



**HAL**  
open science

# Réduction dimensionnelle de type PGD pour la résolution des écoulements incompressibles

Antoine Dumon

► **To cite this version:**

Antoine Dumon. Réduction dimensionnelle de type PGD pour la résolution des écoulements incompressibles. Mécanique des fluides [physics.class-ph]. Université de La Rochelle, 2011. Français. NNT : . tel-00644565v1

**HAL Id: tel-00644565**

**<https://theses.hal.science/tel-00644565v1>**

Submitted on 24 Nov 2011 (v1), last revised 16 Apr 2012 (v2)

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

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

*Ecole Doctorale Sciences pour l'Environnement Gay-Lussac (GL)*

LABORATOIRE : LEPTIAB

**THÈSE** présentée par :

**Antoine DUMON**

soutenue le : **3 Juin 2011**

pour obtenir le grade de : **Docteur de l'université de La Rochelle**

Discipline : **MECANIQUE DES FLUIDES**

**Réduction dimensionnelle de type PGD pour la  
résolution des écoulements incompressibles**

Dirigée par Cyrille **ALLERY** & Amine **AMMAR**

---

**JURY :**

**Rapporteurs :**

A. HUERTA

Professeur, Universitat Politècnica de Catalunya

P. SAGAULT

Professeur, Université Paris VI

**Examineurs :**

C. ALLERY

Maître de Conférence, Université de La Rochelle

A. AMMAR

Professeur, ENSAM, Angers

M. AZAIEZ

Professeur, ENSCPB, Bordeaux

F. CHINESTA

Professeur, Ecole Centrale de Nantes

A. CIMETIERE

Professeur, Université de Poitiers

A. HAMDOUNI

Professeur, Université de La Rochelle



# Réduction dimensionnelle de type PGD pour la résolution des écoulements incompressibles

## Résumé

L'objectif de ce travail consiste à développer la méthode de résolution PGD (Proper Generalized Decomposition), qui est une méthode de réduction de modèle où la solution est recherchée sous forme séparée, à la résolution des équations de Navier-Stokes. Dans un premier temps, cette méthode est appliquée à la résolution d'équations modèles disposant d'une solution analytique. L'équation de diffusion stationnaire 2D et 3D, l'équation de diffusion instationnaire 2D et les équations de Burgers et Stokes sont traitées. Nous montrons que dans tous ces cas la méthode PGD permet de retrouver les solutions analytiques avec une précision équivalente au modèle standard. Nous mettons également en évidence la supériorité de la PGD par rapport au modèle standard en terme de temps de calcul. En effet, dans tous ces cas, la PGD se montre beaucoup plus rapide que le solveur standard (plusieurs dizaines de fois). La résolution des équations de Navier-Stokes isothermes et anisothermes est ensuite effectuée par une discrétisation volumes finis sur un maillage décalé où le couplage vitesse-pression a été géré à l'aide d'un schéma de prédiction-corrrection. Dans ce cas une décomposition PGD sur les variables d'espace uniquement a été choisie. Pour les écoulements incompressibles 2D stationnaire ou instationnaire, de type cavité entraînée et/ou différentiellement chauffée, les résultats obtenus par résolution PGD sont similaires à ceux du solveur standard avec un gain de temps significatif (la PGD est une dizaine de fois plus rapide que le solveur standard). Enfin ce travail introduit une première approche de la résolution des équations de transferts par méthode PGD en formulation spectrale. Sur les différents problèmes traités, à savoir l'équation de diffusion stationnaire, l'équation de Darcy et les équations de Navier-Sokes, la PGD a montré une précision aussi bonne que le solveur standard. Un gain de temps à été observé pour le cas de l'équation de Poisson, par contre, concernant le problème de Darcy ou les équations de Navier-Stokes les performances de la PGD en terme de temps de calcul peuvent encore être améliorées.

Mots-clés : Proper Generalized Decomposition (PGD), Réduction de modèles (ROM), Séparation de variables, Ecoulements isothermes/anisothermes, Formulation spectrale, Mécanique des fluides numérique.

# Dimensional reduction of type PGD for solving incompressible flows

## Abstract

Motivated by solving the Navier-Stokes equations, this work presents the implementation and development of a reduced order model, the PGD (Proper Generalized Decomposition). Firstly, this method is applied to solving equations models with an analytical solution. The stationary diffusion equation 2D and 3D, 2D unsteady diffusion equation and Burgers equations and Stokes are processed. We show that in all these cases, the PGD method allows to find analytical solutions with a good accuracy compared to the standard model. We also demonstrate the superiority of the PGD relative to the standard model in terms of computing time. Indeed, in all these cases, PGD was much more rapid than the standard solver (several dozen times). The Navier-Stokes 2D and 3D thermal and isothermal isotherms are then processed by a finite volume discretization on a staggered grid where the velocity-pressure coupling was handled using a prediction-correction scheme. In this case a decomposition of the space variables only was chosen. The results in 2D for Reynolds numbers equal to 100, 1000 and 10,000 are similar to those of the solver standard with a significant time saving (PGD is ten times faster than the solver standard). Finally, this work introduces a first approach to solving the Navier-Stokes equations with a spectral method coupled with the PGD. Different cases were dealt, the stationary diffusion equation, the Darcy equation and the Navier-Sokes equations. PGD showed a good accuracy compared with the standard solver. Saving time was observed for the case of the Poisson equation, on the other hand, about Darcy's problem or Navier-Stokes' equations, performance of the PGD in terms of computing time may yet be improved.

Keywords : Proper Generalized Decomposition (PGD), Reduced order models (ROM), Separation of variables, Flow insulated/isothermal , Spectral formation, Computational fluid mechanics.

## **Remerciements**

*Quand on écrit cette page, c'est que tout est au mieux dans le meilleur des mondes... La thèse est (enfin!) rédigée et soutenue et on peut commencer à se détendre... C'est pourquoi je tiens à remercier Mr Huerta et Mr Sagault d'avoir accepté d'être les rapporteurs de ma thèse ainsi que Mr Cimetière et Mr Azaiez d'avoir accepté de participer à mon jury.*

*Si cette thèse a pu être menée au bout c'est surtout et avant tout grâce à Cyrille, encadrant de tout les instants, qui a fait en sorte que ce travail puisse s'accomplir dans des conditions optimales tout en corrigeant mes légères sautes de motivation (si si, il y en a eu...). Bref, Cyrille je te remercie énormément d'avoir passé du temps et de t'être investi comme tu l'a fait afin de faire de moi un docteur. Je remercie également Amine Ammar qui m'a toujours très bien reçu à Grenoble (et à Angers) pour me débloquer sur des points cruciaux. Merci Amine pour ta gentillesse, ton calme et ta disponibilité quand j'avais besoin de toi. Je tiens également à adresser mes remerciements à Fransisco Chinesta et Aziz Hamdouni pour leur implication dans mon comité de direction de thèse. Je souhaite également remercier tous les membres du LEPTIAB qui, de près ou de loin, ont contribué au bon déroulement de ce travail et plus particulièrement Francis Allard de m'avoir accueilli au sein du laboratoire. Je ne peux malheureusement citer tout le monde, mais je remercie tous les professeurs, maitres de conférences, secrétaires, chercheurs invités, post-docs et tout le personnel du LEPTIAB et de l'Université de la Rochelle en général.*

*Et maintenant, honneur aux fourmis du laboratoires, je veux bien entendu parler des doctorants (ou jeunes docteur). J'espere que je ne vais pas en oublier... Un gros merci à Marx, Laura et Misère (surtout Misère), à Max et Yasmine (Le King René vous salue bien bas), Luc et Camille (mes parents Rochelais), Dhaluin et Coluchette (la dream Team), Nazir (mon pote de galère), Minou (ah les blagues de minou...), Eddy (comme son nom l'indique), Contador (pris la main dans le sac) et encore Justin, Axel, Abdelkrim, Hassan, Issa, Nissrine, Mireille, Camilia, Ibrahim, Phuto, Rabbat, Massi, Walid, Aubin ... J'espere ne pas avoir été trop dur à supporter. J'ai une pensée toute particuliere pour Aliou, Pierrot et Muru qui sont en plein rush à l'heure où j'écris ces quelques lignes...*

*Je remercie également ma famille et mes amis qui, bien que n'ayant toujours pas compris mon sujet de thèse (et surtout à quoi ça sert...) m'ont toujours soutenu.*

*Enfin, un mot pour la plus belle, mon amoureuse Nathalie (Thalie, Pic, Pouc, Chouck... c'est selon...) sans qui, et là c'est pour de vrai, je n'aurais certainement jamais atteint le sommet de cette montagne que représente une thèse. C'est grâce à elle que le matin je me reveillais vaillant pour attaquer une nouvelle journée, et que je pouvais décompresser le soir en rentrant dans mon nid douillet. Bref, mon amoureuse, si on pouvait mettre une photo en couverture ce serait la tienne, et ton nom mérite autant que le mien d'y figurer.*

*A vous tous, Merci...*

*“Douter de tout ou tout croire sont deux solutions également commodes, qui l’une et l’autre nous dispensent de réfléchir.” (H. Poincaré)*

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Etat de l'art</b>	<b>5</b>
I Méthodes de réduction a posteriori . . . . .	7
I .1 La méthode POD(Proper Orthogonal Decomposition) . . . . .	7
I .2 La méthode SVD (Singular Value Decomposition) . . . . .	11
I .3 La méthode CVT (Central Voronoi Tessellations) . . . . .	13
I .4 Méthode basée sur les modes propres de l'opérateur de Navier-Stokes . .	14
I .5 Autres méthodes a posteriori . . . . .	15
II Méthodes a priori . . . . .	16
II .1 Méthode de réduction a priori (APR) . . . . .	16
II .2 Proper Generalized Decomposition (PGD) . . . . .	20
III Description de la méthode PGD . . . . .	24
III .1 PGD en décomposition spatio-temporelle . . . . .	24
III .2 PGD progressive pondérée appliquée à des problèmes multidimensionnels	32
III .3 PGD progressive pondérée en formulation tensorielle . . . . .	35
<b>2 Formulation de la PGD sur des équations modèles</b>	<b>39</b>
I Equation de diffusion stationnaire 2D . . . . .	40
I .1 Formulation du problème . . . . .	40
I .2 Résultats . . . . .	42
II Equation de diffusion stationnaire 3D . . . . .	47
II .1 Formulation du problème . . . . .	47
II .2 Résultats . . . . .	48
III Equation de diffusion instationnaire 2D . . . . .	52
III .1 Utilisation d'un schéma de discrétisation temporel classique . . . . .	52
III .2 Formulation PGD espace-temps . . . . .	53
III .3 Résultats . . . . .	54
IV Equation de Burgers . . . . .	59
IV .1 Problème étudié . . . . .	59
IV .2 Résultats . . . . .	60
V Equation de Stokes . . . . .	63
V .1 Problème étudié . . . . .	63

V .2	Résultats	64
VI	Conclusions du chapitre	68
<b>3</b>	<b>Equations de Navier-Stokes isothermes</b>	<b>69</b>
I	Formulation des schémas de prédiction-correction	71
I .1	Les schémas de correction de pression non-incrémentaux.	71
I .2	Les schémas de correction de pression incrémentaux standards.	72
I .3	Les schémas de correction de pression incrémentaux rotationnels.	73
I .4	Généralisation	74
II	Discrétisation spatiale	76
III	Algorithme PGD associé aux schémas de prédiction correction	77
IV	Applications de la PGD	79
IV .1	Problème de "Taylor-Green"	79
IV .2	Écoulement dans la cavité entraînée 2D	82
IV .3	Écoulement dans la cavité entraînée 3D	98
V	Conclusions du chapitre	106
<b>4</b>	<b>Equations de Navier-Stokes anisothermes</b>	<b>107</b>
I	Introduction	108
II	Écoulement dans une cavité entraînée 2D anisotherme	109
II .1	Equations du problème	109
II .2	Résultats	110
III	Transport de masse	121
III .1	Equations du problème	121
III .2	Résultats	122
IV	Conclusions du chapitre	126
<b>5</b>	<b>Vers une discrétisation spectrale de la PGD</b>	<b>127</b>
I	Approximation $\mathbb{P}_N - \mathbb{P}_{N-2}$ en formulation spectrale Tchebychev	129
I .1	Principe de l'approximation $\mathbb{P}_N - \mathbb{P}_{N-2}$	129
I .2	Expression de l'opérateur de dérivée première dans $\mathbb{P}_N$	130
I .3	Expression de l'opérateur de dérivée première dans $\mathbb{P}_{N-2}$	132
I .4	Expression des opérateurs bidimensionnels	133
II	Méthode $\mathbb{P}_N - \mathbb{P}_{N-2}$ associée au schéma de prédiction de Van-Kan	134
II .1	Rappel de l'algorithme de Van-Kan	134
II .2	Etape de prédiction	135
II .3	Etape de correction	136
III	Formulation PGD du problème	139
III .1	Etape de prédiction	139
III .2	Etape de correction	140
IV	Résultats	141
IV .1	Equation de diffusion	142
IV .2	Problème correspondant à l'étape de correction	143
IV .3	Problème de Taylor-Green	145

IV .4	Ecoulement dans une cavité 2D . . . . .	146
V	Conclusions du chapitre . . . . .	151
<b>Conclusion</b>		<b>153</b>
<b>A Formulation <math>\psi - \omega</math></b>		<b>157</b>
I	La formulation ligne de courant-vorticité . . . . .	158
II	Résolution par PGD . . . . .	159
III	Ecoulement dans une cavité entraînée 2D . . . . .	160
	III .1 Régime stationnaire ( $Re = 100$ et $Re = 1000$ ) . . . . .	160
	III .2 Régime périodique ( $Re = 10.000$ ) . . . . .	165
IV	Conclusions . . . . .	168
<b>B PGD appliquée à l'équation de Poisson 2D stationnaire</b>		<b>169</b>
I	Etape d'enrichissement . . . . .	171
II	Etape de projection . . . . .	172
III	Etape d'estimation de la convergence . . . . .	172
<b>Bibliographie</b>		<b>173</b>



# Introduction

Les méthodes numériques sont souvent utilisées en physique pour la résolution numérique des équations aux dérivées partielles (EDP) décrivant des phénomènes physiques. Dans le cas d'un écoulement fluide, les EDP non-linéaires concernées sont les équations de Navier-Stokes. Malheureusement, dans la grande majorité des cas aucune solution analytique de ces équations n'est disponible. C'est pourquoi la résolution numérique des équations de Navier-Stokes est de plus en plus utilisée. Si l'on souhaite modéliser un écoulement fluide sur un grand domaine et sur des durées longues, les ressources de calculs aussi bien en terme de rapidité que de capacité de stockage peuvent très rapidement devenir insuffisantes. Ainsi, ces dernières années est apparue une forme de résolution des EDP permettant de rechercher la solution sur des bases de taille réduite. Le principe des méthodes de réduction consiste en un premier temps à construire une base réduite de taille  $N$  bien inférieure au nombre de degrés de libertés du problème à résoudre et, dans un second temps à projeter le problème sur cette base réduite de façon à ramener le problème initial à la résolution d'un système d'équations différentielles couplées de taille  $N$ , dont la résolution est très rapide. Selon la méthode de réduction utilisée, le principe de construction de la base sera différent. Parmi les différentes méthodes de réduction de modèle (ou ROM pour 'Reduced Order Models') il est possible de distinguer deux grandes familles, celles dites 'a-posteriori' qui nécessitent une connaissance préalable de l'écoulement afin de construire la base réduite, et celles dites 'a-priori' qui ne nécessitent aucune information préalable sur l'écoulement que l'on souhaite simuler. Les méthodes 'a-posteriori' ont été beaucoup étudiées ces vingt dernières années et en particulier l'une d'entre elle, la Proper Orthogonal Decomposition (POD). Cependant comme nous l'avons indiqué, ces méthodes nécessitent des connaissances sur la nature de l'écoulement avant de pouvoir construire la base de réduction. Un autre désavantage de ce type de méthode est que la base construite à partir des informations préalables sur le fluide, n'est valable que dans des conditions proches de celles où ces informations ont été tirées.

Afin de remédier à ces problèmes, des méthodes de réduction permettant de s'affranchir de cette connaissance préalable de l'écoulement ont été développées. Ces méthodes portent le nom de méthodes 'a-priori'. Parmi celles-ci, nous nous intéresserons à une en particulier, la Proper Generalized Decomposition (PGD) qui consiste à rechercher la solution d'une EDP de manière itérative sous forme séparée. Cette technique de résolution, initialement appelée 'radial time-space approximation', fut introduite pour résoudre des problèmes de mécanique non-linéaire à la fin des années 1990 en recherchant la solution sous la forme d'une décomposition espace-temps. Ainsi en couplant cette forme de décomposition à la méthode LATIN (LArge Time INcrement method), il a été possible de traiter des cas inaccessibles jusque là. Elle a ensuite été reprise et améliorée par différents auteurs jusqu'à sa version multidimensionnelle. Dans ce cas la PGD consiste à rechercher la solution d'un problème comme une somme de produits de fonctions de chacune des variables du problème. Ces variables peuvent être les variables d'espace classique, le temps, ou encore les paramètres de contrôle. De cette manière, les deux inconvénients des méthodes 'a-posteriori' peuvent être dépassés. En effet, la base est calculée de manière itérative, donc aucune information préalable sur l'écoulement n'est nécessaire. De plus, la dépendance de la base au choix du nombre de Reynolds de l'écoulement par exemple, peut être résolue en choisissant de prendre le nombre de Reynolds comme paramètre.

La PGD a fait l'objet d'un grand nombre d'études, dans des domaines très variés, tels que la mécanique des solides, la rhéologie, la simulation de matériaux composites, l'homogénéisation ou encore la résolution de problèmes stochastiques. Elle n'a cependant jamais été appliquée à des problèmes de mécanique des fluides, et en particulier à la résolution des équations de Navier-Stokes. L'objectif de ce travail de thèse est donc de tester les capacités de la PGD pour résoudre les écoulements fluides isothermes et anisothermes, aussi bien en terme de précision que de temps de simulation.

Ainsi, ce mémoire est organisé de la façon suivante :

- Dans le premier chapitre, un état de l'art (non exhaustif) sur les différentes méthodes de réduction de modèles est effectué en distinguant les méthodes 'a priori' de celles 'a posteriori'. La méthode PGD est plus particulièrement détaillée car c'est la technique utilisée dans la suite de ce mémoire.
- Dans le chapitre deux, la PGD est appliquée pour résoudre des équations modèles. Les problèmes traités sont les équations de diffusion stationnaire 2D et 3D, l'équation de diffusion instationnaire 2D, le problème de Burgers 2D, et enfin l'équation de Stokes 2D. Dans ce chapitre une décomposition sur les variables d'espace seulement est utilisée sauf dans le cas de l'équation de diffusion instationnaire 2D où la séparation sur les variables spatiales et temporelle a été effectuée. La PGD est étudiée sur l'équation de Burgers 2D en raison du caractère non-linéaire de ce problème. De même le couplage vitesse-pression est analysé à travers le problème de Stokes 2D.
- Le chapitre trois est consacré à la résolution des équations de Navier-Stokes en condition isotherme. Une présentation des méthodes de prédiction-correction permettant d'effectuer le découplage vitesse-pression et la description de la formulation PGD associée sont présentées. La méthode PGD est ensuite appliquée à la résolution du problème de 'Taylor-Green' (où il existe une solution analytique), puis au cas d'un écoulement dans une cavité entraînée 2D et 3D.
- Le chapitre quatre traite quant à lui de la résolution des équations de Navier-Stokes en condition anisotherme avec et sans transport de masse. Les équations des problèmes sont rappelées et la PGD est étudiée sur différents cas tests.
- Enfin, le chapitre cinq développe la méthode PGD pour une formulation spectrale des équations de Navier-Stokes. Une approximation  $\mathbb{P}_N - \mathbb{P}_{N-2}$  en formulation spectrale Tchebychev est tout d'abord détaillée sur un schéma de prédiction-correction. La formulation PGD du problème associé est ensuite exposée. Enfin la résolution de l'équation de diffusion stationnaire, du problème de Darcy et des équations de Navier-Stokes sont traitées.



# Chapitre 1

## Etat de l'art

Très souvent un problème physique peut être représenté par un système d'EDP dépendant de plusieurs variables. De manière générale, cela revient à chercher  $\mathbf{u} \in L^2([0, T]; V)$  tel que

$$\begin{cases} \mathcal{A} : V \longrightarrow W \\ \frac{\partial \mathbf{u}}{\partial t} + \mathcal{A}(\mathbf{u}) = f \text{ dans un ouvert } \Omega \text{ de } \mathbb{R}^n \text{ où } n = 1, 2, 3 \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases} \quad (1.1)$$

où  $V$  et  $W$  sont souvent des espaces de Hilbert de dimension infinie,  $\mathbf{u}(t) \in V$  est l'inconnue du problème physique et  $\mathcal{A}$  l'opérateur différentiel qui décrit le système d'équations gouvernant le problème physique. Si la résolution analytique n'est pas possible (ce qui est généralement le cas pour les équations de Navier-Stokes gouvernant les écoulements) il est alors nécessaire de procéder à une résolution numérique du problème. Pour ce faire, il faut passer par une étape de discrétisation permettant de calculer les valeurs de l'inconnue en chaque point d'un maillage spatial et pour chaque intervalle de temps de la discrétisation. Le problème (1.1) est donc réécrit dans de nouveaux espaces  $V_h$  et  $W_h$  de dimensions finies  $n = O(\frac{1}{h})$  où  $h$  est le paramètre de discrétisation spatiale. Après discrétisation, le problème (1.1) pourra donc s'écrire sous la forme suivante :

$$\begin{cases} \mathcal{A}_h : V_h \longrightarrow W_h \\ \frac{\partial \mathbf{u}_h}{\partial t} + \mathcal{A}_h(\mathbf{u}_h) = f_h \\ \mathbf{u}_h(0) = \mathbf{u}_h^0 \end{cases} \quad (1.2)$$

où  $\mathcal{A}_h$  est l'opérateur discret et  $f_h$  le second membre discret associé à la discrétisation spatiale de pas  $h$ .  $\mathbf{u}_h(t) \in V_h$  est l'inconnue du problème discrétisé. La dimension  $n$  des nouveaux espaces dépend alors de la finesse du maillage utilisé. Pour représenter correctement certains phénomènes physiques il est nécessaire d'avoir un maillage très fin, ce qui implique la résolution de très grands systèmes et donc des temps de simulations élevés. Afin de diminuer la taille du problème à résoudre il est possible d'utiliser des techniques de réduction de modèles.

La réduction de modèle consiste à chercher une approximation de la solution en projetant les équations du problème sur des sous espaces  $\mathcal{V}_h$  de  $V_h$  bien déterminés de taille  $m \ll n$ . On cherche donc généralement la solution  $\mathbf{u}_h$  de (1.2) sous la forme suivante :

$$\mathbf{u}_m(x, t) = \sum_{k=1}^m a_k(t) \phi_k(x) \quad (1.3)$$

où  $m$  sera l'ordre de la décomposition de notre solution, généralement de l'ordre de la dizaine.

La première étape de la réduction de modèle consiste à construire la base  $\{\phi_k(x)\}_{1 \leq k \leq m}$ . C'est cette étape qui différencie généralement les méthodes de réduction. Une fois cette base connue, la méthode consiste à projeter les équations du problème sur cette base afin de construire un système dynamique composé d'équations différentielles dont les inconnues sont les  $\{a_k(t)\}_{1 \leq k \leq m}$ . Ainsi si on suppose que les  $\{\phi_k(x)\}_{1 \leq k \leq m}$  sont connus, un des systèmes dynamiques associés au problème (1.2) s'écrira de la manière suivante, trouver les  $\{a_k(t)\}_{1 \leq k \leq m}$  tels que :

$$\left( \sum_{k=1}^m \frac{da_k(t)}{dt} \phi_k(x), \phi_i(x) \right) + \left( \mathcal{A} \sum_{k=1}^m a_k(t) \phi_k(x), \phi_i(x) \right) = (f_h, \phi_i(x)) \quad \text{pour } i = 1, \dots, m \quad (1.4)$$

Après résolution de ce système d'équations, l'approximation d'ordre  $m$  de la solution s'écrit comme dans (1.3).

Dans ce chapitre nous allons nous intéresser aux différentes techniques de réduction de modèle en différenciant les cas où la construction de la base réduite nécessite une connaissance préalable du phénomène étudié, que l'on appellera les méthodes '*a posteriori*', des cas où la base de calcul est inconnue initialement, que nous appellerons les méthodes '*a priori*'. Dans ce cas la base réduite est construite de façon itérative lors de la résolution du problème étudié. Parmi les méthodes '*a posteriori*' nous étudierons plus particulièrement la méthode de réduction certainement la plus connue et la plus utilisée ces dernières années, à savoir la POD (Proper Orthogonal Decomposition). La SVD (Singular Value Decomposition) et la CVT (Central Voronoi Tessellation) seront également traitées. Enfin nous détaillerons une méthode de réduction basée sur les modes propres de l'opérateur de Navier Stokes. Concernant les méthodes '*a priori*', nous étudierons l'APR (A Priori Reduction) ainsi que la PGD (Proper Generalized Decomposition) sur laquelle nous nous attarderons plus particulièrement du fait que ce sera la méthode développée dans la suite de ce manuscrit.

## I Méthodes de réduction a posteriori

Nous allons tout d'abord nous intéresser aux méthodes dites '*a posteriori*', c'est à dire aux méthodes nécessitant une connaissance préalable de l'écoulement afin de construire la base réduite  $\{\phi_k(x)\}_{1 \leq k \leq m}$ . Ces informations préalables peuvent être des snapshots (clichés de la solution à différents instants) dans le cas de la POD ou de la CVT, ou bien un champ stationnaire dans le cas de la méthode basée sur les modes propres de l'opérateur de Navier-Stokes.

### I.1 La méthode POD(Proper Orthogonal Decomposition)

De toutes les méthodes de réduction de modèle existantes, la POD est incontestablement la plus utilisée dans le domaine de la mécanique des fluides, c'est en tout cas la méthode qui a donné lieu au plus grand nombre de publications ces dernières années. Il est possible d'interpréter la POD de deux manières différentes. D'une part, il est possible de considérer la POD comme une décomposition de Karhunen-Loeve (KLD), et d'autre part, on peut voir la POD comme une technique englobant l'Analyse en Composante Principale (ACP). L'analyse en composante principale consiste à réduire la dimension d'un ensemble de données formé d'un très grand nombre de variables, en transformant et en ordonnant les variables en un ensemble incorrélaté de nouvelles variables, de manière à ce que les premières composantes contiennent la plupart des variations contenues dans les variables originelles. La méthode KLD est quant à elle, une extension de l'ACP à des espaces de dimension infinie, comme par exemple, l'espace

des fonctions continues en temps. L'analyse KLD utilise des fonctions à un seul paramètre à la place de vecteurs et des fonctions à deux paramètres pour représenter les autocorrélations à la place de matrices. Bien qu'il n'existe ni démonstration complète, ni preuve théorique et systématique de l'équivalence de ces trois méthodes, des connexions étroites existent. Les problèmes aux valeurs propres issus de ces méthodes sont équivalents et on obtient la même base de vecteurs propres.

Ces méthodes ont été utilisées dans de nombreux domaines comme l'analyse du signal, la compression de données ou des processus d'identification et de contrôle en chimie, l'océanographie et enfin en mécanique des fluides... C'est en 1967 que Lumley[Lumley, 1967] introduit la POD dans des problèmes de mécanique des fluides. Il associe les structures cohérentes d'un écoulement aux modes propres issus de la décomposition orthogonale aux valeurs propres. En 1988, Aubry et al.[Aubry et al., 1988] utilisent pour la première fois la base POD pour construire un système dynamique capable de décrire la dynamique des structures d'un écoulement à l'intérieur d'un canal. Depuis ce papier initiateur, la POD a été utilisée un dans un grand nombre de cas avec plus ou moins de succès dans [Berkooz et al., 1993, Aubry et al., 1993] [Bergooz et Titi, 1993, Holmes et al., 1996, Holmes et al., 1997, Park et Lee, 1997] ou encore [Allery et al., 2004, Allery et al., 2005, Allery et al., 2008, Liberge et Hamdouni, 2010].

### a) POD classique

Dans la suite on considère des fonctions réelles  $u(x, t)$ <sup>1</sup> définies par

$$\begin{aligned} u : \Omega \times ]0, T[ &\rightarrow \mathbb{C}^n \\ (x, t) &\rightarrow u(x, t) \end{aligned} \quad (1.5)$$

où  $\Omega$  est un sous espace de  $\mathbb{R}^n$  et  $n = 2, 3$ .

La POD consiste donc à rechercher une fonction déterministe  $\phi$  approximant au mieux en moyenne un champ aléatoire  $u$ . On recherche donc  $\phi$  appartenant à un espace de Hilbert  $H$  tel que :

$$\frac{\langle (\phi, u)^2 \rangle}{(\phi, \phi)} = \max_{\Phi \in H} \frac{\langle (\Phi, u)^2 \rangle}{(\Phi, \Phi)} \quad (1.6)$$

où  $(\bullet, \bullet)$  désigne le produit scalaire et  $\langle \bullet, \bullet \rangle$  l'opérateur moyenne.

Le problème (1.6) peut se ramener, en utilisant le calcul des variations, au problème aux valeurs propres suivant

$$\langle (\phi, u)u \rangle = \lambda \phi \quad (1.7)$$

où  $\lambda$  est un multiplicateur de Lagrange qui a été introduit lors du calcul.

Ce problème revient, en utilisant le produit scalaire associé à l'espace des fonctions  $L^2(\Omega)$ , à rechercher les fonctions  $\phi$  et les valeurs propres  $\lambda$  qui vérifient :

$$\left\langle \int_{\Omega} u(x', t) \cdot \phi(x', t) dx' u(x, t) \right\rangle = \lambda \phi(x) \quad (1.8)$$

---

<sup>1</sup>La POD est en général présentée pour des fonctions complexes pour tenir compte des transformées de Fourier qui apparaissent dans les directions de périodicité.

En introduisant le tenseur de corrélation en deux points noté  $\mathcal{R}(x, x')$ , et défini par  $\mathcal{R}(x, x') = \langle u(x, t) \otimes u(x', t) \rangle$ , on est amené à résoudre l'intégrale de Fredholm :

$$\int_{\Omega} \mathcal{R}(x, x') \phi(x') dx' = \lambda \phi(x) \quad (1.9)$$

Les valeurs propres et fonctions propres possèdent les propriétés suivantes :

- Les fonctions propres sont orthogonales et peuvent être normalisées

$$(\phi_n, \phi_m) = \int_{\Omega} \phi_n(x) \phi_m(x) dx = \delta_{nm} \quad (1.10)$$

- Toute réalisation du champ aléatoire  $u(x, t)$  peut s'écrire de la manière suivante

$$u(x, t) = \sum_{i=1}^{\infty} a_n(t) \phi_n(x) \quad (1.11)$$

où les coefficients temporels sont déterminés en projetant le champ  $u(x, t)$  sur les fonctions propres correspondantes

$$a_n(t) = (u(x, t), \phi_n(x)) \quad (1.12)$$

- Par construction la base POD est optimale au sens énergétique, ainsi en général un faible nombre de modes permet de capturer la quasi-totalité de l'énergie.
- Les fonctions propres  $\phi_n$  vérifient les conditions aux limites, respectent les symétries spatiales et sont à divergence nulle dans le cas d'un écoulement incompressible.

## b) La snapshot POD

Pour déterminer les fonctions propres POD il faut résoudre l'équation intégrale de Fredholm (1.9). La dimension du noyau de  $\mathcal{R}(x, x')$  est dans le cas 2D  $4.n_1.n_2$  et, dans le cas 3D  $9.n_1.n_2.n_3$  ( $n_1$ ,  $n_2$  et  $n_3$  sont le nombre de points du maillage dans la direction  $x_1$ ,  $x_2$  et  $x_3$  du repère associé au domaine). De plus comme  $\mathcal{R}(x, x')$  est une matrice pleine, la résolution de (1.9) peut devenir très coûteuse et peut même très vite dépasser la capacité de calcul des ordinateurs actuels. Une des alternatives possibles est d'utiliser la méthode des snapshots, introduite par Sirovich dans [Sirovich, 1987], qui consiste à traiter des clichés temporels (snapshots) de l'écoulement. Il suppose que seulement  $M$  réalisations ( $M$  temps) de l'écoulement ( $M < N_m.n_c$  où  $N_m$  est le nombre de points de mesures possibles et  $n_c$  le nombre de composantes de vitesses) sont nécessaires pour décrire de manière précise le problème étudié. Ainsi le problème à résoudre par la méthode des snapshots est de dimension  $M \times M$  alors que celui de la méthode classique est de  $N_m.n_c \times N_m.n_c$ .

Soit  $u(x, t_i)_{i=1, \dots, M}$  l'ensemble des  $M$  clichés de l'écoulement et  $\phi(x)$  un vecteur propre de la décomposition orthogonale. On cherche alors les coefficients  $a_k$  tels que :

$$\phi(x) = \sum_{k=1}^M a_k u(x, t_k) \quad (1.13)$$

Sous l'hypothèse d'ergodicité (la moyenne peut être assimilée à une moyenne temporelle) le tenseur des corrélations s'écrira de la manière suivante :

$$\mathcal{R}(x, x') = \langle u(x, t) \otimes u(x', t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T u(x, t) u(x', t) dt \approx \frac{1}{M} \sum_{i=1}^M u(x, t_i) u(x', t_i) \quad (1.14)$$

De cette manière le problème aux valeurs propres (1.9) peut être réécrit :

$$\frac{1}{M} \sum_{i=1}^M \sum_{k=1}^M (u(t_k), u(t_i))_{L^2(\Omega)} a_k u(x, t_i) = \lambda \sum_{k=1}^M a_k u(x, t_k) \quad (1.15)$$

On obtient ainsi la condition suffisante<sup>2</sup> suivante pour que  $\{a_k, k = 1, \dots, M\}$  soit solution de l'équation précédente :

$$\frac{1}{M} \sum_{k=1}^M C_{ki} a_k = \lambda a_i \quad (1.16)$$

avec

$$C_{ki} = (u(t_k), u(t_i))_{L^2(\Omega)} \quad (1.17)$$

La résolution de ce nouveau problème aux valeurs propres donne accès aux coefficients  $a_k$ . Une fois ces coefficients calculés il suffit de construire les modes POD en se servant de la forme décrite dans l'équation (1.13)

### c) Construction du système dynamique dans le cas des équations de Navier-Stokes

A ce stade, les fonctions spatiales  $\{\phi_i\}_{i=1, \dots, M}$  de la décomposition de  $u$  sont connues. L'objectif est de calculer les coefficients temporels  $\{a_i\}_{i=1, \dots, M}$  en projetant les équations du problème étudié sur chaque élément de la base spatiale. Nous illustrons cette démarche dans le cas de équations de Navier-Stokes.

Les équations de Navier-Stokes incompressibles s'écrivent de la manière suivante :

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{f} & \text{dans } \Omega \\ \operatorname{div} \mathbf{u} = 0 \end{cases} \quad (1.18)$$

où  $\mathbf{u} = \mathbf{u}(x, t) = (u(x, t), v(x, t))$  est la vitesse du fluide,  $p$  représente la pression,  $\rho$  la masse volumique du fluide étudié et  $\nu$  sa viscosité cinématique.  $\mathbf{f}$  représente l'action des forces de volume extérieures. Après avoir calculé la base réduite des  $\{\phi_k(x)\}_{1 \leq k \leq M}$ , une approximation d'ordre  $N$  de la vitesse peut s'écrire de la manière suivante :

$$\mathbf{u}_N = \sum_{i=1}^N a_i(t) \phi_i(x) \quad (1.19)$$

---

<sup>2</sup>Si les  $\{u(x, t_i), i = 1, \dots, M\}$  forment une famille linéairement indépendante, c'est à dire si les  $u(x, t_i)$  sont incorrélés, c'est également une condition nécessaire.

Généralement très peu de modes contiennent la quasi-totalité de l'énergie. Ainsi en conservant les  $N < M$  premiers modes significatifs du point de vue énergétique, l'équation du mouvement est réécrite en injectant (1.19) dans (1.18), soit, pour un champ de pression connu :

$$\sum_{i=1}^N \frac{da_i(t)}{dt} \phi_i(x) - \nu \sum_{i=1}^N a_i(t) \nabla^2 \phi_i(x) + \sum_{i=1}^N \sum_{j=1}^N a_i(t) a_j(t) (\phi_i(x) \cdot \nabla) \phi_j(x) + \frac{1}{\rho} \nabla p = \mathbf{f} \quad (1.20)$$

Après projection de Galerkin sur les fonctions  $\{\phi_k\}_{1 \leq k \leq N}$ , le système dynamique associé aux équations de Navier Stokes peut donc s'écrire pour  $1 \leq k \leq N$

$$\sum_{i=1}^N G_{ki} \frac{da_i(t)}{dt} - \nu \sum_{i=1}^N K_{ki} a_i(t) + \sum_{j=1}^N \sum_{i=1}^N N_{kij} a_i(t) a_j(t) + D_k = (\mathbf{f}, \phi_k(x)) \quad (1.21)$$

avec

$$G_{ki} = \int_{\Omega} \phi_k \phi_i d\Omega \quad K_{ki} = \int_{\Omega} \phi_k \nabla^2 \phi_i d\Omega \quad N_{kij} = \int_{\Omega} \phi_k \cdot (\phi_i \cdot \nabla) \phi_j d\Omega \quad D_k = -\frac{1}{\rho} \int_{\Gamma} \phi_k p \mathbf{n} dS \quad (1.22)$$

Le terme  $D_k$  associé à la pression est gênant. En effet, la base spatiale calculée à partir des snapshots n'a aucune raison de convenir pour décomposer la pression. Dans de nombreuses applications, ce terme s'annule exactement (c'est le cas si les  $\phi_k$  sont nulles sur les bords). Si ce n'est pas le cas il est souvent négligé et parfois modélisé.

Le système dynamique construit à partir d'un faible nombre de modes a généralement des difficultés à dissiper l'énergie, car ce sont les petites échelles du système représentées par les modes faiblement énergétiques qui dissipent cette énergie. La non-prise en compte de ces petites échelles peut rendre le système instable. Afin de le stabiliser, plusieurs méthodes ont été mises en place. L'une de ces méthodes consiste à modéliser l'influence des petites échelles par l'introduction d'une viscosité artificielle, aussi nommée viscosité tourbillonnaire. Cette méthode consiste à remplacer la viscosité  $\nu$  intervenant dans le système dynamique par  $\nu + \nu_{add}$  où  $\nu_{add}$  est choisit par l'utilisateur de façon à ce que les résultats obtenus soient le plus proche possible de ceux obtenus dans les simulations utilisées pour la construction de la base.  $\nu_{add}$  peut également être fonction du mode POD  $i$ , dans ce cas on a  $\nu_{add} = \sum_{i=1}^N \nu_{add}^i$ . Il est également possible de rechercher les  $a_k(t)$  de manière à ce que l'erreur commise entre l'approximation POD et les snapshots utilisés pour construire la base spatiale soit minimale. Couplet et al. [Couplet et al., 2005] ramène ce problème à une minimisation de fonctionnelle. Une autre méthode consiste à améliorer la base ou à modifier la manière de calculer cette base. Ainsi Rempfer [Rempfer, 2000], propose de compléter la base POD avec des éléments appartenant au noyau de l'opérateur de corrélation associé à la POD et de déterminer les modes propres instables éventuels par le biais d'une analyse de Floquet.

## I.2 La méthode SVD (Singular Value Decomposition)

La SVD est une puissante méthode de factorisation de matrices introduite par Beltrami et Jordan dans les années 1870, et étendue au cas des matrices non carrées par Ekardt et Young

en 1939. Cette méthode peut être interprétée comme une extension de la décomposition aux valeurs propres pour les matrices rectangulaires. Contrairement à ce qui a été fait dans le cadre de la POD, la SVD sera traitée sous forme matricielle.

Si l'on dispose de  $M$  snapshots du champ  $u$  définis par  $u_i^j = u(x_j, t_i)_{i=1, \dots, M; j=1, \dots, N}$ , où  $t \in [0, T]^M$  et  $x \in \mathbb{R}^N$ . Il est possible de définir la matrice des snapshots  $A$  telle que

$$A = \begin{pmatrix} u_1^1 & u_2^1 & \cdots & u_M^1 \\ u_1^2 & u_2^2 & \cdots & u_M^2 \\ \vdots & \ddots & \ddots & \vdots \\ u_1^N & u_2^N & \cdots & u_M^N \end{pmatrix} \quad (1.23)$$

Effectuer une SVD de la matrice  $A$  de taille  $(N \times M)$  consiste à rechercher cette matrice sous la forme :

$$A = U \Sigma V^T \quad (1.24)$$

où

- $U$  est une matrice orthogonale de dimension  $(N \times N)$  contenant les vecteurs singuliers à gauche de  $A$ . Les colonnes de  $U$  constituent une base orthonormale de l'espace engendré par les lignes de  $A$ .
- $V$  est une matrice orthogonale de dimension  $(M \times M)$  contenant les vecteurs singuliers à droite de  $A$ . Les colonnes de  $V$  constituent une base orthonormale du temps engendré par les colonnes de  $A$ .
- $\Sigma$  est une matrice diagonale de dimension  $(N \times M)$  contenant  $p = \min(N, M)$  nombres  $\sigma_i$  appelés valeurs singulières de  $A$ . Par convention, ces valeurs sont rangées dans un ordre décroissant.
- l'exposant  $T$  indique la transposée de la matrice.

A partir de la décomposition SVD de  $A$  on peut écrire,

$$AA^T = (U \Sigma V^T)(U \Sigma V^T)^T = U \Sigma V^T V \Sigma^T U^T \quad (1.25)$$

La matrice  $V$  étant orthogonale,  $V^T V$  est égal à l'identité, et comme  $\Sigma$  est une matrice diagonale, on obtient l'égalité suivante :

$$AA^T = U \Sigma^2 U^T \quad (1.26)$$

où l'on reconnaît un problème aux valeurs propres<sup>3</sup> associé à la matrice  $AA^T$  de taille  $(N \times N)$ . Le même raisonnement peut être mené avec la matrice  $A^T A$  de taille  $(M \times M)$ .

Si l'on applique la SVD à la matrice  $A$  contenant les snapshots, les lignes de la matrice  $U$  seront homogènes à une base spatiale des données de  $A$  et seront donc vues ici comme les  $\{\phi_k\}_{k=1, \dots, N}$ . De même les colonnes de la matrice  $V$  seront homogènes à une base temporelle des données de  $A$  et seront donc vus ici comme les  $\{a_k\}_{k=1, \dots, M}$ .

En utilisant l'une ou l'autre des bases ainsi calculées, il est aisé d'obtenir une décomposition spatio-temporelle de notre champ  $u(x, t)$  et d'utiliser ensuite cette décomposition à travers un système dynamique par exemple comme cela a été fait dans la POD.

---

<sup>3</sup>il suffit de multiplier l'équation (1.26) à droite par  $U$

**Remarque 1**

La SVD est intéressante ici car, au même titre que la POD, la base de vecteur engendrée par cette décomposition constitue une base optimale au sens énergétique. C'est à dire que les  $r$  premiers vecteurs  $\phi_i$  et  $a_i$  obtenus par cette décomposition minimisent l'erreur :

$$\frac{1}{M} \sum_{k=1}^M \|u_k - \sum_{j=1}^r a_j \phi_j\|^2 \quad (1.27)$$

De plus cette décomposition peut être utilisée dans le cas d'une décomposition spatio-temporelle, mais aussi dans le cas d'une séparation espace-espace. Tout dépendra de la définition de la matrice  $A$ . C'est pourquoi dans la suite nous utiliserons cette décomposition afin d'étudier l'optimalité de la méthode PGD dans le cas de décomposition sur les variables d'espace seulement.

**I.3 La méthode CVT (Central Voronoi Tessellations)**

La CVT qui est une autre forme de réduction de modèle a posteriori a été utilisée avec succès dans un grand nombre d'applications allant de la compression de données au traitement d'image en passant par la résolution des équations de Navier-Stokes [Du et al., 1999, Du et Gunzburger, 2002b, Du et Gunzburger, 2002a, Burkardt et al., 2006]. Burkardt et al. [Burkardt et al., 2006] comparent notamment les méthodes POD et CVT dans le cas d'un écoulement dans un canal en T. Les résultats observés sont comparables pour des bases de même taille. Le calcul de la base CVT serait plus rapide que celui de la base POD, mais la méthode n'a pas été testée sur de grands nombres de Reynolds et le choix de la taille de la base n'est pas encore bien défini.

**a) Définitions**

Soit une famille de fonctions notée  $U = \{u_n\}_{n=1}^M$  appartenant à  $\mathbb{R}^J$ . Une famille  $\{V_k\}_{k=1}^N$  est définie comme une partition de  $U$  si  $\{V_k\}_{k=1}^N$  est une subdivision d'ensemble disjoints de  $U$ , c'est à dire si pour  $k = 1, \dots, N$ , on a

$$\begin{cases} V_k \subset U \\ V_k \cap V_i = \emptyset & k \neq i \\ \bigcup_{k=1}^N V_k = U \end{cases} \quad (1.28)$$

Soit une famille de fonctions  $\{\phi_k\}_{k=1}^N$ , alors les  $V_k$  qui vérifient la propriété suivante forment la *région de Voronoï* associée au fonctions  $\{\phi_k\}$ .

$$V_k = \{u \in U : |u - \phi_k| \leq |u - \phi_i|, \text{ pour } i = 1, \dots, N; i \neq k\} \quad (1.29)$$

où  $|\cdot|$  est la norme euclidienne.

La famille  $\{V_k\}_{k=1}^N$  est aussi appelée *diagramme de Voronoï* ou *partition de Voronoï* de  $U$  associée aux fonctions  $\{\phi_k\}_{k=1}^N$ . Les fonctions  $\phi_k$  sont également appelées les génératrices du diagramme de Voronoï.

### b) Construction du système dynamique

Dans le contexte de réduction de modèle, notons  $U = u(x, t)_{k=1, \dots, M}$  l'ensemble des  $M$  snapshots (clichés temporels de l'écoulement à l'instant  $t_k$ ). On a maintenant le résultat suivant :

*Construire la région de Voronoï de  $U$  associée aux fonctions  $\{\phi_k\}_{k=1}^N$  revient à rechercher la famille des  $\{\phi_k\}_{k=1}^N$  qui minimise l'énergie CVT exprimée par :*

$$E(\Phi) = \sum_{i=1}^N \sum_{u \in V_k} |u - \phi_i|^2 \quad (1.30)$$

où  $\{V_k\}_{k=1}^N$  est la région de Voronoï associée aux snapshots  $\{u_k\}_{k=1}^M$ .

La CVT consiste donc à construire la région de Voronoï associée aux snapshots, où les générateurs constitueront la base CVT d'ordre réduit. Pour ce faire, il existe des algorithmes efficaces, comme par exemple la méthode de Lloyd qui est un algorithme déterministe ou la méthode de Mac Queen qui est un algorithme probabiliste. Pour en savoir plus sur ces méthodes de construction, le lecteur pourra se référer à [Du et al., 1999, Ju et al., 2002].

De la même manière que pour la POD, une fois la base  $\{\phi_k\}_{k=1}^N$  connue, on recherche la solution du problème sous la forme :

$$u_N(x, t) = \sum_{i=1}^N a_i(t) \phi_i(x) \quad (1.31)$$

Les coefficients  $\{a_k\}_{k=1}^N$  sont obtenus par résolution du modèle réduit (1.21). Il est à noter ici que dans le cas de la CVT aucune informations sur l'orthogonalité des fonctions  $\phi_k$  n'est disponible.

## I.4 Méthode basée sur les modes propres de l'opérateur de Navier-Stokes

Une autre méthode de réduction, qui a notamment été appliquée dans le cadre du contrôle actif par Gadoin et al. [Gadoin et al., 2001], consiste à utiliser comme base, les modes propres de l'opérateur de Navier-Stokes pour résoudre les équations de Navier-Stokes au voisinage d'une solution stationnaire  $\bar{\mathbf{u}}$  en recherchant la solution  $\mathbf{u} = (u, v, p)$  sous la forme

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \tilde{\mathbf{u}}(\mathbf{x}, t) \quad (1.32)$$

où  $\bar{\mathbf{u}}$  est connue et calculée de manière traditionnelle, et  $\tilde{\mathbf{u}}$  est une perturbation.

$\mathbf{u}(\mathbf{x}, t)$  est donc solution des équations de Navier-Stokes dont l'équation de quantité de mouvement est réécrite sous la forme :

$$\frac{\partial \mathbf{u}}{\partial t} = (L - N_{\mathbf{u}}) \cdot \mathbf{u} \quad (1.33)$$

où  $L$  représente la partie linéaire et  $N_{\mathbf{u}}$  la partie non linéaire du problème, auquel il faut rajouter la contrainte de continuité.

En introduisant (1.32) dans (1.33), cette décomposition conduit, en négligeant les termes du second ordre au problème linéarisé :

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = \mathcal{L}_{\bar{\mathbf{u}}}\tilde{\mathbf{u}} \quad (1.34)$$

où  $\mathcal{L}_{\bar{\mathbf{u}}} = (L - N_{\bar{\mathbf{u}}})$  est le Jacobien de l'opérateur de Navier-Stokes évalué en  $\bar{\mathbf{u}}$ . Le champ  $\tilde{\mathbf{u}}$ , solution du problème linéarisé, est recherché sous la forme :

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_{k=1}^N a_k(t)\phi_k(\mathbf{x}) \quad (1.35)$$

où les  $\phi_i$  sont les vecteurs propres du Jacobien de l'opérateur de Navier-Stokes  $\mathcal{L}_{\bar{\mathbf{u}}}$ , et les  $a_i(t)$  les coefficients temporels correspondants.

Si l'on injecte cette forme de la solution dans les équations de Navier-Stokes (1.33), cela nous donne

$$\sum_{k=1}^N \frac{da_k(t)}{dt}\phi_k(\mathbf{x}) = \sum_{k=1}^N \lambda_k a_k(t)\phi_k(\mathbf{x}) - \sum_{l=1}^N \sum_{m=1}^N a_l a_m \phi_l(\mathbf{x}) \cdot \nabla \phi_m \quad (1.36)$$

où  $\lambda_k$  correspond à la valeur propre de  $\phi_k$  associée au vecteur propre  $\phi_k$ .

Comme le Jacobien  $\mathcal{L}_{\bar{\mathbf{u}}}$  n'est pas auto-adjoint, les vecteurs propres  $\phi_k$  ne sont pas orthogonaux entre eux, mais sont orthogonaux aux vecteurs propres de l'adjoint du Jacobien notés  $\psi_l$ . Ainsi le système réduit s'obtient en projetant (1.36) sur les  $\psi_l$ .

$$\frac{da_k}{dt} = \lambda_k a_k - \sum_{l=1}^N \sum_{m=1}^N \mu_{klm} a_l a_m \quad k = 1, \dots, K \quad (1.37)$$

où  $\mu_{klm} = (\psi_k, \phi_l \cdot \nabla \phi_m)$  représente le terme de couplage non-linéaire.

Comme l'indiquent Dowell et al. [Dowell et al., 1997], et Hall et al. [Hall et al., 1995] la méthode est extrêmement coûteuse pour un nombre de degrés de liberté trop élevé. En effet, la méthode a besoin d'un nombre de modes élevés pour décrire correctement le problème. De plus il est difficile de mettre en place une stratégie permettant de sélectionner les modes, ceux-ci n'étant pas à valeurs décroissante. Même si cette méthode a permis de retrouver des résultats corrects, on constate que son usage reste limité à certaines configurations, de plus la méthode se limite au cas où l'on recherche une solution proche d'une solution stationnaire.

