aperçu sur les différents types de machine parallèle. Une analyse du temps de code était nécessaire pour déterminer les parties les plus consommatrices en temps de calcul. Après, nous avons mis l'accent sur l'algorithmique parallèle utiliser. En fin de ce chapitre, nous avons présenté des résultats des cas test pour montrer l'efficacité du calcul parallèle.

4.2 Position du problème

La modélisation du processus de l'aération reste physiquement raisonnable si on adopte un modèle Navier –Stokes incompressible. Une telle approche est basée sur la modélisation de la dynamique créée dans le lac par l'effet des bulles. L'injection des bulles est remplacée par des conditions aux limites sur la vitesse au niveau de l'injecteur. Le système à résoudre est :

Trouver u et p solution de :

$$\begin{cases} \rho \left(\frac{\partial u}{\partial t} + (u \nabla)u \right) - \mu \Delta u + \nabla p = F & \text{dans } Q_T \\ \operatorname{div} \Theta & \text{dans } Q_T \\ u = u_d & \text{sur } \Sigma_T \\ u(., t = 0) = u^0(.) & \text{dans } \Omega \end{cases} \quad (4.1)$$

4.3 Présentation du calcul parallèle

Sans rajouter encore une définition du parallélisme, ce qui n'est ni notre but ni notre vocation, nous tentons ici à donner une description, certes incomplète mais admissible, du calcul parallèle, suivie d'une revue sur l'état de l'art en la matière. Ainsi le calcul parallèle, ou plus généralement, le calcul distribué peut être décrit comme une action faisant coopérer plusieurs tâches, s'exécutant généralement sur des processeurs différents, afin de résoudre numériquement un algorithme donné. Le développement du Calcul Parallèle s'est accompagné certes par l'émergence et la conception d'architecture parallèle, mais l'algorithmique parallèle est devenue une phase prépondérante dans la préparation de l'ossature d'un nouveau code numérique ou dans la parallélisation d'un code existant. Avec l'implémentation de MPI/PVM, les programmeurs disposent maintenant d'une norme quasi standard de programmation qui permet une quasi-portabilité des codes d'une machine parallèle virtuelle ou non à une autre.

Avant de voir les possibilités du récent environnement MPI, il est bon de rappeler les différentes architectures existantes ; leur classification est basée sur la façon de manipuler les instructions et le flot de données. Ainsi, nous pouvons distinguer quatre principales classes.

4.3.1 Classification des ordinateurs

- SISD : « Single Instruction Single Data » Ce sont les machines conventionnelles qui contiennent 1 CPU exécutant en série un flot d'instructions ; actuellement, il y a de gros mainframes ayant plusieurs CPU mais chacun exécute indépendamment du reste ces instructions. Ces configurations doivent être vues comme un ensemble de machines SISD saisissant des espaces de données différents (Bull DPX 5000, Control Data 4000, Workstation DEC-HP-IBM-SUN etc...)
- SIMD « Single Instruction Multiple Data » : Ces machines contiennent souvent un grand nombre de processeurs allant de 1024 à 16384. Elles exécutent la même instruction sur des jeux de données différents en mode *lockstep*. Ainsi, une seule instruction agit sur plusieurs instances de données en parallèle. Exemple de machines SIMD : CPP-DAP Gamma et la MasPar MP-2. Une

sous-classe des machines SIMD concerne celles ayant 1 processeur vectoriel. Celui-ci, ayant une structure CPU spéciale, agit sur des tableaux de donnée. Lorsque les données sont manipulées comme des vecteurs, les résultats sont délivrés à un taux de un, deux et jusqu'à trois dans certains cas par cycle d'horloge. Dans ce cas, ils sont plusieurs fois plus rapides que lorsqu'ils travaillent en mode scalaire. Parmi ces machines, on peut citer entre autres : Convex C410 et Hitachi S3600.

- MISD : « Multiple Instruction Single Data » Théoriquement, dans ce type de machines, plusieurs instructions devraient agir sur le même flot de données. Mais aucune machine de cette classe n'a pu être construite jusqu'à ce jour à cause de la difficulté de conception d'un tel système.
- MIMD : « Multiple Instructions Multiple Data » Ces machines exécutent plusieurs instructions différentes en parallèle sur un ensemble de données différentes. La différence en comparaison avec les multiprocesseurs SISD est dû au fait que les instructions et les données représentent des parties différentes de la même tâche à exécuter. Donc, les systèmes MIMD peuvent exécuter plusieurs sous – tâches en parallèle dans le but de réduire le temps de restitution de la tâche principale.

Dans la suite, nous allons nous pencher particulièrement sur cette dernière classe des machines MIMD, très répandues dans les milieux de recherche et d'industrie.

Il existe une large variété de système MIMD, et spécialement pour ce type d'architecture, la taxinomie Flynn (1972) s'est avérée incomplète : des systèmes qui ont des comportements différents se trouvent dans la même classe. Par exemple, une machine Cray Y-MP T94 à quatre processeurs et une machine nCube qui possède des milliers de processeurs appartenant à cette classe. De ce fait, on introduira dans ce qui suit un autre critère pour différencier les architectures, à savoir le type de la mémoire principale des données. Celle-ci peut être de deux types partagée (*Shared Memory (SM)*), ou distribuée (*Distributed Memory (DM)*).

- SM : Tous les processeurs se partagent de manière égale l'accès aux adresses de la mémoire globale i.e. l'utilisateur n'a pas à s'occuper de la localisation ou du stockage d'une donnée. Les systèmes SM peuvent être des deux classes SIMD et MIMD. Les processeurs vectoriels à un seul CPU peuvent être vus comme SM-SIMD, tandis que les multi – processeurs sont considérés des SM – MIMD.
- DM : Dans les systèmes à mémoire distribuée, chaque CPU possède sa propre mémoire. Les CPUs sont connectés d'une certaine manière et peuvent échanger des données entre leur mémoire respective par le biais de bibliothèque *passage de message*. Ainsi l'utilisateur doit contrôler la localité d'une donnée dans la mémoire locale et la communiquer ou la distribuer explicitement. Pour les DM-MIMD, on distingue deux classes : ceux dont les processeurs sont interconnectés suivant une topologie fixe, et ceux dont la topologie est flexible et peut varier d'une tâche à l'autre.

Historiquement, les SIMD sont DM et les MIMD sont SM mais aujourd'hui une grande partie des MIMD sont DM. Même si la différence entre les DM et les SM paraît claire, ce n'est pas toujours le cas du point de vue utilisateur. Par exemple, les systèmes de Kendall Square Research utilisent l'idée de la mémoire partagée virtuelle (*Virtual Shared Memory (VSM)*) au niveau du matériel (*Hardware*). Les VSM peuvent être simulées au niveau logiciel (*Software*). Le langage de programmation *High Performance Fortran (HPF)* distribue les données sur les processeurs disponibles pour une tâche à travers des directives du compilateur Fortran. Ainsi, la machine qui est DM physiquement, utilisant *HPF* aura un comportement SM pour l'utilisateur. Autres systèmes *Massively Parallel Processing (MPP)* tels que Convex, Cray et IBM supportent des modèles propriétaires pour simuler le comportement SM sur des machines qui sont physiquement DM. De plus, des logiciels tels que TreadMarks fournissent des environnements SM sur les réseaux de stations de travail.

Par ailleurs, la notion de DM évolue vers le *Distributed Processing (DP)*. Au lieu des systèmes actuels contenant des processeurs identiques, on peut connecter dorénavant des processeurs de type différent via Ethernet ou FDDI et les faire travailler en parallèle sur les tâches d'un même programme. Le concept des DP est identique au DM-MIMD mais il faut des bibliothèques « généralistes » de *passage de message* capables de gérer la communication entre processeurs hétérogènes qui devient naturellement plus lente que celle sur une DM-MIMD. Plusieurs packages existent pour réaliser ce type de *calcul distribué* (terme plus global que le *calcul parallèle*), les plus connus sont le *Parallel Virtual Machine (PVM)*, le *Message Passing Interface (MPI)* et le *Parasoft's Express*.

Le PVM et le MPI, produits non commerciaux, ont été adoptés par Convex, Cray, IBM et Intel dans la phase de transition entre les DP et les MPP sur leurs *Clusters* favoris de processeurs et ils sont disponibles pour une grande partie de système DM-MIMD et même sur des SM-MIMD pour des raisons de compatibilité. En plus, il y a une tendance vers les systèmes SM, par exemple par HIPPI Channels afin d'obtenir des systèmes d'une très grande puissance de calcul. Silicon a fourni déjà des systèmes de ce type, Intel Paragon avec le MP (Multi Processor), et le NEC SX4 ont aussi cette structure. Le Convex Exempler SPP-2000 peut être considéré comme un exemple plus intégré (l'environnement logiciel est plus complet et permet l'adressage SM).

Ainsi le développement de l'informatique répartie permet actuellement, via ses bibliothèques « passage de message » (Message Passing), de résoudre des problèmes de grandes tailles, domaine jusque là réservé aux machines multiprocesseurs (MPP).

Face à des besoins croissants en puissance de calcul, les constructeurs ont conçu de nouvelles générations de machines parallèles ayant pour objectif l'obtention de performance de l'ordre de téraflops. Ces super-calculateurs sont construits autour de processeurs puissants interconnectés par des liens de communication de plus en plus rapides (>200 Mflops).

Les nouvelles architectures mises en place s'orientent vers des modèles de type MIMD où la mémoire est distribuée sur chaque nœud du réseau et où les communications sont gérées par des circuits de routage spécialisés.

Plusieurs types de ces machines ont déjà vu le jour, on peut citer entre autres la Paragon (Intel), la SP2 (IBM), la T3E (Cray), l'Archipel volvox LS – 860, la CS – 2 (CPI) ou la CM – 5 TM. Il est important de noter que ces machines sont fréquemment construites autour de composants standards (1860, Sparc, Risc) et que les nœuds du réseau sont quasiment des ordinateurs à eux seuls (plusieurs Méga octets de mémoire, unité vectorielle, interface réseau, processeur de calcul, processeur de communication,...).

Ces tendances générales sont en grande partie les résultats de l'évolution récente des composants tant au niveau de l'intégrité qu'au niveau des technologies employées ou des matériaux utilisés.

La montée en puissance et en vitesse des processeurs et des systèmes de communication ont aussi largement profité aux réseaux de stations de travail. Il n'est pas rare de nos jours de trouver les mêmes processeurs sur une station de travail que dans une machine parallèle. Certaines stations sont elles – même multi – processeurs et développent des puissances de calcul non négligeables. De plus l'architecture d'un réseau de stations de travail est comparable à celle d'une machine DM – MIMD constituée de processeurs interconnectés par un BUS de communication et possédant leur propre mémoire. Une telle machine offre une puissance potentielle élevée (puissances cumulées) répondant tout à fait aux besoins de certaines applications de calcul lourd. Une approche de type parallélisme virtuel, par le biais de logiciels, permet alors d'exploiter à moindre frais cette puissance et d'utiliser des ressources qui sont la plupart du temps sous – exploitées. Toutes ces raisons laissent entrevoir les réseaux locaux de stations de travail comme une alternative des machines parallèles intrinsèques.

4.4 Problème à résoudre

Malgré les différentes simplifications apportées aux différents modèles proposés ultérieurement (*M. Abdelwahed*, 2002 ; *M. Abdelwahed et al.*, 2002 ; *F. Dabaghi*, 2000), la simulation numérique du processus d'aération sur un cas d'application réel reste toujours limitée et ce même en 2D ; ceci est dû naturellement à la grande taille des systèmes matriciels à résoudre pour les problèmes traités et ce malgré les ressources disponibles très importantes en capacité de calcul et en place mémoire. Prenons par exemple le cas d'une simulation dans une coupe réelle d'un lac (*M. Abdelwahed*, 2002 ; *M. Abdelwahed et al.*, 2004) dont les dimensions moyennes sont de l'ordre de $500m \times 200m$: L'injecteur d'aération de forme cylindrique de diamètre 10 cm et de longueur 1000 cm comporte une centaine de trous de diamètre de 1 cm chacun. Si nous voulons détecter convenablement l'effet de l'injection des bulles, il est nécessaire de faire un maillage avec des éléments de l'ordre du volume de la bulle ce qui donnera un maillage d'une dizaine de millions

de noeuds. Si nous proposons de faire un maillage fin sur une zone proche de l'injecteur et grossier ailleurs, nous restons toujours dans l'ordre de quelques millions de noeuds.

Par exemple, pour un code Navier-Stokes 2D en formulation $\psi - \omega$, un calcul sur un maillage de $6 \cdot 10^5$ noeuds nécessite environ 1h10mn pour une itération en temps sur une machine RISK HP 750Mhz et 2Go de mémoire RAM i.e. aucun effet de pagination. Pour 1 heure de temps de simulation avec un pas de temps d'une minute, il faut 60 itérations en temps soit 70 heures de calcul (≈ 3 jours). Ainsi, il est évident que pour des simulations plus fines en 2D ou grossières en 3D, la parallélisation du code est incontournable surtout si nous ajoutons l'injection de l'air qui peut prendre plusieurs jours pour bien entamer l'effet de l'oxygénation.