## I .5 Autres méthodes a posteriori

Il existe de nombreuses autres méthodes de réduction a posteriori dans la littérature. On pourra citer ici la balanced truncation ou méthode de troncature équilibrée introduite par Moore [Moore, 1981] pour des problèmes issus de l'automatique et du contrôle afin de construire des méthodes d'ordre réduits conservant la stabilité asymptotique. Cette méthode a été étudiée et développée ensuite par de nombreux auteurs [Antoulas, 2005a, Antoulas, 2005b, Gugercin et Antoulas, 2006, Stykel, 2006]. Dans [Ilak et Rowley, 2006] et [Rowley, 2005] Ilak

et Rowley introduisent la décomposition aux valeurs propres équilibrée combinant la POD et la méthode de troncature équilibrée. Une étude comparative est réalisée entre la POD classique et la 'balanced-POD' et les résultats indiquent que la balanced-POD est plus efficace que la POD classique sur des problèmes linéaires. On doit cependant noter que les approches du type balanced-truncation ou balanced-POD n'ont été traitées que dans le cas de problèmes linéaires. D'autres méthodes de réduction permettant de résoudre des problèmes dépendant d'un paramètre ont également été développées. Ces méthodes consistent à établir une base évoluant en fonction de ces paramètres. Cette base peut être construite de différentes manières, soit en utilisant des sous-espaces de Lagrange [Peterson, 1989], [Ito et Ravindran, 1998], des sous-espaces de Taylor [Peterson, 1989] ou bien des sous-espaces de Hermitte [Ito et Ravindran, 1998]. Ces méthodes ont été utilisées afin de résoudre des problèmes de mécanique des fluides dans des cavités entraînées à faibles nombres de Reynolds avec des applications dans le domaine du contrôle.

Ces méthodes a posteriori ont donc été développées dans un souci de réduire le temps de simulation des problèmes. Cependant on peut noter quelques inconvénients à ce type de méthode. D'une part la construction des bases réduites nécessite d'avoir des informations préalables sur la solution (snapshots, solution stationnaire, ...), ce qui est contraignant. D'autre part, les calculs effectués sont corrects sur une plage de temps ou gamme de paramètre (nombre de Reynolds, ...) proche des caractéristiques contenues dans les snapshots, en l'occurrence pour la POD, sur l'intervalle de temps où les snapshots ont été réalisés.

## II Méthodes a priori

Ce paragraphe est consacré aux méthodes dites a priori qui ne nécessitent pas d'informations initiales sur la solution recherchée, et donc pas de phase d'échantillonnage de l'écoulement. Avec ce type de méthodes, la base est construite de manière itérative jusqu'à obtenir une base représentative de l'écoulement. La méthode APR dont l'enrichissement est basé sur les sous-espaces de Krylov sera tout d'abord présentée. Dans un second temps, nous détaillerons la PGD (Proper Generalized Decomposition), qui consiste à rechercher la solution sous forme séparée et qui est la méthode utilisée dans ce document.

### II .1 Méthode de réduction a priori (APR)

#### a) Généralités

La méthode de réduction a priori (APR) est, comme son nom l'indique, une méthode qui ne nécessite aucune information sur la solution, au contraire de la POD ou de la CVT. C'est une méthode incrémentale basée sur un enrichissement de la base à l'aide des sous-espaces de Krylov sur un intervalle de temps donné. Elle a été développée par Ryckelnyck [Ryckelnyck, 2002, Ryckelnyck, 2005] pour résoudre des problèmes thermomécaniques puis appliquée dans le cas de problèmes d'élasticité non-linéaire [Ammar et al., 2006b] ainsi que de problèmes d'éléments de frontières [Ryckelnyck et al., 2005]. La méthode a également été utilisée pour résoudre les équations de transfert [Verdon et al., 2006, Verdon et al., 2011, Allery et al., 2011], pour résoudre les équations de Navier-Stokes [Verdon, 2007], pour traiter des problèmes élasto-

visco-plastiques [Ryckelynck et Benziane, 2010] et pour modéliser la propagation de fissures [Kerfriden et al., 2010].

On rappelle que l'on souhaite résoudre le problème suivant :

$$\text{Trouver } u(\mathbf{x}, t) \text{ solution de } \begin{cases} \frac{\partial u}{\partial t} + \mathcal{A}(u) = f & \text{dans } \Omega \times [0, T] \\ u(0) = u^0 \end{cases} \quad (1.38)$$

L'APR consiste tout comme la POD, à rechercher la solution du problème sous forme de décomposition de fonctions spatiales  $\{\phi_k\}_{k=1}^N$  et temporelles  $\{a_k\}_{k=1}^N$ , de dimension très faible comparée à la taille du problème discrétisé de sorte que la décomposition d'ordre  $N$  de la solution s'écrive :

$$u(\mathbf{x}, t) \approx u_N(\mathbf{x}, t) = \sum_{k=1}^N a_k(t) \phi_k(\mathbf{x}) \quad (1.39)$$

où le nombre de fonctions de la décomposition  $N$  n'est pas prédéfini.

La philosophie de la méthode APR est une correction de la base  $\phi$  de manière itérative jusqu'à en obtenir une qui décrit le mieux possible la solution  $u(\mathbf{x}, t)$  sur tout l'intervalle de temps considéré. L'algorithme APR nécessite trois étapes : une étape d'initialisation de la base, suivie d'une étape de résolution du système réduit, puis une étape d'enrichissement et d'orthogonalisation de la base. Ces deux dernières étapes sont répétées jusqu'à convergence.

## b) Description des différentes étapes de l'APR

### 1) Initialisation de la base

Comme il s'agit d'une méthode itérative, cette méthode nécessite une initialisation des vecteurs spatiaux. En règle générale les vecteurs de la base spatiale  $\phi$  sont initialisés de manière à satisfaire les conditions initiales du problème et donc contiennent au moins la solution initiale elle-même.

### 2) Construction et résolution du système réduit

On suppose que nous sommes à la  $j^{\text{ème}}$  itération, c'est à dire que la solution peut s'écrire :

$$u_{N_j}(\mathbf{x}, t) = \sum_{k=1}^{N_j} a_k(t) \phi_k(\mathbf{x}) \quad (1.40)$$

Pour construire le modèle réduit cette forme de la solution est introduite dans l'expression du problème (1.38) :

$$\sum_{k=1}^{N_j} \frac{da_k(t)}{dt} \phi_k(\mathbf{x}) + \sum_{k=1}^{N_j} a_k(t) \mathcal{A}(\phi_k(\mathbf{x})) = f + Res_{N_j} \quad (1.41)$$

où le résidu  $Res_{N_j}$  apparait du fait de la discrétisation.

Une projection de Galerkin est ensuite effectuée, ce qui revient à projeter l'expression (1.41) sur chacun des vecteurs de la base spatiale en imposant que le résidu  $Res_{N_j}$  soit orthogonal à ces vecteurs. Le modèle réduit qui en résulte s'écrit pour  $n = 1, \dots, N_j$  :

$$\sum_{k=1}^{N_j} (\phi_k(\mathbf{x}), \phi_n(\mathbf{x})) \frac{da_k(t)}{dt} + \left( \mathcal{A} \left[ \sum_{k=1}^{N_j} a_k(t) \phi_k(\mathbf{x}) \right], \phi_n(\mathbf{x}) \right) = (f, \phi_n(\mathbf{x})) \quad (1.42)$$

où  $(\cdot, \cdot)$  désigne le produit scalaire euclidien ou le produit scalaire  $L^2$  selon le problème étudié.

La résolution de ce système d'équations différentielles permet d'obtenir les coefficients temporels  $a_k(t)$  sur tout l'intervalle de temps considéré et par suite la solution  $u_{N_j}(\mathbf{x}, t)$ .

### 3) Vérification de la validité de la base et enrichissement

Le résidu associé à l'équation (1.41) va représenter la qualité de la base considérée et donc définir le critère de convergence de l'algorithme APR. Ainsi, si à chaque instant  $t_i$  ( $i = 1, \dots, M$ ), la norme de ce résidu définie par

$$Res_{N_j}(t_i) = \sum_{k=1}^{N_j} \frac{da_k(t_i)}{dt} \phi_k(\mathbf{x}) + \sum_{k=1}^{N_j} a_k(t_i) \mathcal{A}(\phi_k(\mathbf{x})) - f \quad (1.43)$$

est inférieure à  $\epsilon$  (fixé par l'utilisateur), alors l'algorithme est convergé. Si ce n'est pas le cas la base doit être orthogonalisée et enrichie.

#### i) Orthogonalisation de la base.

L'orthogonalisation de la base se fait en appliquant une décomposition de Karhunen-Loève (ou SVD) aux coefficients temporels  $a_k(t_i)$  ( $i = 1, \dots, M$  et  $k = 1, \dots, N_j$ ). Ainsi il faut résoudre le problème aux valeurs propres suivant :

$$[K]\mathbf{V} = \lambda\mathbf{V} \quad (1.44)$$

où la matrice  $[K]$  est la matrice de corrélation temporelle de taille  $N_j \times N_j$ , définie par :

$$K_{ij} = \frac{1}{M} \sum_{k=1}^M a_i(t_k) a_j(t_k) \quad (1.45)$$

et  $M$  est le nombre de pas de temps.

La résolution de ce problème donne  $N_j$  valeurs propres  $\lambda_1 > \lambda_2 > \dots > \lambda_{N_j}$ . Seules les  $q$  premières valeurs propres vérifiant le critère suivant :

$$\lambda_i > \eta \lambda_1 \quad ; \forall i = 1, \dots, q$$

sont conservées.  $\eta$  est le critère de sélection des modes propres fixé par l'utilisateur.

Ainsi la nouvelle base orthogonalisée  $\tilde{\Phi} = \{\tilde{\phi}_1, \dots, \tilde{\phi}_q\}$  contient  $q$  vecteurs définis par :

$$\tilde{\phi}_k = \sum_{i=1}^{N_j} \phi_i \mathbf{V}_i^k \quad \text{pour } k = 1, \dots, q \quad (1.46)$$

où  $V^k$  est le vecteur propre associé à la valeur propre  $\lambda_k$ .

### ii) Enrichissement

Après avoir conservé et orthogonalisé les fonctions de la base spatiale qui méritaient de l'être (selon le critère fixé par l'utilisateur), il faut maintenant enrichir la base. Précédemment on a remarqué que pour une certaine valeur du temps le critère sur le résidu n'était pas vérifié. Notons  $t^+$  cette valeur. On a donc

$$\forall t < t^+, \|Res_{N_j}(t)\| < \epsilon, \quad \text{et} \quad \|Res_{N_j}(t^+)\| = R^+ > \epsilon \quad (1.47)$$

On peut interpréter cela comme le fait que la base est efficace jusqu'au temps  $t^+$  exclus, et que passé ce temps, la base n'est plus capable de fournir de bons résultats. Pour enrichir la base, le vecteur qui va être retenu sera le premier résidu pour lequel le critère de convergence n'est pas vérifié, soit  $R^+$ . A partir de ce résidu, l'espace de Krylov d'ordre  $m$ ,  $\mathcal{K}^m$ , est construit de la manière suivante :

$$\mathcal{K}^m = \{R^+, JR^+, J^2R^+, \dots, J^{m-1}R^+\} \quad (1.48)$$

où  $J$  est la jacobienne du problème (1.38). En pratique le sous-espace de Krylov est d'ordre 1 ou 2. De plus, dans ce cas les vecteurs de la base ne sont plus nécessairement orthogonaux mais ils peuvent l'être en utilisant une procédure de Gram-Schmidt.

La nouvelle base à l'étape  $j+1$  sera donc définie par la concaténation de la base orthogonalisée  $\tilde{\Phi}$  et de la base enrichie  $\mathcal{K}^m$  :

$$\Phi^{(j+1)} = \{\tilde{\Phi}, \mathcal{K}^m\} \quad (1.49)$$

Les étapes 2 et 3 sont répétées jusqu'à ce que le critère de convergence défini par l'équation (1.43) soit vérifié.

### c) Algorithme APR

L'algorithme suivant résume la méthode APR.

1.	<i>Initialisation de la base <math>\Phi^1 = \{\phi^1\}</math> où <math>\phi^1</math> est choisi pour satisfaire au moins les conditions initiales.</i>
2.	<b><i>tant que</i></b> ( $N_j \leq N_{max}$ et $\ Res_{N_j}(t_i)\  \geq \epsilon$ ( $i = 1, \dots, N_j$ )) <b><i>faire</i></b>
4.	<i>Construction du modèle réduit de taille <math>N_j</math> (Eq. 1.42)</i>
5.	<b><i>pour</i></b> $i = 1, M$ <b><i>faire</i></b>
6.	<i>Résolution du ROM de façon à obtenir les coefficients temporels <math>a_k(t_i)</math> pour <math>k = 1, \dots, N_j</math>.</i>
7.	<i>Reconstruction de la solution <math>u_{N_j}(\mathbf{x}, t)</math> en utilisant l'équation (1.40).</i>
8.	<i>Calcul de <math>\ Res_{N_j}(t_i)\ </math> avec l'équation (1.43).</i>
9.	<b><i>fin pour</i></b>
10.	<i>Recherche du premier résidu tel que <math>\ Res_{N_j}(t_i)\  &gt; \epsilon</math>. (<math>\epsilon</math> est fixé par l'utilisateur et ce résidu est noté <math>R^+</math>)</i>
11.	<i>Orthogonalisation de la base <math>\Phi^j</math>, ce qui donne <math>\tilde{\Phi}</math> (Eq. 1.46).</i>
12.	<i>Calcul du sous espace de Krylov <math>\mathcal{K}^m</math> associé à <math>R^+</math> (Eq. 1.48).</i>
13.	<i>Obtention de la nouvelle base <math>\Phi^{j+1} = \{\tilde{\Phi}, \mathcal{K}^m\}</math>.</i>
14.	<b><i>fin tant que</i></b>

TAB. 1.1 – Algorithme APR ( $\epsilon$  est fixé par l'utilisateur)

## II .2 Proper Generalized Decomposition (PGD)

La méthode PGD (Proper Generalized Decomposition) au même titre que l'APR est une méthode dite '*a priori*' dans le fait que les fonctions de la base d'approximation sont construites sans avoir d'information sur la forme de la solution. Cette méthode consiste à rechercher la solution  $u$  sous la forme séparée suivante :

$$u(x_1, \dots, x_N) \approx u_m(x_1, \dots, x_N) = \sum_{i=1}^m F_1^i(x_1) \dots F_N^i(x_N),$$

où les variables  $x_1, \dots, x_N$  peuvent représenter des variables spatiales, temporelles, ou même des paramètres de contrôle (comme le nombre de Reynolds) du problème à résoudre.

La philosophie PGD, initialement appelée 'radial time-space approximation', à été introduite pour la première fois par Ladèveze ([Ladèveze, 1999]) dans le contexte de la méthode LATIN (LARGE Time INcrement method) afin de réduire les coûts de calcul aussi bien en terme de stockage mémoire qu'en terme de temps de simulation. La méthode LATIN qui a été développée pour traiter des problèmes instationnaires de mécanique non-linéaire consiste à résoudre un problème non-linéaire local suivi d'un problème linéaire global, et cela de manière itérative jusqu'à convergence vers la solution du problème. La résolution du problème global linéaire pouvant s'avérer très lourde, l'introduction de la PGD pour la résolution de ce problème à permis l'utilisation de la méthode LATIN dans des cas où une résolution classique aurait été impossible. Dans ce cas la solution  $u(\mathbf{x}, t)$  du problème défini sur un domaine

spatio-temporel est recherchée sous la forme suivante

$$u(\mathbf{x}, t) \approx u_m(\mathbf{x}, t) = \sum_{i=1}^m F_1^i(\mathbf{x}) F_2^i(t)$$

Ainsi, Champaney et al. utilisent la méthode dans le cadre de problèmes d'assemblages avec un grand nombre de contact [Champaney et al., 1999] et pour faire une analyse paramétrique de joints conçus pour des applications aérospatiales [Champaney et al., 2008]. Guidault et al. [Guidault et al., 2008] appliquent cette notion de problème local/global pour étudier la propagation de fissure. La méthode a également été appliquée par Cresta et al. [Cresta et al., 2007], pour simuler des comportements de matériaux en flambage. Caignot et al. [Caignot et al., 2010] ont développé un outil de test afin de prédire l'apparition d'humidité dans les joints du lanceur d'Ariane 5. Enfin, Violeau et al. [Violeau et al., 2009] appliquent la méthode pour la simulation de matériaux composites dont le comportement peut être décrit sur des échelles microscopiques. La PGD s'avère également efficace pour résoudre les problèmes mécaniques impliquant différentes échelles. Ces problèmes correspondent aux cas où le milieu étudié est fortement hétérogène et où la solution du problème local implique des phénomènes sur de très courtes longueurs d'ondes aussi bien temporelles que spatiales. Une manière de traiter ces problèmes vise à mettre en place un échange d'informations entre les échelles microscopiques et macroscopiques aussi bien en temps qu'en espace en utilisant la théorie d'homogénéisation. L'utilisation de la PGD couplée à la méthode LATIN permet d'obtenir des résultats très satisfaisants en diminuant fortement le temps de calcul et le stockage [Ladevèze et Nouy, 2003, Nouy et Ladevèze, 2004]. Plus récemment, et toujours dans le cadre de problèmes d'homogénéisation, Lamari et al. [Lamari et al., 2010] ont appliqué la PGD afin de déterminer dans le cas linéaire le tenseur de conductivité thermique homogénéisé pour une microstructure composée de cellule en choisissant de définir la conductivité de ces cellules comme une coordonnée supplémentaire du problème.

La PGD a également été appliquée pour résoudre des équations aux dérivées partielles stochastiques. Dans ce contexte la solution  $u$  d'un problème stochastique donnée est recherchée sous la forme suivante :

$$u \approx \sum_{i=1}^m \lambda^i F^i$$

où les  $F^i$  sont des fonctions déterministes pondérées par les coefficients aléatoires  $\lambda^i$ . Nouy [Nouy, 2007, Nouy, 2008] applique la méthode pour résoudre des problèmes stochastiques linéaires (dans ces travaux, la PGD est nommée Generalized Spectral Decomposition) en considérant la décomposition comme la solution d'un problème d'optimisation de fonctionnelle qui peut être interprété comme un quotient de Rayleigh 'étendu' associé à un problème aux valeurs propres. Nouy montre que la méthode permet d'économiser du temps de calcul et de l'espace de stockage comparé aux méthodes de type Krylov traditionnellement utilisées dans ce contexte. Dans [Nouy, 2008], la méthode est étendue à un plus grand nombre d'équations stochastiques et la stratégie numérique de résolution a également été révisée. Enfin, Nouy et Le Maître [Nouy et Le Maître, 2007] ont proposé une extension de la PGD à la résolution d'équations différentielles stochastiques non-linéaires. Contrairement à ce qui avait

été fait dans [Nouy, 2008], la construction de la décomposition de la solution est directement construite à partir du problème non-linéaire initial.

Un autre domaine où la PGD a montré ses capacités est la résolution de problème multidimensionnels. Dans ce cas, la solution  $u(x_1, \dots, x_N)$  d'un problème donné est recherchée sous forme totalement séparée. On définit donc l'approximation d'ordre  $m$  de la solution comme :

$$u_m(x_1, \dots, x_N) = \sum_{i=1}^m F_1^i(x_1) \dots F_N^i(x_N),$$

où les variables  $x_1, \dots, x_N$  peuvent représenter des variables spatiales, temporelles, ou même des paramètres de contrôle (comme le nombre de Reynolds) du problème à résoudre. En effet, le fait de rechercher la solution sous forme de décomposition sur chacune des variables peut rendre le problème beaucoup plus léger (en terme de stockage et de temps de calcul) à résoudre. C'est pourquoi, Ammar et al. [Ammar et al., 2006a, Ammar et al., 2007, Ammar et al., 2010e] ont introduit la PGD pour traiter des problèmes de description de fluides à l'échelle microscopique par le biais de la résolution de l'équation de Fokker-Planck. Dans ce genre de description le nombre de degré de libertés associé à une molécule (représentée comme un simple ressort) est de sept (trois dimensions spatiales, une dimension temporelle, et trois autres coordonnées servant à définir l'orientation et l'allongement des ressort). Si de plus on considère qu'une molécule est constituée de  $S$  ressorts connectés entre eux, le problème est donc de dimension  $(3+1+3S)$ . Dans ce genre de problème, l'utilisation de méthodes déterministes à maillages fixes semble être la méthode la plus adaptée. En effet, si l'on souhaite modéliser le comportement d'une molécule dans un espace de dimension  $N = 80$  (cela correspond à moins de trente ressorts interconnectés alors qu'en réalité il en faudrait plusieurs milliers), et que chaque dimension est maillée par dix éléments, le problème à résoudre est de taille  $10^{80}$ , ce qui représente le nombre supposé de particules élémentaires dans l'univers. C'est pourquoi une stratégie permettant de traiter ces problèmes fortement multidimensionnels est nécessaire pour pouvoir résoudre ce type de problèmes. Dans [Ammar et al., 2006a], la PGD est introduite comme une méthode permettant de rechercher la solution sous forme d'une décomposition tensorielle sur chacune des variables en résolvant de manière itérative des problèmes 1D dans chacune des directions. La méthode a permis de résoudre l'équation de Fokker-Planck en utilisant un nombre d'inconnues égales à  $10^5$  alors que la résolution classique aurait nécessité un maillage de taille  $10^{40}$ . Dans [Ammar et al., 2007], le temps est intégré comme une variable supplémentaire de la décomposition et différents cas test sont traités. Chinesta et al. [Chinesta et al., 2008] passent en revue différentes descriptions physiques impliquant des échelles différentes (mécanique quantique, dynamique moléculaire, dynamique Brownienne, théorie cinétique) et mettent l'accent sur les difficultés rencontrées dues au grand nombre de degrés de liberté qui composent ces problèmes, et donc sur les avantages de la méthode PGD à traiter ces problèmes. Chinesta et al. [Chinesta et al., 2010a] traitent le cas d'une décomposition sur des variables spatiales et temporelles et rajoutent un paramètre de contrôle correspondant à un coefficient d'une équation. Cette représentation leur permet, en effectuant une seule simulation, de traiter le cas de différentes espèces (une espèce par paramètre) dans un modèle cinétique. Toujours dans le cas de la résolution de problèmes dépendant de paramètres, Prulière et al. [Pruliere et al., 2010] appliquent la PGD

à la résolution d'équations aux dérivées partielles paraboliques non linéaires. De même Chinesta et al. [Chinesta et al., 2010b] analysent un problème parabolique couplé à un système d'équations cinétiques locales. Une séparation de type PGD couplée avec la méthode LATIN est réalisée en rajoutant comme variable du problème local une coordonnée dépendant des différentes espèces mises en jeu dans le modèle cinétique. Enfin la méthode a été appliquée à la résolution de l'équation de Schrodinger en traitant des cas avec un seul noyau et jusqu'à 50 électrons. Dans [Ammar et al., ], la méthode a été appliquée en enrichissant certaines zones du domaine de résolution d'une discrétisation éléments finis afin de pouvoir capter des comportements locaux. Les auteurs montrent en effet que la PGD peut avoir quelques difficultés à capturer des comportements extrêmement locaux. Une manière de résoudre ce problème est d'utiliser une méthode hybride combinant une résolution éléments finis et une résolution PGD. En effet, en rajoutant aux endroits où se produisent ces phénomènes un maillage type éléments finis et en résolvant le modèle complet localement, il est possible de simuler ces comportements. Cette adaptation étant faite dans l'objectif de capter des discontinuités dans les solutions, ce qui est impossible dans la forme la plus simple de la PGD. Cette formulation hybride présente également l'avantage de pouvoir imposer des types de conditions limites différentes sur une surface. En effet la formulation la plus simple de la PGD présente le désavantage de ne pouvoir imposer les conditions limites que sur des 'tranches', c'est à dire que si l'on suppose une décomposition dans un domaine  $\Omega = ]-L, L[^2$ , le type de conditions limites (Dirichlet ou Neumann) que l'on pourra imposer sur une bande  $(x_0, y_j); j = 1, \dots, N$ , ne pourra être différents d'un point à l'autre. Cette particularité vient directement de la forme de la décomposition. Ce problème a été traité par Bonithon et al. [Bonithon et al., 2011] où une méthode hybride combinant la PGD et la BEM (Boundary Element Method) a été développée.

Parmi les travaux les plus récents concernant la PGD, Nouy [Nouy, 2010] effectue une description détaillée de la méthode et des différentes définitions de la décomposition, l'une basée sur une formulation de Galerkin, l'autre basée sur une minimisation du résidu. Les résultats obtenus par PGD sont ensuite comparés aux résultats obtenus en appliquant une résolution POD. Nouy montre que dans de nombreux cas (diffusion pure, advection-diffusion, advection-diffusion-dispersion, advection-diffusion-réaction) les propriétés de convergence de la POD et de la PGD sont similaires, mais que, pour certains problèmes (advection-diffusion), la PGD ne retrouvait pas exactement les mêmes modes que ceux de la POD. Cette différence pourrait être due aux algorithmes utilisés pour résoudre le pseudo problème aux valeurs propres associé à la PGD. Gonzalez et al. [Gonzalez et al., 2010] ont traité des problèmes disposant de conditions aux limites non-homogènes en définissant un changement de variable approprié ainsi que des problèmes à géométrie complexe en utilisant la méthode des fonctions caractéristiques. On peut également citer les travaux d'Ammar et al. [Ammar et al., 2010d] où l'identification inverse de comportement de fluides complexes est menée pour une grande gamme de paramètres. Dans ce cas, les variables du problème sont les coordonnées spatiales, le temps et les différents paramètres. Les résultats obtenus semblent indiquer que la PGD peut être une alternative intéressante aux méthodes déjà existantes pour la résolution de ce type de problèmes. Dans [Chinesta et al., 2010c], Chinesta et al. dressent un inventaire des avancées récentes effectuées dans le domaine de la résolution de problèmes multidimensionnels par PGD.

D'un point de vue plus théorique, Ammar et al. [Ammar et al., 2010c] ont démontré que la solution du problème obtenue par PGD convergeait toujours vers la solution du modèle com-

plet. Dans [Ammar et al., 2010a] une première approche de la caractérisation de la précision de la méthode PGD est effectuée en montrant que le résidu du problème n'est pas le meilleur critère et en définissant un nouvel estimateur d'erreur basé sur la solution du problème dual.

### III Description de la méthode PGD

En pratique il existe trois façons d'utiliser la PGD pour résoudre un problème transitoire :

- 1) La première consiste en une décomposition spatio-temporelle :

$$u(\mathbf{x}, t) = \sum_i F_{\mathbf{x}_i}^i(\mathbf{x}) F_{t_i}^i(t)$$

Dans ce cas, la solution est recherchée sous la même forme qu'avec la décomposition APR mais en utilisant une approche non-incrémentale. Ici, la solution est directement recherchée sous forme de décomposition espace-temps. L'inconvénient majeur de cette formulation réside dans le fait que l'on doit recourir à une discrétisation spatiale sur toute la grille de calcul afin de déterminer les fonctions  $F_{\mathbf{x}_i}^i$  et donc à la résolution de problèmes de grande taille.

- 2) Afin de contourner cette difficulté, la seconde possibilité consiste à écrire une décomposition sur chacune des variables de l'espace physique :

$$u(x, y, t) = \sum_i F_{x_i}^i(x) F_{y_i}^i(y) F_{t_i}^i(t)$$

En pratique cette décomposition conduit à une augmentation significative du nombre de terme de la somme.

- 3) Enfin, la dernière possibilité, qui sera utilisée dans la suite de ce travail, consiste à garder une approche incrémentale en temps et d'effectuer la séparation, à chaque pas de temps, uniquement sur les variables spatiales :

$$u^t(x, y) = \sum_i F_{x_i}^i(x) F_{y_i}^i(y)$$

Nous allons maintenant décrire plus en détail deux types de décomposition, la décomposition espace-temps (1) en utilisant un formalisme proche de celui utilisé par Nouy dans [Nouy, 2010], et la décomposition sur chacune des variables (2-3) permettant de traiter des problèmes multi-dimensionnels. Enfin, dans une dernière partie, nous expliciterons une formulation algébrique du problème.

#### III.1 PGD en décomposition spatio-temporelle

On cherche à résoudre le problème suivant où  $\mathcal{A}$  est linéaire<sup>4</sup> :

$$\text{Trouver } u(x, t) \in \Omega \times ]0, T] \text{ tel que } \begin{cases} \frac{\partial u}{\partial t} + \mathcal{A}(u) = f \\ u(0) = u^0 \end{cases} \quad (1.50)$$

---

<sup>4</sup>s'il ne l'est pas il est nécessaire de le linéariser

Dans le cas d'une décomposition spatio-temporelle, la PGD consiste à rechercher une décomposition de la solution  $u(\mathbf{x}, t)$  d'un problème donné sous la forme

$$u(\mathbf{x}, t) \approx u_m(\mathbf{x}, t) = \sum_{i=1}^m \phi_i(\mathbf{x}) a_i(t) \quad (1.51)$$

où les  $\phi_i \in \Omega$  sont les fonctions spatiales et les  $a_i \in ]0, T]$  les fonctions temporelles. De cette manière la famille des  $a_i$  (respectivement des  $\phi_i$ ) peut être interprétée comme une base réduite de fonctions temporelles (respectivement de fonctions spatiales).

Dans la suite nous allons décrire quelques variantes de la méthode PGD. Dans un premier temps nous décrirons la PGD dite progressive, ensuite nous nous intéresserons à la PGD dite optimale, enfin nous étudierons un compromis des deux que nous appellerons la PGD progressive pondérée.

### a) Méthode PGD progressive

La décomposition à l'ordre  $m - 1$  est supposée connue telle que :

$$u_{m-1} = \sum_{i=1}^{m-1} \phi_i a_i$$

Pour définir la décomposition à l'ordre  $m$  un nouveau couple  $(\phi_m, a_m) \in \Omega \times ]0, T]$  est recherché, tel que

$$u_m = u_{m-1} + \phi_m a_m = \sum_{i=1}^{m-1} \phi_i a_i + \phi_m a_m \quad (1.52)$$

En introduisant (1.52) dans (1.50), on obtient :

$$\frac{\partial u_{m-1}}{\partial t} + \mathcal{A}(u_{m-1}) + \phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m) = f + Res^m \quad (1.53)$$

où  $Res^m$  est le résidu d'ordre  $m$  qui apparait naturellement car  $u_m$  est une approximation de  $u$ . L'équation précédente peut être réécrite :

$$\phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m) = f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}) + Res^m \quad (1.54)$$

Afin de déterminer  $\phi_m$  et  $a_m$ , une projection de (1.54) sur chacune des inconnues  $a_m$  et  $\phi_m$ , est effectuée, soit

$$\left( \phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m), a_m \right)_{L^2(]0, T])} = \left( f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}), a_m \right)_{L^2(]0, T])} + (Res^m, a_m)_{L^2(]0, T])} \quad (1.55)$$

et

$$\left( \phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m), \phi_m \right)_{L^2(\Omega)} = \left( f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}), \phi_m \right)_{L^2(\Omega)} + (Res^m, \phi_m)_{L^2(\Omega)} \quad (1.56)$$

où  $(\bullet, \bullet)_{L^2(\Omega)}$  (respectivement  $(\bullet, \bullet)_{L^2([0, T])}$ ) sont les produits scalaires définis de manière usuelle sur  $L^2(\Omega)$ <sup>1</sup> (respectivement  $L^2([0, T])$ ).

On impose alors que le résidu  $Res^m$  soit orthogonal aux  $a_m$  et aux  $\phi_m$ , on a donc

$$\left( \phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m), a_m \right)_{L^2([0, T])} = \left( f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}), a_m \right)_{L^2([0, T])} \quad (1.57)$$

$$\left( \phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m), \phi_m \right)_{L^2(\Omega)} = \left( f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}), \phi_m \right)_{L^2(\Omega)} \quad (1.58)$$

Pour obtenir les nouvelles fonctions  $a_m$  et  $\phi_m$ , les équations (1.57) et (1.58) doivent être vérifiées simultanément. L'algorithme naturel permettant de faire cela est l'algorithme des puissances itérées (ou du point fixe). L'algorithme associé à la méthode PGD progressive est résumé dans la table 1.2. Les fonctions  $S_n$  et  $T_n$  intervenant dans cet algorithme sont définies de la façon suivante :

- $S_m : [0, T] \rightarrow \Omega$  est l'application qui associe à une fonction temporelle  $a \in [0, T]$  une fonction spatiale  $\phi = S_m(a) \in \Omega$  définie par l'équation (1.57).
- $T_m : \Omega \rightarrow [0, T]$  est l'application qui associe à une fonction spatiale  $\phi \in \Omega$  une fonction temporelle  $a = T_m(\phi) \in [0, T]$ , définie par l'équation (1.58).

<ol style="list-style-type: none"> <li>1. <b>tant que</b> (<math>m \leq m_{max}</math> et <math>\ Res^m\  \geq \epsilon</math>) <b>faire</b></li> <li>2.     Initialisation de <math>a^0</math> (généralisé de façon aléatoire)</li> <li>3.     <b>tant que</b> (<math>k \leq k_{max}</math> et <math>\ Err^k\  \geq \eta</math>) <b>faire</b></li> <li>4.         Calcul de <math>\phi^k = S_m(a^{k-1})</math></li> <li>5.         Normalisation de <math>\phi</math> (optionnel)</li> <li>6.         Calcul de <math>a^k = T_m(\phi^k)</math></li> <li>7.         Calcul de <math>\ Err^k\  = \ \phi^k a^k - \phi^{k-1} a^{k-1}\ _{L^2(\Omega \times [0, T])}</math></li> <li>8.     <b>fin tant que</b></li> <li>9.     On pose <math>\phi_m = \phi</math> et <math>a_m = a</math></li> <li>10.    On pose <math>u_m = u_{m-1} + \phi_m a_m</math> et calcul de <math>\ Res^m\ </math></li> <li>11. <b>fin tant que</b></li> </ol>
---

TAB. 1.2 – Algorithme associé à la méthode PGD progressive ( $\epsilon$  et  $\eta$  définis par l'utilisateur)

### Remarque 1

On peut montrer que la démarche permettant d'aboutir à la résolution simultanée des équations (1.57) et (1.58) est équivalente à la formulation variationnelle suivante :

<sup>1</sup>on aurait aussi bien pu choisir un autre produit scalaire

$$\int_{\Omega \times [0, T]} u^* \left( \frac{\partial u}{\partial t} + \mathcal{A}(u) \right) = \int_{\Omega \times [0, T]} u^* f \quad \forall u^* \in H_0^1(\Omega) \times L^2([0, T])$$

où  $u^* = \phi^* a + \phi a^*$ .

Effectuer un point fixe sur les inconnues  $a$  et  $\phi$  de l'équation précédente est donc équivalent à effectuer un point fixe sur les équations (1.57) et (1.58) définies auparavant.

### Remarque 2

Ici, le critère de convergence de la méthode est basé sur la minimisation du résidu au sens de la norme de  $L^2$ . Cela est équivalent à minimiser l'erreur au sens de la norme de l'opérateur. En effet, soit le problème suivant à résoudre :

$$Au = f \tag{1.59}$$

L'erreur associée à ce problème est définie par :

$$u - A^{-1}f = 0 \tag{1.60}$$

Minimiser cette erreur par rapport à la norme de l'opérateur revient à écrire :

$$\begin{aligned} \frac{d}{du} [(u - A^{-1}f)^T A(u - A^{-1}f)] &= 0 \\ \Rightarrow \frac{d}{du} [u^T Au - u^T f - f^T A^{-1T} Au - f^T A^{-1T} f] &= 0 \end{aligned}$$

et donc, dans le cas où  $A$  est symétrique

$$Au - f = 0$$

On a montré ici que, dans le cas où l'opérateur est symétrique, minimiser l'erreur du problème par rapport à la norme de l'opérateur revenait à minimiser la norme  $L^2$  du résidu du problème (1.59).

L'équivalence précédente est valable, dans le cas où l'opérateur  $A$  est symétrique. Si ce n'est pas le cas, l'erreur (1.60) peut être minimisée par rapport à la norme associée à l'opérateur  $A^T A$ , on obtient :

$$\frac{d}{du} [(u - A^{-1}f)^T A^T A(u - A^{-1}f)] = 0$$

et donc

$$\frac{d}{du} [(Au - f)^T (Au - f)] = 0$$

d'où

$$A^T Au - A^T f = 0$$

Cette fois-ci, minimiser l'erreur par rapport à la norme de l'opérateur ( $A^T A$ ) revient à minimiser la norme  $L^2$  du résidu du problème préconditionné suivant :

$$A^T Au = A^T f$$

Dans notre cas, les opérateurs seront toujours symétriques, nous nous contenterons donc de la première formulation.

### b) Méthode PGD Galerkin

Généralement la décomposition précédente n'est pas optimale au sens de la projection de Galerkin. Pour qu'elle le soit il faut imposer que le résidu de la décomposition soit simultanément orthogonal à l'ensemble des fonctions spatiales  $\Phi_m = \{\phi_1, \dots, \phi_m\}$  et à l'ensemble des fonctions temporelles  $A_m = \{a_1, \dots, a_m\}$ . Ces deux critères d'orthogonalité sont formulés dans la suite au travers de la définition des deux fonctions  $T$  et  $S$  suivantes :

- Soit  $S : ([0, T])^m \rightarrow \Omega^m$  qui associe aux fonctions temporelles  $A_m \in ([0, T])^m$  les fonctions spatiales  $\Phi_m = S(A_m) \in \Omega^m$  et définie par

$$\left( \left[ \sum_{i=1}^m \phi_i \frac{da_i}{dt} + \mathcal{A} \left( \sum_{i=1}^m \phi_i a_i \right) \right], a_k \right)_{L^2([0, T])} = (f, a_k)_{L^2([0, T])} \quad \text{pour } 1 \leq k \leq m \quad (1.61)$$

- Soit  $T : \Omega^m \rightarrow ([0, T])^m$  qui associe aux fonctions spatiales  $\Phi_m \in \Omega^m$  les fonctions temporelles  $A_m = T(\Phi_m) \in ([0, T])^m$  et définie par

$$\left( \left[ \sum_{i=1}^m \phi_i \frac{da_i}{dt} + \mathcal{A} \left( \sum_{i=1}^m \phi_i a_i \right) \right], \phi_k \right)_{L^2(\Omega)} = (f, \phi_k)_{L^2(\Omega)} \quad \text{pour } 1 \leq k \leq m \quad (1.62)$$

La décomposition  $u_m = \Phi_m \cdot A_m$  qui vérifiera les équations (1.61) et (1.62) simultanément sera donc considérée comme la décomposition optimale au sens de la projection de Galerkin.

De la même manière que dans le cas précédent, il nous faut résoudre par la méthode du point fixe, successivement le système d'équations (1.61) et (1.62), afin d'obtenir les familles de fonctions  $\Phi_m$  et  $A_m$  qui permettront de reconstruire la solution à l'ordre  $m$ . Il est clair par contre que recalculer toutes les fonctions de la base à chaque enrichissement sera beaucoup plus coûteux que dans le cas progressif, où seulement la dernière fonction de la base était calculée à chaque enrichissement. L'algorithme de la méthode PGD Galerkin est résumé dans la table 1.3.

<ol style="list-style-type: none"> <li>1. <b>tant que</b> (<math>m \leq m_{max}</math> et <math>\ Res^m\  \geq \epsilon</math>) <b>faire</b></li> <li>2.     Initialisation de <math>A_m^0</math> (généralé aléatoirement)</li> <li>3.     <b>tant que</b> (<math>k \leq k_{max}</math> et <math>\ Err^k\  \geq \eta</math>) <b>faire</b></li> <li>4.         Calcul de <math>\Phi_m^k = S(A_m^{k-1})</math></li> <li>5.         Calcul de <math>A_m^k = T(\Phi_m^{k-1})</math></li> <li>6.         Calcul de <math>\ Err^k\  = \ \Phi_m^k \cdot A_m^k - \Phi_m^{k-1} \cdot A_m^{k-1}\ _{L^2(\Omega^m \times [0, T]^m)}</math></li> <li>7.         <b>fin tant que</b></li> <li>8.     On pose <math>u_m = \Phi_m A_m</math> et calcul de <math>\ Res^m\ </math></li> <li>9.     <b>fin tant que</b></li> </ol>
---

TAB. 1.3 – Algorithme associé à la méthode PGD Galerkin avec itérations successives sur les sous-espaces ( $\epsilon$  et  $\eta$  sont définis par l'utilisateur)

### c) Méthode PGD Galerkin avec mise à jour

L'algorithme précédent n'est pas utilisable en l'état, puisque l'application  $S$  est très lourde à mettre en place. En effet cela revient à résoudre un système d'équations aux dérivées partielles couplées dépendant du temps. On cherche donc à avoir une "décomposition optimale" sans avoir à se servir de l'application  $S$ . Pour ce faire, le sous espace  $\Phi_m$  est construit à l'aide des étapes de l'algorithme de la méthode PGD progressive, puis le sous espace  $A_m$  est construit par le biais de l'application  $T$ . On obtient alors l'algorithme résumé dans la table 1.4.

Le problème est résolu en enrichissant la base pour le calcul des fonctions spatiales (la base spatiale est calculée de manière progressive), mais la base temporelle est quant à elle entièrement modifiée à chaque étape (elle est calculée de manière optimale).

<ol style="list-style-type: none"> <li>1. <b>tant que</b> (<math>m \leq m_{max}</math> <b>et</b> <math>\ Res^m\  \geq \epsilon</math>) <b>faire</b></li> <li>2.     <i>Initialisation de <math>a^0</math> (généralisé aléatoirement)</i></li> <li>3.     <b>tant que</b> (<math>k \leq k_{max}</math> <b>et</b> <math>\ Err^k\  \geq \eta</math>) <b>faire</b></li> <li>4.         <i>Calcul de <math>\phi^k = S_m(a^{k-1})</math></i></li> <li>5.         <i>Calcul de <math>a^k = T_m(\phi^k)</math></i></li> <li>6.         <i>Calcul de <math>\ Err^k\  = \ \phi^k a^k - \phi^{k-1} a^{k-1}\ _{L^2(\Omega \times [0, T])}</math></i></li> <li>7.     <b>fin tant que</b></li> <li>8.     <i>On pose <math>\Phi_m = (\Phi_{m-1}, \phi^k)</math></i></li> <li>9.     <i>Calcul de <math>A_m = T(\Phi_m)</math></i></li> <li>10.    <i>On pose <math>u_m = \Phi_m \cdot A_m</math> et on calcul de <math>\ Res^m\ </math></i></li> <li>11. <b>fin tant que</b></li> </ol>
--

TAB. 1.4 – Algorithme associé à la méthode PGD Galerkin avec mise à jour ( $\epsilon$  et  $\eta$  sont définis par l'utilisateur)

### d) PGD progressive pondérée

La méthode présentée dans ce paragraphe n'est qu'une extension de la PGD progressive dans laquelle une normalisation des fonctions calculées et un calcul de coefficients de pondération par le biais d'une projection sont ajoutés. L'ajout de ces coefficients de pondération permet de rendre la décomposition "optimale" au sens de la projection de Galerkin. La méthode consiste à calculer les fonctions spatiales et temporelles de manière progressive puis à pondérer chacun des couples  $\phi_i a_i$  par un coefficient  $\alpha_i$  (ces coefficients sont tous recalculés à chaque itération) en résolvant le problème associé à la condition d'orthogonalité du résidu par rapport à chacune des fonctions spatiales et temporelles. La solution est donc recherchée sous la forme

$$u_m = \sum_{i=1}^m \alpha_i \phi_i(\mathbf{x}) a_i(t)$$

où  $\phi_i \in \Omega$ ,  $a_i \in [0, T]$ , et  $\alpha_i \in \mathbb{R}$ .

On suppose que la décomposition à l'ordre  $(m-1)$  est connue. Pour définir la décomposition à l'ordre  $m$  un nouveau couple  $(\phi_m, a_m) \in \Omega \times [0, T]$  et un nouveau jeu de coefficients

$(\alpha_i)_{1 \leq i \leq m} = \Lambda_i$  est recherché. Dans un premier temps, les fonctions  $(\phi_m, a_m)$  sont cherchées en imposant que  $\alpha_m = 1$ , ce qui revient à écrire :

$$u_m = u_{m-1} + \phi_m a_m \quad (1.63)$$

En injectant  $u_m$  dans le problème initial (1.50), cela donne

$$\phi_m \frac{da_m}{dt} + \mathcal{A}(\phi_m a_m) = f - \frac{\partial u_{m-1}}{\partial t} - \mathcal{A}(u_{m-1}) + Res^m \quad (1.64)$$

Ce problème à déjà été résolu dans le cas de la PGD progressive. Les fonctions  $(\phi_m, a_m)$  sont calculées exactement de la même manière. Une fois les fonctions  $\{\phi_i\}_{1 \leq i \leq m}$  et  $\{a_i\}_{1 \leq i \leq m}$  connues, les coefficients  $\{\alpha_i\}_{1 \leq i \leq m}$  sont recherchés tels que :

$$u_m = \sum_{i=1}^m \alpha_i \phi_i a_i \quad (1.65)$$

En injectant cette forme de la solution dans le problème (1.50), on obtient :

$$\sum_{i=1}^m \alpha_i \phi_i \frac{da_i}{dt} + \mathcal{A}\left(\sum_{i=1}^m \alpha_i \phi_i a_i\right) = f \quad (1.66)$$

Le système d'équations linéaires vérifié par les coefficients  $\{\alpha_i\}_{1 \leq i \leq m}$  s'obtient en projetant l'équation précédente sur chaque  $a_k \phi_k$ . Ainsi, le système s'écrit pour  $1 \leq k \leq m$  :

$$\left( \sum_{i=1}^m \alpha_i \phi_i \frac{da_i}{dt}, a_k \phi_k \right)_{L^2([0,T] \times \Omega)} + \left( \mathcal{A}\left(\sum_{i=1}^m \alpha_i \phi_i a_i\right), a_k \phi_k \right)_{L^2([0,T] \times \Omega)} = (f, a_k \phi_k)_{L^2([0,T] \times \Omega)} \quad (1.67)$$

L'algorithme de résolution de cette version de la méthode PGD est répertorié dans la table 1.5.

<ol style="list-style-type: none"> <li>1. <b>tant que</b> (<math>m \leq m_{max}</math> et <math>\ Res^m\  \geq \epsilon</math>) <b>faire</b></li> <li>2.     Initialisation de <math>a_m^0</math> (généralé aléatoirement)</li> <li>3.     <b>tant que</b> (<math>k \leq k_{max}</math> et <math>\ Err^k\  \geq \eta</math>) <b>faire</b></li> <li>4.         Calcul de <math>\phi^k = S_m(a^{k-1})</math></li> <li>5.         Calcul de <math>a^k = T_m(\phi^k)</math></li> <li>6.         Calcul de <math>\ Err^k\  = \ \phi^k a^k - \phi^{k-1} a^{k-1}\ _{L^2(\Omega \times [0,T])}</math></li> <li>7.     <b>fin tant que</b></li> <li>8.     Normalisation des fonctions <math>\phi^k, a^k</math></li> <li>9.     On pose <math>\Phi_m = (\Phi_{m-1}, \phi^k)</math></li> <li>10.    On pose <math>A_m = (A_{m-1}, a^k)</math></li> <li>11.    Calcul de <math>\Lambda_m = U(A_m, \Phi_m)</math></li> <li>12.    On pose <math>u_m = \sum_{i=1}^m \alpha_i \phi_i a_i</math> et calcul de <math>\ Res^m\ </math></li> <li>13. <b>fin tant que</b></li> </ol>
---

TAB. 1.5 – Algorithme associé à la méthode PGD progressive pondérée ( $\epsilon$  et  $\eta$  sont définis par l'utilisateur)

Où nous avons noté  $U : ([0, T])^m \times \Omega^m \rightarrow \mathbb{R}^m$  l'application qui associe aux fonctions temporelles  $A_m = \{a_i\}_{1 \leq i \leq m} \in ([0, T])^m$  et spatiales  $\Phi_m = \{\phi_i\}_{1 \leq i \leq m} \in \Omega^m$ , les coefficients

$\Lambda_m = \{\alpha_i\}_{1 \leq i \leq m} \in \mathbb{R}^m$ , définie par l'équation (1.67). De cette manière le calcul des fonctions de la base spatiale et temporelle est relativement simple, puisque effectué de manière progressive. Quand au calcul des coefficients qui servent à orthogonaliser le résidu par rapport aux espaces engendrés par les fonctions spatiales et temporelles, il est peu coûteux. En effet,  $\Lambda_m$  contient  $m$  éléments, cela revient donc à résoudre un système linéaire de taille  $m \times m$ , ce qui n'était pas le cas dans la méthode optimale présentée précédemment.

### e) Synthèse des différentes méthodes PGD

Les quatre méthodes décrites précédemment, à savoir, la PGD progressive, la PGD Galerkin, la PGD Galerkin avec mise à jour et la PGD pondérée sont résumées dans la table 1.6.

PGD progressive	PGD Galerkin
1. Initialisation de $a^0$ 2. point fixe $\begin{cases} \phi^{(k)} = S_m(a^{(k-1)}) \\ a^{(k)} = T_m(\phi^{(k)}) \end{cases}$ 3. $\phi_m = \phi^{(k)}$ , $a_m = a^{(k)}$ 4. $u_m = u_{m-1} + \phi_m a_m$ jusqu'à convergence	1. Initialisation de $a^0$ 2. point fixe $\begin{cases} \Phi^{(k)} = S(A^{(k-1)}) \\ A^{(k)} = T(\Phi^{(k-1)}) \end{cases}$ 3. $\Phi_m = \Phi^{(k)}$ , $A_m = A^{(k)}$ 4. $u_m = \Phi_m \cdot A_m$ jusqu'à convergence
PGD Galerkin avec mise à jour	PGD progressive pondérée
1. Initialisation de $a^0$ 2. point fixe $\begin{cases} \phi^{(k)} = S_m(a^{(k-1)}) \\ a^{(k)} = T_m(\phi^{(k)}) \end{cases}$ 3. $\Phi_m = (\Phi_{m-1}, \phi^{(k)})$ 4. $A_m = T(\Phi_m)$ 5. $u_m = \Phi_m \cdot A_m$ jusqu'à convergence	1. Initialisation de $a^0$ 2. point fixe $\begin{cases} \phi^{(k)} = S_m(a^{(k-1)}) \\ a^{(k)} = T_m(\phi^{(k)}) \end{cases}$ 3. $\phi_m = \frac{\phi^{(k)}}{\ \phi^{(k)}\ }$ , $a_m = \frac{a^{(k)}}{\ a^{(k)}\ }$ 4. $\Phi_m = (\Phi_{m-1}, \phi_m)$ 5. $A_m = (A_{m-1}, a_m)$ 6. $\Lambda_m = U(A_m, \Phi_m)$ 7. $u_m = \sum_{i=1}^m \alpha_i \phi_i a_i$ jusqu'à convergence

TAB. 1.6 – Synthèse des différentes méthodes PGD

### III .2 PGD progressive pondérée appliquée à des problèmes multidimensionnels

Dans la suite de ce mémoire, la méthode utilisée est la 'PGD progressive pondérée'. Nous allons maintenant détailler cette méthode dans le cas où la solution est recherchée non plus comme le produit de deux fonctions mais comme le produit de plusieurs fonctions, dépendant chacune d'une seule variable. Ces variables peuvent aussi bien être des variables temporelles, spatiales ou des paramètres du problème.

Soit le problème suivant :

$$\text{Chercher } u(x_1, x_2, \dots, x_N), \text{ tel que } \begin{cases} \mathcal{A}(u) = f \\ u(0) = u^0 \end{cases} \quad (1.68)$$

où  $\mathcal{A}$  est l'opérateur linéaire du problème,  $(x_1, x_2, \dots, x_N) \in \Omega_1 \times \Omega_2 \times \dots \times \Omega_N$ . La PGD multidimensionnelle consiste donc à rechercher  $u$  sous la forme séparée suivante :

$$u(x_1, \dots, x_N) \approx u_m(x_1, \dots, x_N) = \sum_{i=1}^m \alpha^i \prod_{j=1}^N F_j^i(x_j) \quad (1.69)$$

où  $F_j^i$  est la  $i^{\text{eme}}$  fonction de base, normée, dépendant uniquement de la variable  $x_j$  et  $\alpha^i \in \mathbb{R}$  des coefficients de pondération.

De la même manière que dans le paragraphe précédent, le calcul des fonctions  $F_j^i$ , et des coefficients  $\alpha^i$  se fait de manière itérative où à chaque itération les trois étapes suivantes sont nécessaires :

- La première étape appelée **Enrichissement de la base** consiste à calculer les fonctions  $F_j^n$ .
- La seconde, appelée **Etape de projection** consiste à calculer les  $n$  coefficients  $\alpha^i$ .
- Enfin, la dernière étape nommée **Estimation de la convergence** consiste à calculer le résidu du problème pour savoir si l'on a convergé.