Nous présentons dans ce travail une parallélisation d'un modèle 2D tenant compte de l'effet de l'aération par le biais d'une correction des équations de mouvement de la phase eau écrites en formulation vitesse-pression. Nous rappelons brièvement dans la suite le problème à résoudre.

Soit Ω un domaine 2D représentant une coupe du lac et $\Gamma = \Gamma_1 \cup \Gamma_2$ la frontière de Ω avec, $\Gamma_1 = \Gamma_1^1 \cup \Gamma_1^2 \cup \Gamma_1^3$, n le vecteur normal unitaire sur Γ , le temps $T \in \mathbb{R}^*$, $Q_T = \Omega \times [0, T]$, $\Sigma_T^1 = \Gamma_1 \times [0, T]$, $\Sigma_T^2 = \Gamma_2 \times [0, T]$ et $\Sigma(u_d) = \{(x, t) \in \Sigma_T^1 : u_d \cdot n < 0\}$.

Soit : $u_L^0 : \Omega \rightarrow \mathbb{R}^2$ et $u_d : \Sigma_T \rightarrow \mathbb{R}^2$.

$u_L(., t=0) = u_L^0(.)$ dans Ω , $u_L = u_d$ sur Σ_T^1 et $\mu_L \frac{\partial u_L}{\partial n} - p_L n = 0$ en Σ_T^2 .

Trouver : $u_L : Q_T \rightarrow \mathbb{R}^2$ et $p_L : Q_T \rightarrow \mathbb{R}$ tel que :

$$\begin{cases} \rho \left(\frac{\partial u}{\partial t} + (u \nabla) u \right) - \mu \Delta u + \nabla p = F & \text{dans } Q_T \\ \text{div } \Theta & \text{dans } Q_T \end{cases} \quad (4.2)$$

La discrétisation en temps du problème (4.1) utilise la méthode des caractéristiques et implique à chaque étape de temps t_n le problème de type Quasi – Stokes suivant :

$$\begin{cases} \frac{1}{\Delta t} \rho u_L^{n+1} - \mu_L \Delta u_L^{n+1} + \nabla p_L^{n+1} = F^{n+1} & \text{dans } \Omega \\ \text{div} u_L^{n+1} = 0 & \text{dans } \Omega \end{cases} \quad (4.3)$$

où :

$$F^{n+1} = \rho g + \frac{1}{\Delta t} \rho u_L^n \circ X^n \quad (4.4)$$

et dont la formulation variationnelle associée s'écrit :

Pour tout $v \in H_0^1(\Omega)$ et $q \in L_0^2(\Omega)$:

Trouver $u_L \in X^d$ et $p \in L^2(\Omega)$ tel que

$$\begin{cases} \frac{\rho}{\Delta t} \int_{\Omega} u_L \, v \, d\Omega + \mu_L \int_{\Omega} \nabla u \nabla v \, d\Omega - \int_{\Omega} p \nabla \cdot v \, d\Omega = \int_{\Omega} F v \, d\Omega \\ - \int_{\Omega} q \nabla \cdot u_L \, d\Omega = 0 \end{cases} \quad (4.5)$$

avec, $X^d = \{v \in H^1(\Omega)^d, v = u_d \text{ p.p. sur } \Gamma\}$.

On montre que les équations (4.3)- (4.5) ont une solution unique (chapitre 3, paragraphes 2.5 et 2.6). L'approximation spatiale- éléments finis du problème (4.5) conduit au système linéaire suivant :

$$\begin{cases} A_h u_h + B_h p_h = f_h \\ B_h^T u_h = 0 \end{cases} \quad (4.6)$$

On voit donc que à chaque pas de temps la résolution du système (4.1) passe par deux étapes :

- 1- La discrétisation temporelle suivant les caractéristiques et après,
- 2- La résolution du système linéaire (4.5) en utilisant un gradient conjugué d'Uzawa.

Nous montrerons dans les paragraphes qui suivent que la résolution du système linéaire consomme la majorité du temps CPU de code, ce qui nous a amené à paralléliser cette partie en omettant la partie des caractéristiques.

4.5 Analyse du temps de code

Le code Navier – Stokes incompressible 2D est composé en gros de 3 parties :

- 1- Lecture et préparation des données nécessaires au solveur (lecture des constantes, lecture de maillage, calcul des matrices élémentaires ...),
- 2- Calcul des caractéristiques et linéarisation des termes convectifs ainsi que le calcul des termes inhérents au linéarisation des termes de transport ou bien convectifs,
- 3- Le système linéaire obtenu est résolu par un algorithme du gradient conjugué d'Uzawa.

Nous avons bâti notre parallélisme sur le code Navier – Stokes séquentiel déjà existant. Ce qui fait que la détermination des parties parallélisables et leurs poids relatifs en temps d'exécution est déterminante dans le choix de la stratégie de parallélisme.

Ci-après, nous donnons les poids relatifs des différentes étapes (1, 2 et 3) sachant que dans le code Navier – Stokes incompressible les opérations 2 et 3 sont répétées à chaque pas de temps tandis que la première est faite une seule fois.

L'analyse du temps a été faite sur plusieurs cas. Les données de la présentation ci-après sont :

- Le nombre de Reynold, $Re = 1000$
- Le nombre des nœuds approximativement 20000.

4.5.1 Représentation graphique des pourcentages des temps dans le code principal et le solveur (STOBU1)

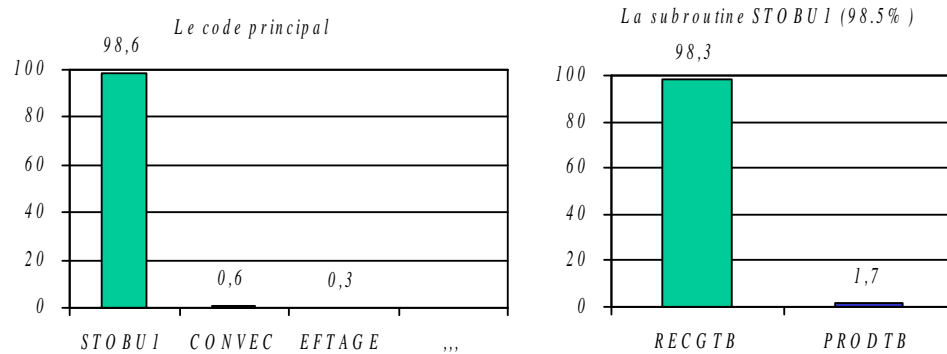


Figure 4-1 : Pourcentages des temps dans les code total et le solveur

4.5.2 Représentation graphique des pourcentages des temps dans la subroutine RECGTB et PRODTB

Dans la figure 4-2, nous présentons le pourcentage de temps dans les sousroutines RECGTB et PRODTB.

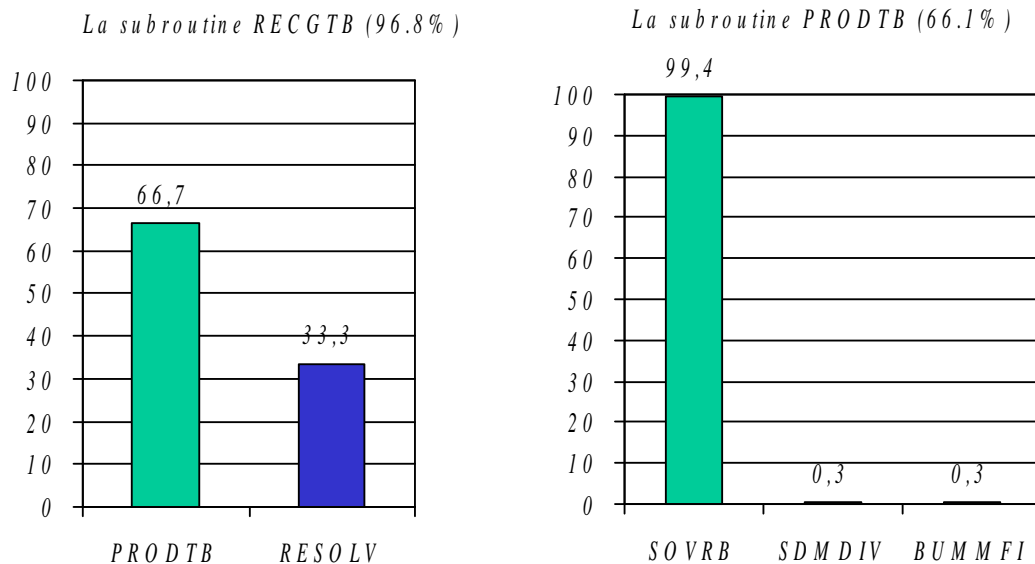


Figure 4-2 : Pourcentages des temps dans les sousroutines RECGTB et PRODTB

4.5.3 Représentation graphique des pourcentages des temps dans la subroutine SOVRB et ICHR1R

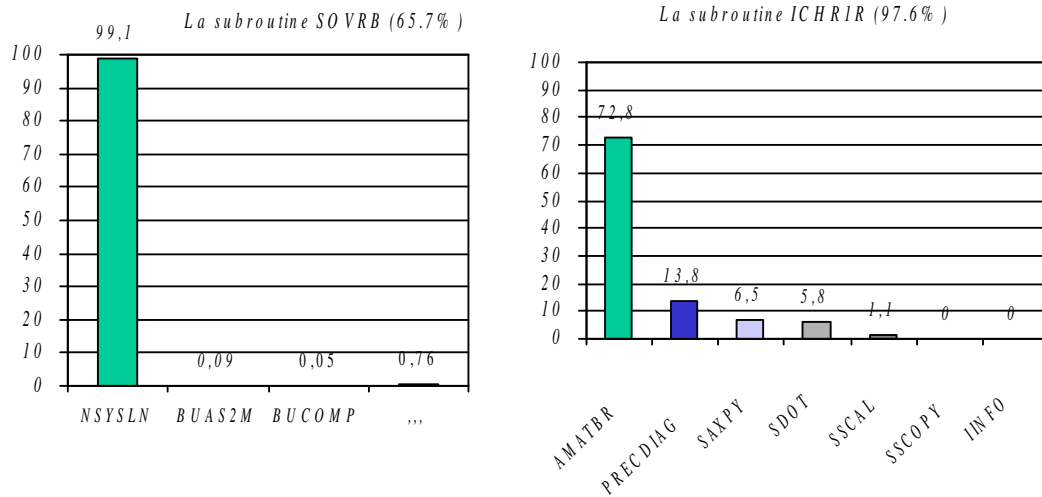


Figure 4-3 : Pourcentages des temps dans les subroutines SOVRB et ICHR1R

La procédure NSYSLN est constituée 100% de la procédure RESOLV, cette dernière est constituée pratiquement de la procédure ICHR1R, ce qui fait que l'analyse de la subroutine NSYSLN revient à analyser la subroutine ICHR1R.

4.6 Arborecence des appels qui consomme la majorité du temps

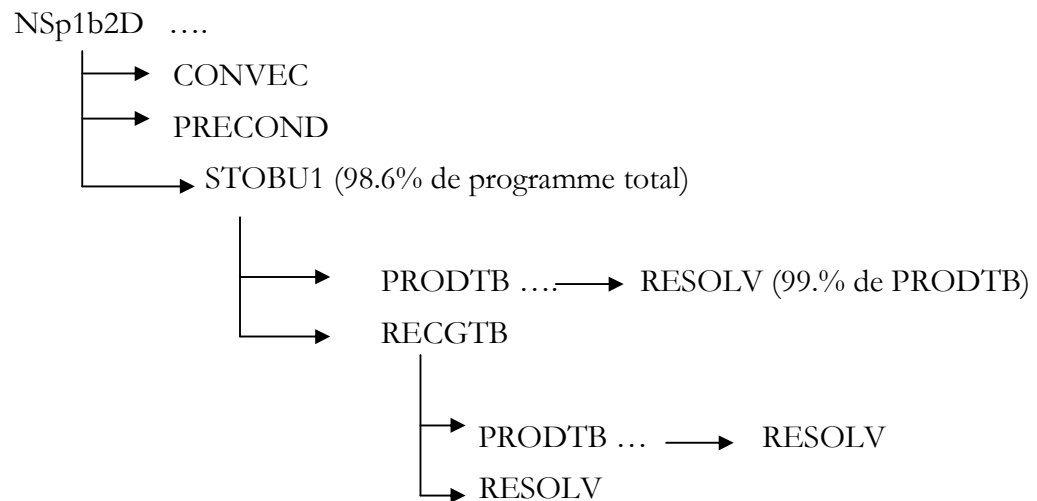


Figure 4-4 : Structure globale du code

Une analyse sur différents cas test (maillages variant entre 10000 et 50000 noeuds), a montré que cette partie solveur, consomme toujours plus de 90% en temps CPU. Le solveur utilisé est basé essentiellement sur l'algorithme d'Uzawa. Ce dernier est constitué principalement par le gradient conjugué préconditionné classique.

Cette subroutine de gradient conjugué préconditionné classique, consomme plus de 90% en temps CPU.

Il est alors raisonnable d'investir dans la parallélisation de gradient conjugué préconditionné classique.

En effet, la résolution du système linéaire (4.6) est donnée par l'algorithme du gradient conjugué suivant où les parties parallélisées sont mises en gras italique:

- Initialisation
 - Choisir $p_{b,0}$
 - Calculer $u_{b,0}$, solution de

$$\mathbf{A}_h \mathbf{u}_{b,0} = \mathbf{f}_h - \mathbf{B}_h \mathbf{p}_{b,0}$$
 - $r_{b,0} = \mathbf{B}_b^T u_{b,0}$
 - Calculer $z_{b,0}$ solution de

$$\mathbf{C}_h \mathbf{z}_{b,0} = \mathbf{r}_{b,0}$$
 - $y_{b,0} = z_{b,0}$
- Etape $n+1$: $p_{b,n}$, $r_{b,n}$, $y_{b,n}$ étant calculées,
 - Calculer $x_{b,n}$ solution de :

$$\mathbf{A}_h \mathbf{x}_{b,n} = \mathbf{B}_h \mathbf{y}_{b,n}$$
 - $\beta_n = (\mathbf{z}_{b,n}, \mathbf{r}_{b,n}) / (\mathbf{B}_b^T \mathbf{x}_{b,n}, \mathbf{z}_{b,n})$
 - $p_{b,n+1} = p_{b,n} + \beta_n \mathbf{x}_{b,n}$
 - $r_{b,n+1} = r_{b,n} - \beta_n \mathbf{B}_b^T \mathbf{x}_{b,n}$
 - Calculer $z_{b,n+1}$ solution de

$$\mathbf{C}_h \mathbf{z}_{b,n+1} = \mathbf{r}_{b,n+1}$$
 - $\gamma_n = (\mathbf{r}_{b,n}, \mathbf{z}_{b,n}) / (\mathbf{r}_{b,n}, \mathbf{z}_{b,n})$
 - $y_{b,n+1} = z_{b,n+1} + \gamma_n y_{b,n}$

Figure 4.5 : Algorithme du gradient conjugué d'Uzawa (les parties parallélisées sont montrées en gras italique)

4.7 Parallélisation

Le parallélisme est fait à l'aide de la bibliothèque MPI (Message Passing Interface). Avant de commencer la présentation de l'algorithmique utiliser, nous donnons un petit aperçu sur les différentes bibliothèques et langage dédiés spécialement au calcul parallèle en mettant l'accent sur le MPI.

4.7.1 Quelques bibliothèques de communications

Une bibliothèque de communication regroupe un ensemble de fonctions permettant d'envoyer des messages entre processeurs soit d'une même machine ou interconnectés sur un réseau. Nous donnons ci-après une brève description des trois bibliothèques (Open MP, HPF, PVM) ; la description de MPI est donnée après ce paragraphe.

a. *Open MP*

Open MP est une implémentation propriétaire de SCI. Open MP est un ensemble d'indications que l'on donne au compilateur C ou Fortran pour lui indiquer quelles parties du code peuvent être exécutées en parallèle. Le compilateur se charge ensuite de la parallélisation. Open MP n'est pas seulement une bibliothèque de communication, mais permet également de paralléliser facilement des programmes.

b. *HPF*

HPF (High Performance Fortran) est une extension data – parallèle du Fortran 90, qui offre des directives de compilation. Ces directives permettent au programmeur de donner au compilateur des indications sur le programme, des stratégies d'implantations (alignement et distribution des données).

c. *PVM*

PVM permet d'utiliser un réseau de stations de travail hétérogènes comme une machine parallèle MIMD. PVM est une bibliothèque de communication qui gère la transcription des données entre les différentes architectures. PVM permet de créer dynamiquement des processus.

4.7.2 MPI – Message Passing Interface

MPI, acronyme de *Message Passing Interface*, est un standard d'échange de messages sur machines à mémoire distribuées. L'environnement de passage de message est l'outil le plus utilisé pour la programmation des machines parallèles à mémoire distribuée. Cela revient, d'une part, à la simplicité de ce modèle puisque c'est la manière la plus proche de ce qui se passe réellement dans ces machines et d'autre part, à son implantation efficace sur une grande variété d'ordinateurs. En outre, les commandes utilisées se font grâce à des procédures stockées dans une librairie accessible depuis un code Fortran ou C, ce qui reste familier au programmeur.

Ce standard est implémenté de diverses manières, soit par des implémentations gratuites comme *mpich*, soit par des implémentations propriétaires, qui seront en général optimisées pour une architecture. Une application MPI est un ensemble de

processus autonomes, qui exécutent chacun leur code et communiquent par des appels à des sous-programmes de la bibliothèque MPI.

Les atouts de MPI sont :

- Spécifier une interface de programmation portable sur différentes plateformes,
- Permettre des communications efficaces et fiables (éviter la recopie mémoire, permettre le recouvrement du calcul et des communications),
- Permettre l'utilisation d'environnements hétérogènes,
- Offre une bibliothèque en C et Fortran,

De plus, MPI essaie de coller aux interfaces existantes (PVM, P4, NX, EXPRESS, etc) de telle sorte que certaines implémentations peuvent être vues comme un sur – ensemble d'autres systèmes notamment PVM. Les sous paragraphes qui suivent donne une petite introduction au MPI :

a. Fonctions de gestion de l'environnement

Un exécutable MPI doit obligatoirement contenir les trois éléments suivants :

- inclusion du fichier mpif.h,
- appel à la procédure `Mpi_Init`, avant tout autre appel à une procédure MPI,
- appel à la procédure `Mpi_Finalize`, après tout autre appel à une procédure MPI.

Ces trois éléments n'ont aucune action particulière sur la parallélisation effective de l'application, mais ils sont nécessaires.

`MPI_INIT()` permet d'initialiser l'environnement MPI

`integer, intent(out) :: code`

`call MPI_INIT(code)`

`MPI_FINALIZE()` permet de désactiver cet environnement :

`integer, intent(out) :: code`

`call MPI_FINALIZE(code)`

`MPI_INIT` initialise le communicateur par défaut, `MPI_COMM_WORLD` qui contient tous les processus actifs.

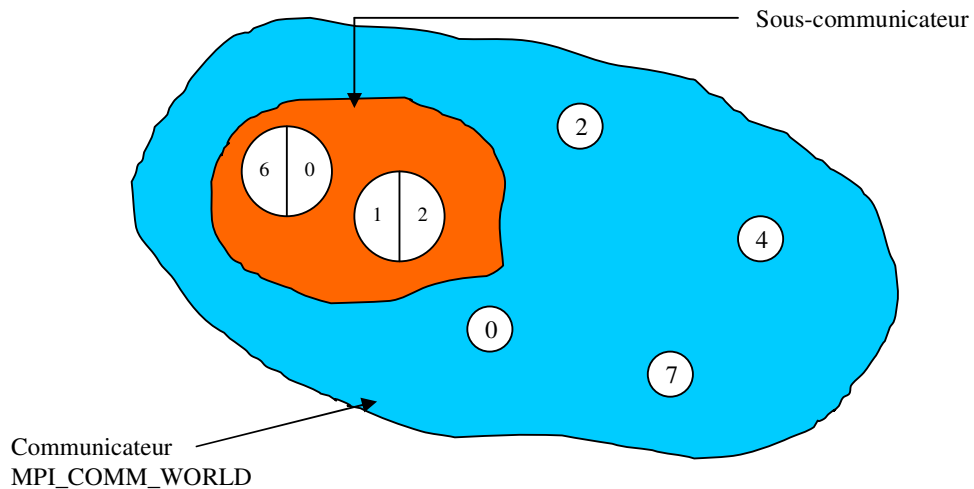


Figure 4-6 : Schématisation des communicateurs

b. Informations de base

Il est à noter que toutes les opérations effectuées par MPI portent sur des communicateurs.

- `MPI_Comm_Size` : Renvoie le nombre de processus dans un communicateur. En lui spécifiant le communicateur `MPI_COMM_WORLD` comme paramètre, elle permet de connaître le nombre total des processus s'exécutant dans l'environnement MPI.
- `MPI_Comm_rank` : Permet au processus appelant de connaître son rang dans le communicateur dont il fait partie.

c. Compilation et exécution de processus en MPI

MPI est une librairie C portable permettant d'échanger des messages entre processus repartis sur un ensemble de processeurs.

Un processus commence toujours par définir un monde commun qui contient l'ensemble des processus du programme parallèle.

La compilation d'un code se fait à l'aide de la commande `mpij77` ou `mpij90 nom` et la compilation se fait par la commande `mpirun -np (nb processus) nom de l'exécutable`.

d. Communicateurs et communications point à point

Les processus composant notre application parallèle sont regroupés dans des communicateurs. Nous pouvons dire donc qu'un communicateur est un ensemble de processus qui peuvent communiquer entre eux. Il existe par défaut un communicateur, appelé `Mpi_Comm_World`, contenant l'ensemble des processus.

Chaque processeur possède un rang unique dans la machine. Les N processeurs sont numérotés de 0 à N-1. Un processus peut appartenir à plusieurs communicateurs avec un rang différent.

Ces quelques notions nous permettent de comprendre l'exemple traditionnel suivant :

```
Program Bonjour
Include 'mpif.h'
Integer rang
Integer ierr
Call Mpi_Init(ierr)
Call Mpi_Comm_Rank(Mpi_Comm_World,rang,ierr)
Print *,'Bonjour, ici le processus #',rang
Call Mpi_Finalize(ierr)
End
```

L'exécution de 3 processus comme celui-ci sur 4 processeurs produit le résultat suivant :

```
Bonjour, ici le processus # 0
Bonjour, ici le processus # 1
Bonjour, ici le processus # 2
```

Quand la communication se fait entre deux processus, on dit qu'on a une communication point à point. Le processus qui envoie le message c'est l'émetteur et qu'il le reçoit c'est le récepteur. L'émetteur et le récepteur sont identifiés par leur rang dans le communicateur au sein duquel se fait la communication. Les procédures standard d'envoi et de réception de message point à point sont : MPI_SEND et MPI_RECV. Elles s'utilisent comme suit :

Pour l'envoi, nous avons ; MPI_SEND (valeur, taille, type, destination, étiquette, comm, code)

Cela signifie qu'un message composé de la variable de la valeur « valeur », de la taille « taille », de type « type », portant l'identifiant « étiquette », est adressé au processus portant le numéro « destination » dans le communicateur « comm ».

Et celle de la réception ;

MPI_RECV (valeur, taille, type, source, étiquette, comm, statut, code)

Ce qui signifie qu'un message de la taille « taille », de type « type », portant l'identifiant « étiquette », provenant du processus portant le numéro « source » dans le communicateur « comm », sera stocké dans la variable « valeur ».

La variable erreur contient un code d'erreur, et la variable status indique l'état de la réception du message.

Voici un exemple d'utilisation d'envoi et de réception par deux processus :

```
Program Bonjour
Include 'mpif.h'
Integer code
Integer message
Integer rang
Integer recu
Integer status(Mpi_Status_Size)
Integer taille
Call Mpi_Init(code)
Call Mpi_Comm_Rank(Mpi_Comm_World,rang,code)
Call Mpi_Comm_Size(Mpi_Comm_World,taille,code)
If(rang.Eq.0) Then
Print *,'Nombre de processus : ',taille
message = 2
Call Mpi_Send(message,1,Mpi_Integer,1,0,Mpi_Comm_World,code)
Print *,'Bonjour, ici le processus #',rang,
& ', message envoye = ', message
Endif
If(rang.Eq.1) Then
Call Mpi_Recv(recu,1,Mpi_Integer,0,0,Mpi_Comm_World,
& status,code)
Print *,'Bonjour, ici le processus #',rang,
& ', message recu = ',recu
Endif
Call Mpi_Finalize(code)
End
```

L'exécution de ce programme sur deux processeurs produit le résultat suivant :

```
Nombre de processus : 2
Bonjour, ici le processus # 0, message envoye = 2
Bonjour, ici le processus # 1, message recu = 2
```

Ces procédures d'envoi et de réception sont bloquantes, ce qui signifie que l'exécution du code est bloquée jusqu'à ce que l'envoi ou respectivement la réception soit terminée. En outre, il existe des commandes d'envoi et de réception non bloquante. Contrairement aux routines bloquantes, les routines de la communication non bloquante n'attendent pas que l'opération de communication

en cours soit terminée. En général, cela se produit dès l'envoi de message. Le retour de cette fonction au programme appelant se fait avant que la réception soit terminée. Ainsi, il existe une version non bloquante des *send/recv*. *MPI_IRecv(...)* et *MPI_IRecv (...)* avec un paramètre supplémentaire : le *handle*. Celui-ci manipule les requêtes pendantes créées et permet par exemple de s'assurer plus tard que la communication s'est bien passée. Les principales routines utilisées dans ce mode sont les suivantes :

- *MPI_IRecv* : Envoi non bloquant d'un message.
- *MPI_IRecv* : Permet la réception non bloquante d'un message.
- *MPI_Wait* : Permet d'attendre la fin d'une opération.
- *MPI_Test* : Tester une opération non bloquante.
- *MPI_Free_Request* : Libérer une requête avant de la réutiliser.

e. Communications collectives et communicateurs

Les communications collectives permettent de faire en une seule opération une série de communications point à point et elles concernent toujours les processus d'un communicateur. La gestion des étiquettes dans ces communications est transparente et à la charge du système. Elles ne sont donc jamais définies explicitement lors de l'appel à ces sous-programmes. Cela a entre autres pour avantage que les communications collectives n'interfèrent jamais avec les communications point à point.