Ces trois étapes sont décrites en détails dans la suite :

#### a) Enrichissement de la base de fonctions

Nous sommes à l'étape  $m$ , on suppose la solution  $u_{m-1} = \sum_{i=1}^{m-1} \alpha^i \prod_{j=1}^N F_j^i(x_j)$  à l'ordre  $m-1$  connue. La solution à l'ordre  $m$  est recherchée sous la forme suivante :

$$u_m = \sum_{i=1}^{m-1} \alpha^i \prod_{j=1}^N F_j^i(x_j) + \prod_{j=1}^N R_j(x_j) \quad (1.70)$$

Si l'on injecte cette forme de la solution dans (1.68), on obtient

$$\mathcal{A}(u_{m-1} + \prod_{j=1}^N R_j) = f + Res^m \quad (1.71)$$



1.	<b>tant que</b> ( $k \leq k_{max}$ <b>et</b> $\ Err^k\  \geq \eta$ ) <b>faire</b>
2.	Calcul de $R_1^{(k)} = S_1(R_2^{(k-1)}, \dots, R_N^{(k-1)})$
3.	Calcul de $R_2^{(k)} = S_2(R_1^{(k)}, R_3^{(k-1)}, \dots, R_N^{(k-1)})$
4.	$\vdots$
5.	Calcul de $R_i^{(k)} = S_i(R_1^{(k)}, \dots, R_{i-1}^{(k)}, R_{i+1}^{(k-1)}, \dots, R_N^{(k-1)})$
6.	$\vdots$
7.	Calcul de $R_N^{(k)} = S_N(R_1^{(k)}, \dots, R_{N-1}^{(k)})$
8.	Calcul de $Err^k = \ \prod_{j=1}^N R_j^{(k)}(x_j) - \prod_{j=1}^N R_j^{(k-1)}(x_j)\ $
9.	<b>fin tant que</b>
10.	On pose $F_j^m = \frac{R_j^{(k)}(x_j)}{\ R_j^{(k)}(x_j)\ }$

TAB. 1.7 – Algorithme d’enrichissement associé à la PGD progressive pondérée appliquée à des problèmes multidimensionnels ( $\eta$  est fixé par l’utilisateur)

### b) Projection de la solution sur la base

Les fonctions  $F_j^i$  ( $i = 1, \dots, m$  et  $j = 1, \dots, N$ ) sont connues, les coefficients  $\alpha^i$  sont obtenus, en recherchant la solution  $u_m$  sous la forme suivante :

$$u_m = \sum_{i=1}^m \alpha^i \prod_{j=1}^N F_j^i \quad (1.76)$$

En injectant cette forme de la solution dans (1.68), cela donne

$$\mathcal{A}\left(\sum_{i=1}^m \alpha^i \prod_{j=1}^N F_j^i\right) = f + Res^m \quad (1.77)$$

Le système vérifié par les coefficients  $\alpha_i$  est obtenu en projetant l’équation (1.77) sur chaque fonction  $\prod_{j=1}^N F_j^k(x_j)$  (pour  $k = 1, \dots, m$ ) en supposant que le résidu  $Res^m$  soit orthogonal à chacune de ces fonctions. On obtient alors pour  $k = 1, \dots, m$  :

$$\left( \mathcal{A}\left(\sum_{i=1}^m \alpha^i \prod_{j=1}^N F_j^i\right), \prod_{j=1}^N F_j^k \right)_{L^2} = \left( f, \prod_{j=1}^N F_j^k \right)_{L^2} \quad (1.78)$$

où cette fois ci  $(\cdot, \cdot)_{L^2}$  est le produit scalaire sur  $L^2(\Omega_1 \times \dots \times \Omega_N)$ . Par la suite  $U_k$  désignera l’application définie par l’équation (1.78) et qui associe aux fonctions  $(F_1^k, \dots, F_N^k) \in (\Omega_1, \dots, \Omega_N)$  les  $m$  coefficients  $\alpha_i$ .

Après avoir calculé les  $m$  coefficients  $\alpha_i$  comme solution du système ( $\alpha_k = U_k(F_1^k, \dots, F_N^k)$ ,  $1 \leq k \leq m$ ), la solution  $u_m$ , s’écrit comme indiqué dans l’équation (1.69).

**c) Estimation de la convergence**

Si la norme du résidu  $Res^m$  définie dans l'équation (1.77) est inférieure à un  $\epsilon$  fixé par l'utilisateur alors l'algorithme a convergé. La solution  $u(x_1, \dots, x_N)$  peut alors être approximée par  $u_m$  et s'écrit alors comme dans (1.69). Sinon il faut enrichir la base de fonctions afin de construire  $u_{m+1}$ .

L'algorithme (Tab 1.8) résume les différentes étapes de la méthode.

<ol style="list-style-type: none"> <li>1. <b><i>tant que</i></b> (<math>m \leq m_{max}</math> <b>et</b> <math>\ Res^m\  \geq \epsilon</math>) <b><i>faire</i></b></li> <li>2.     <i>On applique les étapes 1 à 11 de l'algorithme d'enrichissement afin d'obtenir les fonctions <math>F_j^m</math></i></li> <li>3.     <i>On calcule les <math>\alpha_k = U_k(F_1^k \dots, F_N^k)</math>, <math>1 \leq k \leq m</math></i></li> <li>4.     <i>On pose <math>u_m = \sum_{i=1}^m \alpha^i \prod_{j=1}^N F_j^i(x_j)</math> et on calcule <math>\ Res^m\ </math></i></li> <li>5. <b><i>fin tant que</i></b></li> </ol>
---

TAB. 1.8 – Algorithme associé à la PGD progressive pondérée pour des problèmes multidimensionnels ( $\epsilon$  est fixé par l'utilisateur)

Nous avons jusqu'à présent utilisé une formulation continue pour détailler la méthode PGD. Comme nous allons utiliser la PGD discrétisée sur chacune des variables, il est important de donner des détails sur cette dernière décomposition. Nous allons donc présenter la PGD sous une forme tensorielle discrète.

**III .3 PGD progressive pondérée en formulation tensorielle**

**a) Notations et préliminaires**

Dans cette partie, dans un souci de simplicité et sans perdre le caractère général de la méthode, nous étudierons un problème 2D, mais la méthode se généralise aisément pour un plus grand nombre de variables. Soit le problème suivant :

$$\text{Trouver } u(x,y) \text{ tel que } \begin{cases} \mathcal{A}(u) = f & \text{dans } \Omega \\ +\text{Conditions limites} \end{cases} \quad (1.79)$$

où  $\mathcal{A}$  est l'opérateur linéaire du problème, et  $f$  le second membre.

Avant de détailler les trois étapes de résolution de la PGD (enrichissement, projection et estimation de la convergence), nous allons préciser quelques notations. Le problème (1.79) peut être écrit sous forme discrète :

$$\mathcal{A}_h(u_h) = f_h \quad (1.80)$$

avec

$$\mathcal{A}_h = \sum_{j=1}^{n_A} A_x^j \otimes A_y^j \quad f_h = \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j \quad u_h \approx u_m = \sum_{i=1}^m \alpha^i \mathbf{F}^i \otimes \mathbf{G}^i \quad (1.81)$$

L'opérateur  $\mathcal{A}_h$  est discrétisé sous forme de produit tensoriel d'opérateurs  $A_x^j$  et  $A_y^j$  dans la direction  $\mathbf{x}$  et  $\mathbf{y}$  respectivement. La taille de l'opérateur discrétisé  $A_x^j$  est  $N_x \times N_x$  et la taille de l'opérateur discrétisé  $A_y^j$  est  $N_y \times N_y$  où  $N_x$  (respectivement  $N_y$ ) représente le nombre de nœuds de discrétisation dans la direction  $\mathbf{x}$  (resp.  $\mathbf{y}$ ). Le second membre est également décomposé comme une somme de produits de vecteurs  $\mathbf{f}_x^j$  de taille  $N_x$  et de vecteurs  $\mathbf{f}_y^j$  de taille  $N_y$ . Finalement l'inconnue  $u_h$  est recherchée comme une somme de produits de vecteurs  $\mathbf{F}^i$  de taille  $N_x$  et de vecteurs  $\mathbf{G}^i$  de taille  $N_y$  pondérée du coefficient  $\alpha^i$ .  $n_A$  représente le nombre de paire d'opérateurs requis afin de représenter sous forme séparée l'opérateur initial  $\mathcal{A}_h$  tandis que  $n_f$  correspond au nombre de vecteurs nécessaires pour représenter le second membre  $f_h$  sous forme séparée.

En tenant compte des propriétés du produit tensoriel, l'équation.(1.80) peut s'écrire de la manière suivante :

$$\sum_{k=1}^{n_A} \sum_{i=1}^m \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \right) = \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j + Res^m \quad (1.82)$$

où  $Res^m$  est le résidu apparaissant naturellement du fait de la discrétisation. A partir de l'équation (1.82) nous pouvons maintenant expliciter dans les détails les trois étapes de l'algorithme PGD qui est une procédure itérative.

## b) Description de l'algorithme

On suppose que nous sommes à l'étape  $m$ . Les  $m - 1$  premières fonctions  $F^i$  et  $G^i$  sont connues ainsi que les  $m - 1$  premiers coefficients  $\alpha^i$ , on cherche alors  $F^m$ ,  $G^m$  et les  $m$  coefficients  $\alpha^i$ .

### 1) Etape d'enrichissement

L'approximation d'ordre  $m$ , est recherchée sous la forme suivante :

$$u_m = \sum_{i=1}^{m-1} \alpha^i \mathbf{F}^i \otimes \mathbf{G}^i + \mathbf{R} \otimes \mathbf{S} \quad (1.83)$$

où les vecteurs  $\mathbf{R}$  et  $\mathbf{S}$  sont inconnus. En introduisant cette forme de la solution dans l'Eq.(1.82), on obtient le système suivant :

$$\sum_{k=1}^{n_A} \left( A_x^k \mathbf{R} \otimes A_y^k \mathbf{S} \right) = \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j - \sum_{k=1}^{n_A} \sum_{i=1}^{m-1} \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \right) + Res^{m-1} \quad (1.84)$$

Afin d'obtenir le système d'équations vérifié par  $\mathbf{R}$  et  $\mathbf{S}$ , l'équation (1.84) est projetée d'une part sur  $\mathbf{R}$  en imposant que  $Res^{m-1}$  soit orthogonal à  $\mathbf{R}$ , et d'autre part sur  $\mathbf{S}$  en imposant que  $Res^{m-1}$  soit orthogonal à  $\mathbf{S}$ .

La projection de (1.84) sur  $\mathbf{S}$  donne le problème suivant :

$$\sum_{k=1}^{n_A} \gamma_k^1 A_x^k \mathbf{R} = \sum_{j=1}^{n_f} \gamma_j^2 \mathbf{f}_x^j - \sum_{k=1}^{n_A} \sum_{i=1}^{m-1} \gamma_{ik}^3 \alpha^i A_x^k \mathbf{F}^i \quad (1.85)$$

avec

$$\gamma_k^1 = {}^t \mathbf{S} A_y^k \mathbf{S} \in \mathbb{R} \quad \gamma_j^2 = {}^t \mathbf{S} \mathbf{f}_y^j \in \mathbb{R} \quad \gamma_{ik}^3 = {}^t \mathbf{S} A_y^k \mathbf{G}^i \in \mathbb{R} \quad (1.86)$$

La projection de (1.84) sur  $\mathbf{R}$  donne le problème suivant :

$$\sum_{k=1}^{n_A} \beta_k^1 A_y^k \mathbf{S} = \sum_{j=1}^{n_f} \beta_j^2 \mathbf{f}_y^j - \sum_{k=1}^{n_A} \sum_{i=1}^{m-1} \beta_{ik}^3 \alpha^i A_y^k \mathbf{G}^i \quad (1.87)$$

avec

$$\beta_k^1 = {}^t \mathbf{R} A_x^k \mathbf{R} \in \mathbb{R}, \quad \beta_j^2 = {}^t \mathbf{R} \mathbf{f}_x^j \in \mathbb{R}, \quad \beta_{ik}^3 = {}^t \mathbf{R} A_x^k \mathbf{F}^i \in \mathbb{R} \quad (1.88)$$

Les équations (1.85) et (1.87) sont résolues en utilisant une stratégie du point fixe. Ainsi pour calculer  $\mathbf{R}$ , on fixe  $\mathbf{S}$  égal à celui obtenu à l'itération précédente et on résout l'équation (1.85). De même, pour calculer  $\mathbf{S}$ , on fixe  $\mathbf{R}$  à celui obtenu à l'itération précédente et on résout l'équation (1.87). La procédure du point fixe s'arrête à la  $k^{ieme}$  itération lorsque le critère suivant est vérifié :

$$\|(\mathbf{RS})^{(k)} - (\mathbf{RS})^{(k-1)}\| \leq \epsilon \quad (1.89)$$

où  $\epsilon$  est un paramètre fixé par l'utilisateur. Les nouvelles fonctions  $\mathbf{F}^m$  et  $\mathbf{G}^m$  sont définies après normalisation des vecteurs  $\mathbf{R}$  et  $\mathbf{S}$  par :

$$\mathbf{F}^m = \frac{\mathbf{R}}{\|\mathbf{R}\|} \quad \mathbf{G}^m = \frac{\mathbf{S}}{\|\mathbf{S}\|} \quad (1.90)$$

## 2) Etape de projection

A cette étape, les  $m$  fonctions  $\mathbf{F}^i$  et  $\mathbf{G}^i$  sont connues, il faut maintenant calculer les  $\alpha^i$  ( $1 \leq i \leq m$ ). Pour ce faire, l'équation (1.82) est projetée sur chaque fonction  $\mathbf{F}^j \mathbf{G}^j$  (pour  $j = 1, \dots, m$ ) en imposant que  $Res^m$  soit orthogonal à chaque couple  $\mathbf{F}^j \mathbf{G}^j$ . On obtient alors le système suivant :

$$\mathbb{M} \boldsymbol{\alpha} = \mathbf{b} \quad \text{avec} \quad {}^t \boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_m\} \quad (1.91)$$

où les composantes de  $\mathbb{M}$  et de  $\mathbf{b}$  sont définies par :

$$\mathbb{M}_{ij} = \sum_{k=1}^{n_A} {}^t \mathbf{F}^j A_x^k \mathbf{F}^i \cdot {}^t \mathbf{G}^j A_y^k \mathbf{G}^i \quad \text{et} \quad \mathbf{b}_j = \sum_{k=1}^{n_f} {}^t \mathbf{F}^j \mathbf{f}_x^k \cdot {}^t \mathbf{G}^j \mathbf{f}_y^k \quad (1.92)$$

## 3) Etape d'estimation de la convergence

Afin d'estimer la convergence de l'algorithme, nous estimons le résidu  $Res^m$  à l'étape  $m$  de l'équation (1.82) défini par :

$$Res^m = \sum_{k=1}^{n_A} \sum_{i=1}^m \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \right) - \sum_{k=1}^{n_f} \mathbf{f}_x^k \otimes \mathbf{f}_y^k \quad (1.93)$$

Lorsque la norme euclidienne de ce résidu devient inférieure ou égale à un paramètre  $\epsilon$  fixé par l'utilisateur, l'algorithme a convergé, et la solution du problème s'écrit :

$$u_m = \sum_{i=1}^m \alpha^i \mathbf{F}^i \otimes \mathbf{G}^i \quad (1.94)$$

Sinon, il faut enrichir à nouveau notre base en ajoutant un nouveau produit de fonctions dépendant de  $x$  et de  $y$ .

Afin d'illustrer la méthode PGD sur un cas simple, l'exemple de l'équation de Poisson stationnaire en deux dimensions a été traité dans l'annexe [B](#).

## Chapitre 2

# Formulation de la PGD sur des équations modèles

L'objectif de ce chapitre est de tester la méthode PGD sur différents cas tests afin d'étudier son comportement général. Il est prouvé que la PGD converge toujours vers la solution du problème complet [Ammar et al., 2010c],[Ammar et al., 2010b], malheureusement aucune information n'est disponible quant au nombre de fonctions nécessaires pour converger vers la solution d'un problème donné, où même quant au nombre d'itérations nécessaires dans le point fixe pour le calcul des nouvelles fonctions. C'est pourquoi il semble intéressant d'étudier des équations modèles afin de se rendre mieux compte des différentes difficultés que l'on pourra rencontrer lorsque l'on souhaitera résoudre les équations de Navier-Stokes. Nous étudierons tout d'abord le cas classique de l'équation de Poisson en deux et en trois dimensions, en stationnaire et en instationnaire. En effet l'équation de Poisson fait partie des problèmes classiques que l'on retrouve dans la résolution de beaucoup de problèmes, entre autre dans la résolution des équations de Navier-Stokes à l'aide d'un schéma de projection ou d'une formulation fonction de courant-vorticité. Pour la résolution instationnaire nous allons tout d'abord chercher la solution sous forme séparée des variables d'espace seulement puis sous forme séparée des variables spatiales et temporelle.

Nous étudierons également les équations de Stokes et Burgers qui sont toutes les deux proches de l'équation de Navier-Stokes, chacune d'une manière différente. En effet, l'équation de Stokes consiste à rechercher un champ de vitesse et de pression sans l'intervention d'un terme de convection. L'étude de cette équation nous permettra d'analyser la capacité de la méthode à traiter le couplage entre le champ de vitesse et la pression. L'équation de Burgers quant à elle, fait seulement intervenir le champ de vitesse, sous la forme d'une équation de convection-diffusion où le terme générateur de la convection est la vitesse elle-même. Ce terme convectif, rend donc l'équation de Burgers non linéaire, elle est donc intéressante à étudier pour analyser le comportement de la PGD pour la résolution de problèmes non linéaires. Les équations de Stokes et de Burgers seront toutes deux traitées dans le cas à deux dimensions.

## I Equation de diffusion stationnaire 2D

Dans la suite de ce chapitre, nous utiliserons une discrétisation spatiale de type volumes finis. Dans la cas de l'équation de diffusion stationnaire 2D nous donnerons la forme des opérateurs intervenant dans la résolution par méthode PGD et traiterons deux problèmes différents avec des conditions aux limites de type Dirichlet et de type Neumann.

### I.1 Formulation du problème

Rappelons le problème de l'équation de Poisson 2D

$$\text{Chercher } T(x,y) \text{ tel que } \begin{cases} \nabla^2 T = f & \text{dans } \Omega \\ \text{+conditions aux limites} \end{cases} \quad (2.1)$$

La discrétisation de cette équation par méthode PGD à déjà été développée dans le premier chapitre. Nous nous contenterons ici de rappeler la formulation tensorielle du problème ainsi que les opérateurs PGD utile à la résolution. La solution  $T(x, y)$  est recherchée sous la forme

séparée suivante :

$$T(x, y) \approx T_m(x, y) = \sum_{i=1}^m \alpha^i F^i(x) G^i(y) \quad (2.2)$$

Cette forme de la solution est introduite dans le problème (2.1), après discrétisation et intégration sur l'espace des équations, en tenant compte des propriétés de la représentation tensorielle, on obtient la formulation suivante :

$$\sum_{k=1}^2 \sum_{i=1}^m \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \right) = \int_{\Omega} \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j d\Omega \quad (2.3)$$

Les opérateurs  $A_x^k$  (respectivement  $A_y^k$ ) sont des matrices carrées de taille  $N_x$  (respectivement  $N_y$ ) où  $N_x$  (resp  $N_y$ ) est le nombre de nœuds de discrétisation dans la direction  $x$  (resp  $y$ ). Ces matrices s'écrivent :

$$A_x^1 = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ b_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_i & a_i & c_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & c_{N_x-1} \\ 0 & \cdots & \cdots & \cdots & 0 & b_{N_x} & a_{N_x} \end{pmatrix} \quad A_y^1 = \text{diag}(\Delta_{y_1}, \dots, \Delta_{y_{N_y}}) \quad (2.4)$$

où

$$a_i = -\frac{1}{x_{i+1} - x_i} - \frac{1}{x_i - x_{i-1}}, \quad b_i = \frac{1}{x_i - x_{i-1}}, \quad c_i = \frac{1}{x_{i+1} - x_i}$$

et

$$A_x^2 = \text{diag}(\Delta_{x_1}, \dots, \Delta_{x_{N_x}}) \quad A_y^2 = \begin{pmatrix} d_1 & f_1 & 0 & \cdots & \cdots & \cdots & 0 \\ e_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e_j & d_j & f_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & f_{N_y-1} \\ 0 & \cdots & \cdots & \cdots & 0 & e_{N_y} & d_{N_y} \end{pmatrix} \quad (2.5)$$

où

$$d_j = -\frac{1}{y_{j+1} - y_j} - \frac{1}{y_j - y_{j-1}}, \quad e_j = \frac{1}{y_j - y_{j-1}}, \quad f_j = \frac{1}{y_{j+1} - y_j}$$

où  $(x_i, y_j)$  est la coordonnée du centre du volume de contrôle  $\Omega_{ij}$ .  $\Delta_{x_i}$  (resp.  $\Delta_{y_j}$ ) est la longueur horizontale (resp. verticale) du volume de contrôle  $\Omega_{ij}$ . On pose  $x_0 = x_{CLW}$  (resp.

$x_{N_{x+1}} = x_{CLE}$ ) la coordonnée horizontale de la frontière ouest (resp. est) du domaine, et  $y_0 = y_{CLS}$  (resp.  $y_{N_{y+1}} = y_{CLN}$ ) la coordonnée verticale de la frontière sud (resp. nord) du domaine. À partir des opérateurs définis dans (2.4) et (2.5), il est aisé de construire les différents problèmes d'enrichissement, de projection et de vérification de la convergence.

Dans ce cas, la procédure d'enrichissement est effectuée par résolution successive de deux systèmes linéaires, l'un de taille  $N_x$  pour calculer  $F^m$  et l'autre de taille  $N_y$  pour calculer  $G^m$ , jusqu'à convergence du point fixe. De même l'étape de projection consiste à résoudre un système linéaire de taille  $m \times m$ .

## I.2 Résultats

Ici nous allons tester la méthode PGD sur le problème simple de l'équation de Poisson 2D stationnaire dont une solution analytique est connue, pour des conditions aux limites de type Dirichlet et Neumann. Les calculs PGD seront comparés aux calculs effectués à l'aide d'un solveur classique type bi-gradient conjugué. Afin de comparer les résultats issus des simulations avec la solution analytique, l'erreur suivante est définie

$$\eta = \frac{\left( \int_{\Omega} [T_{ana}(x, y) - T_{simul}(x, y)]^2 d\Omega \right)^{1/2}}{\left( \int_{\Omega} [T_{ana}(x, y)]^2 d\Omega \right)^{1/2}} \quad (2.6)$$

Dans la suite on suppose que le nombre de nœuds dans les directions  $x$  et  $y$  est le même et sera noté  $N_h$ .

### a) Problème 1

On Considere le problème suivant :

$$\text{Chercher } T(x,y) \in \Omega = ]0, 1[ \times ]0, 1[ \text{ tel que } \begin{cases} \nabla^2 T(x, y) = -8\pi^2 \cos(2\pi x) \sin(2\pi y) \\ + \text{ conditions aux limites} \end{cases} \quad (2.7)$$

où  $\Gamma$  est la frontière de  $\Omega$ . La solution analytique de ce problème est donnée par :

$$T_{ana}(x, y) = \cos(2\pi x) \sin(2\pi y) \quad (2.8)$$

Les conditions aux limites (de type Dirichlet ou Neumann) sont imposées de manière à satisfaire la solution analytique. On constate que le second membre de cette équation peut s'écrire sous une forme séparée en  $x$  et en  $y$ .

#### 1) Conditions aux limites de type Dirichlet

Dans ce paragraphe, le problème (2.7) à été résolu en imposant des conditions aux limites de types Dirichlet. La figure 2.1 (gauche) représente l'évolution du temps de calcul en fonction du nombre de nœuds pour les deux solveurs étudiés. On observe que quelque soit le nombre

de nœuds, la PGD est plus rapide que le bigradient-conjugué. Ceci s'explique par le fait que dans ce cas, **la solution du problème est trouvée en une seule itération PGD**. Ainsi pour  $N_h = 1000$ , la PGD nécessite moins d'une seconde pour converger vers la solution alors que le bigradient-conjugué nécessite 454 secondes.

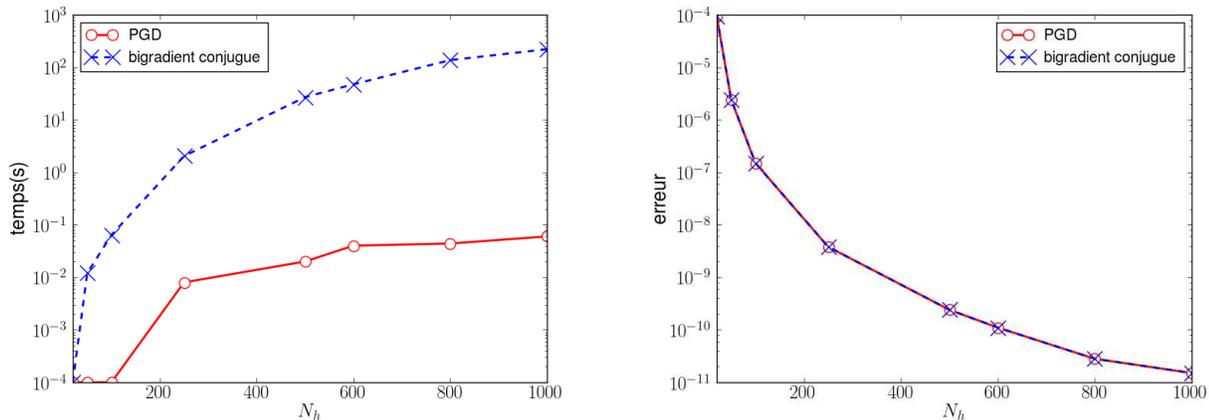


FIG. 2.1 – Temps de calcul (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds  $N_h$  dans chaque direction pour des conditions aux limites de type Dirichlet. (axes des ordonnées en échelle logarithmique)

Sur la figure 2.1 (droite) nous avons tracé l'erreur  $\eta$  définie dans l'équation (2.6) et qui représente la norme  $L^2$  de la différence entre la solution calculée par chacun des solveurs et la solution analytique. On constate que l'erreur est semblable pour les deux solveurs.

## 2) Conditions aux limites de type Neumann

Le problème à résoudre est exactement le même que le cas précédent sauf que les conditions aux limites imposées ici sont des conditions de Neumann qui sont déterminées à l'aide de la solution analytique.

La figure 2.2 (gauche) représente le temps de calcul pour chacun des solveurs. Ici aussi, on constate la supériorité de la méthode PGD par rapport au solveur classique. En effet, quelque soit le nombre de nœuds dans chaque direction, la PGD est plus rapide que le solveur classique. Concrètement, pour  $N_h = 1000$ , il faut moins d'une seconde de calcul pour converger contre 111 secondes pour le bi-gradient conjugué.

Les erreurs entre les solutions calculées avec les solveurs PGD et avec le bigradient conjugué, et la solution analytique sont identiques (cf. figure 2.2 (droite)).

Ainsi, dans le cas de l'équation de diffusion stationnaire, sur un problème où la solution peut se décomposer en un seul produit de fonctions, la PGD est extrêmement intéressante en terme de temps de calcul avec une précision équivalente au solveur étudié. En effet la solution PGD est reconstruite en exactement un produit de fonctions. Cela est vrai en imposant des conditions de Dirichlet ou des conditions de Neumann aux frontières, et quelque soit le nombre de nœuds.

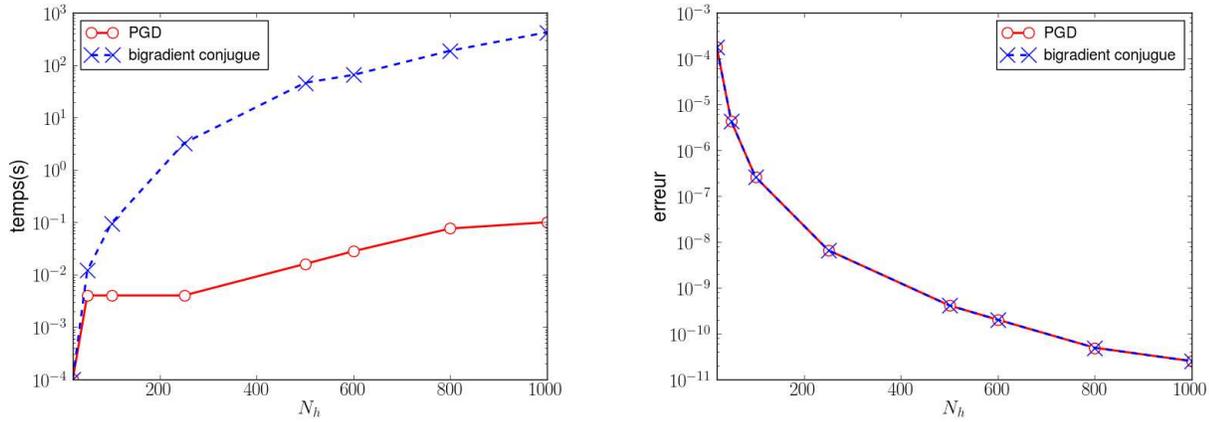


FIG. 2.2 – Temps de calcul (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds dans chaque direction pour des conditions aux limites de type Neumann. (Axes des ordonnées en échelle logarithmique)

## b) Problème 2

Considérons maintenant le problème suivant :

$$\text{Chercher } T(x,y) \in \Omega = ]0, 1[ \times ]0, 1[ \begin{cases} \nabla^2 T(x,y) = 12(x^2 y^4 + x^4 y^2) - 8\pi^2 \cos(2\pi x) \sin(2\pi y) \\ + \text{conditions aux limites} \end{cases} \quad (2.9)$$

La solution analytique de ce problème est donnée par :

$$T_{ana}(x,y) = x^4 y^4 + \cos(2\pi x) \sin(2\pi y) \quad (2.10)$$

De la même manière que pour le problème précédent, les conditions aux bords (de type Dirichlet ou Neumann) seront imposés de manière à satisfaire la solution analytique. Cette fois ci la solution est décomposable en deux produits de fonctions.

### 1) Conditions aux limites de type Dirichlet

Dans ce paragraphe, le problème (2.9) a été résolu en imposant des conditions aux limites de types Dirichlet. La figure 2.3 (gauche) représente l'évolution du temps de calcul en fonction du nombre de nœuds pour les deux solveurs étudiés. On observe ici, que la PGD est plus rapide que le bigradient-conjugué à partir de 200 nœuds dans chaque direction. Ainsi pour  $N_h = 1000$ , la PGD est vingt-deux fois plus rapide que le bigradient-conjugué.

Sur la figure 2.3 (droite) nous avons tracé l'erreur  $\eta$  définie dans l'équation (2.6) en fonction du nombre de nœuds. On constate que l'erreur est semblable pour les deux solveurs. Dans ce cas la PGD nécessite soixante dix fonctions pour converger alors que la solution est décomposable en deux produit de fonctions ce qui implique que la PGD ne converge pas vers la solution de façon optimale.

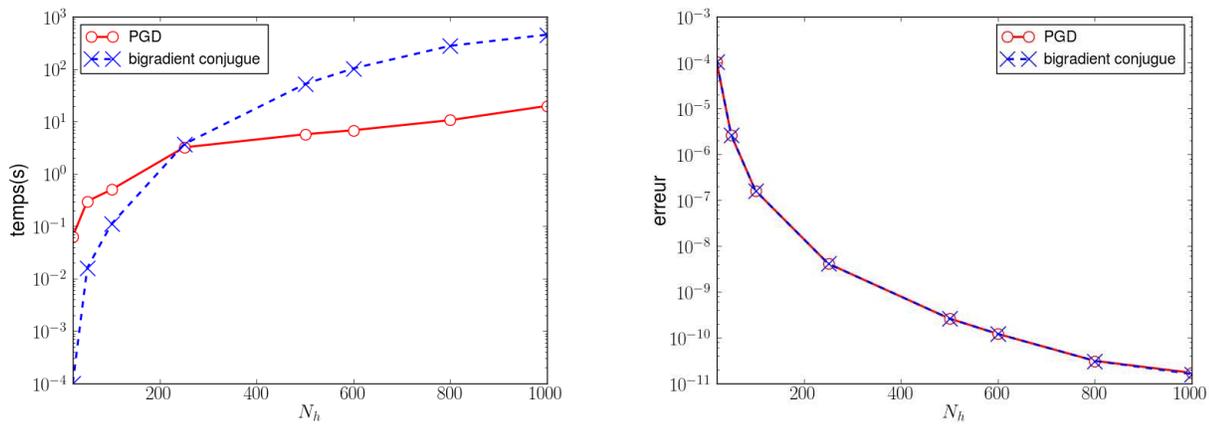


FIG. 2.3 – Temps de simulation (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds  $N_h$  dans chaque direction pour des conditions aux limites de type Dirichlet.

## 2) Conditions aux limites de type Neumann

La figure 2.4 (gauche) indique qu'à partir de cinquante nœuds dans chaque direction, la PGD est plus rapide que le solveur classique. Concrètement quel que soit le maillage étudié ici, il faut moins de dix secondes de calcul pour converger. Pour mille nœuds dans chaque direction la PGD est cinquante fois plus rapide que le bi-gradient conjugué. La PGD, dans ce cas n'a besoin que d'une dizaine de fonctions pour vérifier le critère de convergence.

La figure 2.4 (droite) montre que les erreurs commises en utilisant les solveurs PGD et bigradient conjugué sont similaires même si la PGD a une erreur très légèrement supérieure au solveur classique.

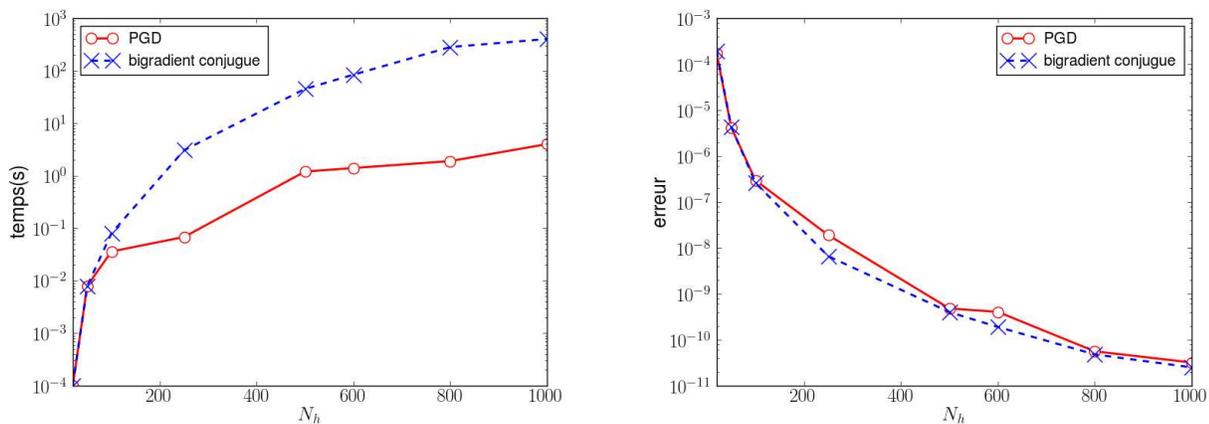


FIG. 2.4 – Temps de simulation (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds dans chaque direction pour des conditions aux limites de type Neumann.

### 3) SVD

Afin de comprendre pourquoi le nombre de fonction PGD est si éloigné du nombre optimal de fonction, que ce soit avec des conditions de Dirichlet ou des conditions de Neumann, une SVD est effectuée sur les résultats obtenus par PGD et sur ceux obtenus à partir du solveur standard. On rappelle que la SVD, définie dans le chapitre 1 section I.2, est une procédure permettant de décomposer de manière optimale un champ 2D en produits de fonctions 1D.

La figure 2.5 représente l'évolution de la norme du résidu en fonction du nombre de fonctions gardé pour la PGD, la SVD appliquée à la solution obtenue par PGD et la SVD appliquée à la solution obtenue par le modèle standard. Cette figure illustre le fait que la PGD ne converge pas de manière optimale vers la solution. En effet, pour des conditions aux limites de type Dirichlet, la norme du résidu pour les fonctions PGD décroît de manière beaucoup plus lente que la norme du résidu calculée à partir des fonctions obtenus par SVD appliquée aux résultats du modèle standard et PGD. Pour ces derniers, on retrouve une norme du résidu inférieure à  $10^{-8}$  en moins de 10 fonctions alors que la PGD nécessite soixante dix fonctions pour une convergence similaire. Les mêmes constatations peuvent être faites pour les résultats obtenus avec des conditions aux limites de types Neumann. Ici la PGD nécessite une quinzaine de fonction pour obtenir une norme du résidu inférieure à  $10^{-5}$  alors que les résultats SVD obtiennent tous deux une norme du résidu inférieure à  $10^{-5}$  en trois fonctions uniquement. Par contre, on observe un caractère strictement décroissant de la norme du résidu en fonction du nombre de fonction PGD, ce qui implique que la PGD peut fournir une solution aussi précise que l'on veut à condition d'augmenter le nombre de fonctions de manière suffisante.

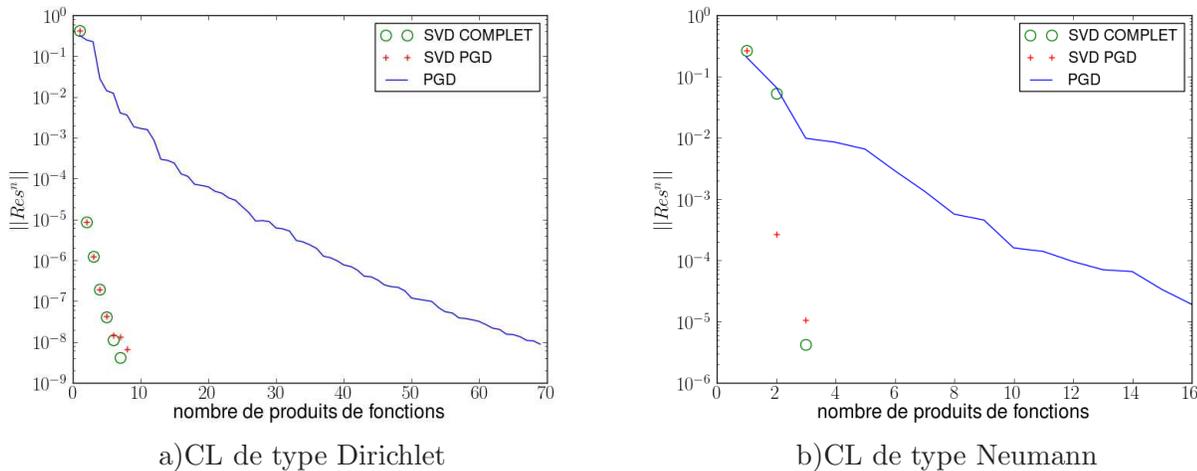


FIG. 2.5 – Norme du résidu en fonction du nombre de fonctions de chacune des décomposition PGD, SVD sur le résultat PGD et SVD sur le résultats du modèle standard pour des conditions aux limites de type Dirichlet et Neumann.

#### Remarque

Une manière d'obtenir la solution PGD sous forme optimale, consiste à rechercher la solution

à chaque itération comme une somme de deux produits de fonctions. Cela revient à appliquer la PGD Galerkin avec itérations successives sur les sous-espaces définie dans le premier chapitre. Les résultats ne sont pas présentés ici, car ce ne sera pas la méthode utilisée par la suite<sup>1</sup>. De plus cette manière de rechercher la solution nécessite une connaissance préalable de la solution.

## II Equation de diffusion stationnaire 3D

Nous allons maintenant nous intéresser à l'équation de diffusion stationnaire en trois dimensions.

### II .1 Formulation du problème

Soit le problème suivant :

$$\text{Chercher } T(x, y, z) \text{ tel que } \begin{cases} \nabla^2 T(x, y, z) = f(x, y, z) & \text{dans } \Omega \in \mathbb{R}^3 \\ T|_{\Gamma} = g(x, y, z) \end{cases} \quad (2.11)$$

où  $\Gamma$  est la frontière du domaine. On suppose que la fonction  $f(x, y, z)$  est à variable séparables<sup>2</sup> et peut s'écrire sous la forme :

$$f(x, y, z) = \sum_{i=1}^{n_f} f_x^i f_y^i f_z^i \quad (2.12)$$

Avant de résoudre ce problème nous allons donner l'expression des opérateurs intervenant dans la formulation algébrique nécessaire à la mise en pratique de l'algorithme PGD. La solution  $T(x, y, z)$  est recherchée sous la forme séparée suivante :

$$T(x, y, z) \approx T_m = \sum_{i=1}^m \alpha^i F^i(x) G^i(y) H^i(z) \quad (2.13)$$

En discrétisant le problème (2.11) par une méthode des volumes finis et en tenant compte de l'écriture de la solution en variables séparées. Le problème peut s'écrire sous la forme suivante :

$$\sum_{k=1}^3 \sum_{i=1}^m \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \otimes A_z^k \mathbf{H}^i \right) = \int_{\Omega} \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j \otimes \mathbf{f}_z^j d\Omega \quad (2.14)$$

où l'opérateur  $A_x$  (respectivement  $A_y, A_z$ ) est une matrice carrée de taille  $N_x$  (respectivement  $N_y, N_z$ ) avec  $N_x$  (resp  $N_y, N_z$ ) le nombre de nœuds de discrétisation dans la direction  $x$  (resp  $y, z$ ). Ces matrices peuvent s'écrire de la manière suivante, en utilisant les notations de (I .1) :

<sup>1</sup>En effet il faut savoir en combien de fonctions la solution peut être décomposée

<sup>2</sup>On rappelle que cette condition n'est pas nécessaire mais facilite l'écriture du problème

$$A_x^1 = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ b_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_i & a_i & c_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & c_{N_x-1} \\ 0 & \cdots & \cdots & \cdots & 0 & b_{N_x} & a_{N_x} \end{pmatrix} \quad A_y^1 = \text{diag}(\Delta_{y_1}, \dots, \Delta_{y_{N_y}}) \quad A_z^1 = \text{diag}(\Delta_{z_1}, \dots, \Delta_{z_{N_z}})$$

$$A_x^2 = \text{diag}(\Delta_{x_1}, \dots, \Delta_{x_{N_x}}) \quad A_y^2 = \begin{pmatrix} d_1 & f_1 & 0 & \cdots & \cdots & \cdots & 0 \\ e_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e_j & d_j & f_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & f_{N_y-1} \\ 0 & \cdots & \cdots & \cdots & 0 & e_{N_y} & d_{N_y} \end{pmatrix} \quad A_z^2 = A_z^1$$

$$A_x^3 = A_x^2 \quad A_y^3 = A_y^1 \quad A_z^3 = \begin{pmatrix} r_1 & t_1 & 0 & \cdots & \cdots & \cdots & 0 \\ s_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & s_j & r_j & t_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & t_{N_z-1} \\ 0 & \cdots & \cdots & \cdots & 0 & s_{N_z} & r_{N_z} \end{pmatrix}$$

où

$$r_i = -\frac{1}{z_{i+1} - z_i} - \frac{1}{z_i - z_{i-1}}, \quad s_i = \frac{1}{z_i - z_{i-1}}, \quad t_i = \frac{1}{z_{i+1} - z_i}$$

## II .2 Résultats

### a) Problème étudié

Soit le problème suivant :

Chercher  $T(x,y,z) \in \Omega = ]0, 1[ \times ]0, 1[ \times ]0, 1[$  tel que

$$\begin{cases} \nabla^2 T = 12(x^2 y^4 z^4 + x^4 y^2 z^4 + x^4 y^4 z^2) - 12\pi^2 \cos(2\pi x) \sin(2\pi y) \sin(2\pi z) \\ + \text{conditions aux limites} \end{cases} \quad (2.15)$$

où les conditions aux limites sont définies de manière à satisfaire la solution analytique suivante :

$$T_{ana}(x, y, z) = x^4 y^4 z^4 + \cos(2\pi x) \sin(2\pi y) \sin(2\pi z) \quad (2.16)$$

De la même manière que dans le cas 2D les résultats obtenus par chacun des deux solveurs étudiés (PGD, bi-gradient conjugué) sont comparés à la solution analytique. Dans la suite on suppose que le nombre de nœuds dans les trois directions  $x$ ,  $y$  et  $z$  est le même et sera noté  $N_h$ .

### b) Conditions aux limites de type Dirichlet

Le problème précédent est donc résolu en considérant des conditions aux bords de type Dirichlet. La figure (2.6)(gauche) indique, pour chaque solveur étudié, le temps de calcul de la solution en fonction du nombre de nœuds  $N_h$  dans chaque direction. Sur cette figure on constate que la PGD est plus rapide que le bigradient conjugué à partir de 30 nœuds dans chaque direction. Avec 200 nœuds dans chaque direction, soit un total de  $8.10^6$  nœuds, la PGD est alors douze fois plus rapide que le bigradient-conjugué. On peut noter ici, que la convergence de la méthode PGD s'effectue en une soixantaine de fonctions, quelque soit le nombre de nœuds.

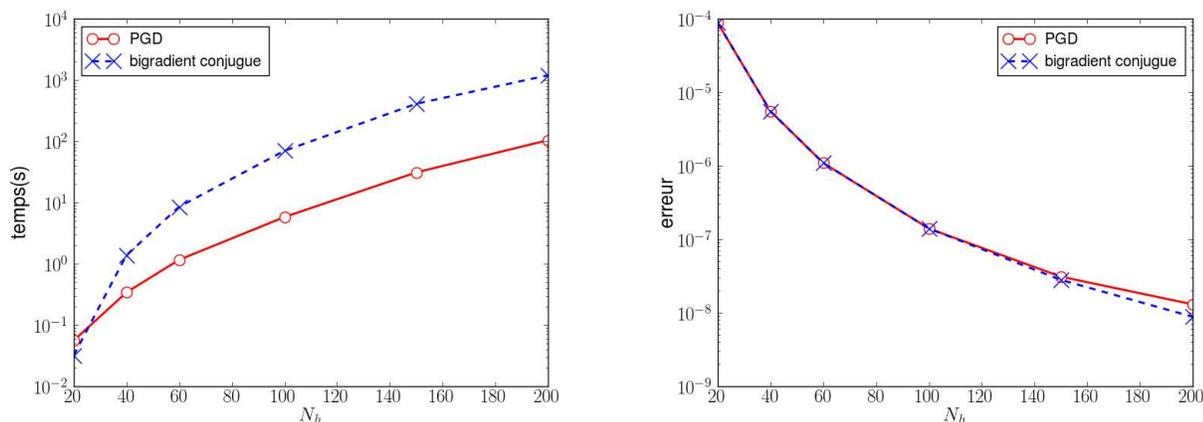


FIG. 2.6 – Temps de simulation (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds dans chaque direction

Nous avons également tracé sur la figure(2.6)(droite) pour chacun des solveurs, les courbes d'erreur relative en norme  $L^2$  entre la solution analytique et les solutions calculées. Cette figure montre que la méthode PGD est aussi précise que le bigradient conjugué.

### c) Conditions aux limites de type Neumann

Les conditions aux limites imposées sont maintenant des conditions de Neumann, qui sont déterminées à l'aide de la solution analytique.

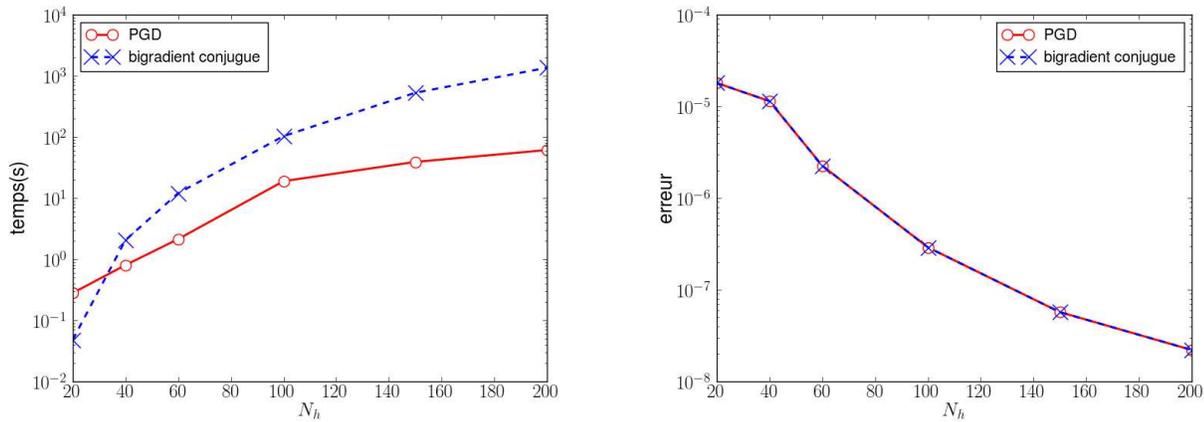


FIG. 2.7 – Temps de simulation (gauche) et erreur  $\eta$  (droite) en fonction du nombre de nœuds dans chaque direction

La méthode PGD dans ce cas est plus rapide que le gradient bi-conjugué à partir de 30 nœuds dans chaque direction (cf. figure 2.7 (gauche)). Avec 200 nœuds dans chaque direction, la PGD est vingt fois plus rapide.

La figure 2.7 (droite) montre que l'erreur par rapport à la solution analytique est la même pour chacun des deux solveurs étudiés. Ici, la résolution par méthode PGD a nécessité une trentaine de fonctions pour converger.

Nous pouvons donc conclure que la PGD est efficace en terme d'erreur et de temps de calcul pour la résolution de l'équation de diffusion stationnaire avec des conditions de Dirichlet et Neumann en trois dimensions. Cependant, là encore le nombre de fonctions PGD nécessaire est supérieur au nombre optimal.

#### d) SVD

De la même manière que dans le cas 2D, le résidu en fonction du nombre de fonction PGD, et du nombre de fonctions SVD (pour une SVD réalisée sur la solution PGD finale, et sur la solution du modèle standard) pour  $N_h = 60$  est tracé sur la figure 2.8. Cette figure montre que lorsque des conditions aux limites de Dirichlet sont imposées, la PGD n'est pas optimale en terme de décomposition. En effet la norme du résidu en fonction du nombre de fonction PGD décroît lentement et nécessite une soixantaine de fonctions pour être inférieure à  $10^{-5}$  alors que dans le cas des SVD, moins de dix fonctions sont nécessaires. Dans le cas où des conditions aux lites de type Neumann sont imposées, on constate que pour obtenir une solution correcte, la SVD nécessite un grand nombre de fonctions malgré que la solution soit décomposable en deux produits de fonctions uniquement. Cette fois ci la PGD semble avoir un comportement semblable à la SVD. Sur les quinze premières fonctions, la PGD semble même être en mesure de fournir une norme du résidu inférieure à celle calculée par SVD. Par contre il faut quand même une trentaine de fonctions PGD pour que la norme du résidu soit inférieur à  $10^{-5}$  contre

une vingtaine pour les SVD. Ces résultats doivent être interprétés avec précaution, du fait que la SVD comme elle est appliquée ici n'est pas optimale au sens de la norme du résidu (voir remarque).

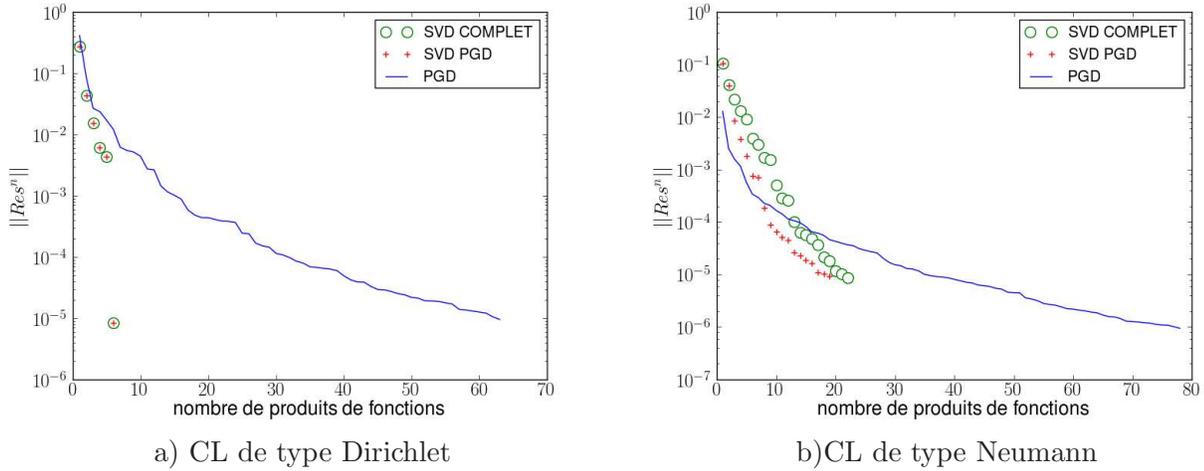


FIG. 2.8 – Norme du résidu en fonction du nombre de fonctions de chacune des décomposition PGD, SVD sur le résultat PGD et SVD sur le résultats du modèle standard pour des conditions aux limites de type Dirichlet et Neumann.