Il existe trois types de fonctions :

1. Celle qui assure les synchronisations globales entre tous les processus : *MPI_Barrier*,
2. Celles qui ne font que transférer des données, parmi lesquelles nous citons :
 - *MPI_Bcast* : Diffuse un message à tous les membres d'un groupe.
 - *MPI_Gather* : Permet à un processus de collecter des données qui appartiennent aux autres.
3. Celles qui, en plus de la gestion des communications, effectuent des opérations sur les données transférées :
 - *MPI_Reduce* : Effectue une opération de réduction, qu'elle soit d'un type prédéfini (somme, produit, maximum, minimum, etc.) ou d'un type personnel.
 - *MPI_AllReduce* : Effectue une opération de réduction et diffuse le résultat.
 -

4.7.3 Modes de communication

Nous allons examiner maintenant différents modes de communications, présents dans les bibliothèques de communications.

a. Communication synchrone

Dans ce mode de communication, l'émission et la réception du message s'effectuent en même temps. L'algorithme d'échange de message nécessite donc, du point de vue d'un observateur extérieur, il existe un intervalle de temps commun entre les deux processus où sont actives les routines d'envoi et de réception. Les deux processus doivent être prêts à communiquer. Ils risquent donc d'attendre. Cette attente permet d'utiliser dans le code du processus émetteur le fait que après l'exécution de la primitive d'envoi, le récepteur a pris en compte le message. L'émetteur connaît donc l'état du processus destinataire. Par exemple, on peut imaginer un programme où un processus envoie n message à chacun de ses n collègues, où n représente pour chaque collègue son identifiant (en supposant que cet identifiant est un entier). Par exemple, le processus 1 reçoit 1 message, le processus 2 en attendra 2, etc... Après avoir reçu ses n messages, le processus se termine. Dans ce mode synchrone, le processus maître sait à tout moment combien de processus sont présents sur la machine.

b. Communication asynchrone

Dans ce mode, l'émetteur peut envoyer son message dès que celui-ci est prêt, et continuer son code de calcul. Le récepteur prendra en compte le message plus tard, quand il sera arrivé à la réception du message dans le déroulement de son code et que le message lui sera effectivement parvenu. Si le récepteur souhaite recevoir un message alors que celui-ci n'est pas encore arrivé, il se mettra probablement en attente. L'émetteur ne doit faire aucune supposition sur l'état d'un processus. Notons que l'on peut simuler une communication synchrone à partir d'une communication asynchrone si l'envoi de tout message de A vers B est suivi de l'envoi d'un acquittement de B vers A.

- ***Mode bloquant*** : Lors d'une communication asynchrone, les primitives d'envoi ou de réception sont bloquantes si elles ne se terminent que lorsque l'opération correspondante est réellement terminée :
 - opération d'envoi : le contenu du buffer d'émission a été transmis dans son intégralité à la couche inférieure chargé de l'acheminement ;
 - opération de réception : un message complet correspondant à celui que l'on souhaite recevoir est disponible et transféré dans le tampon de réception accessible à l'utilisateur.
- ***Mode non bloquant*** : Les primitives non bloquantes se déroulent en deux temps :
 - initiation de l'opération ;
 - terminaison de l'opération. Il est nécessaire de pouvoir indiquer quelle opération on souhaite terminer. Pour cela, lors d'initiation de l'opération, on récupère souvent un identifiant d'opération.

Ceci permet aux processus de masquer les temps de communication si leur algorithme leur permet d'effectuer des calculs avant de prendre en compte le message ou pendant son envoi. Il faut néanmoins se préoccuper explicitement de la terminaison de l'opération. En général, nous espérons que la routine de terminaison de l'opération n'aura pas à attendre. Dans le cas des processus récepteurs, il existe des routines de test de disponibilité d'un message, bloquantes ou non bloquantes.

4.7.4 Présentation de la HP 9000 V-Class

La HP 9000/800 est une des plates formes les plus rapides fournies par Hewlett-Packard (HP). La machine, également appelée serveur V-Class, est une machine MIMD (multiples instructions, multiples données) et a une architecture de mémoire partagée ([72]).

a. Configuration interne de la machine

La V-Class contient 16 processeurs. Les processeurs et le matériel associés forment ce qui s'appelle généralement le nœud. Le nœud utilise une conception symétrique du multiprocesseur (SMP) qui peut exploiter le parallélisme à grain fin. Un schéma fonctionnel conceptuel de l'architecture interne de la machine est donné dans la figure (4.7). Dans ce schéma, on voit que la machine est synchronisée par plusieurs unités de commande.

- Centralement située dans le diagramme, la barre transversale d'hyperplan de HP qui est composée de quatre contrôleurs de connexion de routage (ERAC). Elle permet à tous les processeurs d'accéder à toute la mémoire disponible.
- Les processeurs sont installés sur les agents contrôleurs des processeurs (EPAC). Un EPAC permet au processeur et au sous-ensemble d'Entrée/Sortie (le contrôleur d'interface EPIC) l'accès à la barre transversale d'hyperplan. Jusqu'à deux processeurs sont situés sur chaque EPAC.
- Egalement reliés à la barre transversale d'hyperplan sont les contrôleurs d'accès mémoire (EMAC).
- Les entrées/sorties se relient au système par l'EPIC qui est reliée aux agents de processeur (EPAC).
- L'utilitaire de noyau dans le nœud (ECUB, généralement appelé panneau d'utilitaire) contient une section du matériel (hardware) appelée logique de noyau. Elle fournit des interruptions à tous les processeurs du système par le bus logique de noyau qui se relie à chaque agent de processeur.

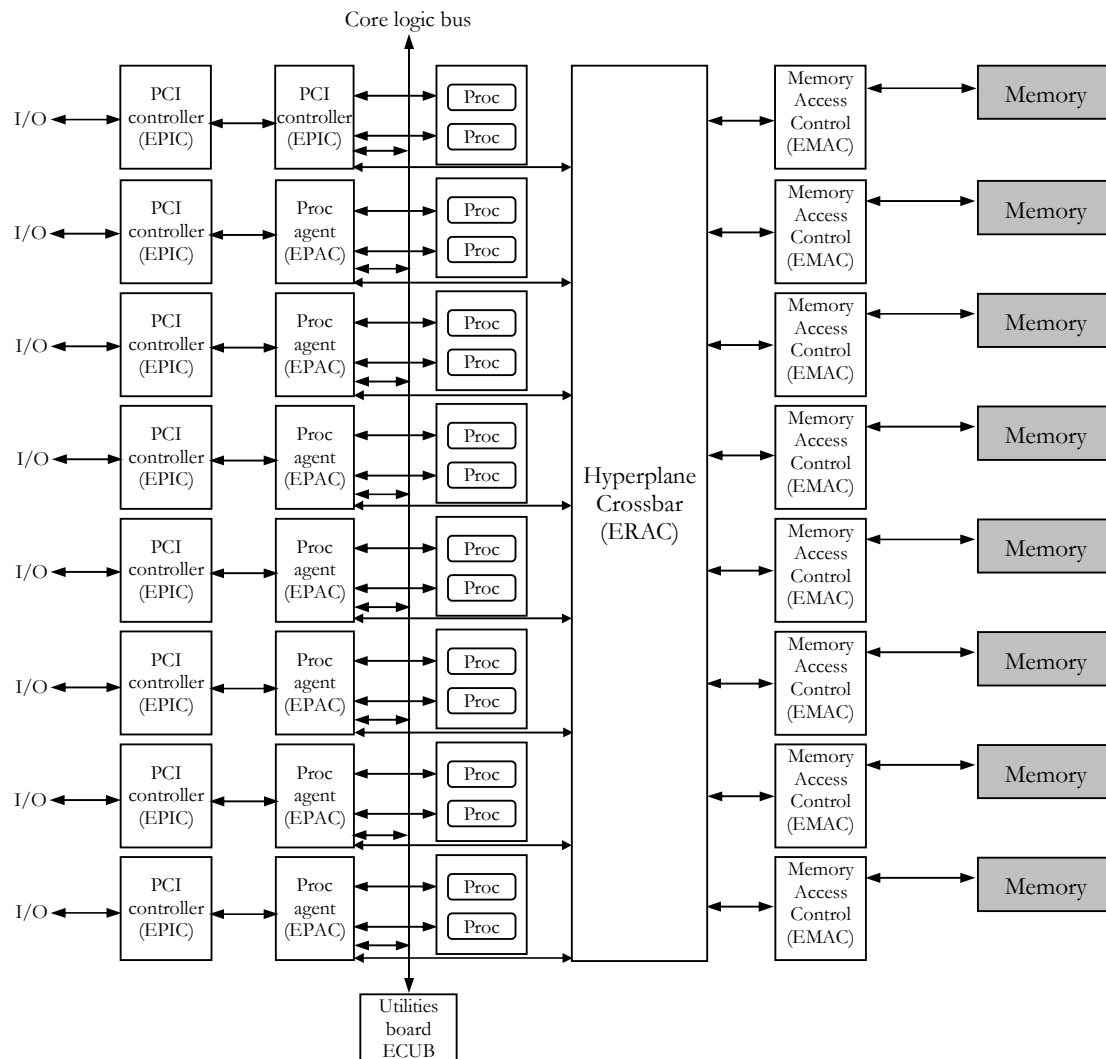


Figure 4-7 : Architecture interne de la V-Class

Le processeur et la mémoire sont reliés par un commutateur à barres croisées utilisant quatre ERAC de sorte que chaque processeur soit relié à toute la mémoire. Chaque ERAC a une largeur de bande de 1,9GB/s (en gigabit par seconde), de sorte que la largeur de bande globale de l'ensemble des quatre ERAC soit 15 GB/s. La figure (4-8) montre un digramme du commutateur à barres croisées.

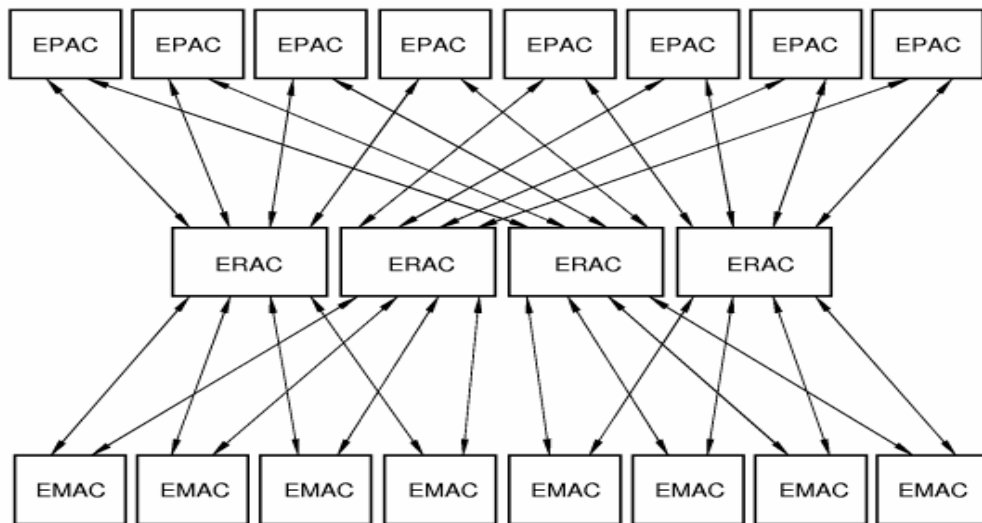


Figure 4-8 : Diagramme de la configuration de l'ERAC

b. Unités de traitement

La V-Class utilise les processeurs PA-8200 de HP. Chaque processeur est basé sur l'architecture RISC (Reduce Instruction Set Computer) à 64 bits de fréquence 240 MHz. Il est doté d'une puissance en crête de 960 MegaFlops (MF : million d'opération en virgule flottante par seconde).

c. Organisation de la mémoire

Mémoire Physique : La mémoire physique ou la RAM (mémoire à accès sélectif) sur la V-Class est de 8GB de stockage (extensible à 16 GB) répartis sur 64 banques de mémoire de 64 Mb. La division de la mémoire (RAM) dans 64 banques de mémoire permet l'accès simultané de la mémoire vers les différentes banques, ce qui ne serait pas possible si la mémoire entière était traitée en tant qu'une unité de mémoire. Chaque EMAC commande un ensemble de huit banques de mémoire avec un total de 512 Mb de mémoire. La mémoire physique a une latence de 500 nanosecondes, approximativement. La mémoire physique contient les programmes à exécuter et les données (appelés « mémoire logique »), mais contient également beaucoup d'informations « système ».

Mémoire virtuelle : La mémoire virtuelle est disponible par l'utilisation d'unités de mémoire appelées pages. Elle permet, par l'accès à des segments de page de mémoire, l'exécution de beaucoup plus de programme que ne peut la mémoire physique. Le compilateur produit des adresses de la mémoire virtuelle beaucoup plus grandes, mais seules les pages s'exécutant sont traduites en mémoire physique. La HP PA-RISC 8200 traduit des adresses de mémoire virtuelles et physiques de 32 bits en adresses de 64 bits.

Cache : Chaque processeur possède 2 Mb de mémoire cache pour les données ainsi que 2 Mb de mémoire cache pour les instructions, soit un total de 4 MB. La

mémoire cache est généralement plus petite, plus rapide et plus proche de l'unité centrale de traitement que la mémoire physique.

d. Logiciels et programme de la V-Class

Le système d'exploitation de la V-Class est le HP-UX 11.0. Sur le plan des logiciels, la plupart des langages de programmation sont accessibles sur cette machine.

Pour la programmation parallèle, le HP MPI est utilisé. Ce dernier fournit un éventail de dispositifs facilitant le développement des applications parallèles. Ces dispositifs incluent

- La conformité à la norme de la version 1.2 de MPI.
- Des styles de programmation SPMD et MPMD.
- Un outil XMPI permettant de tracer et de surveiller les applications et le fonctionnement des processeurs.
- Un outil MPIVIEW permettant de visualiser et d'analyser les performances de chaque processeur.

4.8 Analyse algorithmique et implémentation

4.8.1 Analyse de l'algorithme du gradient conjugué préconditionné

L'analyse précédente du code a montré que La partie solveur consomme plus de 98% en temps CPU. Le solveur utilisé est basé essentiellement sur l'algorithme d'Uzawa. Ce dernier est constitué principalement par le gradient conjugué préconditionné. Ce qui veut dire que le gradient conjugué préconditionné consomme la majorité du temps d'exécution.

Cela nous a ramené à restreindre la parallélisation au gradient conjugué préconditionné. L'algorithme de ce dernier est constitué de 3 opérations principales qui sont : 2 produits scalaires, 3 combinaisons linéaires de deux vecteurs, le produit matrice -vecteur ainsi que l'étape de préconditionnement.

```

Choisir  $u^0$ 
 $r = b - AX^0$ 
 $z = C^{-1}r$ 
 $p = z$ 
 $\sigma = \sigma_{old} = \sigma_0 = (r, z)$ 
repeat
 $v = Ap$ 
 $\alpha = \sigma / (v, p)$ 
 $X = X + \alpha p$ 
 $r = r - \alpha v$ 
 $w = C^{-1}.r$ 
 $\sigma = (r, z)$ 
 $\beta = \sigma / \sigma_{old}$ 
 $\sigma_{old} = \sigma$ 
 $p = z + \beta p$ 
until  $\sqrt{\sigma / \sigma_0} < \text{tolerance}$ 

```

Figure 4-9 : Algorithme du gradient conjugué pour résoudre le système $AX=b$

Pour que la parallélisation soit efficace, il est recommandé de minimiser le nombre de messages échangés entre les différents processus en dérivant un maximum de tâches indépendantes (c'est à dire, chaque tâche peut s'exécuter indépendamment dans chaque processus, sans faire recours aux données existant dans les autres processus).

La dérivation de telles tâches dans les deux premières opérations est directe : Il suffit de couper les vecteurs en nombre de parties égal au nombre de processus.

Toutefois, pour les produits scalaires, une communication sera nécessaire pour que chaque processeur envoie aux autres sa partie et reçoive de chacun leur partie. La routine MPI utilisée c'est `MPI_ALLREDUCE`, elle opère la sommation sur les données réparties sur l'ensemble de processeurs avec récupération du résultat sur tous. La dérivation des tâches indépendantes dans la troisième opération (produit matrice - vecteur) est fonction du stockage morse utilisé. Cette opération est la plus importante en temps de calcul, elle consomme à peu près 70% du temps CPU total, d'où la nécessité de chercher à dériver des tâches indépendantes pour cette opération afin de la paralléliser efficacement.

En effet, le stockage morse utilisé dans le code conserve la partie triangulaire inférieure des matrices utilisées, la réalisation d'un produit matrice - vecteur AX , doit être effectué donc en deux étapes :

Etape 1
pour $i=1$ *a* N *faire*
 $v(i)=0$
pour $j=IT(i-1)+1, IT(i)$
 $Y(i)=Y(i)+RA(j)*X(IT(j))$

Etape 2
pour $i=1$ *a* N *faire*
pour $j=IT(i-1)+1, IT(i)-1$
 $Y(IT(j))=Y(IT(j))+RA(j)*X(i)$

Figure 4-10 : Produit matrice-vecteur A.X

Dans l'étape 1, les itérations sur i sont indépendantes entre elles, et donc peuvent être réalisées en parallèle.

Dans l'étape 2, nous pouvons avoir simultanément deux processeurs distincts, qui exécutent des opérations différentes sur la même composante du vecteur v , c'est-à-dire, nous pouvons écrire dans la même zone de mémoire en même temps, ce qui peut fausser le résultat. Ce conflit empêche une parallélisation efficace. Il est intéressant donc de penser à restreindre l'opération (produit matrice-vecteur) seulement à l'étape 1.

Pour cela, un stockage complet de la matrice est nécessaire. Ce stockage permet, à l'étape 1, le balayage entier de toutes les lignes i .

La stratégie adoptée consiste à équidistribuer par paquets de lignes de la matrice sur les processeurs ; chacun aura la totalité du vecteur X et un bloc A_k de lignes de A (Fig 4.11).

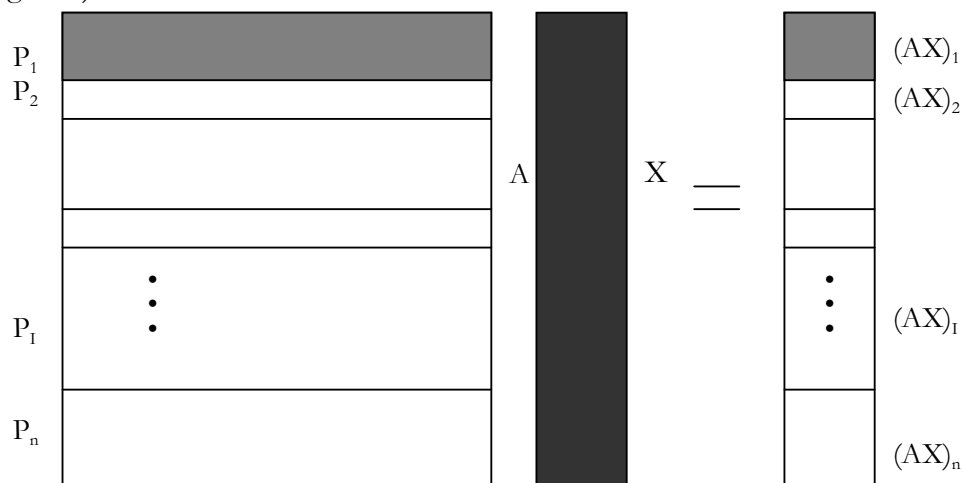


Figure 4-11 : Schématisation du produit matrice -vecteur A.X

Chaque processeur P_k se charge localement de faire le produit du bloc A_k par le vecteur X . Le résultat sera un vecteur $(AX)_k$ de dimension égale au nombre de lignes de A_k . Nous utilisons une fonction de communication synchrone pour la mise à jour du vecteur global après chaque produit matrice-vecteur. Nous utilisons la routine MPI de collecte générale : `MPI_Allgather`.

Une partie importante d'overhead résulte généralement des échanges (envoi et réception) des messages entre tâches lors de la phase de réception/diffusion des données. Pour cela, on commence par la distribution de toutes les données nécessaires au calcul, cela dit, avant de commencer le calcul, une tâche recevra un paquet de données contenant les parties utiles de A , IT , JT et les vecteurs X tous entiers. Tous les échanges effectués après ne concernent que la mise à jour des vecteurs de directions (remontée/descente) du gradient conjugué préconditionné.

4.8.2 Gradient conjugué préconditionné en parallèle

La résolution du système $AX = b$ par l'algorithme parallèle du gradient conjugué préconditionné est donné par :

Choisir X^0

– Calcul local de $r(N_1 : N_2) = b(N_1 : N_2) - A(N_1 : N_2, 1 : N)X^0(1 : N)$

– Calcul local de $z(N_1 : N_2) = C^{-1}(N_1 : N_2)r(N_1 : N_2)$

– Affectation locale de $p(N_1 : N_2) = z(N_1 : N_2)$

– Collecte globale sur tous les processeurs de p

– Affectation locale de $\sigma = \sigma_{old} = \sigma_0 = (r(N_1 : N_2), z(N_1 : N_2))$

– Somme globale de σ dans σ (avec `MPI_Allreduce`)

repeat

– Calcul local de $v(N_1 : N_2) = A(N_1 : N_2, 1 : N)p(1 : N)$

– Calcul local de $(v(N_1 : N_2), p(N_1 : N_2))$

– Somme globale de $(v(N_1 : N_2), p(N_1 : N_2))$

– Calcul sur tous les processeurs de $\alpha = \sigma / (v, p)$

– Calcul local de $X(N_1 : N_2) = X(N_1 : N_2) + \alpha p(N_1 : N_2)$

– Calcul local de $r(N_1 : N_2) = r(N_1 : N_2) - \alpha v(N_1 : N_2)$

– Calcul local de $w(N_1 : N_2) = C^{-1}(N_1 : N_2, 1 : N)r(N_1 : N_2)$

– Calcul local de $\sigma = (r(N_1 : N_2), z(N_1 : N_2))$

– Somme globale de σ dans σ

– Calcul de β sur tous les processeurs $\beta = \sigma / \sigma_{old}$

– Affectation sur tous les processeurs $\sigma_{old} = \sigma$

– Calcul local de $p(N_1 : N_2) = z(N_1 : N_2) + \beta p(N_1 : N_2)$

– Collecte globale sur tous les processeurs de p

until $\sqrt{\sigma / \sigma_0} < \text{tolerance}$

4.8.3 Algorithme général

L'algorithme parallèle général de la résolution du système (4.5) issu de la discrétisation en temps des équations de Navier – Stokes est donné ci-après.

- Initialisation

Nous commençons par l'initialisation de l'environnement de calcul MPI.

- Préparation

Cette étape contient la lecture des données et la préparation des différentes constantes. Après, le processeur maître prépare le système linéaire à résoudre. Ceci correspond aux étapes transport second membre et conditions aux limites dans l'algorithme séquentiel.

- Partage et envoi

Le processeur maître prépare et envoi à chaque processeur esclave les parties de données nécessaires pour effectuer sa tâche de calcul.

- Réception

Les processeurs esclaves reçoivent du processeur maître leurs parts de données nécessaires pour effectuer localement leurs tâches.

- Transport et second membre

L'étape de transport et le calcul de second membre se fait en séquentiel.

- Résolution du système linéaire correspond au système de Stokes généralisé.

Dans cette étape et comme c'est montré dans les paragraphes précédents, la parallélisation est faite que dans les étapes où on résout un système linéaire $AX=b$; ce qui fait donc que la parallélisation est concernée que la partie où on utilise le gradient conjugué préconditionné. Ci-après sont données les étapes paralléliser en gras italique :

- **Initialisation**

Choisir $p_{b,0}$

Calculer $u_{b,0}$, solution de

$$\mathbf{A}_h \mathbf{u}_{b,0} = \mathbf{f}_h - \mathbf{B}_h \mathbf{p}_{b,0}$$

$$\mathbf{r}_{b,0} = \mathbf{B}_h^T \mathbf{u}_{b,0}$$

Calculer $z_{b,0}$ solution de

$$\mathbf{C}_h \mathbf{z}_{b,0} = \mathbf{r}_{b,0}$$

$$\mathbf{y}_{b,0} = \mathbf{z}_{b,0}$$

- **Etape n+1** : $p_{b,n}$, $r_{b,n}$, $y_{b,n}$ étant calculées,

Calculer $x_{h,n}$ solution de :

$$\mathbf{A}_h \mathbf{x}_{h,n} = \mathbf{B}_h \mathbf{y}_{h,n}$$

$$\beta_n = (\mathbf{z}_{h,n}, \mathbf{r}_{h,n}) / (\mathbf{B}_h^T \mathbf{x}_{h,n}, \mathbf{z}_{h,n})$$

$$p_{h,n+1} = p_{h,n} + \beta_n \mathbf{x}_{h,n}$$

$$\mathbf{r}_{h,n+1} = \mathbf{r}_{h,n} - \beta_n \mathbf{B}_h^T \mathbf{x}_{h,n}$$

Calculer $\mathbf{z}_{h,n+1}$ solution de

$$\mathbf{C}_h \mathbf{z}_{h,n+1} = \mathbf{r}_{h,n+1}$$

$$\gamma_n = (\mathbf{r}_{h,n}, \mathbf{z}_{h,n}) / (\mathbf{r}_{h,n}, \mathbf{z}_{h,n})$$

$$\mathbf{y}_{h,n+1} = \mathbf{z}_{h,n+1} + \gamma_n \mathbf{y}_{h,n}$$

- **Collecte et affichage des résultats**

Après la sortie du système linéaire, les résultats sont collectés sur tous les processeurs. Et enfin, ces résultats sont affichés dans les fichiers de sortie dans le processeur maître.