### Remarque

Dans ce cas, par abus de langage, nous avons utilisé la dénomination SVD pour décomposer un champ 3D en somme de produits de fonctions 1D. En effet, la SVD n'est définie que dans le cas 2D. Ce qui est effectué ici, est en fait une résolution PGD du problème suivant :

$$\sum_{k=1}^m (\mathbb{I}_x \otimes \mathbb{I}_y \otimes \mathbb{I}_z)(F_x^k \otimes F_y^k \otimes F_z^k) = SM(x, y, z) \quad (2.17)$$

où  $SM(x, y, z)$  représente le champ que l'on souhaite décomposer,  $\mathbb{I}_x$  (respectivement  $\mathbb{I}_y$  et  $\mathbb{I}_z$ ) est l'opérateur identité dans la direction  $x$  (resp.  $y$  et  $z$ ). Les  $m$  fonctions  $F_x^k$ ,  $F_y^k$  et  $F_z^k$  solutions de ce problème constitueront les fonctions SVD (toujours par abus de langage). Malheureusement dans ce cas, nous n'avons plus aucune information sur l'optimalité de la méthode par rapport au résidu.

### III Equation de diffusion instationnaire 2D

Nous allons maintenant nous intéresser au problème de diffusion instationnaire en 2 dimensions, soit :

$$\text{Chercher } T(x,y,t) \text{ tel que } \begin{cases} \frac{\partial T(x,y,t)}{\partial t} - \nabla^2 T(x,y,t) = f(x,y,t) & \text{dans } \Omega \times ]0, T[ \\ T|_{\Gamma} = g \\ T(x,y,0) = T^0 \end{cases} \quad (2.18)$$

où  $\Omega = ]-L, L[ \times ]-L, L[$ , et  $\Gamma$  est la frontière de ce domaine.

Le problème instationnaire peut être résolu de deux manières différentes par méthode PGD :

- 1) En utilisant un schéma de discrétisation temporel classique et en calculant la valeur de  $T$  de manière itérative en temps. Ce qui implique une séparation sur les variables spatiales  $x$  et  $y$  uniquement.
- 2) En considérant le temps comme une variable pouvant entrer dans notre décomposition, et donc en effectuant une séparation sur les deux variables spatiales  $x$  et  $y$  ainsi que sur le temps  $t$ .

#### III.1 Utilisation d'un schéma de discrétisation temporel classique

Nous sommes au temps  $t$ , il s'agit de trouver la solution au temps  $t + \delta t$ . En utilisant un  $\Theta$ -schéma, le problème (2.18) discrétisé temporellement et intégré sur le domaine s'écrit.

$$\begin{aligned} \int_{\Omega} \left[ \frac{T^{t+1}}{\delta t} - \Theta \nabla^2 T^{t+1} \right] d\Omega &= \int_{\Omega} \left[ \frac{T^t}{\delta t} - (1 - \Theta) \nabla^2 T^t \right] d\Omega \\ &+ \int_{\Omega} (\Theta f^{t+1} + (1 - \Theta) f^t) d\Omega \end{aligned} \quad (2.19)$$

où le choix de  $\Theta$  permet de décider si l'on se place dans le cas d'un schéma explicite ( $\Theta = 0$ ), implicite ( $\Theta = 1$ ) ou de Crank-Nicholson ( $\Theta = \frac{1}{2}$ ). Dans la suite, la partie de droite de l'équation(2.19) sera noté  $S(x, y)$ .

Cette équation sera résolue pour chaque pas de temps en utilisant la méthode PGD. Ainsi, la solution est recherchée sous la forme

$$T^{t+1}(x, y) \approx T_m^{t+1}(x, y) = \sum_{i=1}^m \alpha_{t+1}^i F_{t+1}^i(x) G_{t+1}^i(y) \quad (2.20)$$

L'équation(2.19) peut donc s'écrire sous forme tensorielle de la manière suivante :

$$\sum_{k=1}^3 \sum_{i=1}^m \alpha_{t+1}^i \left( A_x^k \mathbf{F}_{t+1}^i \otimes A_y^k \mathbf{G}_{t+1}^i \right) = \sum_{i=1}^{n_f} \int_X S_x(x) \int_Y S_y(y) \quad (2.21)$$

où l'on a supposé que le second membre pouvait s'écrire sous forme séparée. Les opérateurs  $A_x$  et  $A_y$  sont donc des matrices carrées de taille  $N_x$  et  $N_y$  (nombre de nœuds de discrétisations dans les directions  $x$  et  $y$ ) définis de la manière suivante :

$$\begin{aligned}
 A_x^1 &= \frac{1}{\delta t} \text{diag}(\Delta_{x_1}, \dots, \Delta_{x_{N_x}}) & A_y^1 &= \text{diag}(\Delta_{y_1}, \dots, \Delta_{y_{N_y}}) \\
 A_x^2 &= \begin{pmatrix} a_1 & c_1 & 0 & \dots & \dots & \dots & 0 \\ b_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_i & a_i & c_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & c_{N_x-1} \\ 0 & \dots & \dots & \dots & 0 & b_{N_x} & a_{N_x} \end{pmatrix} & A_y^2 &= -\Theta \cdot \text{diag}(\Delta_{y_1}, \dots, \Delta_{y_{N_y}}) \\
 A_x^3 &= -\Theta \cdot \text{diag}(\Delta_{x_1}, \dots, \Delta_{x_{N_x}}) & A_y^3 &= \begin{pmatrix} d_1 & f_1 & 0 & \dots & \dots & \dots & 0 \\ e_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e_j & d_j & f_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & f_{N_y-1} \\ 0 & \dots & \dots & \dots & 0 & e_{N_y} & d_{N_y} \end{pmatrix}
 \end{aligned}$$

où les  $a_i, b_i, c_i, d_i, e_i$  et  $f_i$  ont été définis dans ce chapitre, section I.1

Ces opérateurs sont ensuite utilisés pour les trois étapes de résolution de l'algorithme PGD. La méthode est ensuite appliquée à chaque pas de temps afin d'obtenir la solution au temps souhaité. Il faut noter que les opérateurs sont définis une fois pour toute au début du processus et que seul le second membre doit être recalculé à chaque itération temporelle.

### III.2 Formulation PGD espace-temps

Lorsque le temps est intégré à la séparation de variable, la démarche est totalement différente de ce qui est fait précédemment. Au lieu de calculer la solution à un temps  $t + 1$ , en se servant des résultats obtenus en  $t$ , la solution pour tous les pas de temps de l'intervalle temporel considéré est directement calculée. La solution est donc recherchée sous la forme :

$$T(x, y, t) \approx T_m(x, y, t) = \sum_{i=1}^m \alpha^i F^i(x) G^i(y) H^i(t) \quad (2.22)$$

On suppose que le second membre peut s'écrire sous la forme :

$$f(x, y, t) = \sum_{j=1}^{n_f} f_x^j(x) f_y^j(y) f_t^j(t) \quad (2.23)$$

En injectant cette forme de la solution dans le problème (2.18) et en intégrant en temps et en espace, on obtient le problème sous la forme tensorielle suivante :

$$\sum_{k=1}^3 \sum_{i=1}^m \alpha^i \left( A_x^k \mathbf{F}^i \otimes A_y^k \mathbf{G}^i \otimes A_t^k \mathbf{H}^i \right) = - \int_{\Omega \times [0, T]} \sum_{j=1}^{n_f} \mathbf{f}_x^j \otimes \mathbf{f}_y^j \otimes \mathbf{f}_t^j d\Omega dt \quad (2.24)$$

où les opérateurs  $A_x$ ,  $A_y$  et  $A_t$  sont des matrices carrées de tailles respectives  $N_x$ ,  $N_y$  et  $N_t$  qui s'écrivent :

$$A_x^1 = \text{diag}(\Delta_{x_1}, \dots, \Delta_{x_{N_x}}) \quad A_y^1 = \text{diag}(\Delta_{y_1}, \dots, \Delta_{y_{N_y}}) \quad A_t^1 = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -1 & 1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}$$

$$A_x^2 = \begin{pmatrix} a_1 & c_1 & 0 & \dots & \dots & \dots & 0 \\ b_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_i & a_i & c_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & c_{N_x-1} \\ 0 & \dots & \dots & \dots & 0 & b_{N_x} & a_{N_x} \end{pmatrix} \quad A_y^2 = -A_y^1 \quad A_t^2 = \text{diag}(\Delta_{t_1}, \dots, \Delta_{t_{N_t}})$$

$$A_x^3 = -A_x^1 \quad A_y^3 = \begin{pmatrix} d_1 & f_1 & 0 & \dots & \dots & \dots & 0 \\ e_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e_j & d_j & f_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & f_{N_y-1} \\ 0 & \dots & \dots & \dots & 0 & e_{N_y} & d_{N_y} \end{pmatrix} \quad A_t^3 = A_t^2$$

où les  $a_i, b_i, c_i, d_i, e_i$  et  $f_i$  ont été définis dans ce chapitre, section I.1

Ces opérateurs sont ensuite utilisés pour effectuer les trois étapes de la méthodes PGD. A convergence de l'algorithme, la solution  $T_m(x, y, t)$  est connue pour tout les points du domaine et tout les temps.

### III.3 Résultats

On souhaite résoudre le problème 2D instationnaire suivant :

Chercher  $T(x, y, t) \in \Omega = ]-1, 1[ \times ]-1, 1[$  tel que

$$\begin{cases} \frac{\partial T}{\partial t} - \nabla^2 T(x, y, t) = f(x, y, t) \\ T(x, y, t) = g(x, y, t) \end{cases} \quad \text{sur } \Gamma \quad (2.25)$$

Le second membre est donné par

$$f(x, y, t) = \frac{x^4 y^4}{12} - x^2 y^4 t - x^4 y^2 t + 4x^2 t - 4y^2 t \quad (2.26)$$

De façon à avoir la solution analytique suivante

$$T_{ana}(x, y, t) = \frac{x^4 y^4}{12} t + 2x^2 t^2 - 2y^2 t^2 \quad (2.27)$$

Les conditions aux limites de type Dirichlet sont choisies de manière à être en accord avec la solution analytique.

Dans la suite, les deux décompositions PGD explicitées avant seront étudiées. On s'intéressera tout d'abord à la décomposition en espace seulement que l'on nommera  $PGD_{XY}$  qui consiste à résoudre le problème associé à chaque pas de temps et à se servir de la solution au pas de temps précédent afin de calculer la solution au pas de temps courant. Puis, nous considérerons la décomposition espace-temps que l'on nommera  $PGD_{XYT}$  qui consiste à calculer la solution sur l'intervalle de temps choisi en un seul calcul. Pour le solveur standard (solveur bigradient-conjugué) et  $PGD_{XY}$  la discrétisation temporelle a été effectuée à l'aide d'un schéma de Crank-Nicholson.

Les performances de ces approches PGD seront comparées au solveur standard du bigradient conjugué. Afin de comparer les résultats par rapport à la solution analytique l'erreur relative est définie.  $T_{simul}$  correspond à la solution obtenue par résolution PGD ou standard et  $T_{ana}$  à la solution analytique.

$$\epsilon_{rel} = \frac{\|T_{simul} - T_{ana}\|_{L^2(\Omega)}}{\|T_{ana(\Omega)}\|_{L^2(\Omega)}} \quad (2.28)$$

Dans la suite, l'influence du pas d'espace, du pas de temps ainsi que l'influence de l'intervalle de temps de simulation sur le comportement des différentes formulations PGD seront étudiés.

### a) Influence du pas d'espace

Dans ce paragraphe le problème (2.25) est résolu sur un intervalle de temps  $I_t = ]0, 1[$  avec un pas de temps  $\delta t = 10^{-3}$  et pour différents maillages.

La figure 2.9 (gauche) représente les temps de calcul nécessaires à la résolution du problème. On constate sur cette figure que le solveur  $PGD_{XYT}$  est le plus lent pour de faibles nombre de nœuds. Par contre à partir de 100 nœuds dans chaque direction, il devient plus rapide que le solveur  $PGD_{XY}$ , et à partir de 150 nœuds dans chaque direction il devient également plus rapide que le solveur standard. En effet pour le solveur  $PGD_{XYT}$ , le temps de simulation

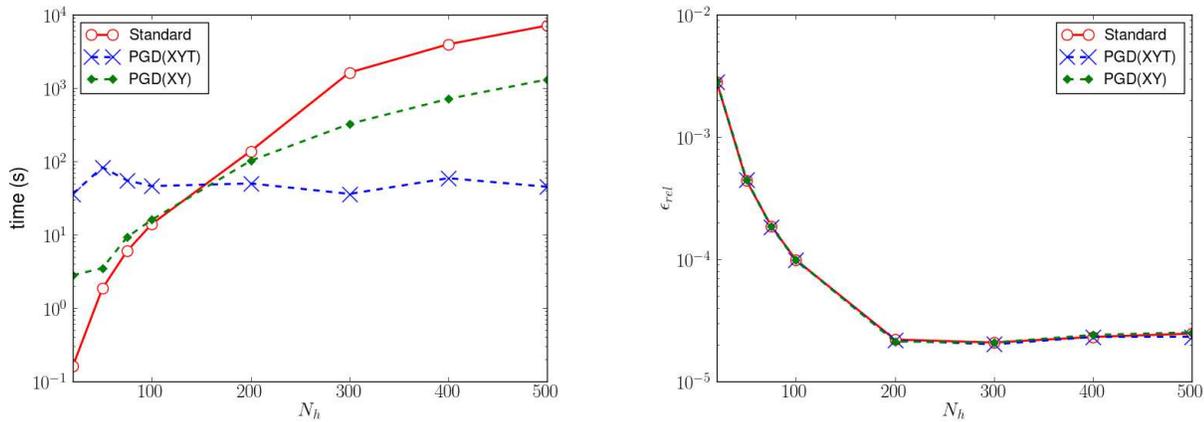


FIG. 2.9 – Temps de calcul (à gauche) et erreur relative (à droite) en fonction du nombre de nœuds dans chacune des directions, avec  $I_t = [0, 1]$  et  $\delta t = 10^{-3}$ .

évolue peu lorsque le nombre de nœuds augmente (entre 40 et 80 secondes) alors que pour les autres solveurs ( $PGD_{XY}$  et solveur standard) le temps de simulation augmente avec le nombre de nœuds. Cette faible évolution du temps de simulation s’explique par le fait que le calcul pour un grand nombre de nœuds dans chaque direction nécessite moins de fonctions pour converger (voir figure 2.10). Le solveur  $PGD_{XY}$  quant à lui devient plus efficace en terme de temps de calcul que le solveur standard à partir de 200 nœuds dans chaque direction, ce qui est en accord avec les résultats obtenus lors de la résolution de l’équation de diffusion stationnaire 2D.

La figure 2.9 (droite) montre l’évolution de l’erreur relative en fonction du nombre de nœuds. On peut constater que le comportement de l’erreur est le même pour les trois schémas étudiés et qu’ils semblent tous les trois atteindre le même palier de convergence.

Ainsi le solveur  $PGD_{XY}$  est plus intéressant que le solveur  $PGD_{XYT}$  pour de petits maillages, même si les temps de calculs observés sont encore bien supérieurs à ceux réalisés par le solveur standard. Par contre pour de grands maillages, la méthode  $PGD_{XYT}$  est plus intéressante que la méthode  $PGD_{XY}$ , elle-même plus intéressante que le solveur standard. En effet pour un maillage  $500 \times 500$  le solveur  $PGD_{XYT}$  nécessite 45 secondes de calcul contre 1307 secondes pour le solveur  $PGD_{XY}$  et 7101 secondes pour le solveur classique.

### b) Influence du pas de temps

Dans ce paragraphe, l’influence du pas de temps est étudié. Pour cela le problème (2.25) est résolu pour un intervalle de temps  $I_t = ]0, 1]$  et pour un maillage  $250 \times 250$  en faisant varier le pas de temps  $\delta t$ .

Comme l’indique, la figure 2.11 (gauche) les comportements du solveur  $PGD_{XY}$  et du solveur standard sont très proche. Il apparaît que le pas de temps a très peu d’influence sur le gain de temps de simulation du solveur  $PGD_{XY}$  par rapport au solveur standard. Ainsi seule la modification du nombre de nœuds rend le solveur  $PGD_{XY}$  plus efficace que le solveur

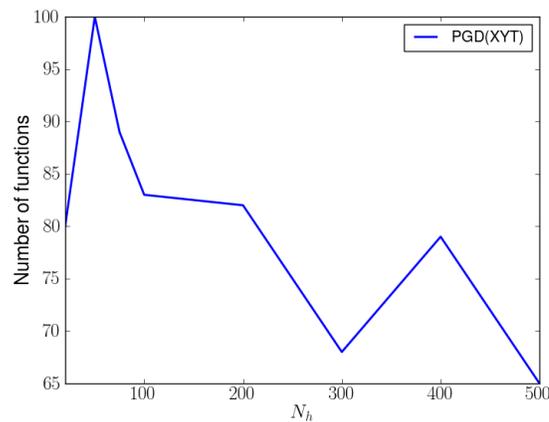


FIG. 2.10 – Nombre de fonctions nécessaires à la convergence de la  $PGD_{XYT}$  en fonction du nombre de nœuds dans chacune des directions avec  $I_t = ]0, 1[$  et  $\delta t = 10^{-3}$  (Axes des ordonnées en échelle logarithmique).

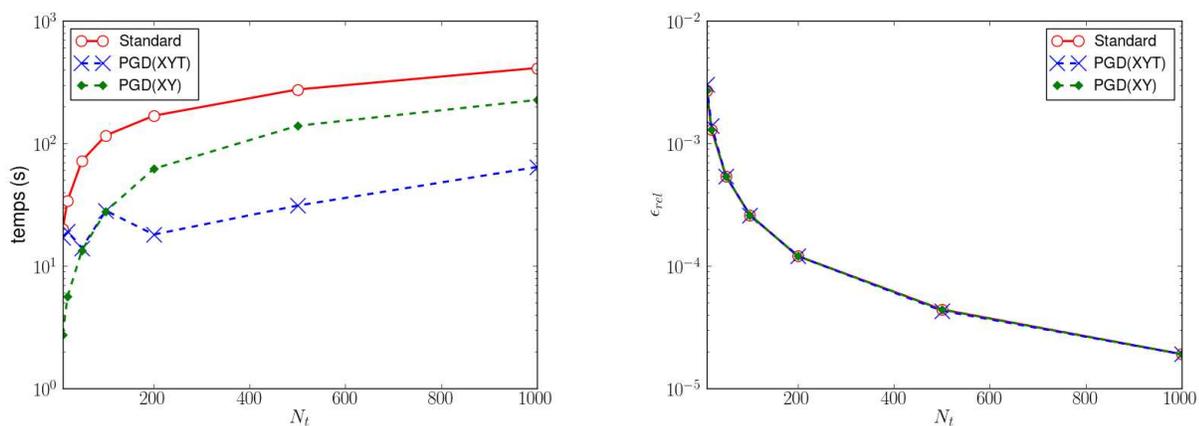


FIG. 2.11 – Temps de calcul (à gauche) et erreur relative (à droite) en fonction du pas de temps pour les solveurs standard,  $PGD_{XY}$  et  $PGD_{XYT}$  avec  $I_t = ]0, 1[$  et  $N_h = 250$ .

classique. Concernant le solveur  $PGD_{XYT}$ , il semble que le pas de temps a une influence plus faible sur le temps de calcul que pour les autres solveurs. On note que pour  $N_t = 1000$  la méthode  $PGD_{XYT}$  est sept fois plus rapide que le solveur standard et quatre fois plus rapide que la méthode  $PGD_{XY}$ .

La figure 2.11 (droite), quant à elle montre une décroissance de l'erreur en fonction du pas de temps utilisé quasiment identique pour chacun des trois solveurs étudiés.

### c) Influence de la durée de simulation

L'influence de l'intervalle de temps de simulation pour chacun des trois solveurs est étudiée. Pour ce faire, le problème (2.25) est résolu pour un pas de temps  $\delta t = 10^{-2}$ , un maillage  $200 \times 200$  et différents intervalles de temps  $I_t$  variant de  $]0; 1]$  à  $]0; 20]$ .

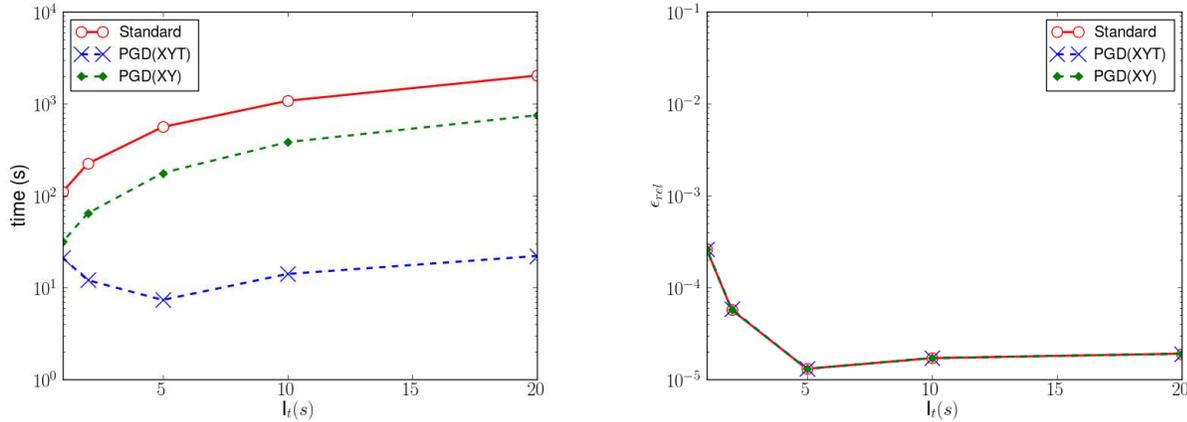


FIG. 2.12 – Temps de calcul (à gauche) et erreur relative (à droite) en fonction de la durée de simulation pour les solveurs standard,  $PGD_{XY}$  et  $PGD_{XYT}$  avec  $\delta t = 10^{-2}$  et  $N_h = 250$ .

La figure 2.12 (gauche) montre que l'évolution du temps de calcul du solveur  $PGD_{XY}$  et du solveur standard est le même. Par contre, contrairement aux solveurs  $PGD_{XY}$  et standard, on constate que l'évolution du temps de simulation pour le solveur  $PGD_{XYT}$  est peu sensible à l'intervalle de temps de calcul.

La figure 2.12 (droite) représente l'erreur relative en fonction de la durée de simulation. On constate ici que le résultat obtenu est exactement le même quelque soit le solveur utilisé. L'erreur diminue jusqu'à atteindre un palier.

### d) Conclusion

Grâce à cet exemple simple, l'efficacité de la PGD par rapport au solveur standard a été mise en évidence. En effet, à partir d'une certaine taille de maillage, quelque soit le solveur PGD utilisé, la PGD est plus rapide que le solveur standard tout en étant aussi précise. De plus lorsque la durée de simulation augmente, la  $PGD_{XYT}$  semble être la plus avantageuse en terme de temps de calcul et cela sans perte de précision sur la solution obtenue.

## IV Equation de Burgers

Nous allons maintenant nous intéresser à l'équation de Burgers qui est une équation de convection diffusion où le terme moteur de la convection est la vitesse elle-même. Cette équation est non-linéaire. Nous allons donc pouvoir juger de la capacité de la méthode à résoudre ce type de problème.

### IV .1 Problème étudié

Le problème de Burgers s'écrit de la manière suivante :

$$\text{Chercher } T(x, y, t) \in \Omega \times [0, T] \text{ tel que } \begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f} \\ \mathbf{u}(t = 0) = \mathbf{u}_0 \\ \mathbf{u} = \mathbf{g}_D \text{ sur } \Gamma \end{cases} \quad (2.29)$$

Nous étudierons cette équation dans le cas à deux dimensions. Dans ce cas,  $\mathbf{u}$  désigne le vecteur vitesse  $(u, v)$ ,  $\mathbf{f}$  le second membre  $(f_x, f_y)$ , et  $\nu$  la viscosité du fluide. Le domaine  $\Omega$  désigne l'espace  $[0.25; 1.25] \times [0.5; 1.5]$ .

Dans la suite on impose le second membre de la façon suivante :

$$\begin{aligned} f_x &= -\pi \sin(4\pi x) * e^{-16\pi^2 \nu t} \\ f_y &= -\pi \sin(4\pi x) * e^{-16\pi^2 \nu t} \end{aligned} \quad (2.30)$$

de façon à avoir la solution analytique suivante :

$$\begin{aligned} u &= -\cos(2\pi x) \sin(2\pi y) e^{-8\pi^2 \nu t} \\ v &= \sin(2\pi x) \cos(2\pi y) e^{-8\pi^2 \nu t} \end{aligned} \quad (2.31)$$

Les conditions aux limites et initiales sont choisies de manière à satisfaire la solution analytique. La figure 2.13 représente le problème étudié ainsi que les conditions aux bords. Le terme de convection est discrétisé en utilisant un schéma d'Adams-Bashforth d'ordre 2. Le terme diffusif quant à lui est discrétisé avec un schéma de Crank-Nicholson<sup>3</sup>. Le problème à résoudre dans la direction  $x$  s'écrit donc :

$$u^{n+1} - \frac{\nu \delta t}{2} \nabla^2 u^{n+1} = u^n + \frac{\nu \delta t}{2} \nabla^2 u^n - C_x(u^n, v^n, u^{n-1}, v^{n-1}) \quad (2.32)$$

où

$$C_x(u^n, v^n, u^{n-1}, v^{n-1}) = \frac{3}{2} \delta t \left[ u^n \frac{\partial u^n}{\partial x} + v^n \frac{\partial u^n}{\partial y} \right] - \frac{1}{2} \delta t \left[ u^{n-1} \frac{\partial u^{n-1}}{\partial x} + v^{n-1} \frac{\partial u^{n-1}}{\partial y} \right] \quad (2.33)$$

---

<sup>3</sup>Nous avons choisi de discrétiser les équations de cette manière car c'est cette discrétisation qui sera utilisée par la suite pour les équations de Navier-stokes.

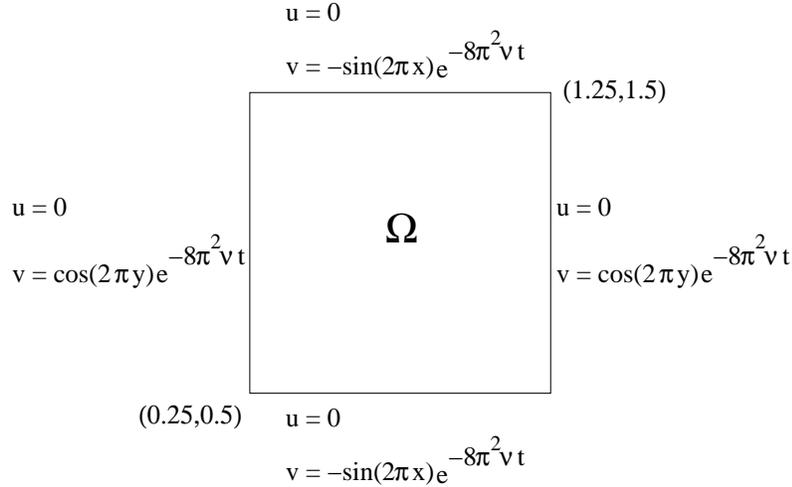


FIG. 2.13 – Geometrie du problème de Burgers.

Dans la direction  $y$  on aura :

$$v^{n+1} - \frac{\nu\delta t}{2}\nabla^2 v^{n+1} = v^n + \frac{\nu\delta t}{2}\nabla^2 v^n - C_y(u^n, v^n, u^{n-1}, v^{n-1}) \quad (2.34)$$

où

$$C_y(u^n, v^n, u^{n-1}, v^{n-1}) = \frac{3}{2}\delta t\left[u^n \frac{\partial v^n}{\partial x} + v^n \frac{\partial v^n}{\partial y}\right] - \frac{1}{2}\delta t\left[u^{n-1} \frac{\partial v^{n-1}}{\partial x} + v^{n-1} \frac{\partial v^{n-1}}{\partial y}\right] \quad (2.35)$$

La PGD est effectuée ici sur les variables spatiales uniquement, cela revient à utiliser un solveur du type  $PGD_{XY}$  (voir section III .1). Cela revient donc à résoudre deux équations de diffusions instationnaire 2D. Les opérateurs à construire pour résoudre le problème de Burgers avec la méthode PGD sont donc ceux explicités dans la partie sur la résolution de l'équation de diffusion instationnaire en utilisant un schéma de discrétisation temporelle classique. Dans les simulations qui suivent, le nombre de nœuds dans chaque direction est le même et est noté  $N_h$ .

## IV .2 Résultats

Le problème de Burgers est résolu en utilisant d'une part la méthode PGD et d'autre part en utilisant un solveur standard (bi-gradient conjugué). Pour ces simulations la viscosité  $\nu$  a été prise égale à  $10^{-2}$ , le pas de temps  $\delta t$  a été pris égal a  $10^{-3}$ . Les calculs ont été effectués sur une durée de simulation  $T=1s$  pour un nombre de nœuds variant de  $N_h = 20$  à  $N_h = 500$ . Les résultats obtenus seront comparés à la solution analytique. Pour cela l'erreur suivante est considérée :

$$\|\mathbf{e}\| = \|\mathbf{u}_{sim} - \mathbf{u}_{ana}\|_{L^\infty(L^2(\Omega))} = \max_{0 < t \leq T} \left[ \int_{\Omega} |\mathbf{u}_{sim} - \mathbf{u}_{ana}|^2 d\Omega \right]^{\frac{1}{2}} \quad (2.36)$$

où  $\mathbf{u}_{sim}$  représente la solution calculée numériquement et  $\mathbf{u}_{ana}$  la solution analytique.

La figure 2.14 représente le champ de vitesse obtenu avec la méthode PGD (à gauche) et la différence entre les champs de vitesse obtenus en utilisant la méthode PGD et la solution analytique (à droite). On remarque que sur cette dernière figure les résultats obtenus par méthode PGD et la solution analytique sont très proches.

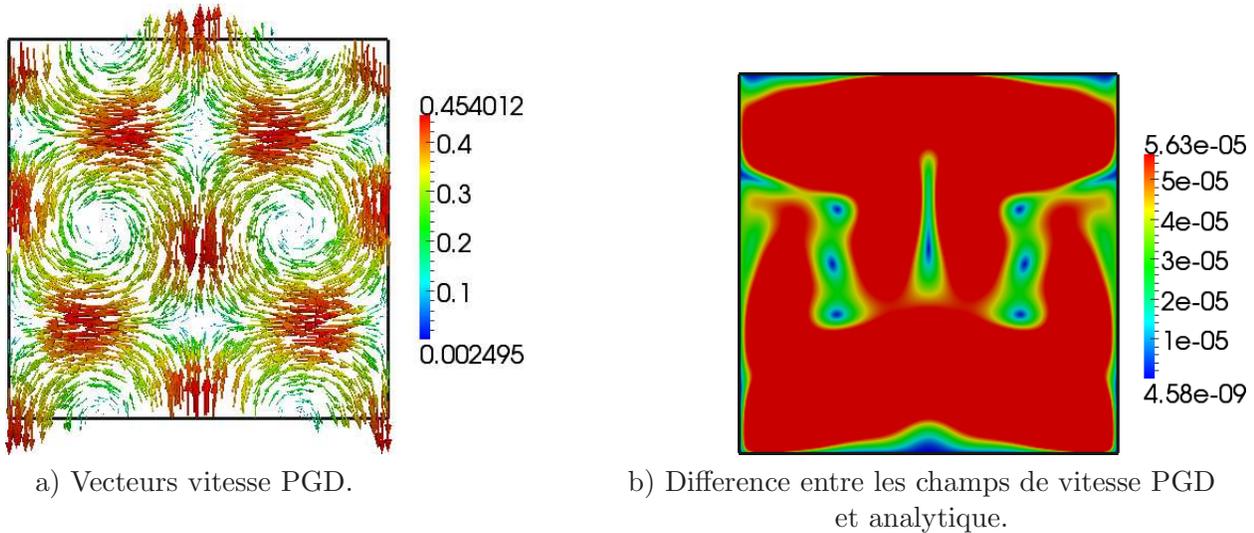


FIG. 2.14 – Vecteurs vitesse calculés par PGD (gauche) et différence entre les champs de vitesse calculés par PGD et la solution analytique à  $T = 1s$  avec  $N_h = 250$ .

La figure 2.15 (gauche) représente la comparaison du temps de calcul nécessaire à la résolution du problème de Burgers, entre la méthode PGD et le solveur standard (une échelle semi-logarithmique a été utilisée). On observe qu'à partir d'un certain nombre de nœuds, (ici  $N_h = 175$ ) la méthode PGD est plus performante en temps de calcul que le solveur standard. Avec un maillage de taille  $500 \times 500$ , la PGD est quatre fois plus rapide que le solveur standard.

La figure 2.15(droite) montre l'évolution de l'erreur en vitesse (Eq. 2.36) entre les solutions calculées par PGD et méthode standard, et la solution analytique en fonction du nombre de nœuds de discrétisation  $N_h$ . On observe que le comportement de la PGD et du solveur standard est à peu près le même, et que cette erreur est de l'ordre de  $10^{-4}$  pour les deux solveurs.

Enfin, il est important de noter que lorsque l'on utilise la méthode PGD, un seul enrichissement est nécessaire à chaque pas de temps. La méthode PGD retrouve donc directement la solution analytique sous sa forme séparée. La figure 2.16 sur laquelle sont tracées les fonctions  $F_1(x)$  et  $G_1(y)$ , solution de la résolution par méthode PGD, ainsi que celles associées à la décomposition de la solution analytique en fonction de  $x$  et de  $y$ . Les courbes se superposent exactement. On peut donc en conclure que la méthode PGD retrouve la solution analytique

sous sa forme séparée en un nombre minimal d'enrichissement.

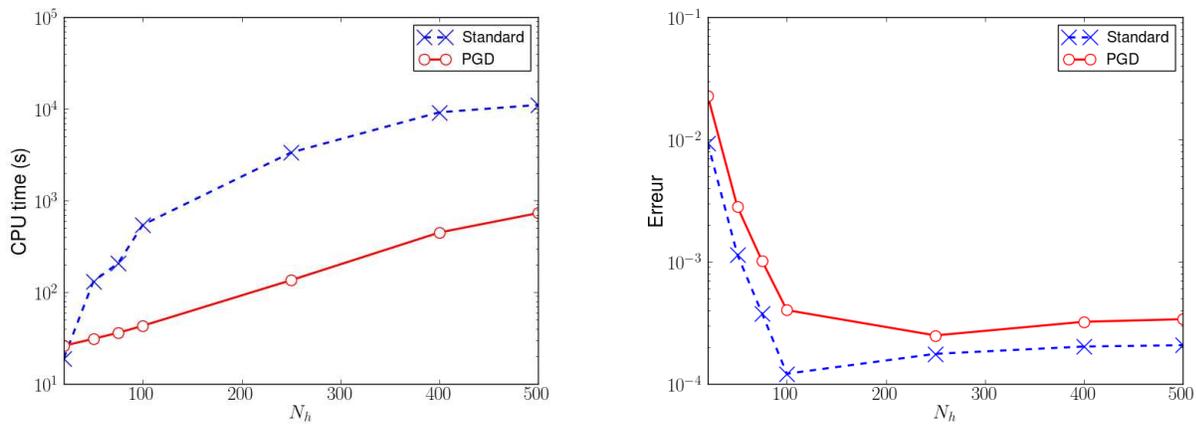


FIG. 2.15 – Temps de calcul (gauche) et erreur en vitesse (droite) en fonction du nombre de nœuds pour une durée de simulation  $T=1s$ .

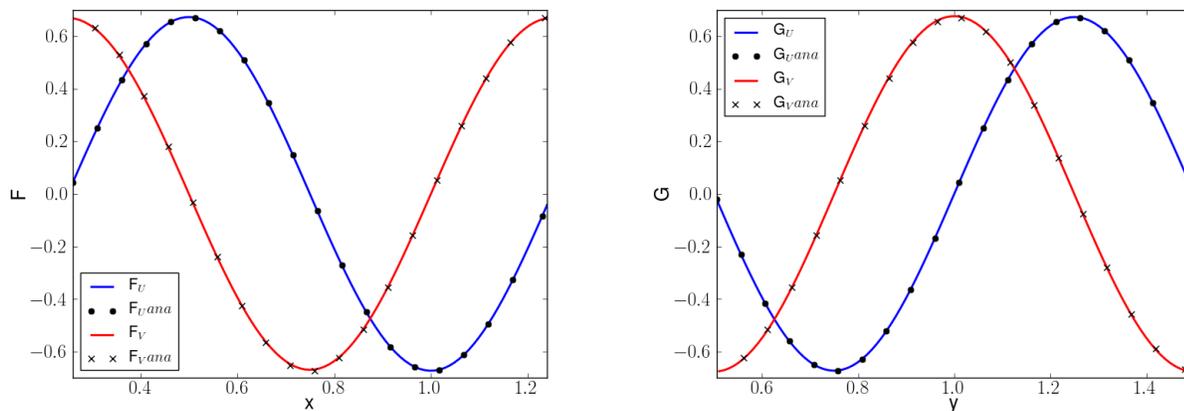


FIG. 2.16 – Fonctions solution de la méthode PGD et issues de la solution analytique représentant les champs  $u$  et  $v$  pour une durée de simulation  $T = 1s$ .

Finalement on observe que le problème de Burgers est bien résolu par la méthode PGD et de manière optimale lorsque la solution analytique est décomposable comme un produit de fonction de  $x$  et de  $y$ . Cela fait de la PGD un bon candidat pour la résolution de problème non-linéaire.

## V Equation de Stokes

Nous allons maintenant nous intéresser au problème de Stokes qui présente la particularité d'être un problème où la vitesse et la pression sont couplées. Nous allons donc pouvoir juger de la capacité de la méthode à résoudre ce type de problème.

### V.1 Problème étudié

En deux dimensions le problème de Stokes s'écrit de la manière suivante :

$$\text{Chercher } T(x, y, t) \in \Omega \times ]0, T[ \begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} = -\nabla p + \mathbf{f} \\ \mathbf{u}(t=0) = \mathbf{u}_0 \\ \mathbf{u} = \mathbf{g}_D \text{ on } \Gamma \end{cases} \quad (2.37)$$

où  $\nu$  représente la viscosité,  $\mathbf{u}$  représente la vitesse du fluide (ici  $\mathbf{u}=(u,v)$ ) et  $\Omega = [0.25; 1.25] \times [0.5; 1.5]$ .

De la même manière que pour l'équation de Burgers, on se fixe un second membre de la manière suivante :

$$\begin{aligned} f_x &= (1 - 2\pi)(8\pi^2\nu - 1) \cos(2\pi x) \sin(2\pi y) e^{-t} \\ f_y &= (1 + 2\pi)(1 - 8\pi^2\nu) \cos(2\pi x) \sin(2\pi y) e^{-t} \end{aligned} \quad (2.38)$$

De façon à avoir la solution analytique suivante :

$$\begin{aligned} u &= \cos(2\pi x) \sin(2\pi y) e^{-t} \\ v &= -\sin(2\pi x) \cos(2\pi y) e^{-t} \\ p &= (1 - 8\pi^2\nu) \sin(2\pi x) \sin(2\pi y) e^{-t} \end{aligned} \quad (2.39)$$

Ce problème et les conditions au bords sont représentés sur la figure 2.17.

Le problème de Stokes est résolu en utilisant un algorithme de prédiction-corréction afin de découpler la vitesse et la pression. Cet algorithme ainsi que les méthodes de prédiction-corréction seront détaillés dans le chapitre suivant pour la résolution des équations de Navier-Stokes (voir chapitre 3 section I). En quelques mots, cette méthode consiste à :

- 1) Calculer un champ de vitesse intermédiaire  $(\tilde{u}, \tilde{v})$  solution d'une équation de diffusion instationnaire avec des conditions au bord de type Dirichlet.
- 2) Calculer un champ de pression intermédiaire  $\tilde{p}$  solution d'une équation de Poisson avec des conditions au bord de type Neumann et un second membre dépendant de la vitesse intermédiaire calculée précédemment.
- 3) Remettre à jour à l'aide de ces champs intermédiaires les vitesses  $u, v$  et la pression  $p$ .

Dans le cas de la résolution PGD, les inconnues  $\tilde{u}, \tilde{v}$  et  $\tilde{p}$  sont recherchées à chaque pas de temps  $k + 1$  sous la forme séparée suivante :

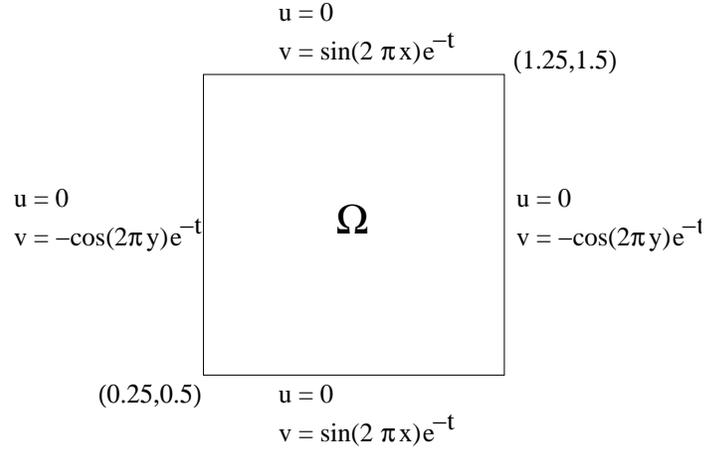


FIG. 2.17 – Geometrie pour le problème de Stokes.

$$\tilde{u}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{u}}^{k+1}} \alpha_{\tilde{u}}^{i,k+1} F_{\tilde{u}}^{i,k+1}(x) G_{\tilde{u}}^{i,k+1}(y) \quad \tilde{v}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{v}}^{k+1}} \alpha_{\tilde{v}}^{i,k+1} F_{\tilde{v}}^{i,k+1}(x) G_{\tilde{v}}^{i,k+1}(y) \quad (2.40)$$

$$\tilde{p}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{p}}^{k+1}} \alpha_{\tilde{p}}^{i,k+1} F_{\tilde{p}}^{i,k+1}(x) G_{\tilde{p}}^{i,k+1}(y) \quad (2.41)$$

Les opérateurs utilisés seront donc ceux de l'équation de diffusion instationnaire et de l'équation de Poisson, déjà construits précédemment.

Dans la suite, le nombre de nœuds dans chaque direction est le même et est noté  $N_h$ .

## V.2 Résultats

De la même manière que pour l'équation de Burgers, le problème de Stokes est résolu à l'aide de la méthode PGD et à l'aide d'un solveur standard (bi-gradient conjugué). Les simulations sont effectuées avec une viscosité  $\nu = 10^{-2}$  et un pas de temps  $\delta t = 10^{-3}$  pour une durée de simulation  $T=1s$ .

La figure 2.18 représente le champ de vitesse obtenu avec la méthode PGD (à gauche) et la différence entre les champs de vitesse obtenus en utilisant la méthode PGD et la solution analytique (à droite). De même la figure 2.19 illustre le champ de pression calculé par méthode PGD (à gauche) et la différence entre la pression calculée par PGD et la pression analytique. On constate que les résultats obtenus par la méthode PGD sont proches des solutions analytiques (de l'ordre de  $10^{-5}$  pour la vitesse et  $10^{-3}$  pour la pression).

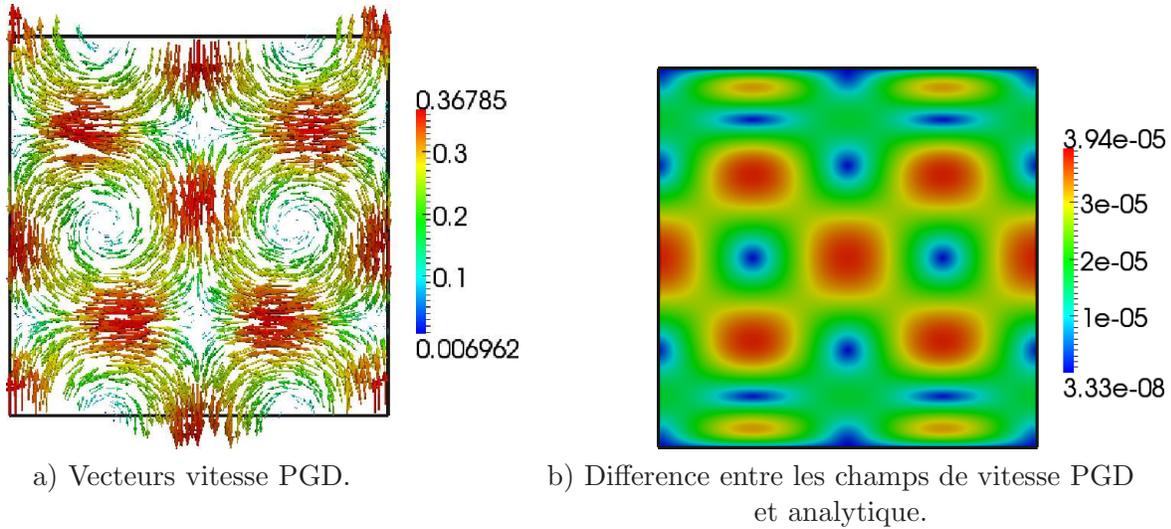


FIG. 2.18 – Vecteurs vitesse calculés par PGD (gauche) et différence entre les champs de vitesse calculés par PGD et la solution analytique à  $T=1s$  avec  $N_h = 250$ .

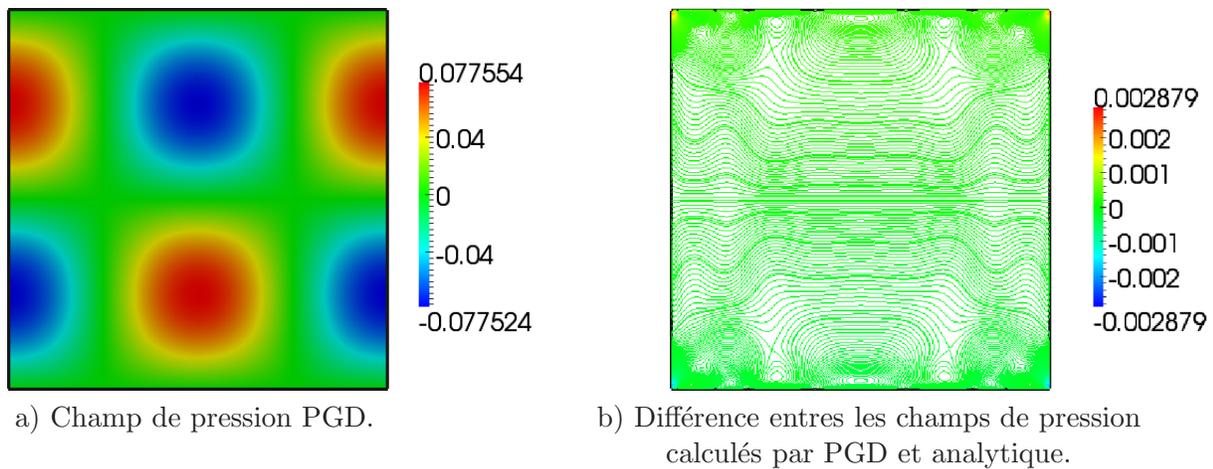


FIG. 2.19 – Champ de pression calculé par PGD (gauche) et différence entre les champs de pression calculés par PGD et la la solution analytique à  $T=1s$  avec  $N_h = 250$ .

Pour pouvoir quantifier l'erreur commise par la simulation numérique l'erreur en vitesse définie lors de l'étude du cas du problème de Burgers dans l'équation (2.36) est réutilisé. Une erreur en pression est également définie de la manière suivante :

$$\|\eta\| = \|p_{sim} - p_{ana}\|_{L^2(L^2(\Omega))} = \left[ \int_0^T \int_{\Omega} |p_{sim} - p_{ana}|^2 d\Omega dt \right]^{\frac{1}{2}} \quad (2.42)$$

La figure 2.20 représente l'évolution de ces erreurs en fonction du nombre de nœuds de discrétisation (la discrétisation est la même dans les deux directions). Sur cette figure on

constate que les erreurs en vitesse pour la PGD et la méthode standard sont très proches. Concernant les erreurs en pression on constate qu'elles ont un comportement identique mais on note une différence quantitative. En effet pour  $N_h = 500$ , l'erreur calculée à partir du solveur standard est de l'ordre de  $10^{-4}$  contre une erreur de l'ordre de  $10^{-3}$  pour la PGD.

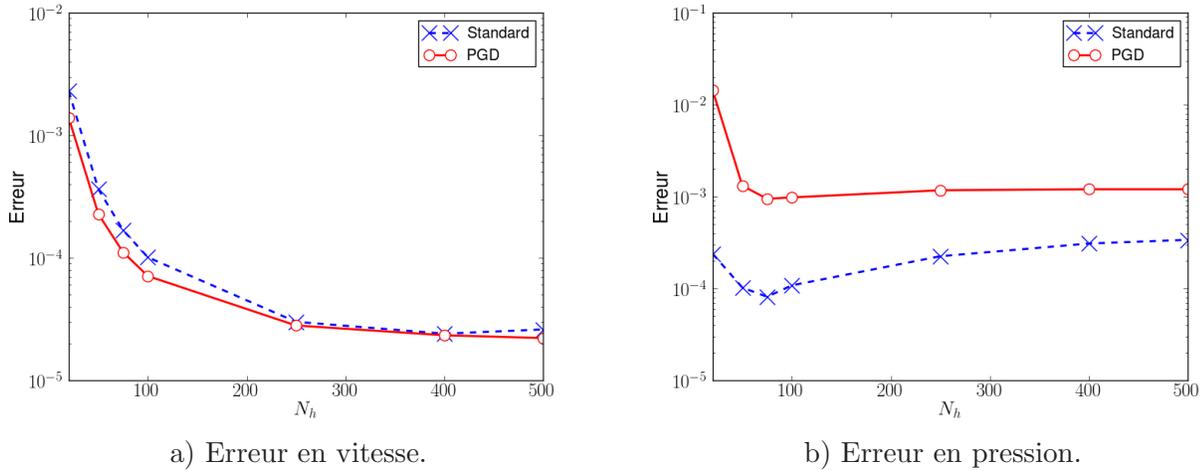


FIG. 2.20 – Evolution des erreurs en vitesse et en pression en fonction du nombre de nœuds pour la méthode PGD et le solveur standard.

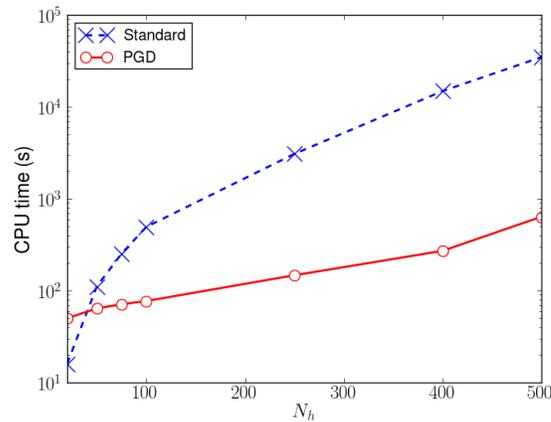


FIG. 2.21 – Comparaison du temps de calcul entre la méthode PGD et le solveur standard.

La figure 2.21 représente les temps de calculs pour la résolution du problème de Stokes en utilisant la méthode PGD et le solveur standard en fonction du nombre de nœuds dans chaque direction pour un temps de simulation  $T=1s$ . On constate sur cette figure qu'à partir d'un certain nombre de nœuds ( $N_h = 50$ ), la méthode PGD devient plus économique en temps de calcul que le solveur standard. En fait plus on augmente le nombre de nœuds dans chaque

direction, plus la méthode PGD devient intéressante par rapport au solveur classique. En effet pour un maillage  $100 \times 100$  la PGD est sept fois plus rapide que le solveur standard, alors que pour un maillage  $500 \times 500$  la PGD est trente-cinq fois plus rapide que le solveur standard.