4.9 Présentation des cas test et Performance

4.9.1 Outils de performances

Les résultats des tests seront analysés à travers les deux types de courbes suivants :

a. Les courbes de temps absolu Elapsed et temps absolu calcul :

Elles nous informent sur l'évolution du temps d'exécution absolue du programme en fonction du nombre de processeurs. Les courbes temps absolus calcul ne prennent en compte que les temps de calcul sans compter le temps de communication par contre les courbes du temps absolus Elapsed, représentent la somme du temps de calcul et celui de communication (sans oublier qu'il dépend étroitement de la charge de la machine). Le temps de communication ou overhead comprend le temps de préparation (latence+surcoût) et le temps de transfert.

b. Les courbes Speed-Up elapsed et calcul

En fait, c'est à travers les courbes de Speed-Up que nous pourrions apprécier l'efficacité d'un programme parallèle et l'influence de la communication entre les processeurs ainsi que l'impact de la granularité de calcul sur les performances.

Si le programme séquentiel résout un problème de taille n en un temps $T_1(n)$ et si l'algorithme parallèle le résout en un temps $T_p(n)$ avec p processeurs, l'accélération (Speed-Up) est donnée par le nombre suivant :

$$S_p(n) = \frac{T_1(n)}{T_p(n)} \quad (4.6)$$

En pratique, $T_1(n)$ est le temps d'exécution de programme parallèle sur 1 processeurs.

Le terme $S_p(n)$ peut être inférieur à 1 si le programme parallèle est pénalisé par le surcoût lié à la communication entre les processeurs dû à l'utilisation des routines de communication.

Nous distinguons deux parties dans le temps de communication :

- l'overhead stable qui représente le coût des send/receive,
- l'overhead instable qui mesure le temps des attentes réseau. Celui-ci est important lors de l'utilisation d'une machine parallèle virtuelle (réseau de stations relié par intranet) mais reste incontrôlable puisqu'il est intimement lié à l'état du réseau (charge, bande passante,...).

Les courbes de Speed-Up sont de deux types, Speed-Up CPU et Speed-Up Elapsed.

4.9.2 Présentation des cas tests

Les résultats des cas tests ont été effectués sur deux cas :

- a- Sur l'algorithme du gradient conjugué préconditionné seul (Figure 4-9).
- b- Sur le code Navier-Stokes tout entier.

a- Présentation des cas test sur l'algorithme du gradient conjugué préconditionné.

La matrice A est la matrice issue de la discrétisation des équations de Navier - Stokes 2D incompressible sur le carré unité. Les tests de performance ont été menés sur 4 discrétisations spatiales du même carré unité suivant le tableau (Tableau 4.1). Nous notons que le passage d'un maillage h_i à h_{2i} nécessite 4 fois plus de données.

Maillage	nœuds	inconnues
h_{32}	1089	3267
h_{64}	4225	12675
h_{128}	16641	49923
h_{256}	66049	198147

Tableau 4.1 : Maillages de la cavité unité (1)

Pour chaque maillage, une série d'exécution avec une partition sur 2, 3, 4, 6, 8, 10, 12 et 14 processeurs a été effectuée.

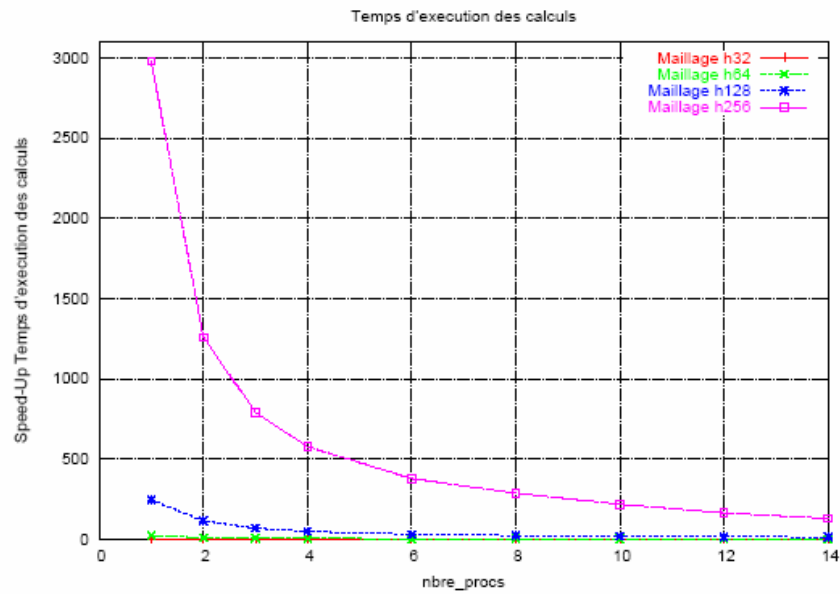


Figure 4-12 : Temps absolu de calcul (gradient conjugué)

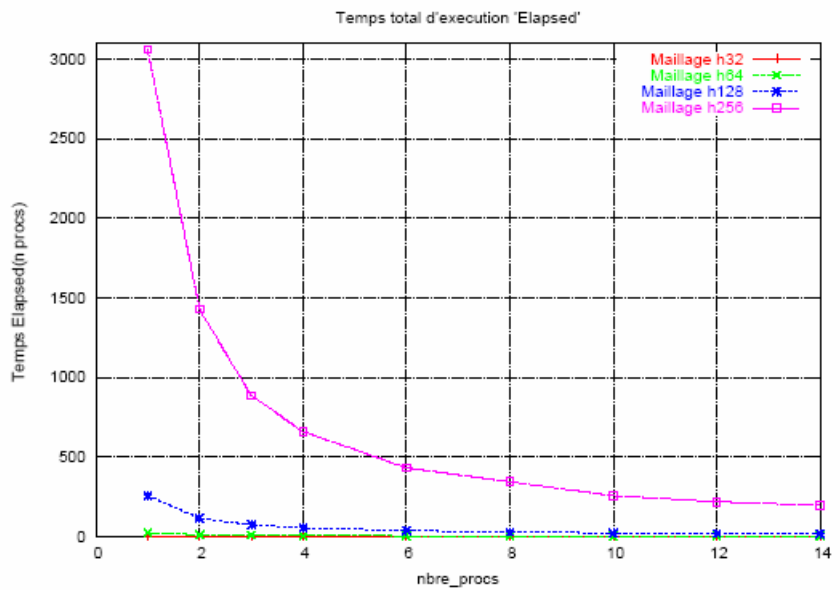


Figure 4-13 : Temps absolu total d'exécution « Elapsed » (gradient conjugué)

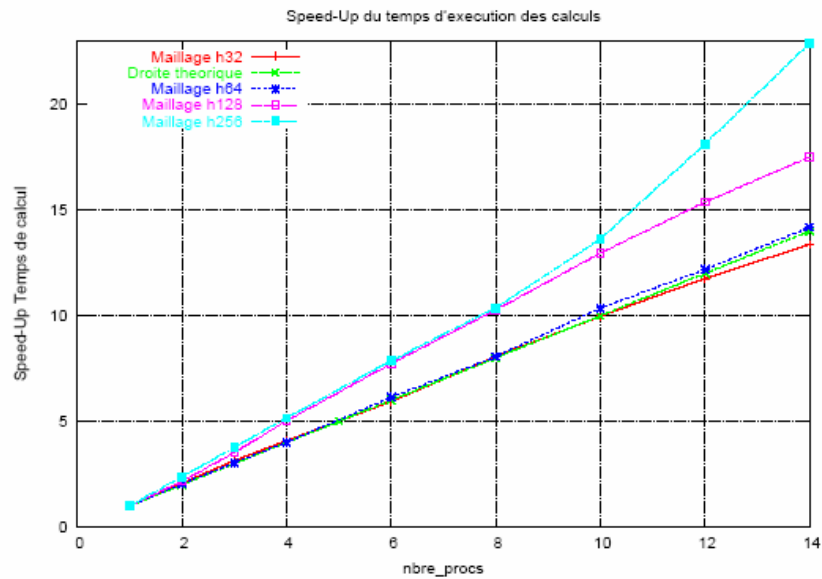


Figure 4-14 : Speed-Up calcul (gradient conjugué)

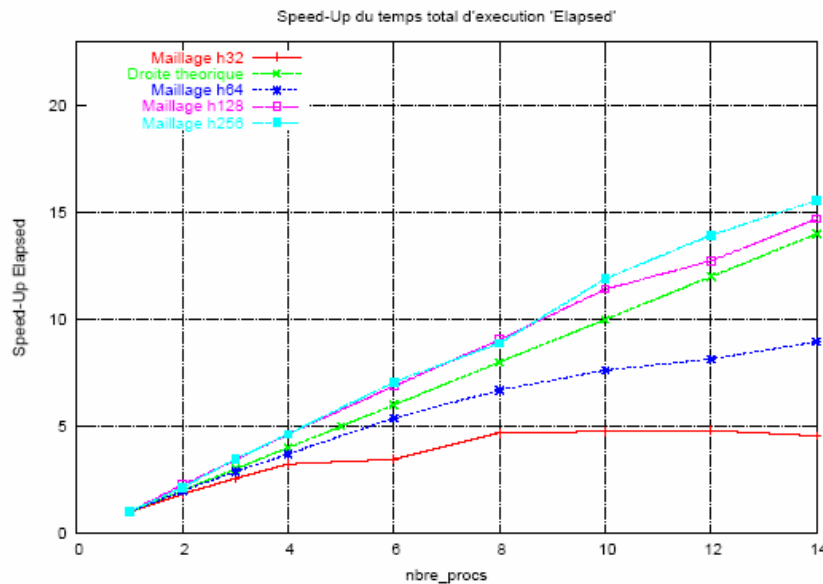


Figure 4-15 : Speed-Up Elapsed (gradient conjugué)

b- Présentation des cas tests sur le code total.

La deuxième série des tests des performances parallèles est faite sur le code Navier-Stokes incompressible 2D entièrement. Le domaine de test est toujours le carré unité. Nous donnons ci-après les différents maillages (Tableau 4.2) :

Maillage	nœuds	inconnues
h ₁₆	289	512
h ₃₂	1089	3267
h ₆₄	4225	12675
h ₁₂₈	16641	49923
h ₂₅₆	66049	198147

Tableau 4.2 : Maillages de la cavité unité (2)

Cette fois-ci, la machine parallèle est la même que la machine précédente, sauf qu'elle est avec six processeurs et sans « Cross bar ». Le calcul est partitionné sur 1, 2, 3, 4, 5 et 6 processeurs.