De la même manière que pour le problème de Burgers, nous avons choisi des solutions analytiques décomposables en un produit de fonctions exactement. On observe lors de la résolution PGD que la solution est obtenue en exactement un enrichissement pour la vitesse et la pression. Les fonctions  $F(x)$  et  $G(y)$  calculées pour la vitesse et la pression par PGD sont comparées avec les fonctions issues des solutions analytiques sur les figures 2.22 et 2.23. On constate effectivement que pour la vitesse les courbes sont parfaitement superposées et que pour la pression les courbes ont un comportement globalement bon.

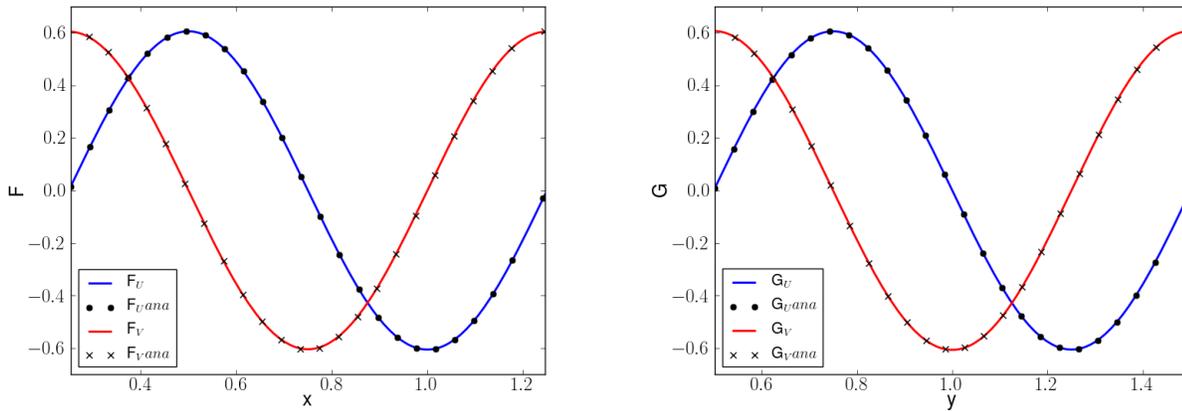


FIG. 2.22 – Fonctions de la décomposition issues de la PGD et de la solution analytique des champs  $u$  and  $v$  après un temps de simulation  $T=1s$  ( $N_h = 250$ ).

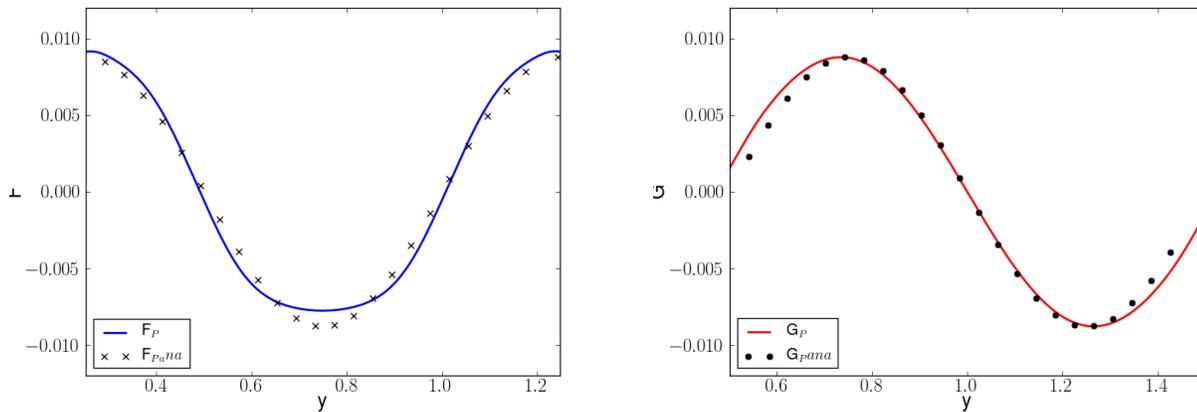


FIG. 2.23 – Fonctions de la décomposition issues de la PGD et de la solution analytique du champ  $p$  après un temps de simulation  $T=1s$  ( $N_h = 250$ ).

Concernant le problème de Stokes, on a montré que la méthode PGD était une alternative intéressante en terme de précision et de temps de calcul pour traiter ce genre de problème, et qu'elle était même optimale dans le cas où la solution est décomposable en un seul produit de fonction. La PGD peut donc être utilisée pour résoudre des problèmes où un couplage entre la vitesse et la pression existe.

## VI Conclusions du chapitre

Le cas des équations de diffusion stationnaire 2D et 3D a permis de montrer l'intérêt de la méthode en terme de temps de calcul (gains de temps d'un facteur vingt avec des conditions aux limites de type Dirichlet et cinquante avec des conditions aux limites de type Neumann pour  $N_h = 1000$ ) à partir d'une certaine taille de maillage, et ce, en deux dimensions comme en trois dimensions (en 3D, nous avons observé des gains de temps d'un facteur douze avec des conditions aux limites de type Dirichlet et vingt avec des conditions aux limites de type Neumann pour  $N_h = 200$ ). De plus l'erreur commise entre la solution analytique et la solution simulée est la même que l'on utilise la PGD ou le solveur standard. Nous avons cependant remarqué que le nombre de fonctions PGD nécessaire à la reconstruction de la solution n'était pas forcément optimal. La séparation en temps et en espace a montré des gains de temps intéressants (gains de temps d'un facteur cent cinquante pour la  $PGD_{XYT}$  et six dans le cas de la  $PGD_{XY}$  pour  $N_h = 500$ ) lorsque le problème est résolu sur de longues durées sans perte de précision par rapport aux schémas incrémentaux classiques, en effet dans ce cas la méthode  $PGD_{XYT}$  semble être la méthode appropriée pour résoudre ce genre de problème. L'équation de Burgers a ensuite été traitée afin d'étudier la capacité de la méthode à traiter des problèmes non linéaires. Dans ce cas, la méthode PGD a montré un gain de temps (la PGD est quatre fois plus rapide que le solveur standard pour  $N_h = 500$ ) par rapport aux méthodes classiques avec une précision correcte comparativement aux solveurs standards. Enfin le problème de Stokes a été traité afin de mettre en évidence le comportement de la PGD face à des problèmes couplés. De la même manière que pour les précédents tests, un gain de temps a été observé (la PGD est trente cinq fois plus rapide que le solveur standard pour  $N_h = 500$ ) malgré une précision sur la pression un peu moins bonne qu'avec le solveur standard. La suite logique de ces tests consiste à traiter le problème de Navier-Stokes qui est un problème à la fois non-linéaire et faisant intervenir un couplage entre la vitesse et la pression. Ce sera donc l'objet du prochain chapitre.

## Chapitre 3

# PGD pour la résolution des équations de Navier-Stokes isothermes

Jusqu'à présent la méthode PGD a été appliquée sur des équations modèles à savoir des équations de diffusion, le problème de Stokes et les équations de Burgers. Dans ce chapitre nous appliquerons la PGD à la résolution des équations de Navier-Stokes isothermes.

Tout d'abord il est important de noter qu'il existe plusieurs manières de traiter le couplage de la vitesse et de la pression. On peut citer tout d'abord les méthodes dites de pénalisation qui ont l'intérêt d'être facile à mettre en œuvre. En effet, sous une technique de pénalisation la condition d'incompressibilité s'écrit :

$$\nabla \cdot \mathbf{u} - \frac{p}{\kappa} = 0 \quad (3.1)$$

où  $p$  est la pression et  $\kappa$  un très grand paramètre. Ce dernier est en général caractérisé par une fonction physique du problème qui peut être la viscosité dynamique  $\mu$  ou encore le nombre de Reynolds  $Re$  (Zienkiewicz et al. [Zienkiewicz et Taylor, 1991]). En général

$$\kappa = c\mu \text{ ou } \kappa = c \cdot \max\{\mu, Re\} \quad (3.2)$$

où  $c$  est une constante sans dimension comprise entre  $10^7$  et  $10^{10}$ .

Une deuxième manière de traiter ce couplage consiste à utiliser des schémas de projection (également appelées méthodes à pas fractionnaire). Ces méthodes, qui sont les plus utilisées, peuvent être séparées en trois grandes classes : (i) les méthodes de correction de pression, (ii) les méthodes de correction de vitesse, et (iii) les méthodes fractionnaires consistantes. Dans la suite de ce document, nous ne détaillerons que la famille de méthodes utilisée à savoir celle de correction de pression. Pour plus d'informations, une revue des différentes méthodes de projection est disponible dans [Guermont et al., 2006].

Une autre famille de méthode, couramment utilisée en mécanique des fluides, consiste à modéliser l'écoulement en utilisant les variables lignes de courant et vorticit . Cette m thode pr sente l'inconv nient de ne pas donner acc s directement au champ de pression hydrostatique, cependant, il arrive que dans certains cas, une connaissance fine des champs de pression ne soit pas n cessaire. Cette m thode a  t  appliqu e dans de nombreux cas acad miques en 2D (voir [Chudanov et al., 1995, Kim et al., 2007, Ramsak et al., 2005]) ainsi qu'en 3D ([Ravnik et al., 2009, Lo et al., 2007]) m me si dans ce cas la formulation est beaucoup plus difficile   mettre en  uvre.

Dans la suite, apr s une description des m thodes de correction de pression et de l'algorithme PGD associ , la m thode PGD sera appliqu e   la r solution du probl me de "Taylor-Green" (dont une solution analytique est connue), puis au cas de la cavit  entrain e 2D en r gime stationnaire et p riodique et enfin au cas de la cavit  entrain e 3D en r gime stationnaire. Dans tous les cas les r sultats obtenus seront compar s en terme de pr cision et de temps de simulation   ceux obtenus par une r solution de type bi-gradient conjugu . La m thode PGD coupl e   une formulation lignes de courant vorticit  a  galement  t  test e sur le cas de la cavit  entrain e 2D, les d tails sur cette formulation ainsi que les r sultats obtenus ont  t  report s dans l'annexe A.

## I Formulation des schémas de prédiction-correction

Comme le terme non-linéaire (terme convectif) des équations de Navier-Stokes n'affecte pas la convergence des schémas de prédiction-correction (voir [Guermont et al., 2006])<sup>1</sup>, la présentation de ces schémas sera appliquée au problème de Stokes instationnaire qui contient le problème de couplage entre les variables de vitesse et de pression. Les équations de Stokes instationnaire s'écrivent :

$$\text{Trouver } \mathbf{u}(x, y, t) \in \Omega \times ]0, T] \left\{ \begin{array}{l} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re} \nabla^2 \mathbf{u} = -\nabla p + \mathbf{f} \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}_D \text{ sur } \Gamma, \text{ et } \mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0 \end{array} \right. \quad (3.3)$$

où  $\mathbf{u} = (u, v)$  et  $p$  sont respectivement les champs de vitesse et de pression,  $\mathbf{f}$  est une force connue,  $Re$  est le nombre de Reynolds,  $\Omega$  un ouvert de  $\mathbb{R}^d$ ,  $\Gamma$  la frontière de ce domaine.

Les méthodes de correction de pression sont des méthodes itératives composées de deux étapes distinctes à chaque pas de temps. Dans un premier temps, une vitesse intermédiaire est recherchée en résolvant les équations de quantité de mouvement en considérant la pression de manière explicite ou en l'ignorant. Dans un second temps, la pression est corrigée en projetant la vitesse intermédiaire calculée à la première étape sur l'espace  $\mathcal{H}$  défini de la manière suivante :

$$\mathcal{H} = \{ \mathbf{u} \in [L^2(\Omega)]^d; \nabla \cdot \mathbf{u} = 0; \mathbf{u} \cdot \mathbf{n}|_{\Gamma} = g_D \cdot \mathbf{n}|_{\Gamma} \} \quad (3.4)$$

où  $d$  est la dimension du problème. Par la suite le pas de temps sera noté  $\delta t$ . Il existe trois grandes familles de méthodes de correction de pression :

- 1) les méthodes non-incrémentales.
- 2) les méthodes incrémentales standards.
- 3) les méthodes incrémentales rotationnelles.

### I.1 Les schémas de correction de pression non-incrémentaux.

Le schéma de correction le plus simple a été proposé pour la première fois par Chorin et Temam ([Chorin, 1968],[Temam, 1969]). En discrétisant la dérivée temporelle à l'aide d'un schéma d'Euler implicite, le problème (3.3) s'écrit :

$$\left\{ \begin{array}{l} \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\delta t} - \frac{1}{Re} \nabla^2 \mathbf{u}^{k+1} = -\nabla p^{k+1} + \mathbf{f}^{k+1} \\ \nabla \cdot \mathbf{u}^{k+1} = 0 \\ \mathbf{u}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{array} \right. \quad (3.5)$$

Une vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$  est alors définie de la manière suivante :

---

<sup>1</sup>De plus, en pratique, le terme convectif est généralement discrétisé de façon explicite par un schéma d'Adams-Bashforth par exemple

$$\mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \delta t \nabla p^{k+1} \quad (3.6)$$

En injectant cette expression dans (3.5) et en négligeant le terme du second ordre  $\frac{\delta t}{Re} \nabla^2 (\nabla p^{k+1})$  (ici on se contente d'un schéma du premier ordre) la vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$  vérifie le problème suivant :

$$\begin{cases} \frac{\tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k}{\delta t} - \frac{1}{Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} = \mathbf{f}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.7)$$

On observe que l'équation (3.6) impose la condition limite  $\nabla p^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0$  qui est une condition artificielle sur la pression. La pression  $p^{k+1}$  est ensuite obtenue en prenant la divergence de l'équation (3.6) et en tenant compte du fait que  $\nabla \cdot \mathbf{u} = 0$ ,

$$\begin{cases} \nabla^2 p^{k+1} = \frac{1}{\delta t} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \\ \nabla p^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (3.8)$$

Le schéma de correction de pression non-incrémental consiste donc à résoudre l'équation (3.7) afin d'obtenir une vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$ , puis, à partir de cette vitesse intermédiaire, à calculer la pression au temps  $t^{k+1}$  à l'aide du système (3.8). Enfin la variable de vitesse est remise à jour à l'aide de la définition de la vitesse intermédiaire donnée par l'équation (3.6).

## I.2 Les schémas de correction de pression incrémentaux standards.

Dans la méthode précédente, le gradient de pression est inexistant dans l'équation (3.7). Or il a été observé, notamment par Goda [Goda, 1979], que rajouter une ancienne valeur du gradient de pression (par exemple  $\nabla p^k$ ) dans la première étape et réajuster la vitesse dans la deuxième étape augmente la précision de la méthode. Cette idée a été rendue populaire par Van-Kan [van Kan, 1986] qui a proposé un schéma de correction de pression incrémental du second ordre. Dans ce cas, un schéma "BDF2" (Backward Difference Formula of 2<sup>nd</sup> order) est utilisé pour discrétiser la dérivée temporelle. Ainsi le problème (3.3) s'écrit :

$$\begin{cases} \frac{3\mathbf{u}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\delta t} - \frac{1}{Re} \nabla^2 \mathbf{u}^{k+1} = -\nabla p^{k+1} + \mathbf{f}^{k+1} \\ \nabla \cdot \mathbf{u}^{k+1} = 0 \\ \mathbf{u}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.9)$$

La vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$  est définie de la manière suivante

$$\mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \frac{2\delta t}{3} \nabla \tilde{p}^{k+1} \quad (3.10)$$

où  $\tilde{p}^{k+1} = p^{k+1} - p^k$  est la correction de pression.

En injectant cette expression dans (3.9) et en négligeant le terme  $\frac{2\delta t}{3Re}\nabla^2(\nabla\tilde{p})$ , la vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$  vérifie le problème suivant

$$\begin{cases} \frac{(3\tilde{\mathbf{u}}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1})}{2\delta t} - \frac{1}{Re}\nabla^2\tilde{\mathbf{u}}^{k+1} + \nabla p^k = \mathbf{f}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.11)$$

De la même manière que pour le schéma non-incrémental, l'équation (3.10) implique l'existence d'une condition aux limite en pression non physique :  $\nabla\tilde{p}^{k+1}.n|_{\Gamma} = 0$ .

L'équation permettant de calculer la correction de pression est déterminée en prenant la divergence de l'équation (3.10), soit

$$\begin{cases} \nabla^2\tilde{p}^{k+1} = \frac{3}{2\delta t}\nabla.\tilde{\mathbf{u}}^{k+1} \\ \nabla p^{k+1}.n|_{\Gamma} = \dots = \nabla p^0.n|_{\Gamma} \end{cases} \quad (3.12)$$

Pour le schéma de correction de pression incrémental standard il faut résoudre successivement le système (3.11) afin d'obtenir une vitesse de correction  $\tilde{\mathbf{u}}^{k+1}$  et le système (3.12) afin d'obtenir la pression de correction  $\tilde{p}^{k+1}$ . Enfin les variables de vitesse et de pression sont remises à jour à l'aide de l'équation (3.10) et de la définition de la correction de pression.

### I.3 Les schémas de correction de pression incrémentaux rotationnels.

Les deux schémas précédents impliquent une condition limite en pression non-physique qui est censé réduire la précision du schéma, c'est pourquoi une nouvelle formulation de l'algorithme a été proposée par Timmermans et al. [Timmermans et al., 1996]. Cette fois ci, la vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$  est définie par :

$$\mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \frac{2\delta t}{3}\nabla\phi^{k+1} \quad (3.13)$$

où

$$\phi^{k+1} = p^{k+1} - p^k + \frac{1}{Re}\nabla.\tilde{\mathbf{u}}^{k+1} \quad (3.14)$$

Dans ce cas l'équation permettant de calculer la vitesse intermédiaire est la même que pour le schéma incrémental standard, mais cette fois ci le terme  $-\frac{1}{Re}\nabla.\tilde{\mathbf{u}}^{k+1} - \frac{2\delta t}{3Re}\nabla^2(\nabla\phi^{k+1})$  a été négligé. On a donc

$$\begin{cases} \frac{(3\tilde{\mathbf{u}}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1})}{2\delta t} - \frac{1}{Re}\nabla^2\tilde{\mathbf{u}}^{k+1} + \nabla p^k = \mathbf{f}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.15)$$

En prenant la divergence de l'équation (3.13), on obtient l'équation sur  $\phi^{k+1}$ , soit

$$\begin{cases} \nabla^2 \phi^{k+1} = \frac{3}{2\delta t} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \\ \nabla \phi^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (3.16)$$

Cette fois, contrairement aux schémas incrémentaux standards, la condition aux limites en pression est physique. En effet en prenant la somme des équations (3.13) et (3.15), on obtient

$$\frac{1}{2\delta t} (3\mathbf{u}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}) - \frac{1}{Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} + \nabla p^{k+1} + \frac{1}{Re} \nabla(\nabla \cdot \tilde{\mathbf{u}}^{k+1}) = \mathbf{f}^{k+1} \quad (3.17)$$

Or on sait que :

$$\nabla(\nabla \cdot \tilde{\mathbf{u}}^{k+1}) = \nabla \times (\nabla \times \tilde{\mathbf{u}}^{k+1}) + \nabla^2 \tilde{\mathbf{u}}^{k+1}$$

De plus, de l'équation (3.13) on a

$$\nabla \times (\nabla \times \tilde{\mathbf{u}}^{k+1}) = \nabla \times (\nabla \times \mathbf{u}^{k+1})$$

(en effet,  $\nabla \times (\nabla \phi^{k+1}) = 0$ ). Ainsi on obtient :

$$\frac{1}{2\delta t} (3\mathbf{u}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}) + \frac{1}{Re} \nabla \times \nabla \times \mathbf{u}^{k+1} + \nabla p^{k+1} = \mathbf{f}^{k+1} \quad (3.18)$$

On en déduit la condition limite en pression :

$$\nabla p^{k+1} \cdot \mathbf{n}|_{\Gamma} = (\mathbf{f}^{k+1} - \frac{1}{Re} \nabla \times \nabla \times \mathbf{u}^{k+1}) \cdot \mathbf{n}|_{\Gamma} \quad (3.19)$$

## I.4 Généralisation

Nous allons maintenant généraliser l'écriture de ces différents algorithmes de correction de pression<sup>2</sup>. On suppose que la dérivée temporelle est discrétisée par un schéma *BDF* (Back Difference Formula) d'ordre<sup>3</sup>  $q$ , d'où

$$\frac{\partial u}{\partial t} = \frac{1}{\delta t} (\beta_q v^{k+1} - \sum_{j=0}^{q-1} \beta_j v^{k-j}) \quad (3.20)$$

De la même manière on notera l'extrapolation d'ordre  $r$  de  $p(t^{k+1})$

$$p^{*,k+1} = \sum_{j=0}^{r-1} \gamma_j p^{k-j} \quad (3.21)$$

En particulier,

---

<sup>2</sup>Il est à noter que ces schémas sont présentés en prenant le terme diffusif en implicite, mais la démarche reste valable lorsque en le discrétisant par un schéma de Crank-Nicholson par exemple. C'est notamment ce que l'on fera dans la suite de ce document.

<sup>3</sup>En prenant  $q = 1$ , on retrouve le schéma d'Euler implicite.

$$p^{*,k+1} = \begin{cases} 0 & \text{si } r = 0 \\ p^k & \text{si } r = 1 \\ 2p^k - p^{k-1} & \text{si } r = 2 \end{cases} \quad (3.22)$$

Avec ces notations, la vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1}$ , est solution du système d'équations :

$$\begin{cases} \frac{1}{\delta t} \left( \beta_q \tilde{\mathbf{u}}^{k+1} - \sum_{j=0}^{q-1} \beta_j \mathbf{u}^{k-j} \right) - \frac{1}{Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} + \nabla p^{*,k+1} = \mathbf{f}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.23)$$

L'étape de correction de pression s'écrit quant à elle,

$$\begin{cases} \frac{\beta_q}{\delta t} (\mathbf{u}^{k+1} - \tilde{\mathbf{u}}^{k+1}) + \nabla \phi^{k+1} = 0 \\ \nabla \cdot \mathbf{u}^{k+1} = 0, \quad \mathbf{u}^{k+1} \cdot \mathbf{n}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.24)$$

avec

$$\phi^{k+1} = p^{k+1} - p^{*,k+1} + \frac{\chi}{Re} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \quad (3.25)$$

Le coefficient  $\chi$  est défini par l'utilisateur et peut être égal à 0 ou 1. Le choix  $\chi = 0$  renvoie à des schémas de correction de pression standards et le choix  $\chi = 1$  renvoie à un schéma de correction de pression rotationnel.

En prenant la divergence de l'équation (3.24) et en considérant que  $\nabla \cdot \mathbf{u}^{k+1} = 0$ , on obtient le problème de Poisson suivant vérifié par  $\phi$  :

$$\begin{cases} \nabla^2 \phi^{k+1} = \frac{\beta_q}{\delta t} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \\ \nabla \phi^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (3.26)$$

En remarquant que

$$\tilde{\mathbf{u}}^{k-j} = \mathbf{u}^{k-j} + \frac{\delta t}{\beta_q} \nabla \phi^{k-j} \quad (3.27)$$

le système d'équation (3.23) peut s'écrire

$$\begin{cases} \frac{1}{\delta t} \left( \beta_q \tilde{\mathbf{u}}^{k+1} - \sum_{j=0}^{q-1} \beta_j \tilde{\mathbf{u}}^{k-j} \right) - \frac{1}{Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} + \nabla \left( p^{*,k+1} + \sum_{j=0}^{q-1} \frac{\beta_j}{\beta_q} \phi^{k-j} \right) = \mathbf{f}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (3.28)$$

Les schémas de correction de pression consistent donc à résoudre le système (3.28) afin de calculer la vitesse intermédiaire, puis l'étape de correction est effectuée en résolvant le système (3.26). Enfin les variables de vitesses et de pression au pas de temps  $k + 1$  sont remises à jour en utilisant les équations (3.27) et (3.25).

## II Discrétisation spatiale

Afin d'éviter notamment l'apparition de gradient de pression nuls qui n'ont rien de physiques, les équations associées aux schémas de prédiction correction seront discrétisées spatialement par la méthode des volumes finis sur un maillage décalé<sup>4</sup>. En effet l'utilisation de ce type de maillage permet de vérifier la condition inf-sup ou LBB (Ladyzenskaia-Babushka-Brezzi).

Le maillage décalé consiste à rechercher les variables de vitesse  $u$  dans la direction  $\vec{x}$ ,  $v$  dans la direction  $\vec{y}$  et la pression  $p$  en des points différents. Ainsi l'équation de mouvement selon  $\vec{x}$  (respectivement selon  $\vec{y}$ ) sera discrétisée sur les volumes de contrôle associés à  $u$  (resp. à  $v$ ) alors que l'équation de Poisson sera discrétisée sur les volumes de contrôle associés à  $p$  (voir la figure 3.1).

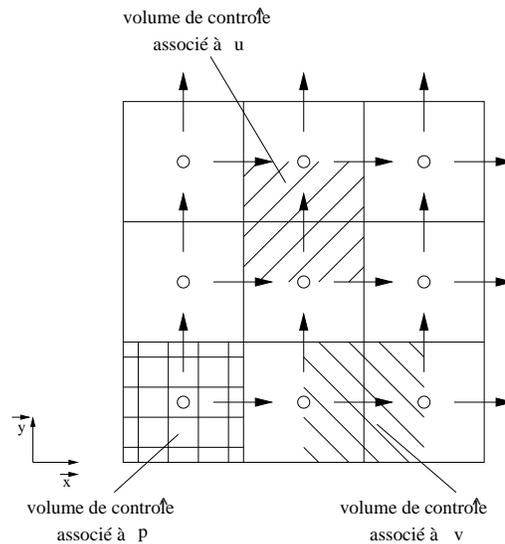


FIG. 3.1 – Localisation des variables de vitesse  $u$ ,  $v$  et de la pression  $p$  dans un maillage 2D décalé.

<sup>4</sup>en pratique, il est possible d'utiliser des grilles colocalisées mais les calculs nécessitent une remise à jour des flux, voir [Peric et al., 1988]. Il est aussi possible d'utiliser un ordre d'interpolation des variables de vitesse plus élevé que celui de la pression.

### III Algorithme PGD associé aux schémas de prédiction correction

Rappelons tout d'abord les équations de Navier-Stokes :

$$\text{Trouver } \mathbf{u}(x, y, t) \in \Omega \times ]0, T] \begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \mathbf{f} \\ \mathbf{u}(t = 0) = \mathbf{u}_0 \\ \mathbf{u} = \mathbf{g}_D \text{ on } \Gamma \end{cases} \quad (3.29)$$

où  $\mathbf{u} = (u, v)$  et  $p$  sont respectivement la vitesse et la pression,  $\mathbf{f}$  est une force connue et  $Re$  est le nombre de Reynolds.

La mise en place de la PGD est détaillée sur le schéma de correction de pression incrémental standard où le terme diffusif est discrétisé à l'aide d'un schéma de Cranck-Nicholson et le terme convectif à l'aide d'un schéma d'Adams Bashforth d'ordre deux. La discrétisation temporelle est effectuée à l'aide d'un schéma d'Euler implicite. L'algorithme présenté ici est une variante du schéma de correction de pression incrémental standard présenté précédemment, il est utilisé pour résoudre les équations de Navier-Stokes par de nombreux auteurs dont Quarteroni [Quarteroni et al., 2000] [Piller et Stalio, 2004] ou encore Faure [Faure, 2003]. Quarteroni, dans [Quarteroni et al., 2000] le nomme 'algorithme de Van-Kan'. Par la suite c'est ainsi que nous le nommerons. C'est ce schéma qui sera utilisé par la suite. L'application aux autres méthodes de prédiction-correction ne pose aucun problème. De plus, il est important de noter que la PGD est effectuée uniquement sur les variables spatiales comme directions de séparations.

L'étape de prédiction s'écrit alors :

$$\frac{\tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k}{\delta t} - \frac{1}{2Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} = \frac{1}{2Re} \nabla^2 \mathbf{u}^k - \nabla p^k + B(\mathbf{u}^k, \mathbf{u}^k - 1), \quad \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \quad (3.30)$$

où

$$B(\mathbf{u}^k, \mathbf{u}^k - 1) = -\frac{3}{2}(\mathbf{u}^k \cdot \nabla) \mathbf{u}^k + \frac{1}{2}(\mathbf{u}^{k-1} \cdot \nabla) \mathbf{u}^{k-1} \quad (3.31)$$

Le système d'équations (3.30) peut s'écrire sous la forme tensorielle suivante :

$$\sum_{j=1}^3 (A_x^j \otimes A_y^j) \tilde{u}^{k+1} = \sum_{j=1}^{n_f} f_x^j \otimes f_y^j \quad \sum_{j=1}^3 (B_x^j \otimes B_y^j) \tilde{v}^{k+1} = \sum_{j=1}^{n_h} h_x^j \otimes h_y^j \quad (3.32)$$

En formulation PGD, la vitesse intermédiaire  $\tilde{\mathbf{u}}^{k+1} = (\tilde{u}^{k+1}, \tilde{v}^{k+1})$  est cherchée sous forme séparée :

$$\tilde{u}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{u}}^{k+1}} \alpha_{\tilde{u}}^{i, k+1} F_{\tilde{u}}^{i, k+1}(x) G_{\tilde{u}}^{i, k+1}(y) \quad \tilde{v}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{v}}^{k+1}} \alpha_{\tilde{v}}^{i, k+1} F_{\tilde{v}}^{i, k+1}(x) G_{\tilde{v}}^{i, k+1}(y) \quad (3.33)$$

Cela revient donc à résoudre deux équations de diffusion instationnaire dont les opérateurs PGD  $\{A_x^j, A_y^j, B_x^j, B_y^j\}_{j=1,2,3}$  ont été définis dans le chapitre 2 section III.1.

La pression intermédiaire est solution de l'équation de Poisson suivante :

$$\begin{cases} \nabla^2 \tilde{p}^{k+1} = \frac{2}{\delta t} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \\ \nabla \tilde{p}^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0, \end{cases} \quad (3.34)$$

Ce problème s'écrit sous forme tensorielle :

$$\sum_{j=1}^2 (C_x^j \otimes C_y^j) \tilde{p}^{k+1} = \sum_{j=1}^{n_l} l_x^j \otimes l_y^j \quad (3.35)$$

En formulation PGD, la pression de correction est recherchée sous la forme séparée suivante :

$$\tilde{p}^{k+1}(x, y) = \sum_{i=1}^{n_{\tilde{p}}^{k+1}} \alpha_{\tilde{p}}^{i,k+1} F_{\tilde{p}}^{i,k+1}(x) G_{\tilde{p}}^{i,k+1}(y) \quad (3.36)$$

Les opérateurs PGD  $\{C_x^j, C_y^j\}_{j=1,2}$  associés à l'équation de Poisson ont été définis dans le chapitre 2 section I.

Enfin les champs de vitesse  $\mathbf{u}^{k+1} = (u^{k+1}, v^{k+1})$  et la pression  $p^{k+1}$  sous forme séparées sont déduits par :

$$\begin{cases} \mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \frac{\delta t}{2} \nabla \tilde{p} \\ p^{k+1} = p^k + \tilde{p} \end{cases} \quad (3.37)$$

et peuvent donc s'écrire sous la forme :

$$\begin{aligned} u^{k+1}(x, y) &= \sum_{i=1}^{n_u^{k+1}} \alpha_u^{i,k+1} F_u^{i,k+1}(x) G_u^{i,k+1}(y) & v^{k+1}(x, y) &= \sum_{i=1}^{n_v^{k+1}} \alpha_v^{i,k+1} F_v^{i,k+1}(x) G_v^{i,k+1}(y) \\ p^{k+1}(x, y) &= \sum_{i=1}^{n_p^{k+1}} \alpha_p^{i,k+1} F_p^{i,k+1}(x) G_p^{i,k+1}(y) \end{aligned} \quad (3.38)$$

A partir de la forme séparée des champs de vitesse et de pression il est aisé de définir les seconds membres  $\{f_x^j, f_y^j\}_{j=1, \dots, n_f}$ ,  $\{h_x^j, h_y^j\}_{j=1, \dots, n_g}$ ,  $\{l_x^j, l_y^j\}_{j=1, \dots, n_h}$  intervenant dans les équations (3.32) et (3.35). A ce stade, il suffit de préciser les opérateurs et les seconds membres intervenant dans les équations (3.32) et (3.35) pour pouvoir résoudre le problème.

## IV Applications de la PGD

### IV .1 Problème de "Taylor-Green"

#### a) Définition du problème

Dans un premier temps, la PGD sera appliquée à la résolution des équations de Navier-Stokes (3.3) dans un cas particulier où une solution analytique est connue, à savoir le problème de "Taylor-Green". Le terme source est supposé nul. La solution analytique du problème est telle qu'à tout instant  $t$ , le champs de vitesse et la pression s'écrivent :

$$\begin{cases} u(x, y, t) = -\cos(2\pi x) \sin(2\pi y) e^{-8\pi^2 \nu t} \\ v(x, y, t) = \sin(2\pi x) \cos(2\pi y) e^{-8\pi^2 \nu t} \\ p(x, y, t) = -\frac{1}{4}(\cos(4\pi x) + \cos(4\pi y)) e^{-16\pi^2 \nu t} \end{cases} \quad (3.39)$$

Les conditions aux limites en vitesse sont prises naturellement égales à la valeur de la solution analytique. La figure 3.2 définit la géométrie et les conditions aux limites utilisée pour ce problème.

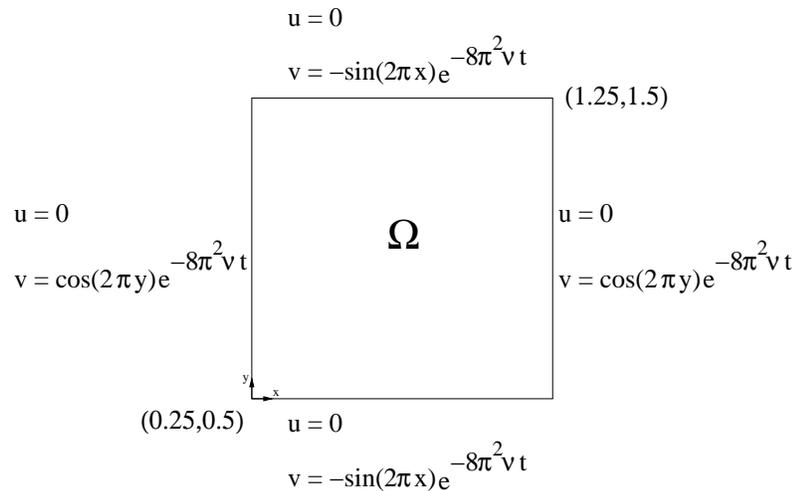


FIG. 3.2 – Geometrie pour le problème de "Taylor-Green".

Le problème de Taylor-Green sera donc résolu à l'aide de la méthode PGD et à l'aide d'un solveur standard (bi-gradient conjugué). Les simulations sont effectuées avec  $\nu = 10^{-2}$ ,  $\rho = 1$  et un pas de temps  $\delta t = 10^{-3}$  sur une durée de simulation  $T=1s$ .

## b) Résultats

La figure 3.3 représente les vecteurs vitesse calculés par méthodes PGD (à gauche), pour un nombre de nœuds dans chaque direction  $N_h = 200$  ainsi que la différence en norme entre le champ de vitesse calculé par PGD et la solution analytique après un temps de simulation  $T=1s$ . Cette figure montre que les vitesses obtenues par PGD sont très proches de la solution analytique (différence de l'ordre de  $10^{-5}$ ). La figure 3.4 quant à elle représente le champ de pression calculé par PGD et la différence entre ce dernier et le champ de pression analytique après un temps de simulation  $T=1s$ . On constate là aussi que les résultats sont très proches (différence de l'ordre de  $10^{-4}$ ).

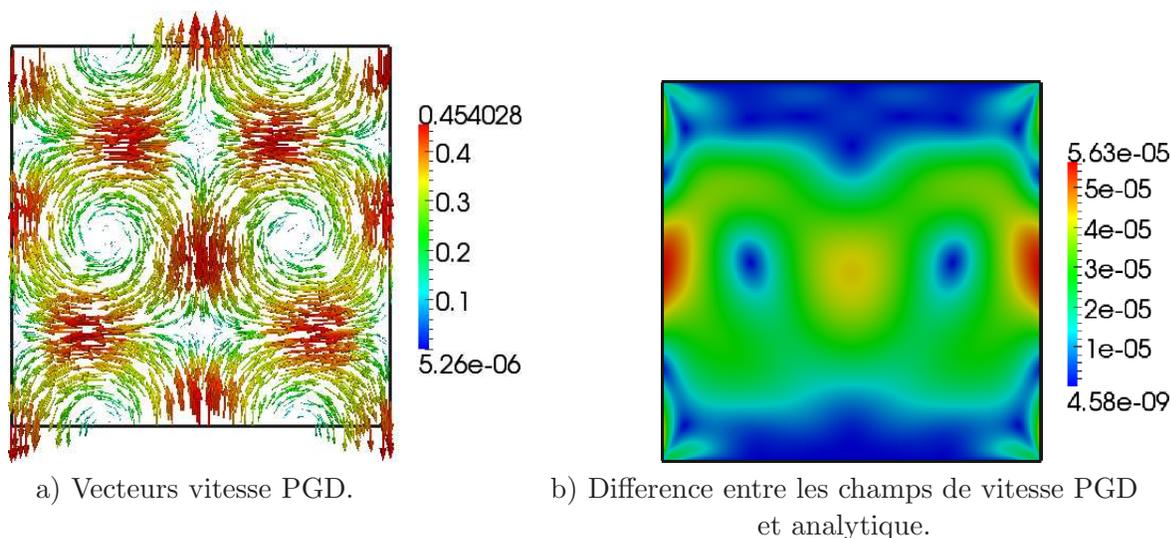


FIG. 3.3 – Vecteurs vitesse calculés par PGD et différence entre les champs de vitesse calculés par PGD et la solution analytique à  $T=1s$  avec  $N_h = 200$ .

La figure 3.5 représente l'évolution des erreurs en vitesse (à gauche) et en pression (à droite) par rapport aux solutions analytiques en fonction du nombre de nœuds de discrétisation. Ces erreurs ont été définies dans le chapitre 3 sections IV et V pour les équations de Burgers et de Stokes. On constate que les erreurs en vitesse et en pression pour la PGD et la méthode standard sont très proches.

La figure 3.6 représente les temps de calcul pour la résolution du problème de Taylor-Green en utilisant la méthode PGD et le solveur standard en fonction du nombre de nœuds dans chaque direction pour un temps de simulation  $T=1s$ . On constate qu'à partir d'un certain nombre de nœuds ( $N_h = 75$ ), la méthode PGD devient plus rapide que le solveur standard. En fait plus on augmente le nombre de nœuds dans chaque direction, plus la méthode PGD devient intéressante par rapport au solveur classique. En effet pour un maillage  $100 \times 100$  la PGD est deux fois plus rapide que le solveur standard, alors que pour un maillage  $500 \times 500$  la PGD est dix huit fois plus rapide que le solveur standard.

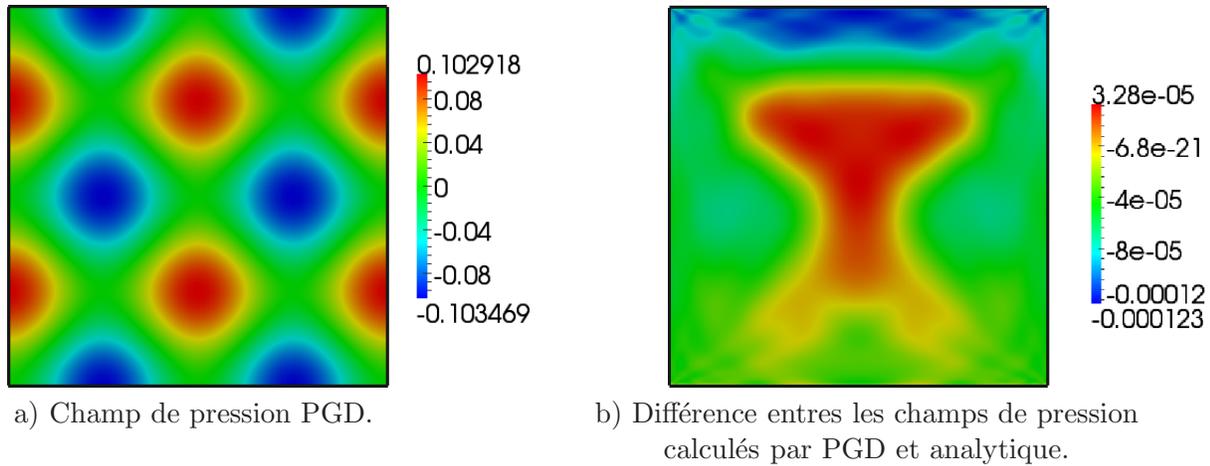


FIG. 3.4 – Pression calculée par PGD et différence entre la pression calculée par PGD et la pression analytique à  $T=1s$  avec  $N_h = 200$ .

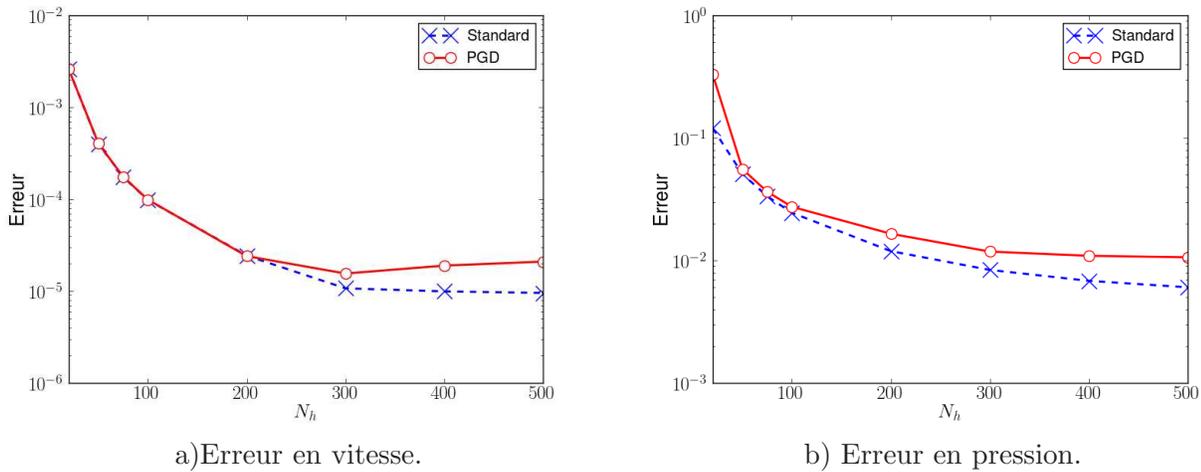


FIG. 3.5 – Évolution des erreurs en vitesse et en pression en fonction du nombre de nœuds pour la méthode PGD et le solveur standard.

Enfin, on observe que lors de la résolution PGD, le champ de vitesse est obtenu en exactement un enrichissement pour chaque composante. Ceci est en parfait accord avec la solution analytique du champ de vitesse qui se décompose en exactement un produit de fonction pour chacune des composantes. Les fonctions  $F(x)$  et  $G(y)$  calculées pour la vitesse par PGD sont comparées avec les fonctions issues des solutions analytiques sur la figure 3.7. On constate effectivement que les courbes sont superposées et qu'en conséquence, la résolution PGD de la vitesse est optimale pour le problème de "Taylor-Green".

La résolution du problème de Taylor-Green par méthode PGD a donc été réalisée de manière efficace en terme de précision et de temps de calcul.

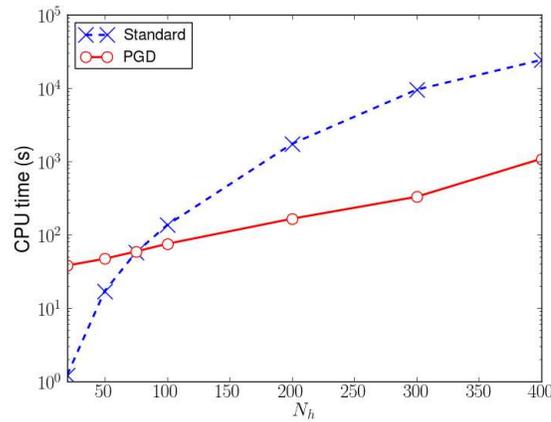
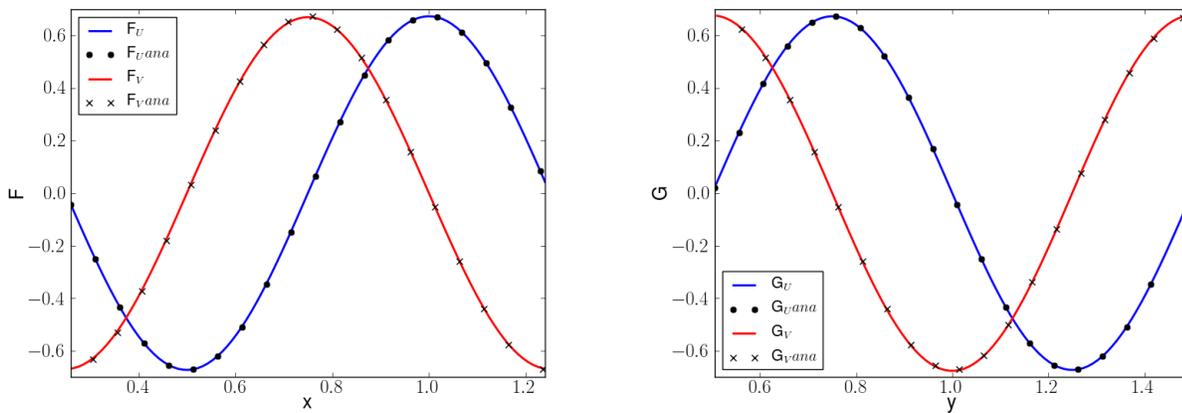


FIG. 3.6 – Comparaison du temps de calcul entre la méthode PGD et le solveur standard.

FIG. 3.7 – Fonctions de la décomposition issues de la PGD et de la solution analytique des champs  $u$  and  $v$  après un temps de simulation  $T=1s$ .

## IV .2 Ecoulement dans la cavité entraînée 2D

Les équations de Navier-Stokes vont maintenant être résolues pour le cas test de la cavité entraînée 2D isotherme. Cette configuration a déjà été étudiée par de nombreux auteurs et permet donc d'avoir d'abondants résultats de référence. La cavité entraînée 2D consiste à étudier un écoulement dans un domaine carré de taille  $d$  où la vitesse sur les bords est nulle sauf sur le bord supérieur où l'on impose une vitesse longitudinale égale à  $U_0$ . Le cas d'étude est représenté par la figure 3.8. Sur cette figure les deux lignes en pointillé correspondent aux endroits où les résultats obtenus par les simulations sont comparés à ceux issus de la littérature.

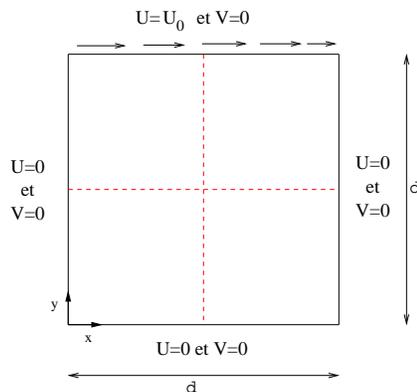


FIG. 3.8 – Géométrie de la cavité entraînée 2D.

Les calculs ont été menés avec un même nombre de nœuds dans chaque direction noté  $N_h$  et pour différents nombres de Reynolds  $Re = \frac{U_0 \times d}{\nu}$ , à savoir  $Re = 100$ ,  $Re = 1000$  et  $Re = 10000$ . Les deux premiers cas convergent vers des solutions stationnaires, alors que le dernier cas  $Re = 10000$  converge vers une solution instationnaire mais périodique. En effet, il est bien connu qu'à partir d'un nombre de Reynolds critique ( $\approx 7700$ ) correspondant à la première bifurcation de Hopf, la solution de la cavité entraînée 2D ne converge plus vers une solution stationnaire. Pour  $Re = 10000$  de nombreux auteurs [Sengupta et al., 2009, Bruneau et Saad, 2006] ont montré que la résolution des équations de Navier-Stokes convergeait vers une solution périodique.

#### a) Cas stationnaire ( $Re=100$ , $Re = 1000$ )

Les simulations ont été effectuées avec un fluide initialement au repos à l'intérieur de la cavité, le pas de temps a été pris égal à  $\delta t = 10^{-3}$  et le critère de convergence suivant a été défini :

$$\frac{\|u_i^k - u_i^{k-1}\|}{\|u_i^k\|} \leq \epsilon_u \quad \text{avec } i = 1, 2 ; \quad \frac{\|p^k - p^{k-1}\|}{\|p^k\|} \leq \epsilon_p \quad (3.40)$$

La solution est estimée convergée, lorsque les deux critères cités ci-dessus sont vérifiés. Dans les simulations qui suivent, les coefficients ont été pris égaux à  $\epsilon_u = 5.10^{-8}$  et  $\epsilon_p = 10^{-4}$ . Les simulations ont été effectuées avec la méthode PGD ainsi qu'avec un solveur classique (bi-gradient conjugué).

Sur la figure 3.9 sont représentées les lignes de courant calculées avec la méthode PGD et avec le solveur standard pour chacun des nombres de Reynolds étudiés pour un maillage de taille  $250 \times 250$ . Pour un nombre de Reynolds égal à 100 on observe la formation de petits vortex secondaires en bas à gauche et en bas à droite de la cavité ainsi que d'un vortex principal au centre de la cavité. Lorsque le nombre de Reynolds est égal à 1000, la taille des vortex secondaires augmente tandis que le vortex principal se recentre au milieu de la cavité. De plus, on constate que les lignes de courants calculées par PGD sont très proches de celles obtenues avec le modèle standard.

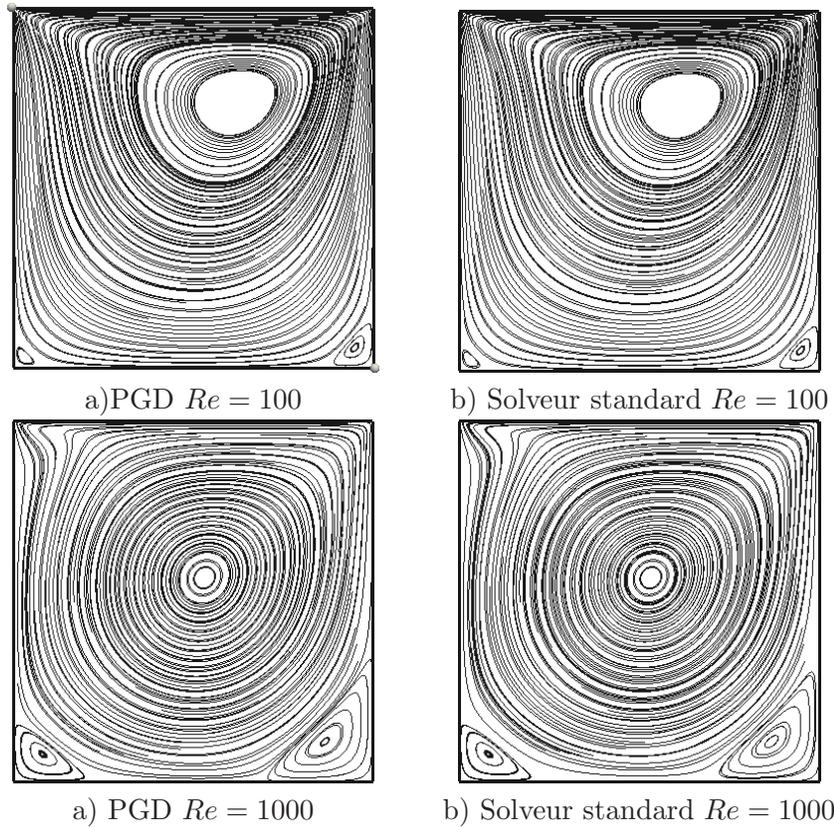


FIG. 3.9 – Lignes de courant calculées par PGD et avec le solveur standard avec  $N_h = 250$  et pour  $Re = 100$  et  $Re = 1000$ .