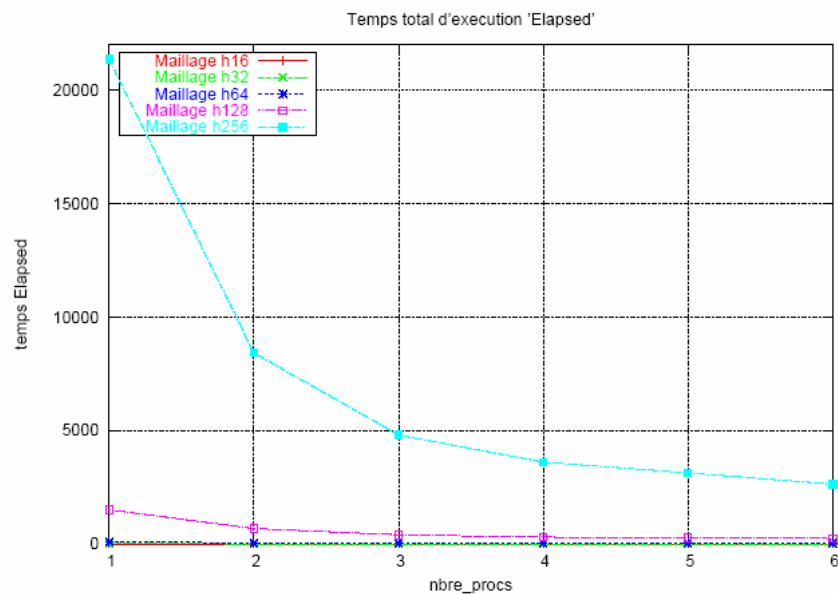


Figure 4-16 : Temps absolu de calcul (gradient conjugué)

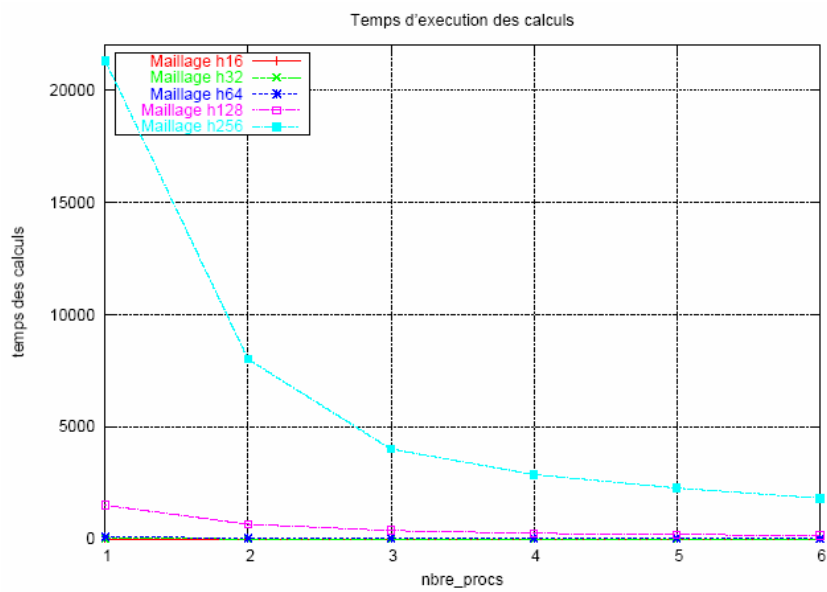


Figure 4-17 : Temps absolu total d'exécution « Elapsed » (gradient conjugué)

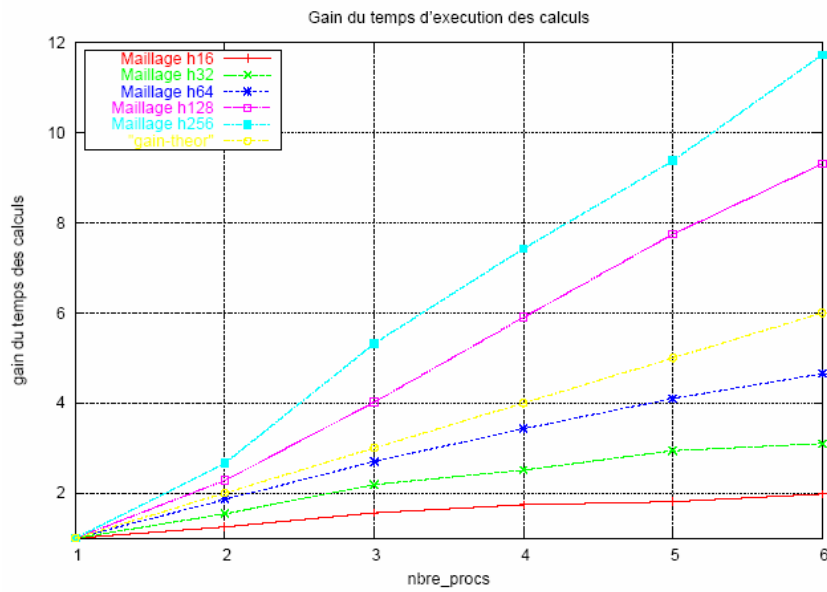


Figure 4-18 : Speed-Up calcul (gradient conjugué)

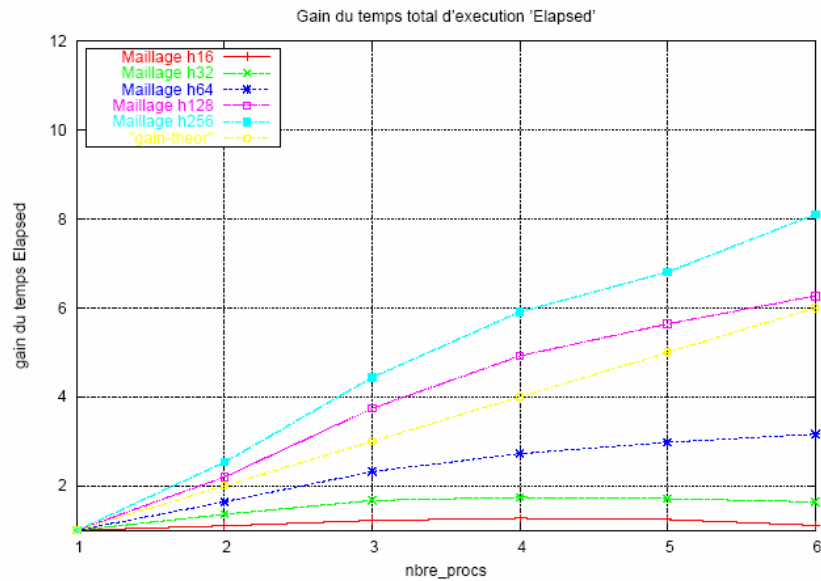


Figure 4-19 : Speed-Up Elapsed (gradient conjugué)

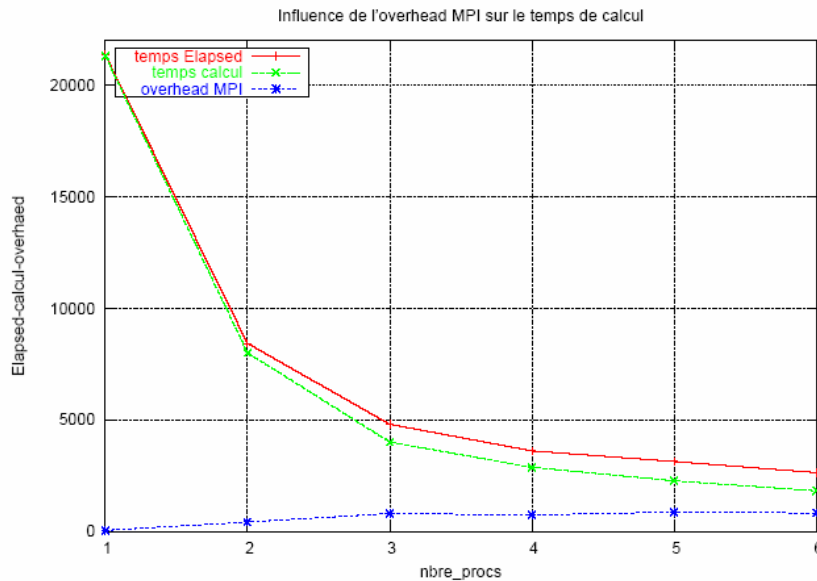


Figure 4-20 : Influence de l'overhead sur le temps d'exécution

4.10 Analyse des performances

Au premier abord, nous remarquons généralement que plus le maillage est fin, plus le gain en performance est significatif et ce quelque soit le nombre de processeurs considéré. Ensuite, nous observons un comportement super linéaire aussi bien en

speed-Up calcul qu'en speed-Up Elapsed (à partir d'un certains nombres de processeurs et du maillage h_3) : deux argumentations justifieraient ce comportement lié intimement à la granularité de calcul.

En effet, pour les temps calculs et pour les test sur le gradient conjugué, la performance est quasi linéaire même pour le cas h_1 ce qui éliminent l'effet négatif de l'overhead, ce qu'est pas le cas pour les test sur le code total (la décroissance des performance pour les petits maillages est due au partie séquentielle non parallélisée). Par contre, pour les temps Elapsed, nous avons procédé à exclure la partie la plus importante de l'overhead résulte de l'échange de communications entre processeurs (exemple : send/receive). Afin de la diminuer, nous procédons dès le départ à distribuer à chaque processeur le maximum de données dont il a besoin en autonome et ainsi minimiser au strict minimum ses échanges avec les autres processeurs.

Ainsi dans notre cas, chaque processeur i reçoit ses parties des différentes matrices (matrice de rigidité, de masse) une fois pour toute tandis que les différents vecteurs seront assujettis à des remises à jour à chaque itération.

Nous notons toutefois que pour un même problème et avec un nombre de processeurs croissant, le temps de l'allocation des registres qui se fait en parallèle sur les processeurs est moins important et la cache est moins gênée par la taille du problème. Et de plus, les registres alloués aux vecteurs X et à la partie A_i des matrices resteront inchangés : ceci accentue la super linéarité en temps calcul et la justifie en Elapsed à partir d'un certain seuil de granularité où ce gain compense l'overhead de communication.

Cela peut expliquer les performances faibles et moyennes avec les cas de faible granularité où le temps de communication emporte le gain du temps dû au partage des données sur un nombre de processeurs croissant.

Ainsi, la contrainte d'overhead en communication entre un nombre important de processeurs disparaît avec les gros cas, ce qui explique un gain en temps considérable.

4.11 Conclusion

Nous avons montré dans ce chapitre la stratégie de parallélisation adoptée dans ce travail. Ensuite, des cas tests ont montré l'efficacité de cette stratégie. Les graphes des Speed-Up présentés montre des gains du temps obtenus très importante. Ce qui confirme l'efficacité de la démarche prise.

Conclusions et perspectives

Dans le cadre de ce mémoire, nous nous sommes intéressés aux divers aspects liés au processus de l'aération mécanique des lacs touchés par le fléau de l'eutrophisation. A travers l'exposition des différents modèles des écoulements diphasiques (modèle physique instantané, modèle à deux fluides, modèle de mélange), nous avons montré les complexités et les difficultés liées au traitement numérique des équations et aussi les difficultés inhérentes à la modélisation des termes turbulents. Ceci nous a conduit à entreprendre l'approche lagrangienne pour aboutir aux modèles monophasique et monophasique corrigé, les équations de ces deux modèles sont de type Navier-Stokes.

L'objectif du premier modèle est de montrer l'effet de l'injection des bulles en se limitant à une condition aux limites aux injecteurs. Quand au modèle monophasique corrigé, il est plus élaboré, il tient compte de l'effet des bulles par la vitesse d'injection et par des termes de correction diphasique dans les équations de mouvement. La méthode choisie pour la résolution de ces deux modèles est la méthode des caractéristiques pour la discrétisation en temps et éléments finis P^1 +Bulle/ P^1 pour la partie de l'espace. La nature des équations de Navier-Stokes et la taille réelle des lacs traités, donnent des systèmes dont la résolution exige des temps de calcul très grand et un espace mémoire important. Ceci, rend nécessaire le traitement parallèle de ces problèmes.

En première approche, nous avons entamé la parallélisation du code de Navier-Stokes 2D incompressible. Ce code est composé de deux parties principales : la partie transport et la partie résolution. La partie transport est difficile et compliquée pour la parallélisation, mais sa consommation en temps de calcul est faible par rapport à l'autre partie. Par contre, la partie résolution consomme la majorité du temps de calcul (en général, plus de 90% voir même plus de 98%) et elle est relativement simple à la parallélisation par rapport à l'autre partie. Cette remarque nous a dicté la stratégie de parallélisation : la partie transport est exécutée en séquentiel et la partie résolution en parallèle.

Plusieurs tests ont été effectués et ont montré des gains en temps considérables, ces gains peuvent même devenir superlinéaire dans les gros cas. Ces résultats encourageants, valident et confirment l'efficacité de la démarche choisie. Mais dans le cas où l'écoulement tend vers un régime stationnaire, le temps de la partie de résolution devient de plus en plus petit, il risque de devenir comparable avec le temps de transport. Il serait donc intéressant de se pencher sur la parallélisation du code tout entier.

Le travail de modélisation, entamé dans ce mémoire, a abouti à la proposition d'un modèle chimique qui donne le développement de la dissolution et de la diffusion du gaz de l'oxygène et du gaz de nitrogène à travers le temps.

Enfin, il reste le traitement complet de processus de l'aération, il nécessite la résolution du modèle chimique couplé avec le modèle hydrodynamique, et comme le modèle chimique est compliqué, sa résolution passe certainement par la recherche des hypothèses simplificatrices qui les rendent plus accessible au traitement numérique.