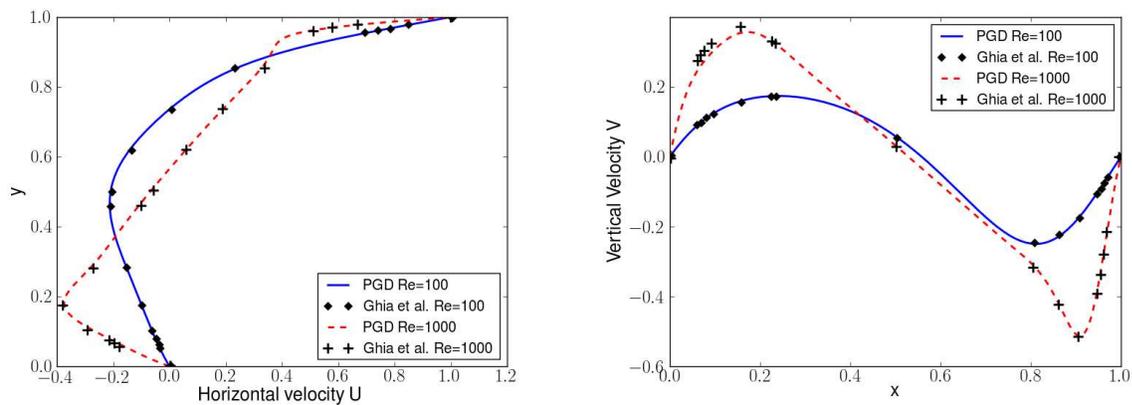


FIG. 3.10 – Comparaison des vitesses horizontales à  $x=0.5$  et des vitesses verticales à  $y=0.5$  avec les résultats de Ghia. and al. pour  $Re = 100$  et  $Re = 1000$  pour  $N_h = 250$ .

Les vitesses horizontale en  $x = 0.5$  et verticale en  $y = 0.5$  (correspondant aux lignes pointillées de la figure 3.8) ont également été comparées à celles issues de la littérature (Ghia et al. [Ghia et al., 1982]) sur la figure 3.10. On constate que les vitesses obtenues sur ces profils correspondent exactement aux résultats issus de la littérature.

Les champs de pression obtenus par PGD et avec le solveur standard sont représentés sur la figure 3.11 pour  $Re = 100$  et pour  $Re = 1000$ . On constate une bonne correspondance entre le calcul standard et le calcul PGD. Pour s'en convaincre, la différence entre ces champs de pression pour chacun des deux nombres de Reynolds étudiés ont été représentés sur la figure 3.12. On constate que l'erreur est de l'ordre de  $10^{-3}$  pour chacun des deux nombres de Reynolds.

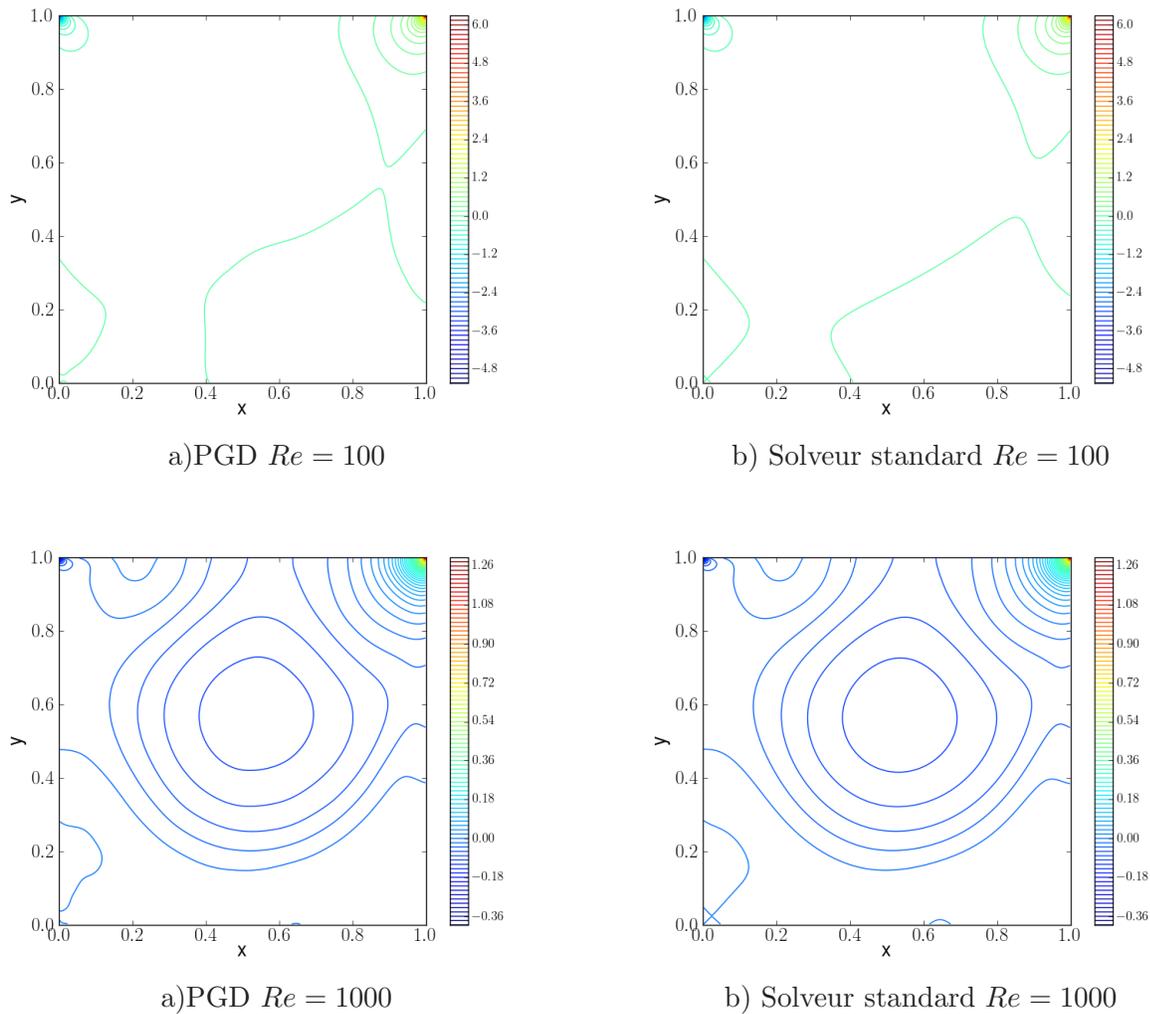


FIG. 3.11 – Pression calculée par PGD et par le solveur standard avec  $N_h = 250$  pour  $Re = 100$  et  $Re = 1000$ .

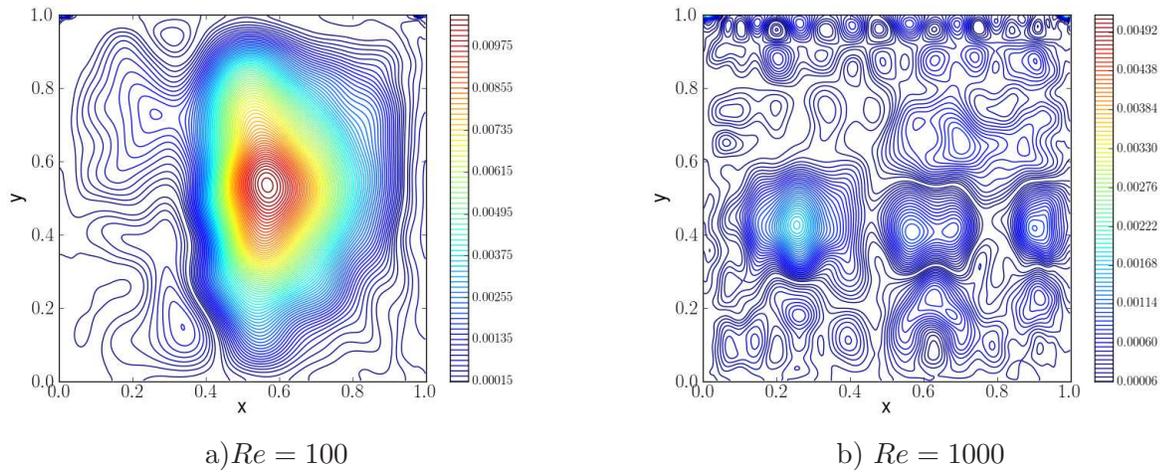


FIG. 3.12 – Différence de pression calculée par PGD et par le solveur standard avec  $N_h = 250$  pour  $Re = 100$  et  $Re = 1000$ .

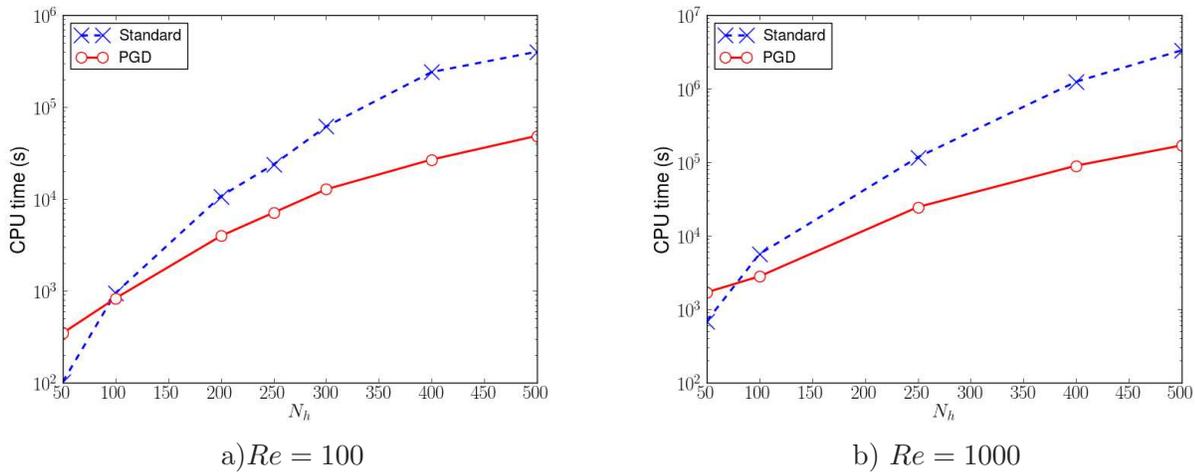
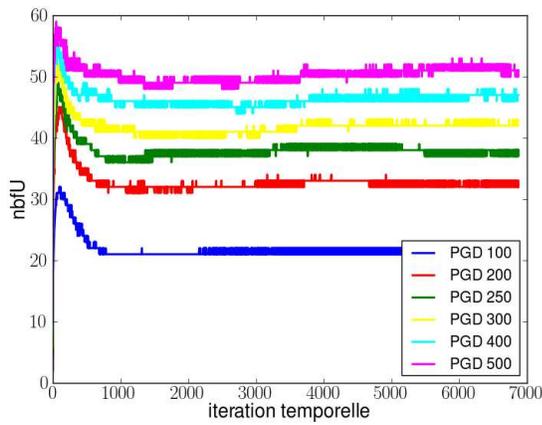


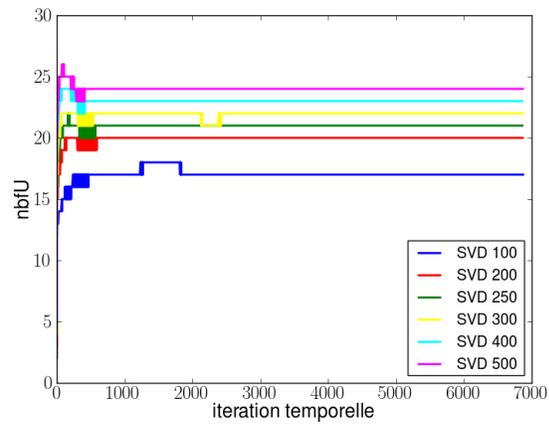
FIG. 3.13 – Comparaison des temps de calcul entre la méthode PGD et la méthode standard pour la résolution de la cavité entraînée à l'aide de schémas de projection pour  $Re = 100$  et  $Re = 1000$ .

La figure 3.13 montre qu'à partir de 100 nœuds dans chaque direction, la PGD devient avantageuse en terme de temps de simulation comparée au modèle standard. Plus précisément pour un maillage de taille  $250 \times 250$ , la PGD est trois fois plus rapide que le modèle standard pour un nombre de Reynolds égal à 100 et cinq fois plus rapide pour un nombre de Reynolds égal à 1000. De la même manière pour un maillage de taille  $500 \times 500$ , le temps de calcul est huit fois plus faible pour  $Re = 100$  et vingt fois plus faible pour  $Re = 1000$ .

Nous allons maintenant nous intéresser à l'optimalité de la méthode PGD. Pour ce faire une SVD (Singular Value Decomposition) a été effectuée à chaque itération de la méthode à pas fractionnaire sur les champs de vitesses et de pression issus de la résolution du solveur standard. On notera 'PGD  $N_h$ ' (respectivement 'SVD  $N_h$ ') le nombre de fonctions issu de la résolution par PGD (resp par SVD sur les résultats du solveur standard) pour un nombre de nœuds égal à  $N_h$  dans chaque direction.

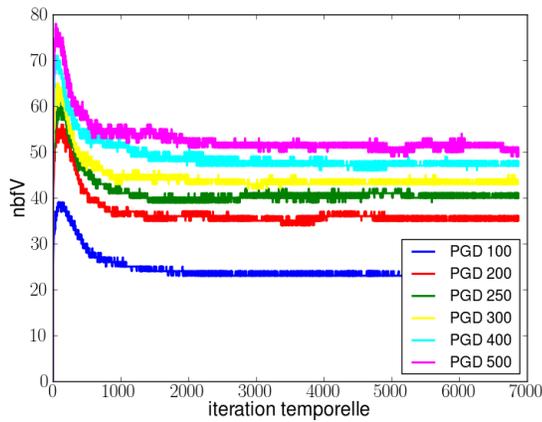


a) PGD  $Re = 100$

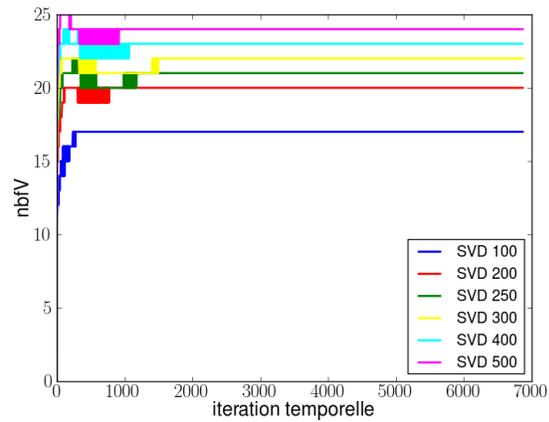


b) SVD  $Re = 100$

FIG. 3.14 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la vitesse horizontale à  $Re = 100$ .



a) PGD  $Re = 100$



b) SVD  $Re = 100$

FIG. 3.15 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la vitesse verticale à  $Re = 100$ .

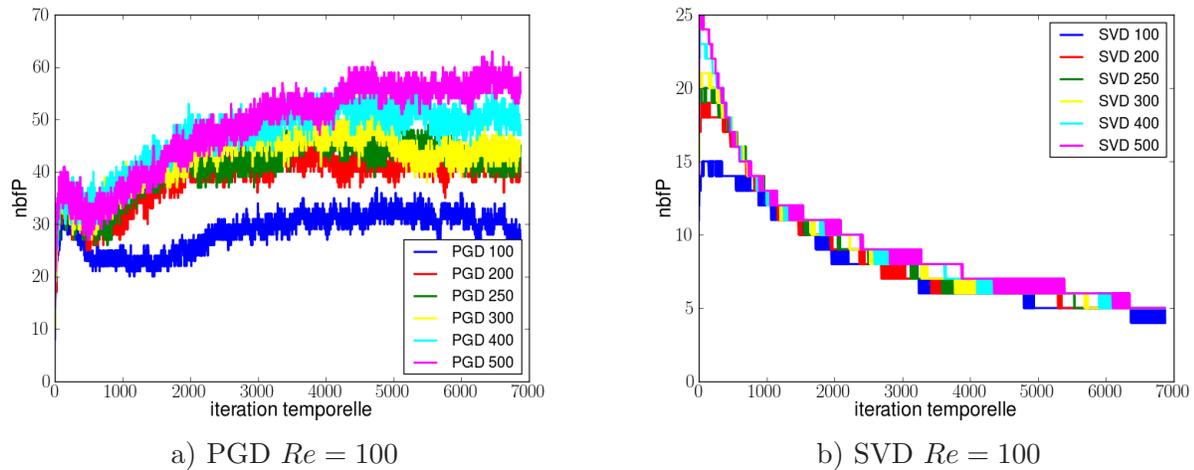


FIG. 3.16 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la pression à  $Re = 100$ .

La figure 3.14 (respectivement 3.15, 3.16) montre le nombre de fonctions SVD et le nombre de fonctions PGD pour la vitesse horizontale (respectivement la vitesse verticale et la pression) à  $Re = 100$  pour différents maillages. On constate tout d’abord que plus la taille du maillage est importante plus le nombre de fonctions requis par le calcul PGD est important. En effet, à convergence, pour le calcul de la vitesse horizontale, le nombre de fonctions PGD nécessaires à la résolution du problème est de l’ordre de la vingtaine pour un maillage  $100 \times 100$  et de l’ordre de la cinquantaine pour un maillage de taille  $500 \times 500$ . Ces figures indiquent également que le nombre de fonctions obtenues par PGD pour le calcul des vitesses augmente lors de la phase transitoire (approximativement les mille premières itérations), alors que dans le cas de la SVD, ce nombre de fonction reste stable. Concernant le nombre de fonctions PGD obtenues pour le calcul de la pression, ce nombre de fonctions est fortement oscillant, au contraire de la SVD des résultats du solveur standard où le nombre de fonction nécessaire est beaucoup plus faible que pour la PGD. A convergence, pour la vitesse horizontale et pour un maillage  $100 \times 100$ , le nombre de fonctions PGD est de l’ordre de la vingtaine contre seize pour la SVD du modèle standard. De même pour un maillage de taille  $500 \times 500$  le nombre de fonctions PGD est de l’ordre de la cinquantaine contre vingt-cinq pour la SVD du modèle standard. La PGD n’est donc pas tout à fait optimale. Cependant, le gain de temps réalisé pour de gros maillages montre que, même si la représentation séparée n’est pas optimale, il est moins coûteux d’effectuer un calcul PGD que de résoudre le problème standard.

De même, la figure 3.17 (respectivement 3.18, 3.19) montre le nombre de fonctions PGD et SVD pour la vitesse horizontale (respectivement la vitesse verticale et la pression) à  $Re = 1000$  pour différents maillages. Les conclusions faites pour  $Re = 100$  restent valables pour  $Re = 1000$ .

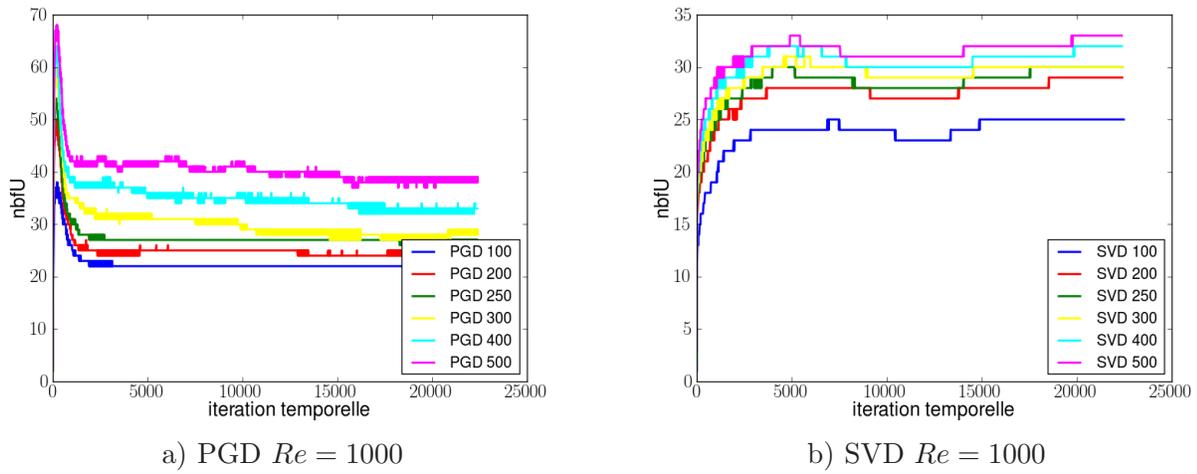


FIG. 3.17 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la vitesse horizontale à  $Re = 1000$ .

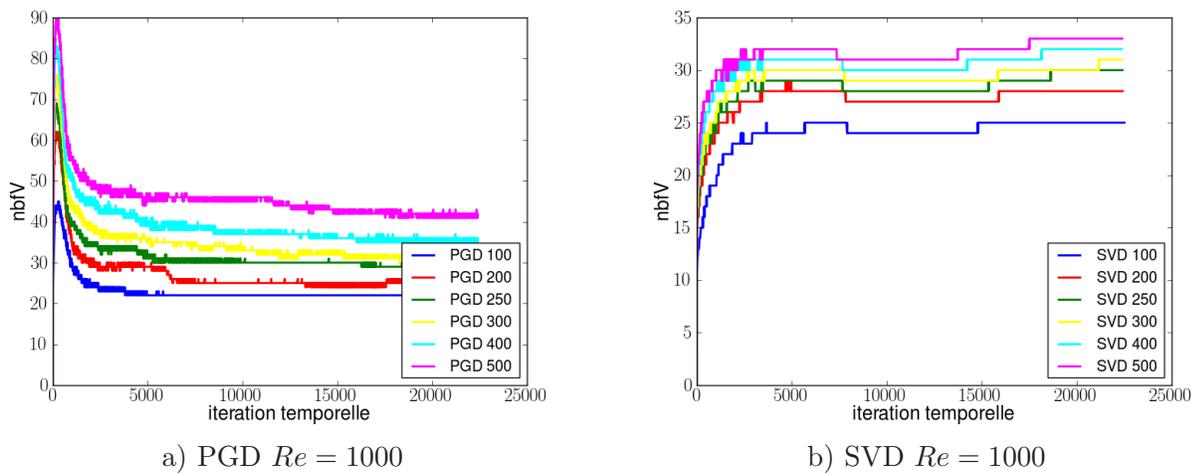


FIG. 3.18 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la vitesse verticale à  $Re = 1000$ .

Nous allons maintenant nous intéresser à l'évolution de la norme du résidu de chacune des

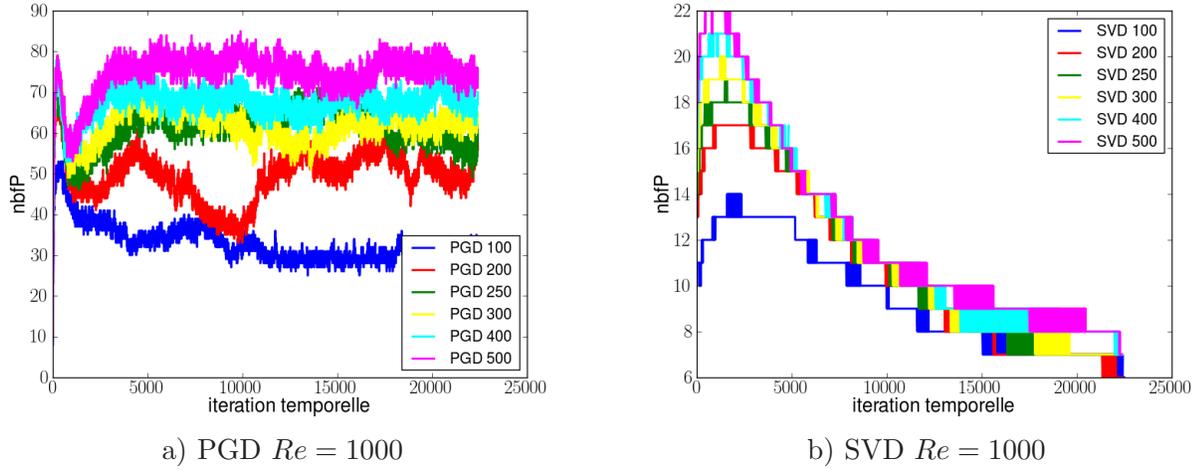


FIG. 3.19 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la pression à  $Re = 1000$ .

équations résolues en fonction du nombre de fonctions PGD ou SVD gardées. Le résidu est défini par :

$$Res_X^n = \mathcal{A}_X \left[ \sum_{i=1}^{n_X} F_X^i(x) G_X^i(y) \right] - SM_X(x, y) \quad (3.41)$$

où  $X$  correspond aux différentes inconnues calculées dans le problème, à savoir les vitesses intermédiaires horizontale et verticale  $\tilde{u}, \tilde{v}$  et la pression de correction  $\tilde{p}$ .  $\mathcal{A}_X$  (respectivement  $SM_X$ ) est l'opérateur (resp. le second membre) associé à l'équation permettant de calculer la variable  $X$ . Les fonctions  $F_X^i, G_X^i$  peuvent être les fonctions PGD de l'inconnue  $X$  à convergence, les fonctions issues d'une SVD sur la solution finale PGD de la variable  $X$  ou encore les fonctions issues d'une SVD sur la solution du modèle standard.

La norme du résidu associé à l'équation vérifiée par  $\tilde{u}$  (resp.  $\tilde{v}$  et  $\tilde{p}$ ) en fonction du nombre de fonctions gardées est tracée sur la figure 3.20 (respectivement 3.21 et 3.22) pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 250$ ). Les figures 3.20 et 3.21 permettent de mettre en évidence la différence de comportement de la PGD entre le cas où  $Re = 100$  et le cas où  $Re = 1000$ . En effet, à  $Re = 100$  on observe que la courbe représentant la norme du résidu pour les fonctions PGD est identique aux courbes représentant la norme des résidus de chacune des SVD jusqu'à une valeur de  $10^{-4}$ . A ce moment, la courbe PGD décroît plus lentement jusqu'à une norme du résidu inférieure à  $10^{-6}$  avec une trentaine de fonctions alors que les SVD ne nécessitent qu'une vingtaine de fonctions pour atteindre la même norme du résidu. On en déduit donc que pour  $Re = 100$ , la résolution PGD des vitesse intermédiaires est optimale jusqu'à une norme du résidu égale à  $10^{-4}$  mais qu'à partir de cette valeur la PGD perd en optimalité. Concernant le cas où  $Re = 1000$ , on observe que la norme du résidu tracée à partir des fonctions PGD est identique à celles tracées à partir des fonctions SVD. Dans ce cas on peut dire que la PGD est optimale en terme de décomposition jusqu'à une norme du résidu égale à  $10^{-6}$ . Les normes du résidu pour la pression de correction à  $Re = 100$  et  $Re = 1000$  sont tracées sur la figure 3.22.

Dans ce cas, la PGD n'est pas du tout optimale. En effet, il suffit de cinq fonctions issues de la SVD sur les résultats du modèle standard pour obtenir une norme du résidu inférieure à  $10^{-2}$  alors que pour obtenir un résidu équivalent, la PGD nécessite quarante-cinq fonctions, et la SVD du résultat PGD nécessite une trentaine de fonction.

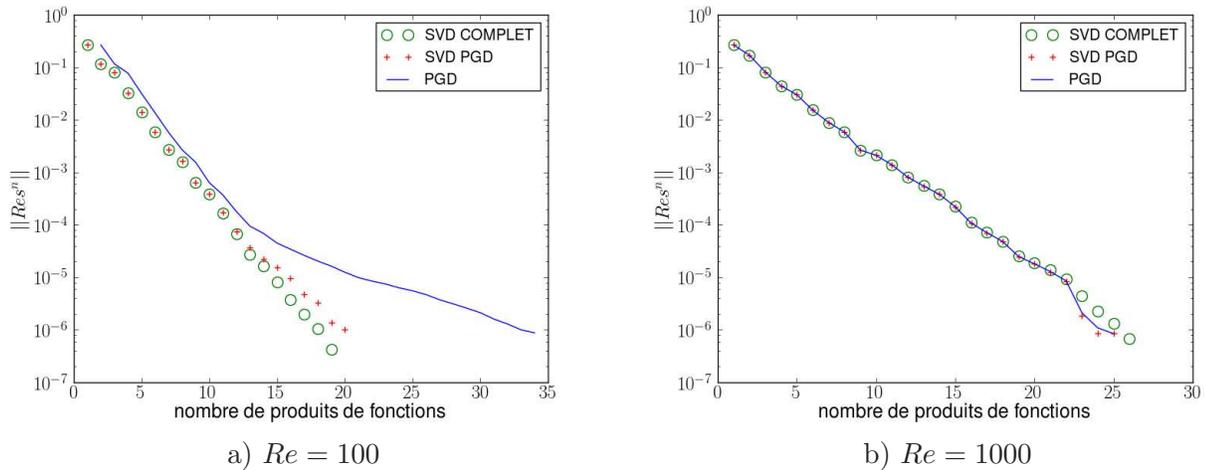


FIG. 3.20 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD, SVD sur le résultat PGD ( SVD PGD) et SVD sur le résultat du modèle standard (SVD COMPLET) pour la vitesse intermédiaire horizontale  $\tilde{u}$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 250$ ).

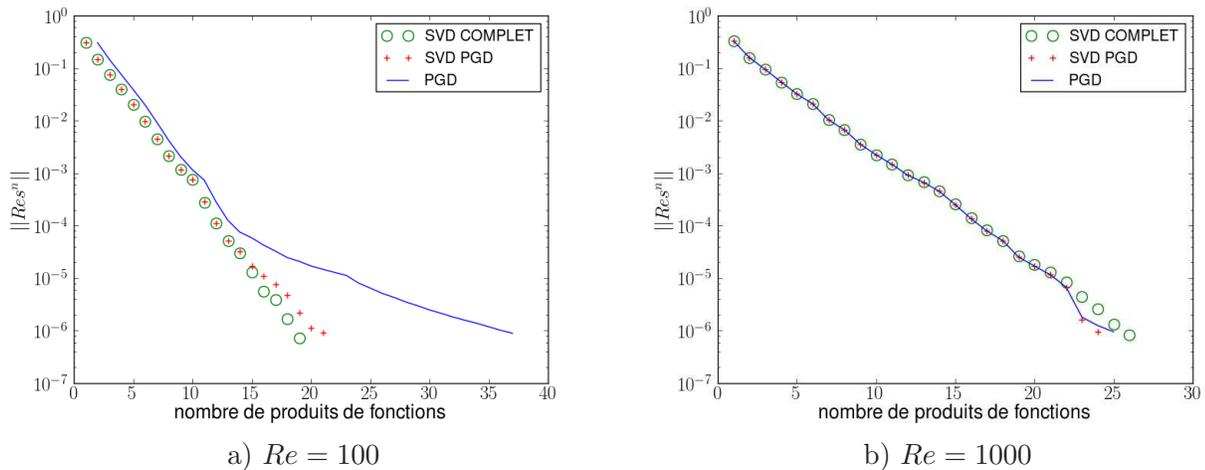


FIG. 3.21 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la vitesse intermédiaire verticale  $\tilde{v}$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 250$ ).

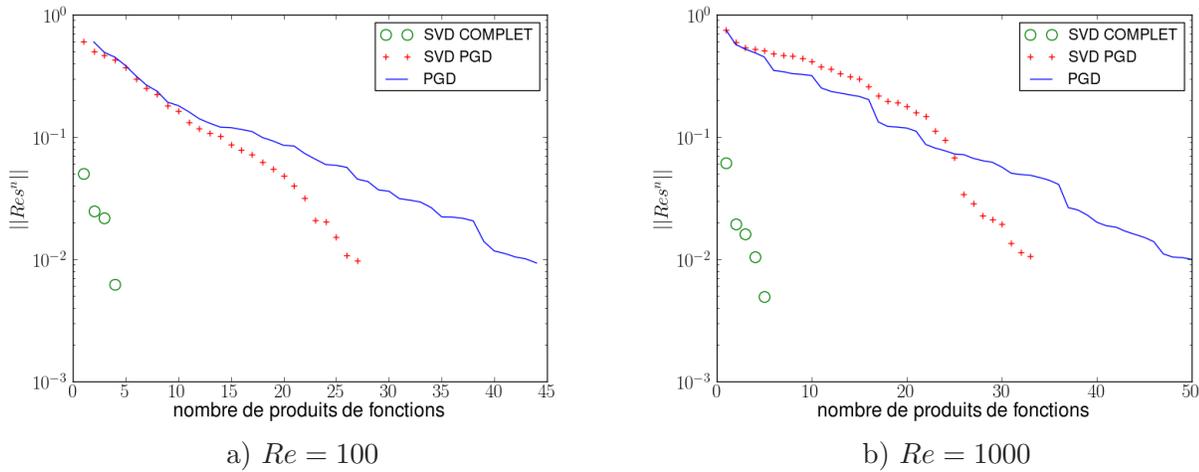


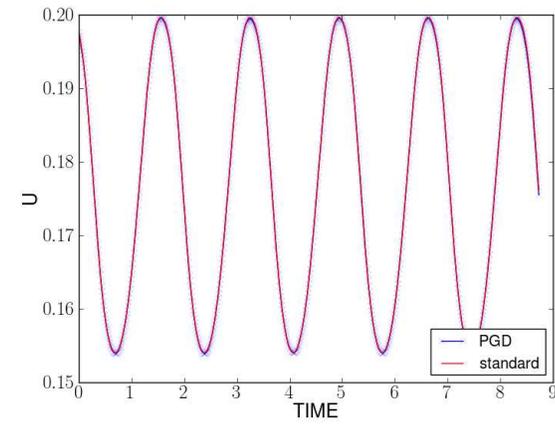
FIG. 3.22 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la pression de correction  $\tilde{p}$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 250$ ).

### b) Cas périodique ( $Re = 10000$ )

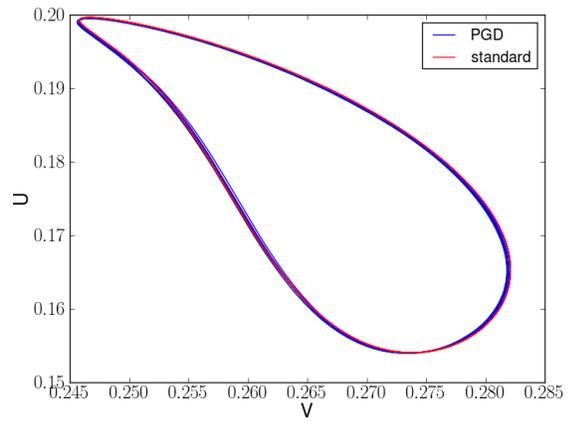
Les simulations ont été effectuées avec un pas de temps  $\delta t = 10^{-3}$  et un maillage de taille  $250 \times 250$  sur une durée de cinq périodes, de plus la taille de l'arrête de la cavité a été prise égale à  $d = 1$ .

L'évolution de la vitesse horizontale située au point de coordonnées  $(x, y) = (2/16, 13/16)$  (coin supérieur gauche de la cavité) est comparée à celle obtenue par le modèle standard sur la figure 3.23-(a). On constate que l'écoulement a bien un comportement périodique de période  $T = 1.69s$ , ce qui est proche de la période de  $1.64s$  obtenue par Bruneau et al. [Bruneau et Saad, 2006]. La similarité des résultats PGD et de ceux obtenus par méthode standard est confirmée par les figures 3.23-(c,e) sur lesquelles ont été tracées l'évolution de la vitesse horizontale en des points situées en bas à gauche de la cavité  $((x, y) = (2/16, 2/16))$  et en haut à droite de la cavité  $((x, y) = (0.95, 0.95))$ . De plus les portraits de phase en chacun de ces points ont été tracés sur les figures 3.23-(b,d,f). Le fait que les courbes soient toutes fermées et parcourent le même chemin au cours des cinq périodes confirme bien le caractère périodique de l'écoulement aux points de contrôle.

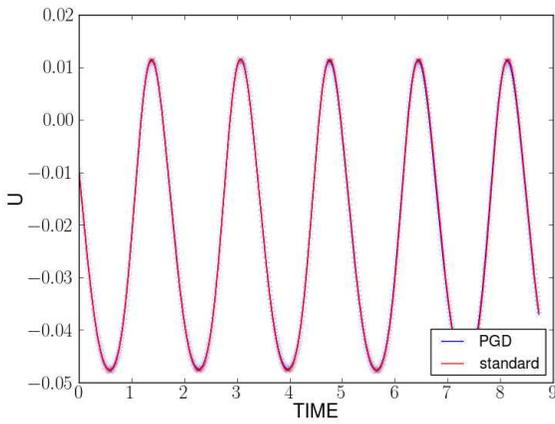
Les lignes de courant sur une période obtenues par PGD et par le solveur classique sont tracées sur les figures 3.24 et 3.25. Elles sont conformes à celles de la littérature [Bruneau et Saad, 2006]. On observe que dans le coin en bas à gauche le vortex secondaire se sépare en deux petits vortex qui se réincorporent de façon périodique. De la même manière le vortex secondaire en haut à gauche à un comportement similaire. Une bonne correspondance entre les résultats obtenus par PGD et ceux obtenus par le solveur standard est observée avec un temps de calcul PGD divisé par six par rapport au solveur standard.



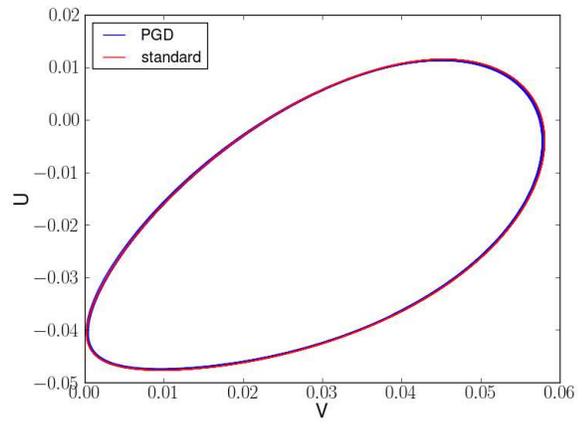
a) Vitesse horizontale en  $(2/16, 13/16)$



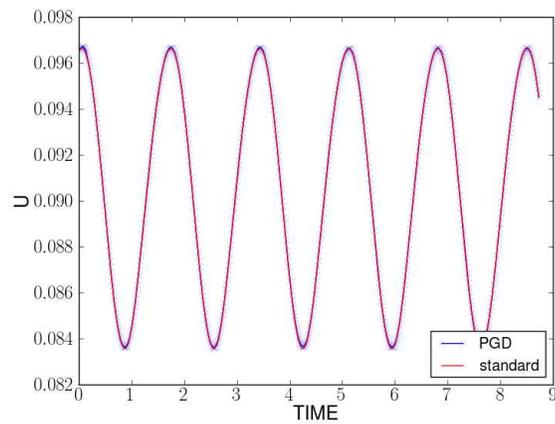
b) Portrait de phase en  $(2/16, 13/16)$



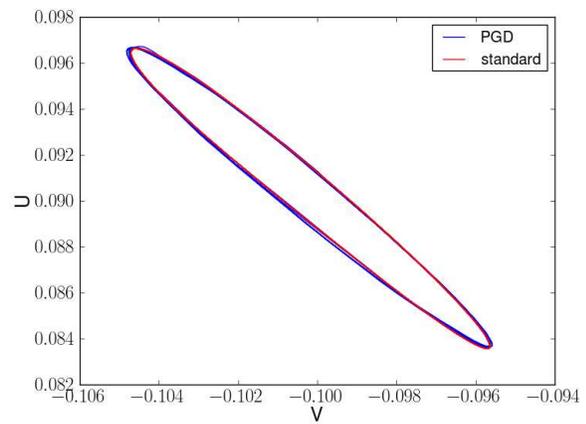
c) Vitesse horizontale en  $(2/16, 2/16)$



d) Portrait de phase en  $(2/16, 2/16)$



e) Vitesse horizontale en  $(0.95, 0.95)$



f) Portrait de phase en  $(0.95, 0.95)$

FIG. 3.23 – Vitesse horizontale et portrait de phase aux points de référence pour  $Re = 10000$ .

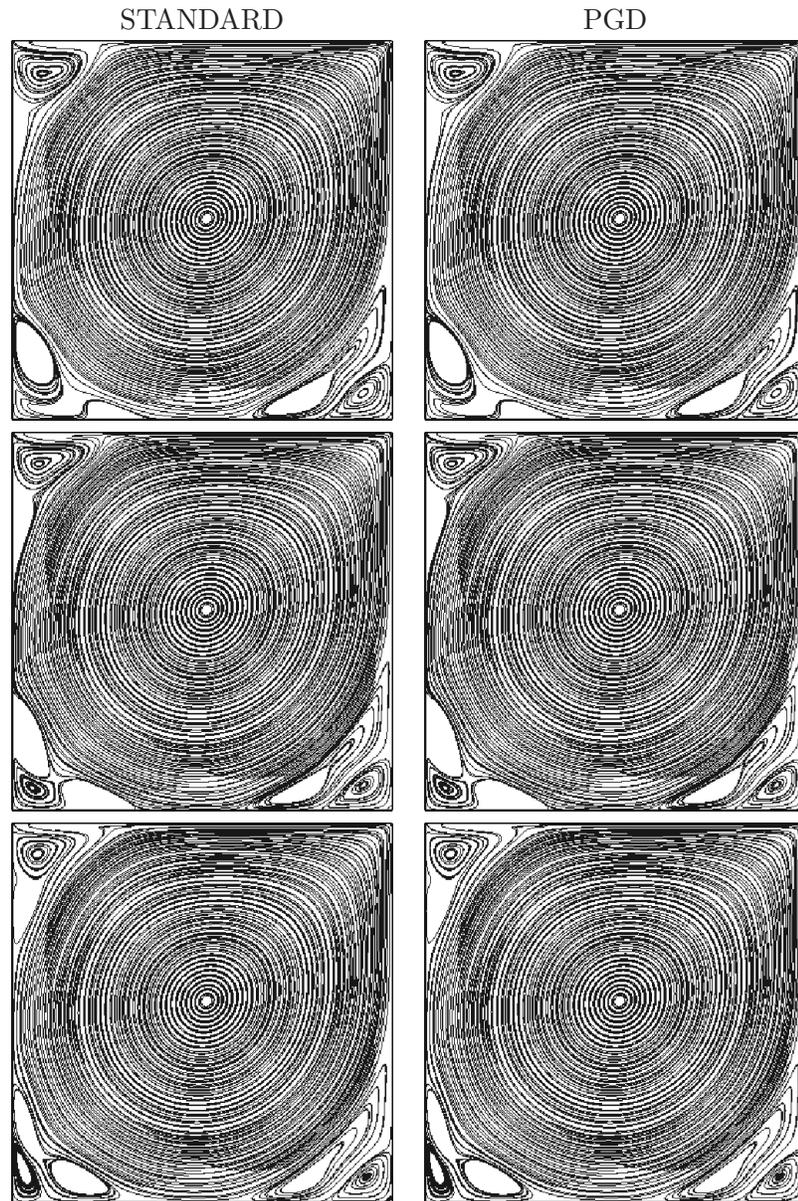


FIG. 3.24 – Evolution des lignes de courant durant une période pour  $Re = 10000$  avec un maillage  $250 \times 250$ . De haut en bas pour chacun des solveurs (standard et PGD), les temps  $t = 0$ ,  $t = 0.2T$  et  $t = 0.4T$  sont représentés ( $T$  est la période égale à  $1.69s$ ).

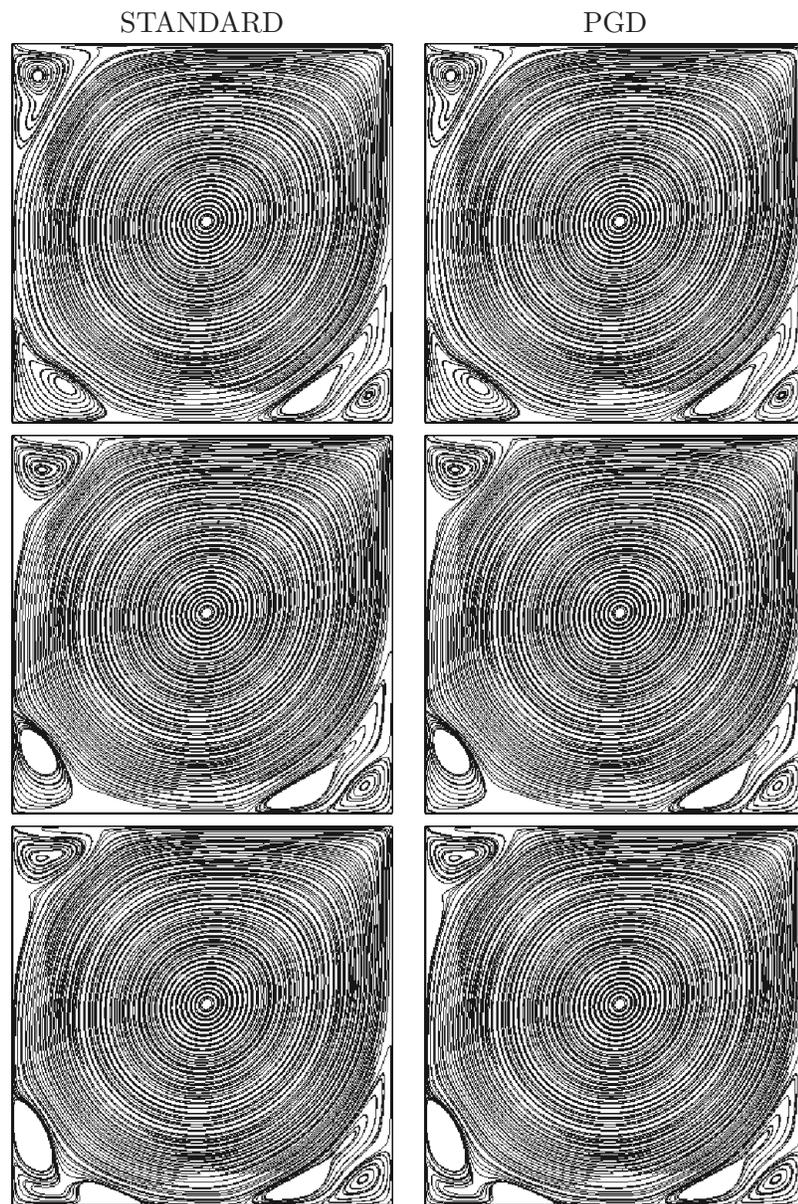


FIG. 3.25 – Evolution des lignes de courant durant une période pour  $Re = 10000$  avec un maillage  $250 \times 250$ . De haut en bas pour chacun des solveurs (standard et PGD), les temps  $t = 0.6T$ ,  $t = 0.8T$  et  $t = T$  sont représentés ( $T$  est la période égale à  $1.69s$ ).

Le nombre de fonctions à chaque itération obtenu par PGD et par SVD sur la solution du modèle standard pour la vitesse horizontale, la vitesse verticale et la pression sont tracés sur les figures (3.26) et (3.27).

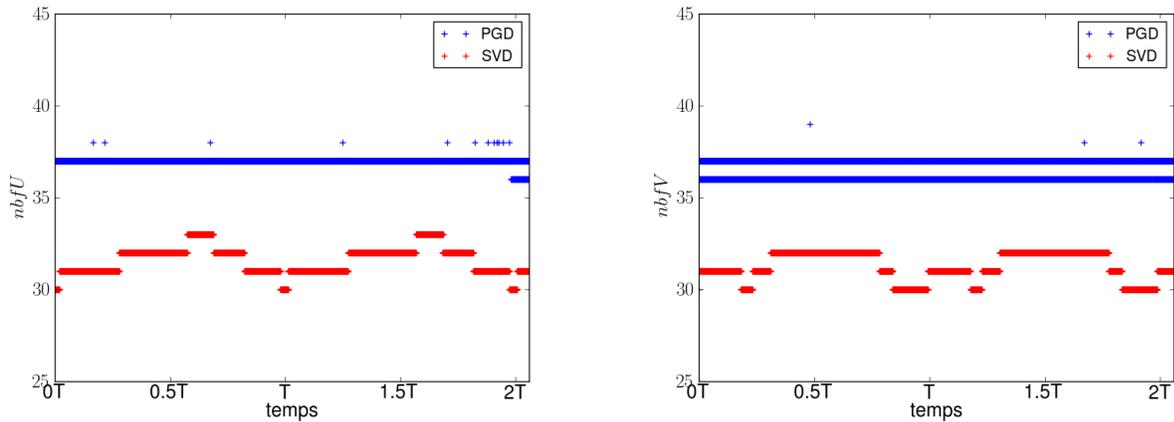


FIG. 3.26 – Nombre de fonctions obtenues par PGD et par SVD de la solution de la méthode standard pour la vitesse horizontale (gauche) et verticale (droite) à  $Re = 10000$  ( $N_h = 250$ ).

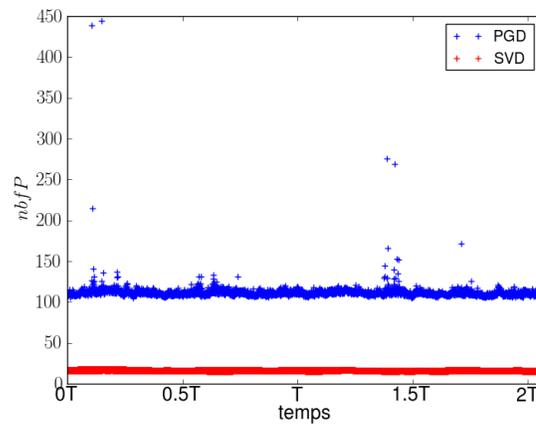


FIG. 3.27 – Nombre de fonctions obtenues par PGD et par SVD de la solution de la méthode standard pour la pression à  $Re = 10000$  ( $N_h = 250$ ).

Concernant le champ de vitesse, la figure 3.26 montre que le nombre de fonctions obtenu par PGD est similaire à celui obtenu par SVD. Dans ce cas la PGD a un comportement quasiment optimal. Par contre, pour le champ de pression, la figure 3.27 indique que le nombre de

fonction obtenu par PGD est cinq fois supérieur à celui obtenu par SVD (30 fonctions SVD contre 150 fonctions PGD). Cette observation avait déjà été faite dans les cas précédents pour des nombres de Reynolds égaux à 100 et 1000. Toujours pour le calcul de la pression, il existe des itérations où la PGD nécessite un très grand nombre de fonctions (de l'ordre de 450). Même si ce cas de figure est extrêmement rare il n'y a pour le moment aucune explication à ce phénomène. Cependant, même si le calcul PGD n'est pas optimal pour le calcul de la pression, on observe quand même un gain de temps par rapport au calcul standard (la PGD est six fois plus rapide que le solveur standard pour une période).

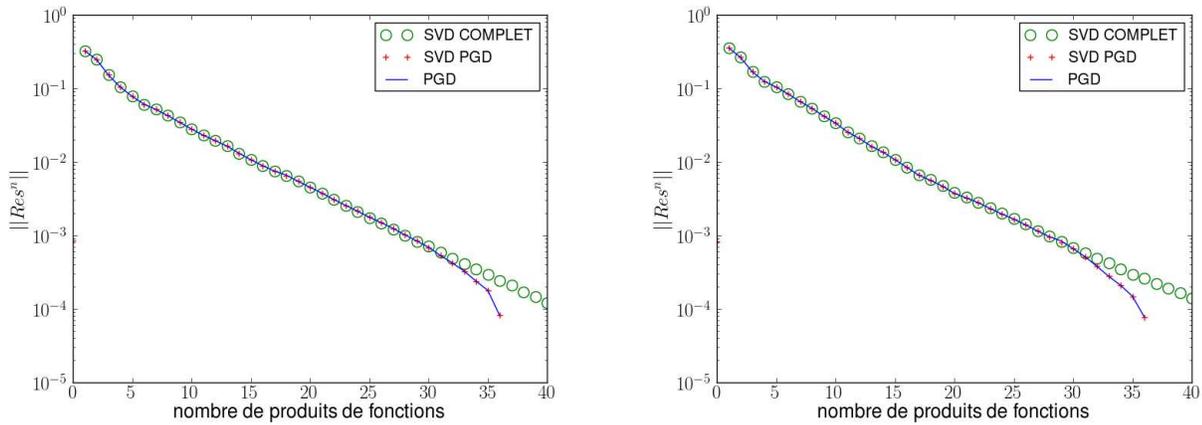


FIG. 3.28 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour les vitesses intermédiaires  $\tilde{u}$  (gauche) et  $\tilde{v}$  (droite) à  $Re = 10000$  ( $N_h = 250$ ).

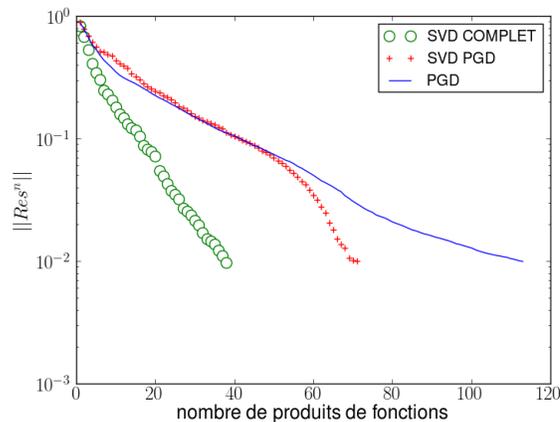


FIG. 3.29 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la pression de correction  $\tilde{p}$  à  $Re = 10000$  ( $N_h = 250$ ).

Comme dans le cas stationnaire, l'évolution de la norme du résidu (définie par l'équation (3.41)) avec le nombre de fonctions PGD et le nombre de fonctions issues de deux SVD (une sur les résultats PGD et l'autre sur les résultats du modèle standard) est tracée sur les figures 3.28 et 3.29 ( $N_h = 250$ ). Ici, la norme des résidus a été tracée au début d'une période de l'écoulement ( $T = 0$ ). La figure 3.28 montre une très bonne correspondance entre la décroissance de la norme du résidu calculé à partir des fonctions PGD et des deux SVD. On en déduit que pour la résolution des vitesse intermédiaires la PGD est optimale. En revanche, la figure 3.29 montre que le calcul de la pression n'est pas optimal. En effet il faut trois fois plus de fonctions PGD que de fonctions issues de la SVD sur les résultats du modèle standard pour atteindre une norme du résidu inférieure à  $10^{-2}$ .

### IV .3 Ecoulement dans la cavité entraînée 3D

#### a) Problème étudié

La cavité entraînée 3D est constituée d'un cube d'arrête  $d$ , sur laquelle on impose une vitesse d'entraînement égale à  $U_0$  sur la face supérieure. C'est en fait une extension du cas traité en 2 dimensions. La figure 3.30 représente la configuration étudiée où les lignes tracées en rouge correspondent aux endroits où seront visualisés les résultats.

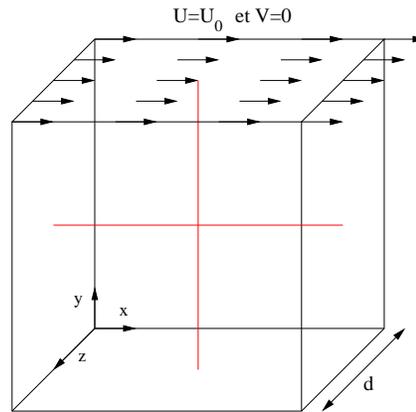


FIG. 3.30 – Geometrie de la cavité entraînée 3D.

Les simulations ont été effectuées pour  $Re = 100$  et  $Re = 1000$  avec un fluide initialement au repos à l'intérieur de la cavité. Le pas de temps a été pris égal à  $\delta t = 10^{-3}$  et le critère de convergence suivant a été défini :

$$\frac{\|w_i^k - w_i^{k-1}\|}{\|w_i^k\|} \leq \epsilon_u \quad \text{avec } i = 1, 2, 3 ; \quad \frac{\|p^k - p^{k-1}\|}{\|p^k\|} \leq \epsilon_p \quad (3.42)$$

La solution est supposée convergée, lorsque les deux critères cités ci-dessus sont vérifiés. Dans les simulations qui suivent, les coefficients ont été pris égaux à  $\epsilon_u = 10^{-8}$  et  $\epsilon_p = 10^{-3}$  et

le nombre de nœuds dans les trois directions est le même et est noté  $N_h$ . Les simulations ont été effectuées avec la méthode PGD ainsi qu'avec un solveur classique (bi-gradient conjugué).

### b) Résultats pour $Re = 100$ et $Re = 1000$

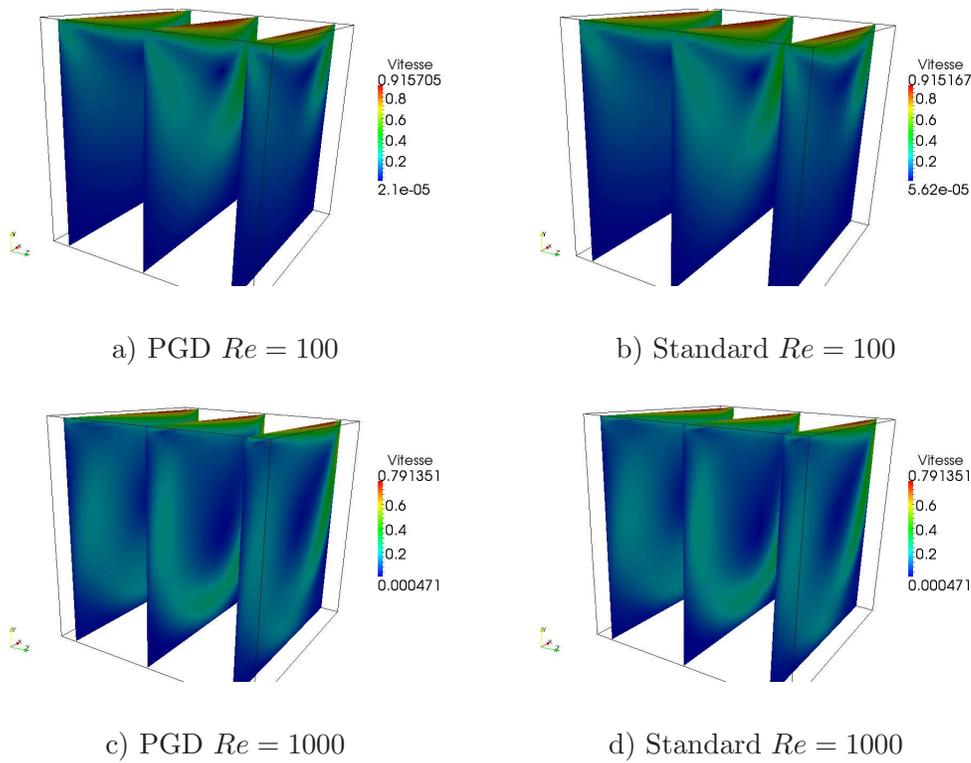


FIG. 3.31 – Champs de vitesse dans les plans  $z = 0.1$ ,  $z = 0.5$  et  $z = 0.9$  obtenus par PGD et par solveur standard pour  $Re = 100$  et  $Re = 1000$ .

La figure 3.31 représente les champs de vitesse dans les plans  $z = 0.1$ ,  $z = 0.5$  et  $z = 0.9$  obtenus par PGD et par solveur standard pour  $Re = 100$  et  $Re = 1000$ . On observe sur cette figure que la PGD permet de trouver les résultats du solveur standard. En effet on observe bien la symétrie de l'écoulement par rapport au plan  $z = 0.5$ , ainsi que la circulation du fluide dans la cavité du fait de la vitesse positive imposée sur la face supérieure. De plus on constate que l'écoulement s'intensifie avec l'augmentation du nombre de Reynolds. En effet pour  $Re = 100$  la circulation du fluide est cantonnée dans la partie supérieure de la cavité alors que pour un nombre de Reynolds égal à 1000, la partie inférieure de la cavité semble atteinte par la circulation du fluide.

Les isocontours de la vitesse dans la direction  $z$  obtenus par PGD et avec le solveur standard pour des nombres de Reynolds égaux à 100 et 1000 sont représentés sur la figure 3.32. Là encore, la symétrie par rapport au plan  $z = 0.5$  est clairement présente. En effet, il apparaît que le fluide entraîné par la face supérieure de la cavité, est soumis à une vitesse de faible intensité dans la direction  $z$  en provenance des bords de la cavité jusqu'au centre

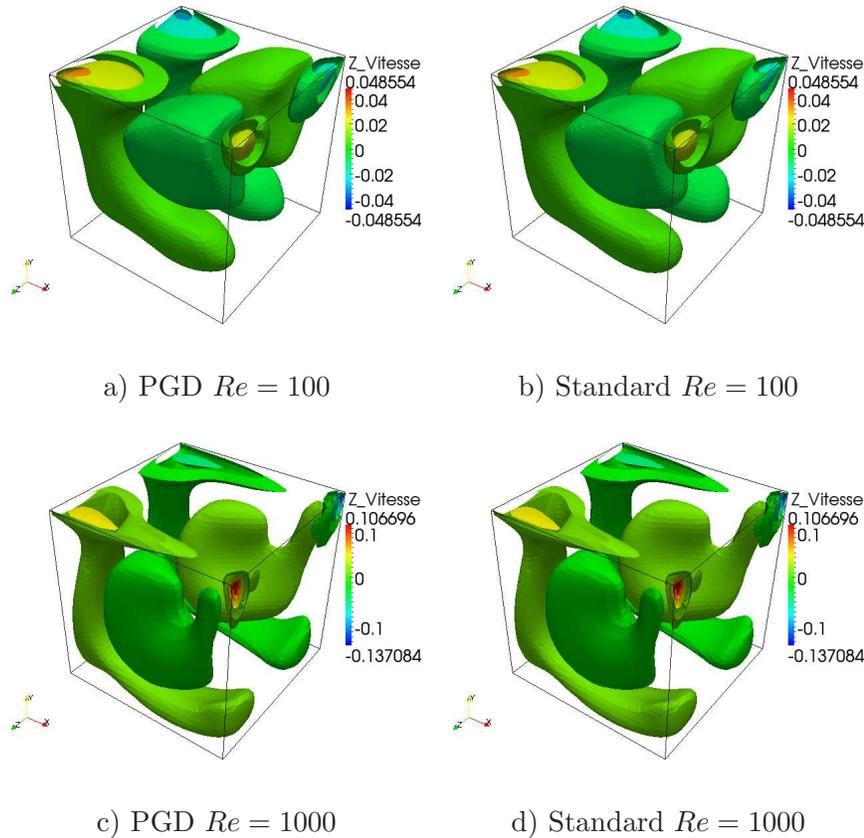


FIG. 3.32 – Isocontours de vitesse  $u_3$  obtenues par PGD et par solveur standard pour  $Re = 100$  et  $Re = 1000$ .

de celle-ci. De cette manière on observe quatre principales zones de recirculation qui viennent s'annuler sur le 'plan de symétrie'. Ces zones de recirculation ont tendance à s'étirer lorsque le nombre de Reynolds augmente mais reste malgré tout dans les mêmes zones de la cavité.

La figure 3.33 montre une bonne correspondance entre les lignes de courant dans le plan  $z = 0.5$  obtenues par PGD et avec le solveur standard pour des nombre de Reynolds égaux à 100 et à 1000. Ces lignes de courants sont semblables à celles obtenues par Ding et al. dans [Ding et al., 2006] où l'on observe que le centre du vortex principal qui se situe en haut à droite à  $Re = 100$  se recentre pour  $Re = 1000$ .

Les vitesses horizontale en  $x = 0.5$  et  $z = 0.5$  et verticale en  $y = 0.5$  et  $z = 0.5$  (correspondant au lignes pointillés de la figure (3.30)) ont été comparées à celles de Ding et al. [Ding et al., 2006] sur la figure (3.34) avec un maillage de taille  $41 \times 41 \times 41$ . On constate que pour les deux nombres de Reynolds considérés, les résultats PGD sont proches de ceux de la littérature. La faible différence observée dans le cas où  $Re = 1000$  pour la vitesse verticale existe également dans le cas de la résolution par le modèle standard. Ainsi, elle est sans doute due aux schémas de discrétisations utilisés et non à la méthode PGD elle-même.

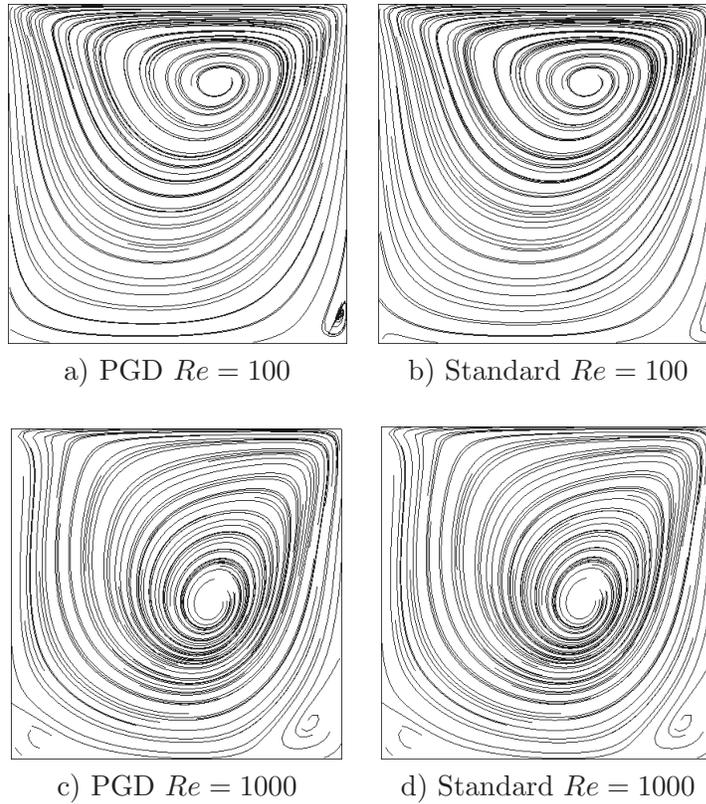


FIG. 3.33 – Lignes de courant dans le plan  $z=0.5$  obtenues par PGD et par solveur standard pour  $Re = 100$  et  $Re = 1000$ .

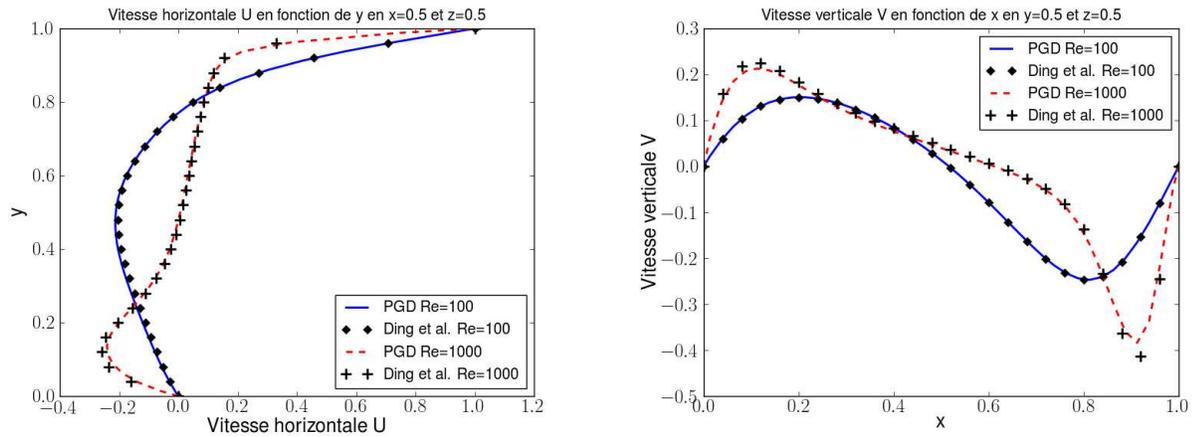


FIG. 3.34 – Comparaison de la vitesse horizontale en  $x=0.5$  et  $z=0.5$  (à gauche) et de la vitesse verticale en  $y=0.5$  et  $z=0.5$  (à droite) avec les résultats de Ding et al. pour  $Re = 100$  et  $Re = 1000$ .

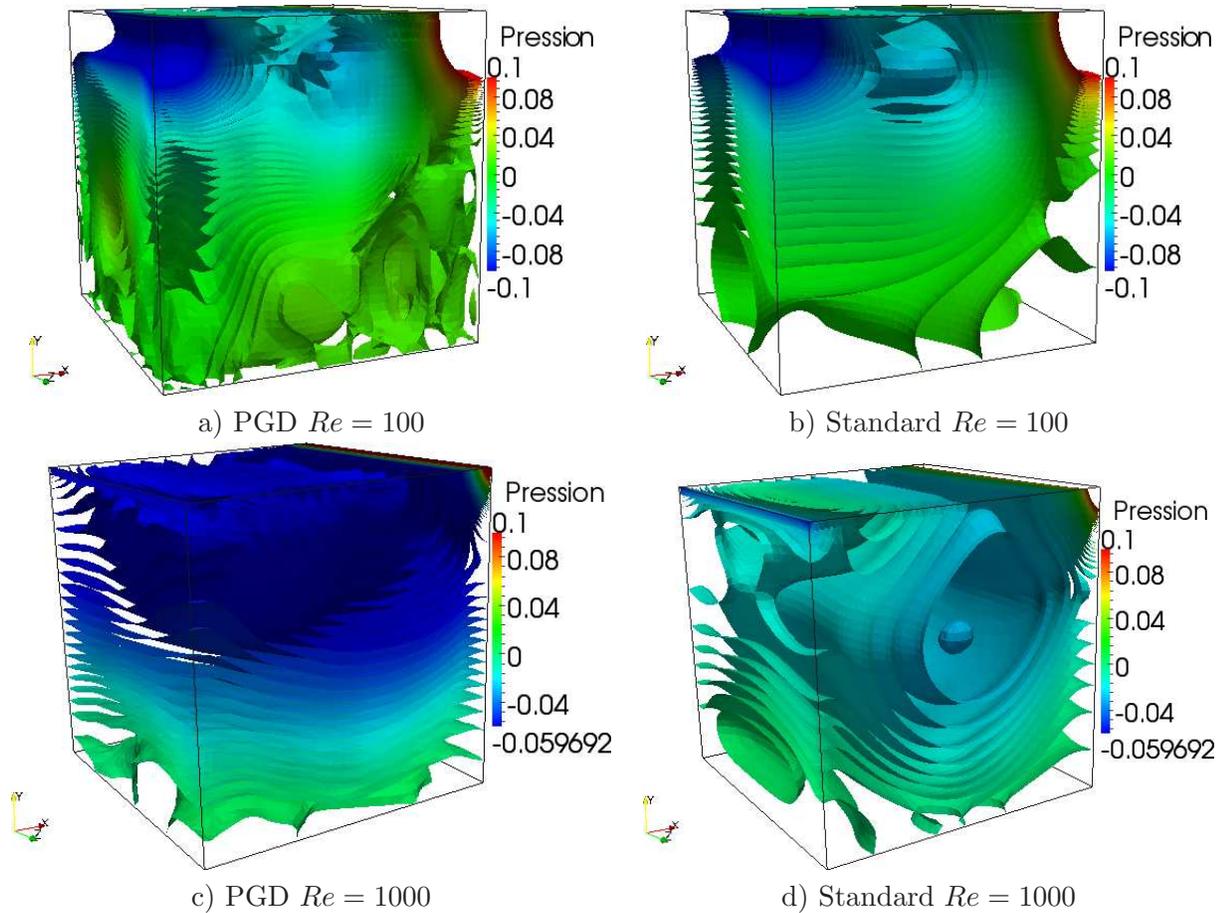


FIG. 3.35 – Isocontours de pression dans le plan  $z=0.5$  obtenues par PGD et par solveur standard pour  $Re = 100$  et  $Re = 1000$ .

Les isocontours de pression obtenus par PGD et par le solveur standard pour  $Re = 100$  et  $Re = 1000$  ont également été comparés sur la figure 3.35. Sur cette figure on constate que la PGD retrouve un bon champ de pression à  $Re = 100$ , par contre, à  $Re = 10000$ , la pression obtenue ne correspond pas à celle obtenue par le solveur standard qui correspond également à celle de la littérature. Cependant les valeurs de la pression sont quasiment nulles dans toute la cavité excepté sur l'arête située en haut à droite. C'est pourquoi, même si en apparence les résultats en pression ne sont pas corrects, le champ de vitesse associé l'est.

La figure 3.36 montre qu'à partir de 60 nœuds dans chaque direction, pour un nombre de Reynolds égal à 100, la PGD devient avantageuse en terme de temps de calcul par rapport au modèle standard. Plus précisément pour un maillage de taille  $100 \times 100 \times 100$  la PGD est deux fois plus rapide que le modèle standard. Cependant pour un nombre de Reynolds égal à 1000, on constate que la PGD n'est pas plus rapide que le modèle standard jusqu'à 100 nœuds dans chaque direction. On remarque quand même une tendance de la PGD à être plus

rapide que le modèle standard pour un nombre de nœuds supérieur à 100. Malheureusement nous n'avons pas réussi ici à atteindre ces tailles de maillages. Cela vient du fait que le code de calcul développé n'est pas totalement optimisé.

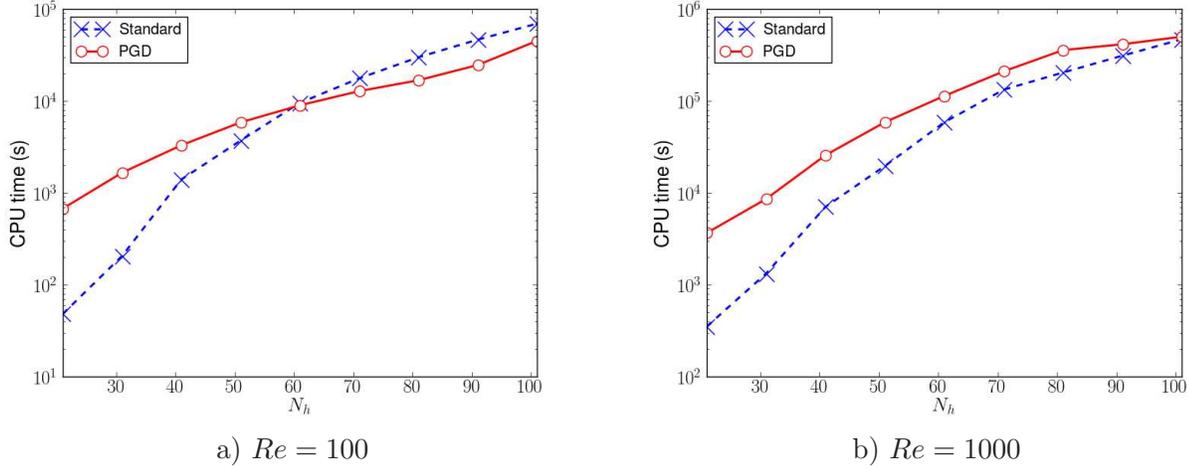


FIG. 3.36 – Comparaison des temps de calcul entre la méthode PGD et la méthode standard pour la résolution de la cavité entraînée 3D à l'aide de schémas de projection pour  $Re = 100$  et  $Re = 1000$ .

L'évolution de la norme du résidu de chacune des équations résolues en fonction du nombre de fonctions PGD ou SVD conservées est maintenant étudiée. Le résidu est défini par :

$$Res_X^n = \mathcal{A}_X \left[ \sum_{i=1}^{n_X} F_X^i(x) G_X^i(y) H_X^i(z) \right] - SM_X(x, y, z) \quad (3.43)$$

où  $X$  correspond aux différentes inconnues calculées dans le problème, à savoir les vitesses intermédiaires horizontales et verticales  $\tilde{u}, \tilde{v}, \tilde{w}$  et la pression de correction  $\tilde{p}$ .  $\mathcal{A}_X$  (respectivement  $SM_X$ ) est l'opérateur (resp. le second membre) associé à l'équation permettant de calculer la variable  $X$ . Les fonctions  $F_X^i, G_X^i$  et  $H_X^i$  peuvent être les fonctions issues du calcul PGD, les fonctions issues d'une SVD sur la solution finale PGD de la variable  $X$  ou encore les fonctions issues d'une SVD sur la solution du modèle standard.

L'évolution de la norme du résidu associé à chacune des équations vérifiées par  $\tilde{u}, \tilde{v}, \tilde{w}$  et  $\tilde{p}$  avec le nombre de fonctions PGD et issues des SVD (une sur la solution finale PGD et l'autre sur la solution du modèle standard) est tracée sur les figures 3.37, 3.38 et 3.39 pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 60$ ). Dans le cas où  $Re = 100$ , on observe que les courbes représentant la norme du résidu pour les fonctions PGD sont identiques à celles représentant la norme des résidus de chacune des SVD pour chacun des problèmes résolus. Par contre les résultats obtenus pour un nombre de Reynolds égal à 1000 sont surprenants. En effet la SVD (ainsi qu'elle a été définie dans la remarque du chapitre 2 section II.2) sur les résultats du modèle standard nécessitent beaucoup plus de fonctions que la PGD ou que la SVD sur les résultats PGD. Dans ce cas, il est clair que la SVD n'est pas optimale. Il est donc difficile de conclure quoi que ce soit dans ce cas.



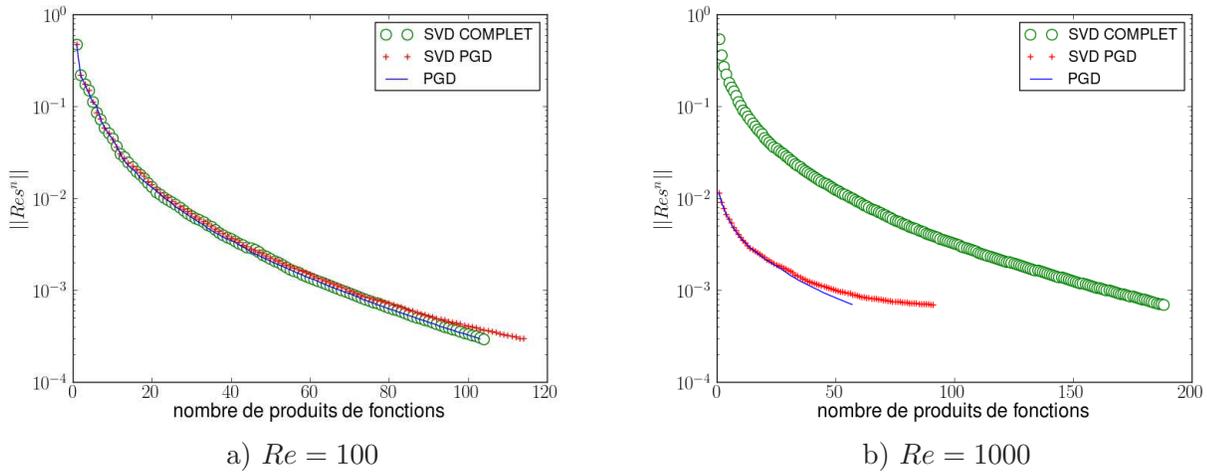


FIG. 3.39 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la vitesse intermédiaire en  $z \tilde{w}$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 60$ ).

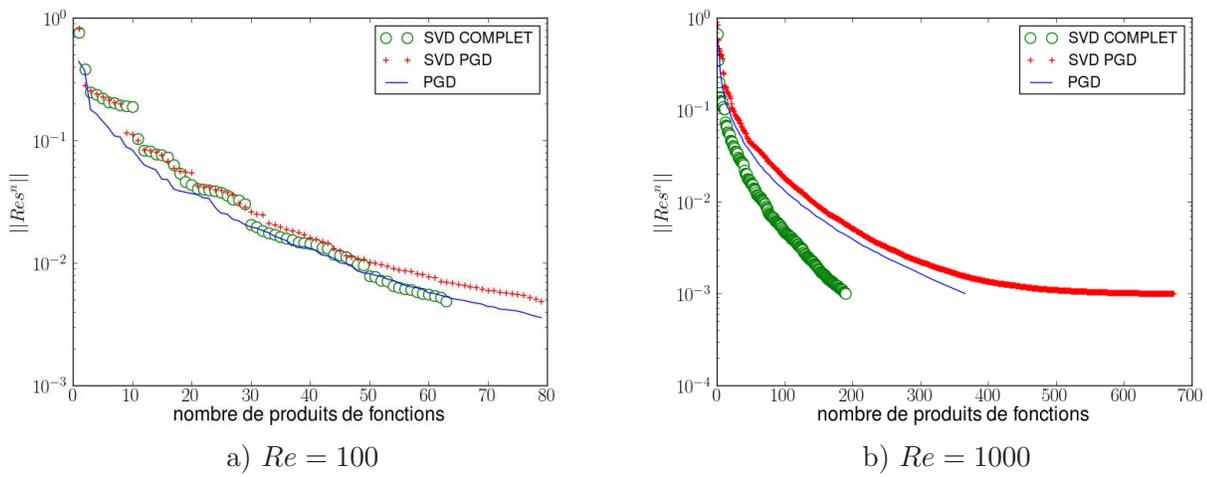


FIG. 3.40 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la pression de correction  $\tilde{p}$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 60$ ).

**Remarque** Afin de ne pas alourdir le manuscrit avec des résultats redondants la formulation ligne de courant vorticité a été détaillée en annexe A. On peut néanmoins rappeler que dans cette annexe, la PGD a été appliquée à la résolution des équations de Navier-Stokes avec une formulation ligne de courant-vorticité pour un écoulement dans une cavité entraînée 2D isotherme. Dans le cas stationnaire ( $Re = 100$  et  $Re = 1000$ ), les résultats obtenus par PGD sont en accord avec ceux obtenus par le modèle complet ainsi qu’avec les résultats de la littérature. De plus, la PGD permet d’obtenir un gain de temps (gain de temps d’un facteur deux pour  $Re = 100$  et quatre pour  $Re = 1000$  pour  $N_h = 500$ ) à partir de 300 nœuds dans chaque direction. Concernant le cas instationnaire ( $Re = 10000$ ), la PGD permet également de retrouver une période similaire à celle obtenue dans la littérature et un comportement des lignes de courants pendant cette même période comparable aux résultats du modèle complet. Cependant un gain de temps peu significatif a été observé.

## V Conclusions du chapitre

Dans ce chapitre, la PGD a été appliquée à la résolution des équations de Navier-Stokes en utilisant un schéma de prédiction correction dans le cadre d’un problème comportant une solution analytique (le problème de Taylor-Green) ainsi que pour un écoulement dans une cavité entraînée 2D isotherme. Concernant le problème de Taylor-Green, la PGD a été capable de retrouver la vitesse analytique en exactement un enrichissement, et ce, avec une erreur comparable à celle obtenue avec le solveur standard. De plus un gain de temps important (de l’ordre de la vingtaine) pour des maillages fins a été mis en évidence. Dans le cas de la cavité entraînée stationnaire ( $Re = 100$  et  $Re = 1000$ ), les résultats obtenus par PGD sont en accords avec ceux obtenus par le modèle standard ainsi qu’avec les résultats de la littérature. De plus, la PGD permet d’obtenir un gain de temps à partir de 150 nœuds dans chaque direction. Concernant le cas instationnaire, la PGD permet de retrouver une période similaire à celle obtenue dans la littérature et un comportement des lignes de courant pendant cette même période comparable aux résultats du modèle standard.

Dans le cas 3D, la PGD est capable de reproduire fidèlement le champ de vitesse au centre de la cavité pour des nombres de Reynolds égaux à 100 et 1000. Par contre, les isocontours de pression à  $Re = 1000$  obtenue par méthode PGD diffèrent de ceux obtenus par le solveur standard qui correspondent à ceux de la littérature alors que dans le cas  $Re = 100$  les résultats sont corrects. De plus pour un nombre de Reynolds égal à 1000, le gain de temps est nul alors qu’à  $Re = 100$  la PGD est seulement deux fois plus rapide que le solveur standard pour un maillage de taille  $100 \times 100 \times 100$ .

## Chapitre 4

# PGD pour la résolution des écoulements anisothermes

## I Introduction

Le problème de la convection mixte pour un écoulement dans une cavité entraînée est rencontré dans de nombreuses applications en ingénierie telles que le refroidissement de composants électroniques, les technologies de lubrification, certains procédés chimiques, les technologies de séchage, etc... Ainsi de nombreuses études concernant les écoulements anisothermes à l'intérieur des cavités ont été réalisées. A partir de ces différentes études, il est possible de classer ces problèmes en deux grands groupes distincts. Le premier groupe concerne les cavités où une des faces, ou deux faces, sont animées d'une vitesse constante (ou sinusoïdale) et où la cavité est différentiellement chauffée (c'est à dire que l'on impose une température chaude sur une des face et une température froide sur une autre face, les deux autres faces étant considérées comme adiabatiques). Ce genre de problème a été traité par de nombreux auteurs en 2D [Oztop et Dagtekin, 2004, Khanafer et al., 2007, Basak et al., 2010, Noor et al., 2009, Cheng et Liu, 2010] et également en 3D [Mohamad et Viskanta, 1995, Ouertatani et al., 2009, Iwatsu et Hyun, 1995]. Le deuxième groupe d'études consiste à reprendre une configuration du cas précédent et à rajouter des ailettes à l'intérieur du domaine dans l'objectif de comprendre l'influence de ces ailettes sur la nature de l'écoulement et éventuellement à effectuer du contrôle en jouant sur les formes et les positionnements de ces ailettes. Ce type de problème a été traité dans [Shi et Khodadadi, 2005, Shi et Khodadadi, 2003, Oztop et al., 2009, Sun et al., 2011].

Le cas de la cavité entraînée avec un transport de masse sera également traité. En effet, après avoir étudié la cavité entraînée isotherme et anisotherme, le fait de rajouter aux équations un terme de transport de masse est justifié du fait des nombreuses applications possible. Ces applications vont de la simulation des phénomènes de séchage, de combustion, d'évaporation à la simulation des procédés chimiques ou agro-alimentaires. La littérature concernant ce sujet est moins abondante que celle concernant les cavités entraînées isotherme ou anisotherme sans transport de masse. On peut cependant citer les travaux de Trevisan et Bejan [Trevisan et Bejan, 1990] ainsi que les travaux de Bennacer et Gobin [Bennacer et Gobin, 1996a, Bennacer et Gobin, 1996b] qui ont traité ces problèmes dans de grandes cavités et en ont déduit des formes de solutions en accord avec les simulations numériques pour différents paramètres. Alleborn et al. dans [Alleborn et al., 1999] ont étudié une cavité rectangulaire entraînée avec transport de masse et de chaleur en faisant varier l'orientation de la cavité. Ces travaux ont montré, entre autre, que la forme de l'écoulement était indépendante de l'orientation de la cavité pour des vitesses d'entraînement importantes, la convection forcée étant prépondérante. Enfin Al-Amiri et al. dans [Al-Amiri et al., 2007] ont étudié l'influence du nombre de Lewis ainsi que du nombre de Richardson sur une cavité entraînée avec transport de masse et de chaleur.

Dans un premier temps, les équations de Navier-Stokes anisothermes seront détaillées puis la PGD sera appliquée à deux types de cavités entraînée différentiellement chauffée où les résultats seront comparés à ceux du solveur standard et à des résultats issus de la littérature. Dans un deuxième temps le phénomène de transport de masse sera ajouté et le problème qui en résulte sera résolu à l'aide de la PGD. Là aussi les résultats obtenus seront comparés à ceux du solveur standard et de la littérature.

## II Ecoulement dans une cavité entraînée 2D anisotherme

### II.1 Equations du problème

Le fluide à l'intérieur du domaine est supposé Newtonien, incompressible et de propriétés physiques constantes. En effet, une approximation de Boussinesq est réalisée, c'est à dire que l'on considère des fluctuations de densité exclusivement dues à des variations de température, et suffisamment faibles pour ne pas remettre en question l'hypothèse d'incompressibilité et modifier la conservation de l'énergie. La température est donc considérée comme un scalaire dans l'équation de l'énergie. Sous ces hypothèses, les équations de la conservation de la masse, de quantité de mouvement, ainsi que l'équation d'énergie peuvent s'écrire :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + g\beta(T - T_c) \cdot \mathbf{y} \\ \frac{\partial T}{\partial t} - \frac{\lambda}{\rho C_p} \nabla^2 T + (\mathbf{u} \cdot \nabla) T = 0 \end{cases} \quad (4.1)$$

où  $\mathbf{u} = (u, v)$  représente la vitesse du fluide,  $p$  la pression et,  $T$  la température.  $\rho$  est la densité du fluide,  $\nu$  la viscosité cinématique,  $g$  la constante de gravité,  $\beta$  le coefficient d'expansion thermique,  $\lambda$  la conductivité thermique et  $C_p$  la chaleur spécifique du fluide. On définit les différentes variables adimensionnées suivantes :

$$\Theta = \frac{T - T_c}{T_h - T_c} \quad X = \frac{x}{d} \quad Y = \frac{y}{d} \quad U = \frac{u}{U_0} \quad V = \frac{v}{U_0} \quad P = \frac{p}{\rho U_0^2} \quad \tau = \frac{t U_0}{d}$$

Le problème (4.1) s'écrit alors sous forme adimensionnelle :

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \\ \frac{\partial \mathbf{U}}{\partial \tau} - \frac{1}{Re} \nabla^2 \mathbf{U} + (\mathbf{U} \cdot \nabla) \mathbf{U} = -\nabla P + Ri\Theta \cdot \vec{y} \\ \frac{\partial \Theta}{\partial \tau} - \frac{1}{RePr} \nabla^2 \Theta + (\mathbf{U} \cdot \nabla) \Theta = 0 \end{cases} \quad (4.2)$$

avec

$$Re = \frac{U_0 d}{\nu} \quad Gr = \frac{g\beta(T_h - T_c)d^3}{\nu^2} \quad Pr = \frac{\nu}{\alpha} \quad (4.3)$$

$$(4.4)$$

où  $\alpha$  désigne la diffusivité thermique,  $Re$  le nombre de Reynolds,  $Gr$  le nombre de Grashof,  $Pr$  le nombre de Prandtl. A partir de ces différents nombres adimensionnels, on peut définir le nombre de Richardson qui jauge le couplage entre les aspects thermiques et dynamiques de la manière suivante :

$$Ri = \frac{Gr}{Re^2} \quad (4.5)$$

Ce nombre caractérise l'importance de la convection naturelle relativement à la convection induite par l'entraînement forcé de la cavité sur la face supérieure, alors que le nombre de Grashoff représente la force de la convection naturelle et le nombre de Reynolds la force de la convection forcée. Pour un nombre de Richardson très petit nous sommes en régime de convection forcée, alors que lorsque le Richardson est grand nous sommes dans un régime dit de convection naturelle.

Afin de comparer nos résultats à ceux issus de la littérature, nous allons définir le nombre de Nusselt sur les frontières droite (est) ( $Nu_e$ ) et gauche (ouest) ( $Nu_w$ ) de la manière suivante :

$$Nu_e = -\frac{(\partial\Theta/\partial X)_e}{\Theta_h - \Theta_c} \quad Nu_w = -\frac{(\partial\Theta/\partial X)_w}{\Theta_h - \Theta_c} \quad (4.6)$$

ainsi que le nombre de Nusselt moyen, calculé en intégrant le Nusselt local dans une direction :

$$\overline{Nu_e} = \int_0^1 Nu_e dY \quad \overline{Nu_w} = \int_0^1 Nu_w dY \quad (4.7)$$

Par la suite, les équations de quantité de mouvement et de conservation de la masse seront résolues en utilisant l'algorithme de prédiction correction décrit dans la section III du chapitre 3 dans lequel le terme lié à la température (prise au pas de temps précédent) sera considérée comme un terme source dans les équations du mouvement. La température est ensuite obtenue en résolvant l'équation d'énergie du problème (4.2) qui correspond à une équation de diffusion instationnaire en utilisant un schéma de Cranck-Nicholson pour le terme diffusif et un schéma d'Adams-Bashfort pour le terme convectif. Le problème sera résolu en utilisant d'une part un solveur standard (bi-gradient conjugué), et d'autre part la méthode PGD. Dans ce cas les champs de vitesse, la pression, et la température seront recherchés sous forme séparée.

## II .2 Résultats

### a) Problème 1

#### 1) Position du problème

Soit une cavité entraînée classique, la face nord de la cavité est animée d'une vitesse horizontale constante égale à  $U_0$ , tandis que la vitesse sur toutes les autres faces est égale à zéro. Un gradient de température de gauche à droite est imposé, c'est à dire qu'une température  $T_h$  est imposée sur la frontière de droite, supérieure à celle imposée sur la frontière de gauche  $T_c$ . Les frontières nord et sud sont considérées comme adiabatique. La figure 4.1 illustre la cavité étudiée.

Les simulations ont été effectuées avec un pas de temps égal à  $\delta t = 10^{-3}$  et le critère de convergence utilisé est le suivant :

$$\frac{\|u_i^k - u_i^{k-1}\|}{\|u_i^k\|} \leq \epsilon_u \quad \text{avec } i = 1, 2, \quad \frac{\|p^k - p^{k-1}\|}{\|p^k\|} \leq \epsilon_p \quad \frac{\|T^k - T^{k-1}\|}{\|T^k\|} \leq \epsilon_T \quad (4.8)$$

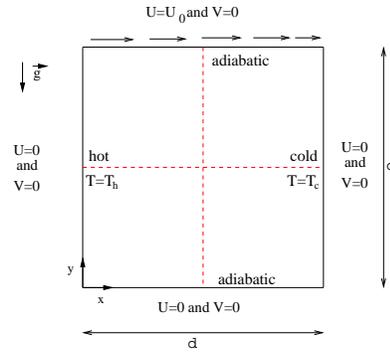


FIG. 4.1 – Geometrie de la cavité entraînée anisotherme 2D.

La solution est estimée convergée, lorsque les critères cités ci-dessus sont vérifiés. Dans les simulations qui suivent, les coefficients ont été pris égaux à  $\epsilon_u = 10^{-10}$ ,  $\epsilon_p = 10^{-8}$  et  $\epsilon_T = 10^{-10}$ . Les simulations ont été effectuées avec la méthode PGD ainsi qu'avec un solveur classique (bi-gradient conjugué) pour un nombre de Grashoff fixé à  $10^5$  et un nombre de Prandtl égal à 0.71 (caractéristique de l'air). Dans la suite, trois nombres de Richardson seront considérés ( $Ri = 0.1$ ,  $Ri = 1$  et  $Ri = 10$ ).

## 2) Résultats

Les nombres de Nusselts moyens sur les frontières de droite (est) et de gauche (ouest) à convergence du calcul par méthode PGD (et par le solveur standard) définis dans l'équation (4.7) sont comparés à ceux obtenus par Sun et al. dans [Sun et al., 2011]. Les résultats sont répertoriés dans le tableau 4.1.

	$\overline{Nu}_{est}$			$\overline{Nu}_{ouest}$		
	Sun et al.	PGD	Standard	Sun et al.	PGD	Standard
$Ri = 10$	4.61	4.49	4.49	4.61	4.54	4.54
$Ri = 1$	5.29	5.19	5.20	5.29	5.20	5.20
$Ri = 0.1$	7.46	7.44	7.45	7.46	7.48	7.48

TAB. 4.1 – Valeurs des Nusselts moyens sur les frontières de droite (est) et de gauche (ouest) issus de la littérature et de la PGD pour des nombres de Richardson égaux à 10, 1 et 0.1.

On note ici que le nombre de Nusselt moyen augmente avec la diminution du nombre de Richardson, ce qui indique que plus les effets dynamiques sont forts plus le transfert thermique sur les frontières droite et gauche de la cavité est important. On observe également que les nombres de Nusselts moyens obtenus par PGD sont proches de ceux de Sun et al., même si une légère différence est observée pour  $Ri = 1$  et  $Ri = 10$ . Cette différence n'est pas due à la PGD, car les mêmes résultats sont obtenus avec le solveur standard.

Les figures 4.2 et 4.3 montrent le champ de vitesse et les isocontours de température obtenus par PGD et ceux obtenus par Sun et al. [Sun et al., 2011]. On constate que pour

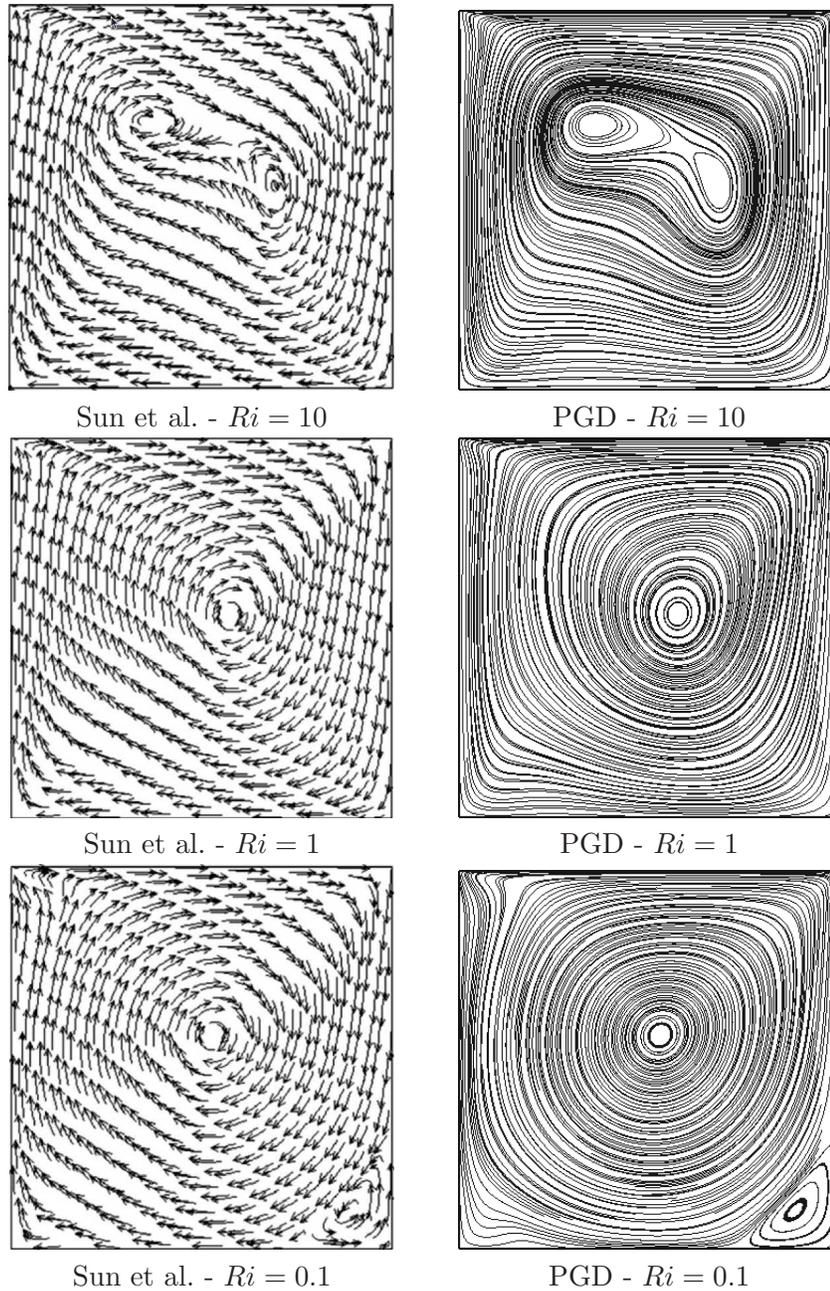


FIG. 4.2 – Vecteurs vitesses issus de Sun et al. [Sun et al., 2011] (à gauche) et lignes de courant issues de la PGD (à droite) pour différents Richardson et pour un nombre de nœuds  $N_h = 100$ .

$Ri = 10$ , l'écoulement est principalement dû au gradient de température et l'on peut observer la création d'un double vortex au centre de la cavité. Pour le cas où le nombre de Richardson est égal à 1, la convection forcée due à la vitesse d'entraînement sur la face supérieure commence à

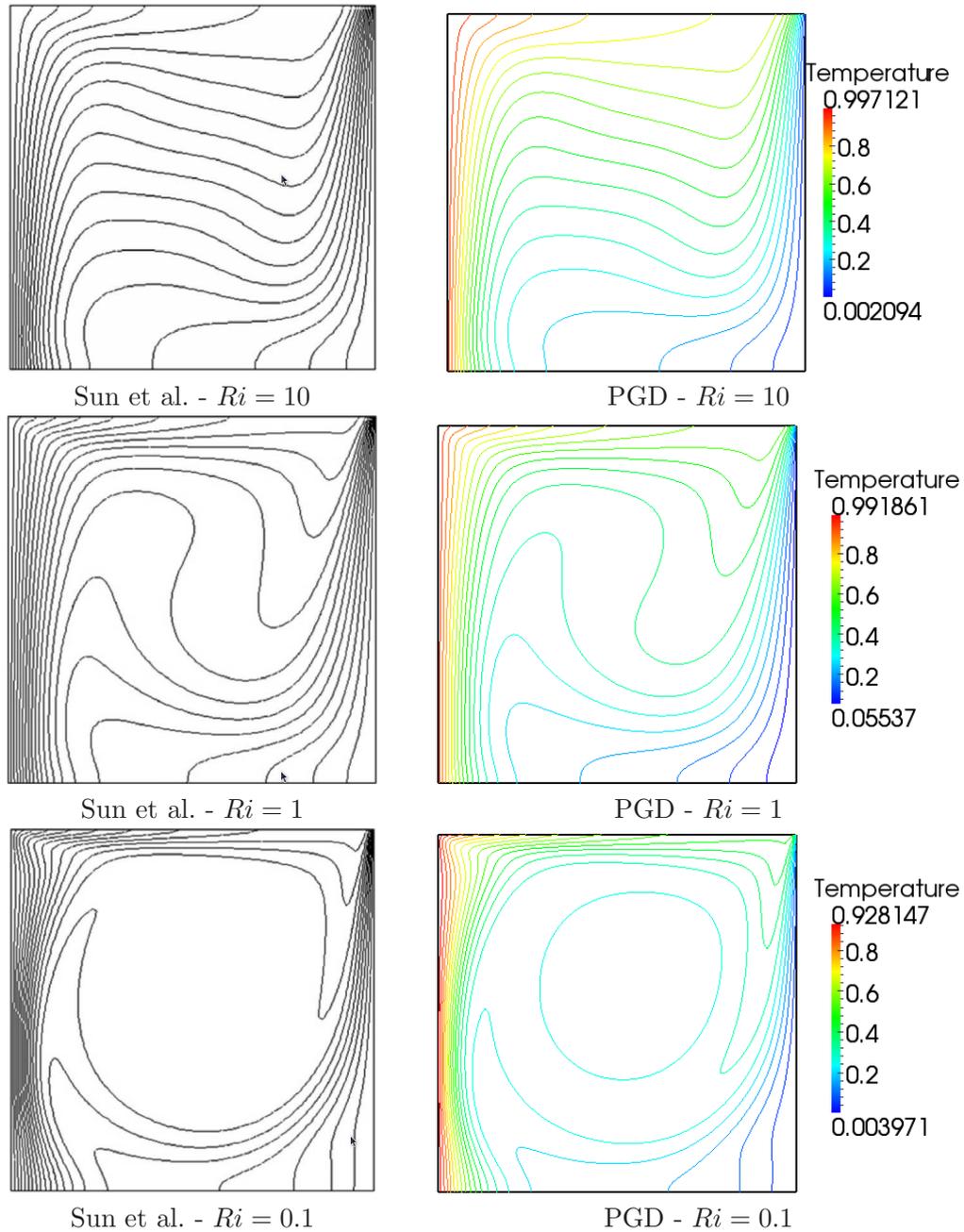


FIG. 4.3 – Isocontours de température issus de la littérature et issus de la PGD pour des nombres de Richardson égaux à 10, 1 et 0.1.

prendre de l'importance, cela se voit particulièrement en observant la taille de la couche limite de température sur la frontière de gauche de la cavité. Enfin pour un nombre de Richardson égal à 0.1, on observe que la convection forcée devient prépondérante par rapport à la température.