Bibliographie

- [1] M. Abdelwahed. *Modélisation et simulation numérique d'écoulement diphasique*. Thèse de l'Université de Pau et de l'École Nationale d'Ingénieurs de Tunis, Octobre 2002.
- [2] M. Abdelwahed, M. Amara, F. Dabaghi. *Finite element approximation of a two phase flow occurring in aeration process*. Soumis à Journal of Scientific Computing, 2004.
- [3] M. Abdelwahed, M. Amara, F. Dabaghi, M. Hasine. *A numerical modelling of a two phase flow for water eutrophication problems*. European Conference on Computational Methods in Applied Sciences, ECOMAS 2000, 18pages, CD Rom Proceedings, 11-14 Septembre 2000, Barcelone.
- [4] M. Abdelwahed, F. Dabaghi. *Numerical simulation eutrophication aspects*. Report D25.1, INRIA/ERCIM 2000.
- [5] M. Abdelwahed, F. Dabaghi. *Two phase flow formulation (2D)*. Report D17, DGXI-European Commission, ESIMEAU Contract, March 2000.
- [6] M. Abdelwahed, F. Dabaghi. *Two phase flow formulation and solver 2D*. Report D25, DGXI-European Commission, ESIMEAU Contract, Jan2001.
- [7] M. Abdelwahed et al. . [*HPCN Implementation Aspects for Flood Propagation*](#). Deliverable: Report D4.3, Availability: R, Release: No. 1- Juillet 15, 2004
- [8] M. Abdelwahed, F. Dabaghi, M. Hassine. *A parallel algorithm implementation of Navier-Stokes type model simulating a two-phase flow*. Proceeding of the 22nd IASTED International Conference on Modelling, Identification, and Control, (MIC'2003), pp. 453-458, [Innsbruck-Austria](#), Fev 2003.
- [9] M. Abdelwahed, F. Dabaghi, M. Hassine. *Analyse algorithmique parallèle et résultats de performance pour un solveur Navier-Stokes ($\psi - \omega$) simulant un modèle d'écoulement diphasique simplifié*. Rapport de recherche No. 4743, INRIA.
- [10] M. Abdelwahed, F. Dabaghi, F. Mezali, B. Nakhlé. *Performance parallèle d'un code E.F.Navier-Stokes 2D en vitesse pression pour la simulation d'écoulements diphasiques*. Papier soumissionner dans la revue ARIMA 2005.
- [11] M. Abdelwahed, F. Dabaghi and D. Ouazar. *An alternative two phase flow model based on corrected water phase flow model for the simulation of eutrophised lake treatment by aeration process*. Int Conf on Thermal Engineering: Theory and Applications, ICTEA-HT4-04, Beirut-Lebanon, May 31-Jun 04, 2004. Selected paper to be published in Int J of Computational Fluid Dynamics.
- [12] M. Abdelwahed, F. Dabaghi, D. Ouazar. *A virtual numerical simulation for aeration effects in lake eutrophication*. International journal on computational fluid dynamics, Vol 16(2), 119-128, 2002.
- [13] R.A. Adams, *Sobolev spaces*. Academic press, 1975.
- [14] M. Amara. Une méthode optimale de classe C^0 d'approximation du bilaplacien. Comptes Rendus de l'Académie des Sciences, Tome319, p.1327-1330, Série I, Paris 1994.

- [15] D. Arnold, F. Brezzi, M. Fortin. *A stable finite element for the Stokes equations*. *Calcolo*, 21(4) p337-344. 1984.
- [16] R. Barret and al. . *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM, 1994.
- <http://www.netlib.org/templates/Templates.html> .
- [17] J. Bear, A. Verruijt. *Modelling groundwater flow and pollution*. D. Reidel Publishing Company, 1987.
- [18] R. Bel F'dhila, O. Simonin. *Eulerian prediction of a turbulent bubbly flow down stream of a sudden pipe expansion*. 2nd International Symposium on Multi phase Flows Modelling and Experimentation, Pise, 23-29 Mai 1999.
- [19] N. Boisson, M. R Malin. *Numerical prediction of two phase flow in bubble columns*. *International journal of numerical methods in fluids*, Vol. 23, p. 1289-1310, 1996.
- [20] H. Chanson. *Air bubble entrainment in free-surface turbulent shear flow*. Academic Press, 1996.
- [21] J. Chergui et al. . *Message Passing Interface (MPI-1)*. Version 2.12, Avril 2004, IDRIS.
- [22] E. Clement. *Dispersion de bulles et modification du mouvement de la phase porteuse dans des écoulements tourbillonnaires*. Thèse de l'INP Toulouse, 1997.
- [23] J. Chahed, L Masbernat, G.Belakel. *k-Epsilon turbulence model for bubbly flow*. 6th work shop on two-phase flow prediction, Erlangen, March 30-April 2, 1992.
- [24] Collection Microsoft Encarta 2005 [CD].
- [25] F. Dabaghi. *Numerical Aspects of Aeration Process Modelling in Eutrophised Water Basins*. *Journal of Systems Analysis Modelling Simulation*, SAMS, Vol.39, p1-23, 2000.
- [26] F. Dabaghi. *WP3.2 : Tutorial HPCN et algorithmique parallèle*. Report D.1, INRIA-ERCIM, 1999.
- [27] F. Dabaghi, M. Kaiss. *WP2.1 : Environnement MPI*. Report D5, INRIA-ERCIM, 1999.
- [28] F. Dabaghi, D. Ouazar. *Validation and calibration through the experimental basin*, Report D9, DGXI-European Commission, ESIMEAU Contract, 1999.
- [29] M. R. Davidson. *Numerical calculations of two-phase flow in a liquid with bottom gas injection: The central plume*. *Applied Mathematic Modelling*, Vol 14, p. 67-76, 1990.
- [30] D. A. Drew. *Mathematical modelling of two-phase flow*. *Ann. Review of Fluid Mechanics*, Vol 15, p261-291, 1983.
- [31] D. A. Drew R. T. Lahey. *The three-dimensional time and volume averaged conservation equations of two-phase flow*. *Advances in Nuclear sciences and Technology*, Vol. 20, 1-69, 1988.

- [32] F. Edeline. *L'épuration physico-chimique des eaux : théorie et technologie*. CEBEDOC, 1996
- [33] S. E. Elghobashi, T. W. Abou-Arab. *A two-equation turbulence model for two-phase flows*. *Physi. Fluid* 26(4), 1983
- [34] H. Enwald, E. Peirano, A. E. Almstedt. *Eulerian two-phase flow theory applied to fluidization*. *J. Multiphase flow* Vol. 22, p.21-66, 1996.
- [35] J. Erhel. *Parallélisation d'un algorithme de gradient conjugué préconditionné*. Rapport de recherche, INRIA, février 2003.
- [36] A. Ern, J.L. Guermond. *Eléments finis : théorie applications, mise en oeuvre*. Editions : Springer, 2002.
- [37] A. Esmaeli, G. Trygvason. *An inverse energy cascade in two-dimensional low Reynolds number bubbly flows*. *J. Fluid Mech.* Vol 314, p. 315-330, 1996.
- [38] G. Fourestey. *Simulation numérique et contrôle optimal d'interaction fluide incompressible/structure par une méthode de Lagrange-Galerkin d'ordre 2. Application aux ouvrages d'art*. Thèse de l'école nationale des ponts et chaussées, 2002.
- [39] J.Ch. Gilbert. *Espaces de Hilbert*. INRIA.
- [40] J.Ch. Gilbert. *Les espaces de Sobolev $W^{m,p}(I)$* . INRIA.
- [41] C. Gomella, H. Gurrée. *Les eaux usées dans les agglomérations urbaines ou rurales*. EYROLLES, deuxième édition 1982.
- [42] W. Gropp, E. Lusk, A. Skjellum. *Using MPI: Portable parallel programming with the message-passing interface*. MIT Press, 1999.
- [43] G. Haase. *Parallelization of numerical algorithms*. Johannes Kepler University of Linz. June 1999.
- [44] F. Hecht, Parés C. NSP1B3: *Un logiciel pour résoudre les équations de Navier-Stokes incompressible 3D*, R1449, INRIA Rocquencourt, 1991.
- [45] K. Helmut. *Control of eutrophication in inland waters*. Ellis Horwood series in water and waste water technology, 1991.
- [46] J. C. R. Hunt, R. J. Perkins, K. Lunde, N. H. Thomas. *Modelling bubbly flows*. Bubble Dynamics and Interface Phenomena, Blake J. Retal, Kluwer Academic Publishers, p. 257-268, 1994.
- [47] M. Jean-Michel Hervouet. *Hydrodynamique des écoulements à surface libre, modélisation numérique avec la méthode des éléments finis*. Thèse d'habilitation à diriger des recherches Université de CAEN / Basse-Normandie U.F.R. Novembre 2001.
- [48] *HP MPI User's Guide*, seventh edition, Helwet Packard, June 2002.
- [49] M. Ishii. *Thermo-fluid dynamic theory of two-phase flow*, Collection de la direction des études de recherché d'électricité de France. EYROLLES 1975.

- [50] E. Jocelyn. *Simulation numérique d'écoulements gravitaires à fortes différences de densité. Application aux avalanches*. Thèse de doctorat, Institut national polytechnique de Grenoble - INPG. Septembre 2004.
- [51] P. JOLY. *D.E.A d'Analyse Numérique : Analyse numérique matricielle*. Université Pierre et Marie Curie, Cours 2001-2002.
- [52] Charles J. Lemckert, Jorg Imberger. *Energetic bubble plumes in arbitrary stratification*. Journal of hydraulic engineering, Vol. 119, No. 6, June, 1993, p. 680-703.
- [53] O. Louisnard, J.J. Letourneau, P. Gaborit. *Initiation au fortran*. Ecole des mines d'Albi Carmaux, 1997-2000.
- [54] Y. Maday. *Méthodes d'approximation variationnelle des EDP vol. 1*. Université Pierre et Marie Curie, Cours de DIA 2001-2002.
- [55] Y. Maday. *Méthodes d'approximation variationnelle des EDP partie 2*. Université Pierre et Marie Curie, Cours de DIA 2001-2002.
- [56] H. Magnin, J.L. Coulomb. *Amélioration des performances d'une implantation parallélo-vecotrielle du gradient conjugué par extension du schéma de stockage matriciel*. J. Phys. III France 3 (1993) 519-529.
- [57] Stephen A. McCord and al.. *Modeling artificial aeration kinetics in ice-covered lakes*. Journal of environmental engineering, January 2000, p. 21-31.
- [58] Daniel F. McGinnis, John C. Little. *Bubble dynamics and oxygen transfer in a speece cone*. Wat. Sci. Tech. Vol. 37, No. 2, 1998, p. 285-292.
- [59] Asterio Sanchez Miron and al. . *Bubble-column and airlift photobioreactors for algal culture*. AIChE Journal, Vol. 46, No. 9, p. 1872-1887, September 2000.
- [60] Pares. *Méthodes des éléments finis pour les fluides*. Masson, 1988.
- [61] F. Pellegrini. *Architectures et systèmes des calculateurs parallèles*. ENSEIRB.
- [62] S. Phelippot. *L'eutrophisation des lacs, rapport de l'Association Française pour l'étude des eaux*, 1975.
- [63] O. Pironneau. *Méthodes des éléments finis pour les fluides*. Masson, 1988.
- [64] O. Pironneau. *Finite element methods for fluids*. Masson, 1989.
- [65] E.E. Prepas, J.M. Burke. *Effects of hypolimnetic oxygenation on water quality in Amisk lake, Alberta, a deep, eutrophic lake with high internal phosphorus loading rates*. Can. J. Fish. Aquat. Sci. 54, 1997, p. 2111-2120.
- [66] H. Roques. *Fondements théoriques du traitement biologique des eaux*. Vol. 1-2. Technique et documentation, deuxième édition 1980.
- [67] Y. Saad. *Iterative methods for sparse linear systems*, PWS series in computer science, 1996.
- [68] P. Saramito. *Modélisation et calcul d'interface fluide-fluide*. Cours DEA maths. applis., Grenoble, Décembre 2001.

- [69] M. P. Schwarz. *Simulation of gas injection into liquid melts*. Appl. Math. Modelling, Vol 20, p 41-51, Janvier 1996.
- [70] M. Summerfeld, S. Decker, G Kohnen. *Time dependent calculation of bubble columns based on Reynolds-averaged Navier-Stokes equation with $k - \epsilon$ turbulence model*. 2nd Japanese-German Symposium on Multi-Phase Flow, Tokyo, 1997.
- [71] Team Cosivie. *COuplage de modèles pour la Simulation numérique de problématiques en VIronnementalEs*. Activity report 2004. INRIA, Rocquencourt.
- [72] *The V-class server architecture*, <http://docs.hp.com>: 80.
- [73] *The TELEMATC modelling system*. Editors: Jean-Michel Hervouet, Paul Bates. WILEY, 2000.
- [74] S. Vialle. DEA Info. TRS-2 – Chapitre 4 : MPI : *programmation et performances*. <http://www.metz.supelec.fr/~vialle>
- [75] L. Viry. *Cours fortran 90/95*. CCH.
- [76] A. Wuest, Norman H. Brooks, Dieter M. Imboden. *Bubble plume modelling for lake restoration*. Water resources research, Vol. 28, No. 12, December 1992, p. 3235-3250.
- [77] <http://www.crihan.fr>