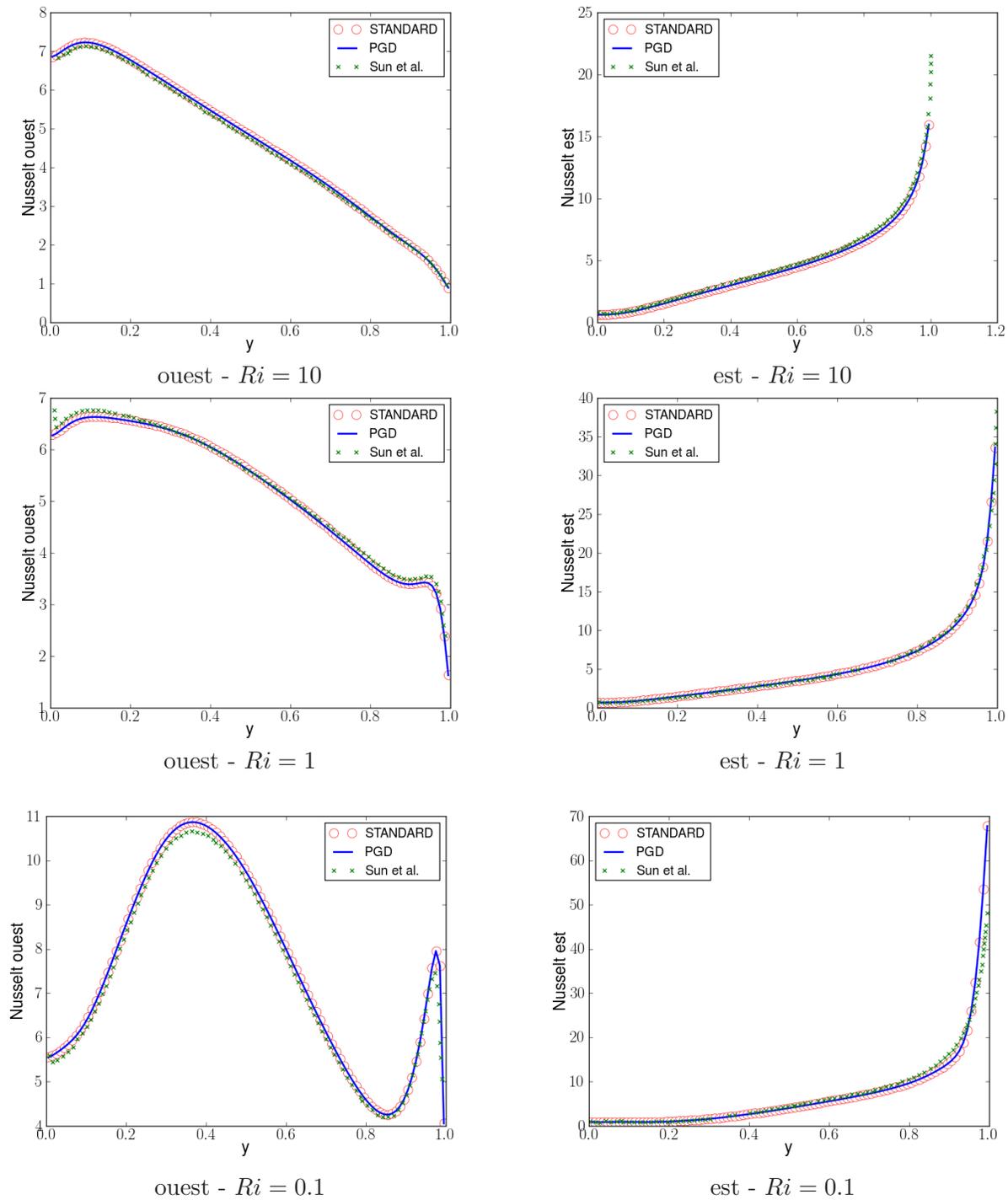


FIG. 4.4 – Valeurs des Nusselts sur les frontières ouest et est issus de la littérature, du modèle standard et de la PGD pour des nombres de Richardson égaux à 10, 1 et 0.1.

Dans ce cas les résultats ressemblent beaucoup à ceux obtenus pour la cavité entraînée 2D isotherme à  $Re = 1000$ <sup>1</sup> même si dans ce cas le vortex secondaire dans le coin inférieur gauche a disparu. Concernant les isocontours de température, on constate que plus le nombre de Richardson est important plus la température a tendance à se stratifier. De même si l'on s'intéresse aux couches limite en température sur les frontières gauche et droite de la cavité, on constate que celles ci ont tendance à se resserrer avec la diminution du nombre de Richardson. Ces figures montrent enfin que quelque soit le nombre de Richardson considéré, la PGD permet de retrouver les résultats de la littérature.

La figure 4.4 où les nombres de Nusselts à gauche et à droite calculés avec le solveur standard et avec la méthode PGD ont été comparés avec ceux de la littérature permet de confirmer ce qui a été dit à propos des résultats de la figure 4.3, à savoir une diminution de la couche limite en température quand le nombre de Richardson diminue, induisant une augmentation du gradient de température sur les frontières verticales. De plus, on observe une bonne correspondance entre les simulations et la littérature, et on remarque aussi que les nusselts calculés avec le solveur standard et la méthode PGD sont toujours identiques.

Nombre de nœuds dans chaque direction	(Temps de calcul Standard)/(Temps de calcul PGD)		
	$Ri = 10$	$Ri = 1$	$Ri = 0.1$
61	0.2	0.14	0.17
101	0.5	0.7	0.57
151	1.25	1.06	0.82
201	4.4	3.67	1.60
301	6.33	5.20	3.7

TAB. 4.2 – Rapport des temps de calcul pour la résolution des équations de Navier-Stokes anisotherme.

Le rapport temps de calcul par simulation avec le solveur standard sur temps de calcul par simulation PGD est consigné dans le tableau 4.2 pour des simulations effectuées avec un nombre de nœuds variant de 61 à 301 dans chaque direction. On observe qu'à partir de 200 nœuds dans chaque direction la PGD est plus rapide que le solveur standard pour chacun des nombres de Richardson considérés.

## b) Problème 2

### 1) Position du problème

Une autre cavité entraînée est maintenant étudié. Dans ce nouveau cas, la frontière gauche de la cavité est entraînée vers le haut tandis que la frontière droite de la cavité est entraînée vers le bas à une vitesse  $V_p$ . Un gradient de température de droite à gauche est également imposé, c'est à dire qu'une température  $T_h$  est imposée sur la frontière droite (est), supérieure à celle imposée sur la frontière de gauche (ouest)  $T_c$ . Les faces nord et sud sont considérées comme adiabatiques. La figure 4.5 illustre la cavité étudiée.

Les simulations ont été effectuées avec la méthode PGD ainsi qu'avec un solveur classique

<sup>1</sup>on rappelle que  $Re = \sqrt{Gr/Ri}$  et donc  $Ri = 0.1 \Rightarrow Re = 1000$

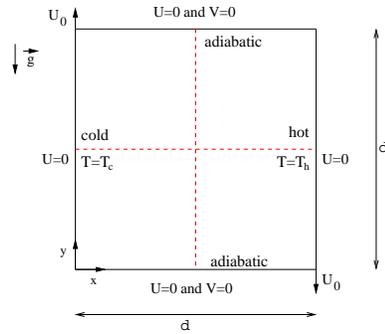


FIG. 4.5 – Geometrie de la cavité entraînée anisotherme 2D.

(bi-gradient conjugué) pour un nombre de Grashoff fixé à  $10^4$  et un nombre de Prandtl égal à 0.71 (caractéristique de l'air) avec un pas de temps  $\delta t = 10^{-3}$ . Dans la suite, trois nombres de Richardson seront considérés ( $Ri = 0.1$ ,  $Ri = 1$  et  $Ri = 10$ ). Les lignes pointillés sur la figure (4.5) représentent les endroits où seront visualisés les vitesses verticales et horizontales. Cette cavité entraînée a été traitée entre autre par Oztop et al. dans [Oztop et Dagtekin, 2004] pour les différents nombres de Richardson cités auparavant.

Les figures 4.6 et 4.7 représentent les lignes de courants et les isocontours de températures obtenus par PGD ( $N_h = 100$ ) comparés à ceux obtenus par Oztop et al. [Oztop et Dagtekin, 2004]. On constate que la PGD retrouve bien les résultats de la littérature. On retrouve pour des nombres de Richardson égaux à 0.1 et 1 un comportement proche de celui d'une cavité entraînée isotherme avec un léger aplatissement du vortex principal pour  $Ri = 1$ . Par contre lorsque l'on augmente la valeur du nombre de Richardson à 10 la configuration des lignes de courants est composée de trois vortex alignés dans le sens vertical. Le fluide remonte le long du mur chaud en raison de la poussée d'Archimède et est entraîné près du mur froid par les forces de cisaillement. Cela implique la création de deux vortex le long des parois rendant le transfert thermique de la droite vers la gauche possible. Concernant les isocontours de température, on remarque que plus le nombre de Richardson augmente, plus le gradient de température près des murs verticaux diminue, ce qui a pour effet de diminuer le transfert thermique. La PGD est donc encore une fois capable de retrouver ce changement de comportement lorsque le nombre de Richardson devient important et donc lorsque l'influence de la température se fait plus forte.

Les vitesses horizontales en  $x = 0.5$  et verticales en  $y = 0.5$  calculées par PGD et à l'aide du solveur standard sont tracées sur la figure 4.8. Sur cette figure on constate que les résultats obtenus par PGD sont similaires à ceux obtenus à l'aide du solveur standard. En accord avec la littérature, on observe également que le profil de vitesse horizontale s'inverse lorsque le nombre de Richardson augmente. En effet pour  $Ri = 0.1$ , la convection forcée est dominante et la vitesse tourne dans le sens négatif. Lorsque le nombre de Richardson augmente, l'influence de la température devient prépondérante et le profil de vitesse horizontale s'inverse à  $Ri = 10$  pour favoriser un écoulement dans le sens positif au centre de la cavité.

De même, la figure 4.9 représente, pour des nombres de Richardson égaux à 0.1, 1 et 10, les

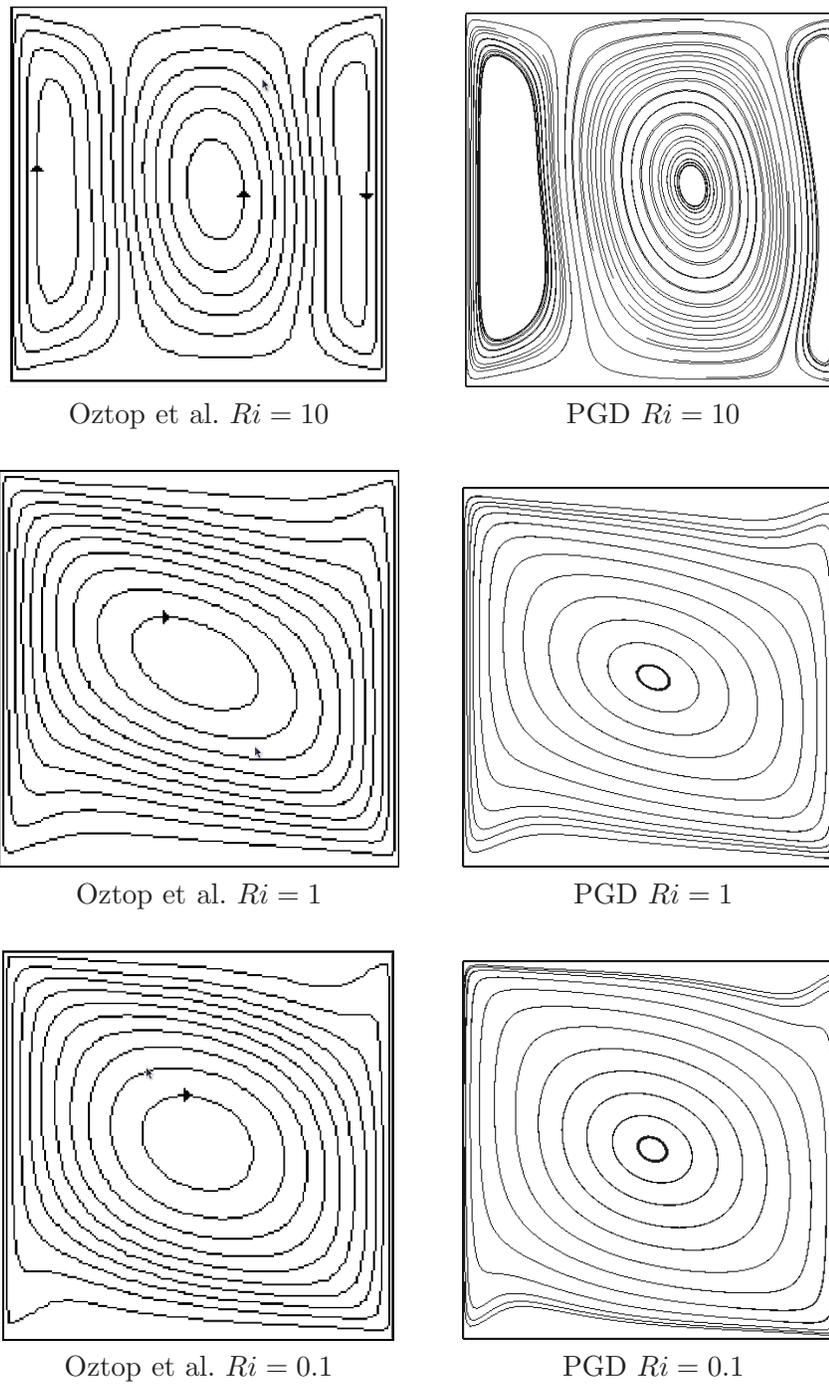


FIG. 4.6 – Lignes de courants issues de Oztop et al. et issues de la PGD pour différents nombres de Richardson .

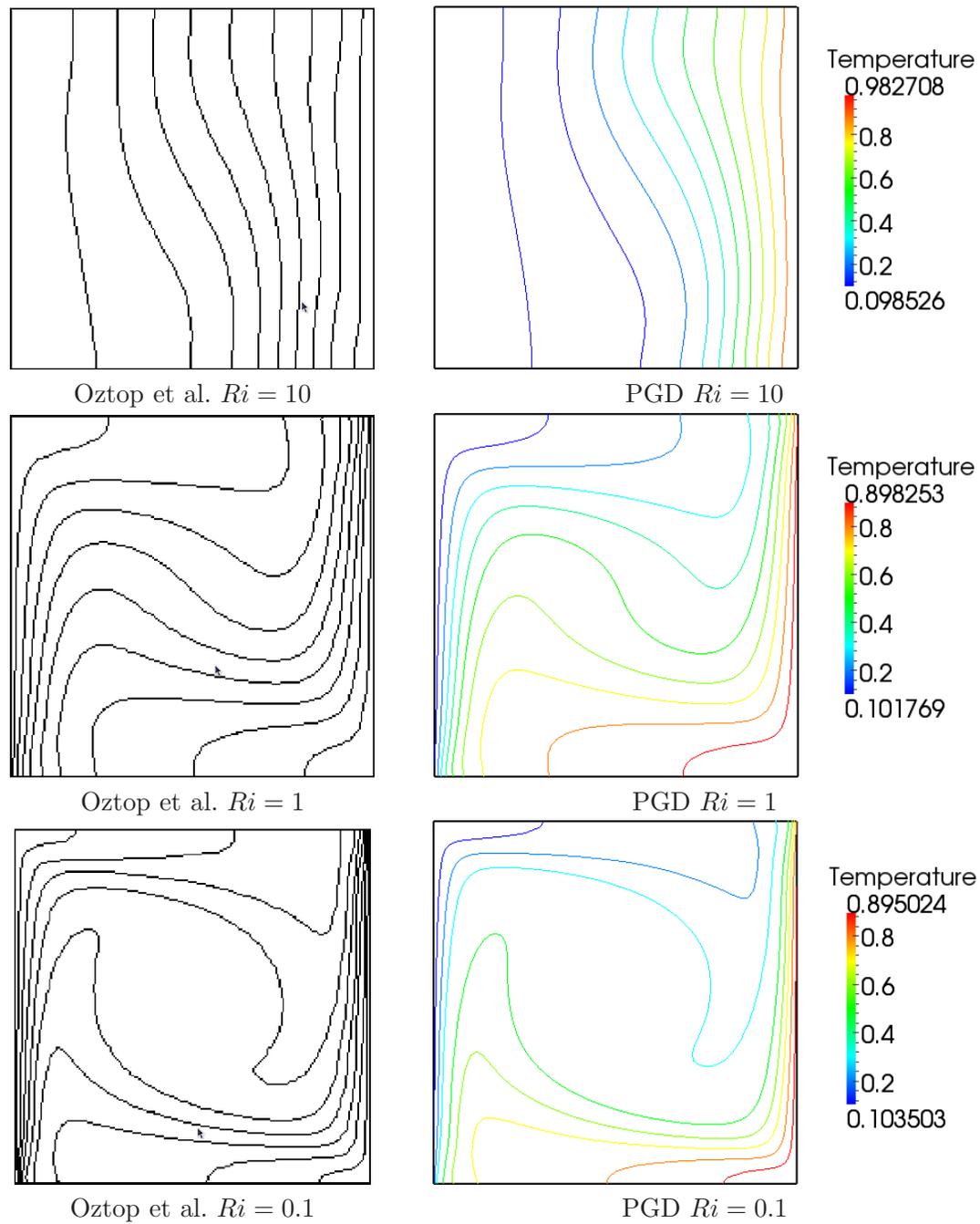


FIG. 4.7 – Isocontours de température issus de Oztop et al. et issus de la PGD pour différents nombres de Richardson.

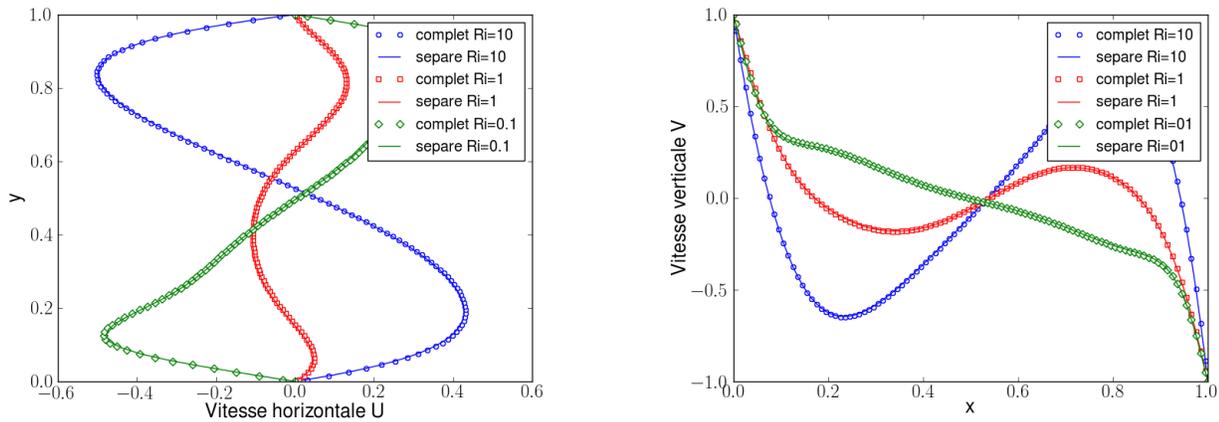


FIG. 4.8 – Comparaison des vitesses horizontales en  $x = 0.5$  (gauche) et des vitesses verticales en  $y = 0.5$  (droite) obtenue par PGD et à l'aide du solveur standard pour  $Ri = 0.1$ ,  $Ri = 1$  et  $Ri = 10$ .

nombre de Nusselts le long des frontières de gauche et de droite obtenus par PGD et à l'aide du solveur standard. Cette figure confirme les observations faite sur la figure 4.7, à savoir, une diminution du gradient de température le long des parois mobiles lorsque le nombre de Richardson augmente et donc une augmentation de la stratification de la température. Les résultats obtenus par PGD sont là aussi en accord avec ceux obtenus par avec le solveur standard.

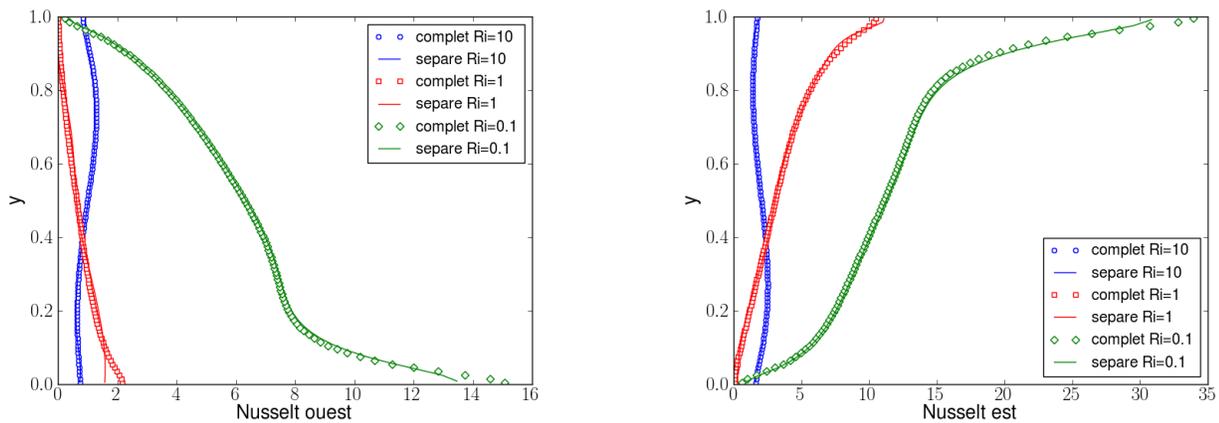


FIG. 4.9 – Comparaison des Nusselts sur les frontières de gauche (gauche) et de droite (droite) obtenue par PGD et à l'aide du solveur standard pour  $Ri = 0.1$ ,  $Ri = 1$  et  $Ri = 10$ .

Enfin, le rapport temps de calcul par simulation avec le solveur standard ( $T_{Stand}$ ) sur temps de calcul par simulation PGD ( $T_{PGD}$ ) est consigné dans le tableau 4.3 pour des simulations

effectuées avec nombre de nœuds variant de 61 à 301 dans chaque direction. On observe qu'à partir de 200 nœuds dans chaque direction la PGD est plus rapide que le solveur standard pour chacun des nombres de Richardson considérés. En effet, pour un maillage de taille  $301 \times 301$  la PGD est sept fois plus rapide que le solveur standard pour  $Ri = 10$ , six fois plus rapide pour  $Ri = 1$  et quatre fois plus rapide pour  $Ri = 0.1$ .

Nombre de nœuds dans chaque direction	$T_{Stand}/T_{PGD}$		
	$Ri = 10$	$Ri = 1$	$Ri = 0.1$
61	0.26	0.32	0.16
101	0.56	0.6	0.58
151	0.96	1.86	0.77
201	3.21	3.67	1.60
251	4.15	4.20	2.68
301	7.19	6.53	4.0

TAB. 4.3 – Rapport des temps de calcul pour la résolution des équations de Navier-Stokes anisotherme.

### III La cavité entraînée 2D anisotherme avec transport de masse

#### III .1 Equations du problème

Dans ce cas, le mélange est supposé parfait et le fluide à l'intérieur du domaine est supposé Newtonien, incompressible et de propriétés constantes excepté la densité du fluide  $\rho$ , qui est supposé varier linéairement avec la température et la concentration [Weaver et Viskanta, 1991a, Weaver et Viskanta, 1991b] telle que  $\rho = \rho_0[1 - \beta_T(T - T_c) - \beta_C(C - C_c)]$ .  $\rho_0$  est la densité caractéristique,  $\beta_T$  et  $\beta_C$  sont respectivement le coefficient de dilatation thermique et le coefficient de dilatation relié à la concentration. Les équations du problème seront écrites sous formes adimensionnées en utilisant les nouvelles variables adimensionnelle suivantes :

$$\begin{aligned} \Theta &= \frac{T - T_c}{T_h - T_c}, & \Phi &= \frac{C - C_c}{C_h - C_c}, & \tau &= \frac{tU_0}{d}, & P &= \frac{p}{\rho_0 U_0^2} \\ X &= \frac{x}{d}, & Y &= \frac{y}{d}, & U &= \frac{u}{U_0}, & V &= \frac{v}{U_0} \end{aligned} \quad (4.9)$$

où  $x$  et  $y$  sont les coordonnées cartésiennes mesurées le long de l'axe horizontal et vertical de la cavité,  $T$  est la température du fluide,  $C$  est la concentration de l'espèce,  $t$  est le temps.  $\Theta$  représente donc la température adimensionnée et  $\Phi$  la concentration adimensionnée. Les équations du problème sous forme adimensionnée peuvent donc s'écrire de la manière suivante :

$$\begin{cases} \nabla \cdot \mathbf{U} = 0. \\ \frac{\partial \mathbf{U}}{\partial \tau} - \frac{1}{Re} \nabla^2 \mathbf{U} + (\mathbf{U} \cdot \nabla) \mathbf{U} = -\nabla P + Ri(\Theta + \frac{Gr_C}{Gr_T} \Phi) \cdot \mathbf{y} \\ \frac{\partial \Theta}{\partial \tau} - \frac{1}{RePr} \nabla^2 \Theta + (\mathbf{U} \cdot \nabla) \Theta = 0 \\ \frac{\partial \Phi}{\partial \tau} - \frac{1}{ReSc} \nabla^2 \Phi + (\mathbf{U} \cdot \nabla) \Phi = 0 \end{cases} \quad (4.10)$$

où  $Re$  est le nombre de Reynolds,  $Sc$  le nombre de Schmidt,  $Ri$  le nombre de Richardson et  $Pr$ <sup>2</sup> le nombre de Prandtl.  $Gr_T = g\beta_T(T_h - T_c)d^3/\nu$  et  $Gr_c = g\beta_C(C_h - C_c)d^3/\nu$  sont respectivement le nombre de Grashof thermique et le nombre de Grashof massique. A partir de ces différents nombres adimensionnels, on peut définir le nombre de Lewis qui est un nombre sans dimension comparant la diffusivité de masse avec la diffusivité thermique. :

$$Le = \frac{Sc}{Pr} \quad (4.11)$$

Par la suite, les équations de quantité de mouvement et de continuité seront résolues en utilisant l'algorithme de prédiction correction décrit dans la section III du chapitre 3 dans lequel les termes liés à la température et à l'espèce (tous les deux pris au pas de temps

<sup>2</sup>des définitions de  $Re$ ,  $Ri$ , et  $Pr$  ont été données dans le paragraphe II .1

précédent) seront considérés comme des termes sources dans les équations du mouvement. La température et la concentration sont ensuite obtenues en résolvant les équations d'énergie et de concentration du problème 4.10 qui correspondent toutes deux à une équation de diffusion instationnaire où un schéma de Cranck-Nicholson pour le terme diffusif et un schéma d'Adams-Bashfort pour le terme convectif sont utilisés. Le problème sera résolu en utilisant d'une part un solveur standard (bi-gradient conjugué), et d'autre part la méthode PGD, dans ce cas les champs de vitesse, la pression, la température et la concentration seront recherchés sous forme séparée.

### III .2 Résultats

#### a) Position du problème

La cavité étudiée ici est telle que sa face nord est animée d'une vitesse horizontale constante égale à  $U_0$ , tandis que la vitesse sur toutes les autres faces est égale à zéro. Un gradient de température du sud au nord est imposé, c'est à dire qu'une température  $T_h$  imposée sur la face sud est supérieure à celle imposée sur la face nord  $T_c$ . De la même manière la concentration en espèce sur la face sud  $C_h$  est supérieure à celle imposée sur la face nord notée  $C_c$ . Les faces est et ouest sont quant à elle considérées comme adiabatiques et imperméables. La figure 4.10 illustre la cavité étudiée.

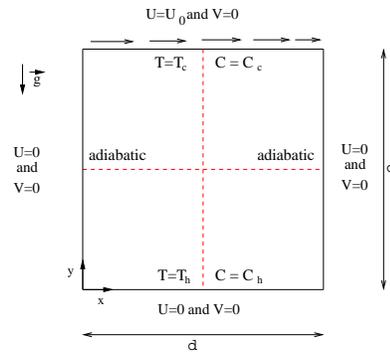


FIG. 4.10 – Geometrie de la cavité entraînée anisotherme avec transport de masse 2D.

Les simulations qui suivent sont toutes effectuées avec  $Pr = 1$  et  $Re = 100$ . Un pas de temps égal à  $\delta t = 10^{-3}$  et un critère de convergence sur la concentration est ajouté à ceux définis précédemment dans le cas anisotherme (Eq. 4.8), à savoir :

$$\frac{\|C^k - C^{k-1}\|}{\|C^k\|} \leq \epsilon_C \quad (4.12)$$

Ici les critères de convergence ont été pris égaux à  $\epsilon_u = 10^{-10}$ ,  $\epsilon_p = 10^{-10}$ ,  $\epsilon_T = 10^{-10}$  et  $\epsilon_C = 10^{-10}$ .

#### b) Résultats

Dans un premier temps on se fixe le nombre de Richardson égal à 0.01 et le nombre de

Grashof à  $10^2$  et on fait varier le nombre de Lewis de 5 à 50.

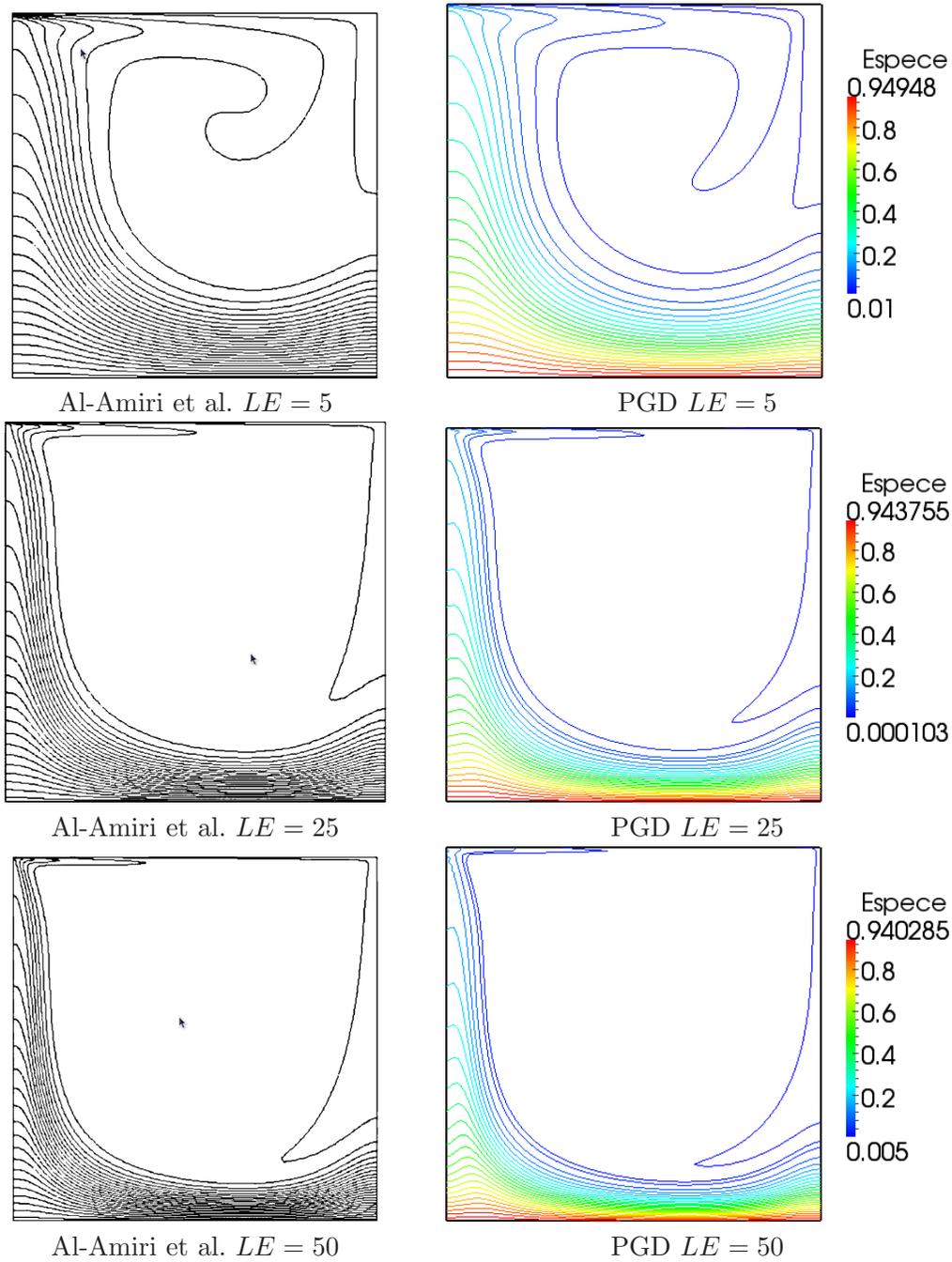


FIG. 4.11 – Isocontours de concentration issus de la littérature et issus de la PGD pour différents nombres de Lewis pour  $N_h = 250$ .

La figure 4.11 sur laquelle sont tracées les isovaleurs de concentrations montre une très

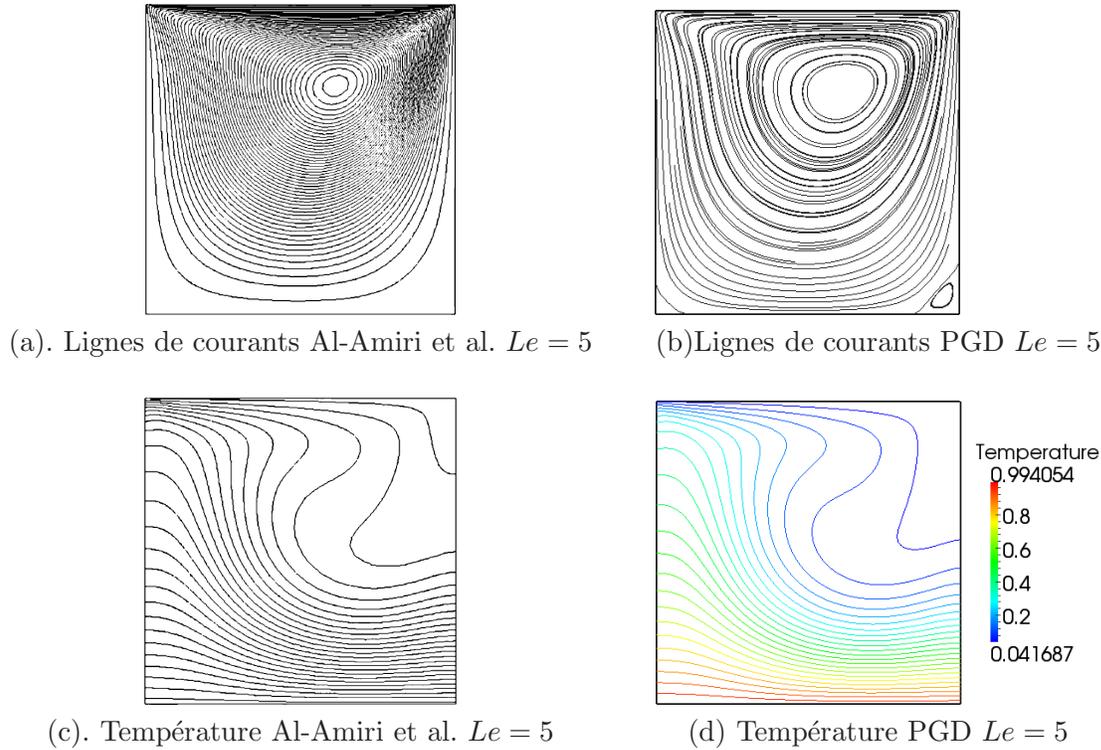


FIG. 4.12 – Lignes de courants issues de Al-Amiri et al. et de la PGD et isocontours de température issus de Al-Amiri et al. et issus de la PGD pour un nombre de Lewis égal à 5.

bonne correspondance entre les résultats PGD et ceux d'Al-Amiri et al [Al-Amiri et al., 2007]. Dans ce cas, on constate que la couche limite sur la frontière du bas semble diminuer avec l'augmentation du nombre de Lewis, ce qui est dû au fait que lorsque le nombre de Lewis augmente, la diffusivité de l'espèce diminue. De plus, quel que soit le nombre de Lewis ( $Le = 5, 25$  ou  $50$ ) le champ de vitesse et la température restent inchangés, en raison du caractère dominant de la convection forcée, ce qui avait déjà été noté par Al-Amiri et al. Ainsi sur la figure 4.12 nous avons uniquement reporté les lignes de courant et les isovaleurs de température pour  $Le = 5$ . Encore une fois on constate que les résultats PGD sont similaires à ceux de la littérature.

Dans un second temps on se fixe le nombre de Richardson à 0.01, le nombre de Reynolds à 100, ce qui impose une valeur du Grashoff thermique égale à 100. On fait varier le nombre de Grashoff massique de façon à ce que le ratio  $Gr_T/Gr_c$  varie entre 0 et  $-100$ . Cela revient donc à faire varier le nombre de Grashoff massique de 0 à  $-10000$ . Les simulations sont effectuées par méthode PGD et avec le solveur standard pour un nombre de Lewis égal à 1, ce qui implique que les champs de température et de concentrations seront identiques. Les valeurs du ratio entre le Grashoff massique et le Grashoff thermique sont prises égales à 0,  $-25$ ,  $-50$  et  $-100$ .

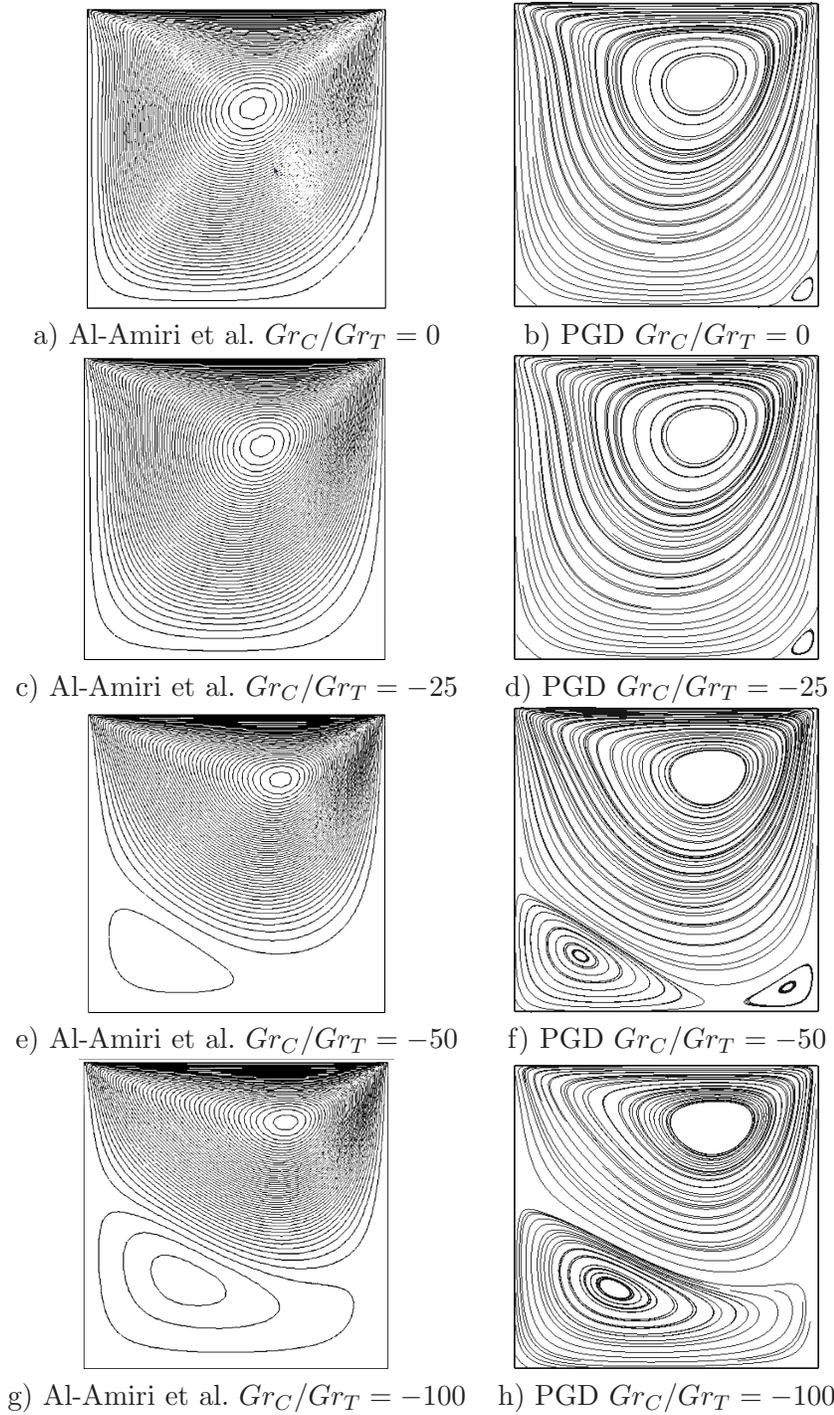


FIG. 4.13 – Lignes de courants issues de la littérature et calculées par PGD pour différents ratios  $Gr_C/Gr_T$ .

Les lignes de courants obtenues par PGD sont comparées à celles obtenues par Al-Amiri et al. [Al-Amiri et al., 2007] sur la figure 4.13. Cette figure montre que là encore la PGD retrouve les résultats de la littérature. Al-Amiri et al. montrent que dans ce cas les champs de température qui sont les mêmes que les champs de concentrations ne sont pas modifiés, c'est pourquoi ils ne sont pas représentés dans la suite, on peut cependant affirmer que la PGD retrouve bien les résultats de la littérature concernant ces champs.

Le rapport du temps de calcul par le solveur standard sur le temps de calcul par PGD nécessaire aux différents cas sont consignés dans la table 4.4. Les simulations ont été effectuées pour un nombre de nœuds allant de 61 à 301 dans chaque direction. On observe qu'à partir de 200 nœuds dans chaque direction la PGD est plus rapide que le solveur standard pour chacun des nombres de Richardson considérés.

Nombre de nœuds dans chaque direction	(Temps de calcul Standard)/(Temps de calcul PGD)			
	$N = 0$	$N = -25$	$N = 50$	$N = -100$
61	0.27	0.18	0.13	0.1
101	0.53	0.45	0.31	0.3
201	1.93	1.83	7.1	7.1
301	5.47	6.42	9.42	9.42

TAB. 4.4 – Rapport des temps de calcul pour la résolution des équations de Navier-Stokes anisotherme.

## IV Conclusions du chapitre

En conclusion de cette partie sur la simulation des équations de Navier-Stokes 2D anisothermes, on peut noter les performances de la PGD à traiter des équations de transfert. En effet quantitativement, on a constaté que dans le premier problème étudié, la PGD était capable de fournir des bonnes valeurs de nusselts sur les frontières non adiabatique, ainsi que des nombres de Nusselts moyens proches de ceux trouvés dans la littérature. Le deuxième cas d'étude a permis de constater que la PGD était également capable de retrouver le changement de comportement des lignes de courants en fonction du nombre de Richardson. De plus le gain de temps réalisé en utilisant la méthode PGD est relativement important par rapport aux solveurs traditionnels. En effet, pour un maillage de taille  $301 \times 301$  on observe des gains de temps de l'ordre de 7 pour un nombre de Richardson égal à 10 et de l'ordre de 4 pour un nombre de Richardson égal à 0.1.

Lorsque le transport de masse est ajouté aux équations, on constate que là encore, la PGD retrouve les résultats de la littérature et du solveur standard. Ceci, en faisant varier le nombre de Lewis, et donc en faisant varier l'importance de l'influence de la température par rapport à l'espèce. De plus, les simulations effectuées en faisant varier le rapport des nombres de Grashoff massique et thermique ont d'une part servi à valider la méthode PGD sur les problèmes de Navier-Stokes anisotherme avec un transport de masse, et d'autre part à constater qu'une fois encore le gain de temps réalisé en utilisant la PGD par rapport à un solveur classique est d'autant plus important que le nombre de nœuds dans chacune des directions augmente.

## Chapitre 5

# Vers une discrétisation spectrale de la PGD

Dans ce chapitre nous allons effectuer une première approche de la résolution des équations de Navier-Stokes par méthode PGD en formulation spectrale. En effet, un des intérêts des méthodes spectrales est de pouvoir résoudre des problèmes en utilisant des maillages plus grossiers que ceux nécessaires avec une formulation en volumes finis, pour une précision équivalente. Il existe au moins deux grandes familles de méthodes de résolution des équations de Navier-Stokes en formulation spectrale. La première repose sur une approche découplée, basée sur l'introduction d'une équation de Poisson sur la pression et sur l'imposition d'une condition limite supplémentaire sur la divergence de la vitesse sur les bords du domaine. Cette méthode ne sera pas traitée ici, mais elle a notamment été traitée par Haldenwang [Haldenwang, 1984] et Tuckerman [Tuckermann, 1989]. La seconde famille de méthodes que nous utiliserons par la suite, repose sur une résolution par des schémas de prédiction-correction, qui, en éliminant la vitesse, conduit à la résolution d'une équation de Poisson sur la pression par une méthode d'Uzawa. Dans ce cas, il faut faire attention à ce que le problème discret en pression et en vitesse soit bien posé. En effet, ceux-ci doivent vérifier une condition de compatibilité, appelée condition inf-sup [Canuto et al., 1986][Canuto et al., 1988], formulée initialement par Brezzi [Brezzi, 1974] pour des problèmes de point-selle.

Dans les premières générations de méthodes spectrales, le champ de vitesse et la pression appartenaient au même espace de polynômes  $\mathbb{P}_{N_x} \otimes \mathbb{P}_{M_y}$ . En effet, la vitesse et la pression étaient définis sur les points de collocation  $\mathcal{GL}_x^N \otimes \mathcal{GL}_y^M$  où  $\mathcal{GL}_x^N = \{x_i, x_i = \cos(\frac{i\Pi}{N}), i = 0, \dots, N\}$  correspond aux  $N + 1$  points de Gauss-Lobato dans la direction  $x$  et  $N$  est le degré maximal des polynômes dans la direction  $x$ <sup>1</sup>. Morchoisne [Morchoisne, 1983] a montré que dans ce cas la matrice de l'opérateur d'Uzawa contient un grand nombre de modes parasites (8) ce qui rend la résolution très difficile. Deux raisons à l'origine de ce problème ont été identifiées : la première vient du fait que les vitesse et la pression sont discrétisées dans le même espace, la seconde est liée au fait que la pression est définie sur les bords du domaine et que la condition d'incompressibilité est imposée sur les bords du domaine également. Cela revient en fait à ne pas vérifier la condition inf-sup.

Pour pallier à cette difficulté, Montigny-Rannou et Morchoisne ont introduit les premières grilles décalées [Montigny-Rannou et Morchoisne, 1987]. Dans ce cas, les vitesses sont prises dans l'espace  $\mathbb{P}_{N_x} \otimes \mathbb{P}_{M_y}$  et sont connues sur les grilles de collocations  $\mathcal{GL}_x^N \otimes \mathcal{GL}_y^M$  tandis que la pression est prise dans l'espace  $\mathbb{P}_{N-1_x} \otimes \mathbb{P}_{M-1_y}$  et est connue sur la grille de collocation  $\mathcal{G}_x^N \otimes \mathcal{G}_y^M$  où  $\mathcal{G}_x^N = \{x_i, x_i = \cos(\frac{(2i-1)\Pi}{2N}), i = 1, \dots, N\}$  sont les  $N$  points de Gauss-Lobato dans la direction  $x$ . Dans ce cas la matrice d'Uzawa contient encore deux modes parasites (dont le mode constant), ce qui rend la méthode insatisfaisante.

Un autre algorithme de grilles décalées introduit par Bernardi et al. [Bernardi et Maday, 1988] consiste à rechercher les deux composantes de vitesses et la pression dans trois espaces polynomiaux différents et donc à définir ces variables sur trois maillages différents. La pression est définie dans l'espace  $\mathbb{P}_{N-1_x} \otimes \mathbb{P}_{M-1_y}$  et est connue sur les points de collocation  $\mathcal{G}_x^N \otimes \mathcal{G}_y^M$  tandis que  $u$  (respectivement  $v$ ) est prise dans l'espace  $\mathbb{P}_{N_x} \otimes \mathbb{P}_{M-1_y}$  (resp.  $\mathcal{G}_x^N \otimes \mathcal{G}_y^M$ ) et définie aux points de collocation  $\mathcal{GL}_x^N \otimes \mathcal{G}_y^M$  (resp.  $\mathcal{G}_x^N \otimes \mathcal{GL}_y^M$ ). Dans ce cas le noyau de l'opérateur d'Uzawa est constitué uniquement du mode constant et la méthode est donc satisfaisante. Cependant, elle est difficile à mettre en œuvre, en particulier si l'on souhaite l'étendre à 3

<sup>1</sup>  $\mathcal{GL}_y^M$  est défini de la même manière et  $M$  est le degré maximal des polynômes dans la direction  $y$

dimensions.

Une autre possibilité consiste à traiter le problème de la condition inf-sup en recherchant la pression dans l'espace  $\mathbb{P}_{N-2_x} \otimes \mathbb{P}_{M-2_y}$  tandis que la vitesse est recherchée dans l'espace  $\mathbb{P}_{N_x} \otimes \mathbb{P}_{M_y}$ . En effet en réduisant la dimension de la pression d'un degré supplémentaire dans chaque direction par rapport à ce qu'avait imaginé Montigny-Rannou et Morchoisne, il est possible de faire disparaître les modes parasites de pression. Cette nouvelle méthode de grilles décalées a été introduite par Patera, Maday et al. [Maday et al., 1987, Maday et al., 1993, Rønquist, 1990, Maday et al., 1992]. Elle a également été utilisée, entre autres, par Azaïez [Azaïez, 1990] et Botella [Botella, 1996] pour la résolution des équations de Navier-Stokes. Dans la suite cette méthode sera nommée la méthode  $\mathbb{P}_N - \mathbb{P}_{N-2}$ .

Dans la suite, après une présentation détaillée de la méthode  $\mathbb{P}_N - \mathbb{P}_{N-2}$  en formulation spectrale Tchebychev, nous détaillerons comment cette méthode peut être appliquée pour résoudre les équations de Navier-Stokes à travers un schéma de prédiction-correction. Ensuite, la formulation PGD de la méthode  $\mathbb{P}_N - \mathbb{P}_{N-2}$  sera explicitée. Enfin les résultats obtenus sur une équation de Poisson stationnaire, un problème de Darcy (correspondant à l'étape de correction du schéma de prédiction-correction), le problème de 'Taylor-Green' et la cavité entraînée 2D seront présentés.

## I Approximation $\mathbb{P}_N - \mathbb{P}_{N-2}$ en formulation spectrale Tchebychev

Comme il a été précisé dans l'introduction, la discrétisation spatiale sera effectuée en utilisant une formulation spectrale Tchebychev  $\mathbb{P}_N - \mathbb{P}_{N-2}$ . Cette méthode présente l'avantage de ne nécessiter qu'une grille de calcul pour la vitesse et la pression (deux en réalité, mais la grille de pression correspond à la grille de vitesse à laquelle on a retiré les nœuds des frontières). Les équations de Navier-stokes seront donc discrétisées sur la grille des points de Tchebychev-Gauss-Lobato usuels  $\overline{\Omega}_c = (x_i, y_j)$  où les points de collocation  $x_i$  et  $y_j$  sont définis par :

$$\begin{cases} x_i = \cos(i\frac{\pi}{N}) & \text{pour } i = 0, \dots, N \\ y_j = \cos(j\frac{\pi}{M}) & \text{pour } j = 0, \dots, M \end{cases} \quad (5.1)$$

Notons, de plus,  $\Gamma_c$  l'ensemble des points de collocation du bords du domaine, et  $\Omega_c$  l'ensemble des points de collocation intérieurs au domaine  $\overline{\Omega}_c$ .

### I.1 Principe de l'approximation $\mathbb{P}_N - \mathbb{P}_{N-2}$

Afin de simplifier la présentation de la méthode nous nous placerons sur le domaine 1D,  $\Lambda = ]-1, 1[$ . Dans la suite une simple transformation affine nous permettra de nous placer dans un intervalle quelconque  $]a, b[$ . La méthode  $\mathbb{P}_N - \mathbb{P}_{N-2}$  consiste à rechercher la vitesse  $\mathbf{u}$  comme un élément appartenant à l'espace vectoriel des polynômes de degrés au plus  $N$  sur  $\Lambda$ ,  $\mathbb{P}_N(\Lambda)$ . La pression quant à elle sera recherchée comme un élément de  $\mathbb{P}_{N-2}(\Lambda)$ . Définissons la grille de collocation des points de Tchebychev Gauss Lobato  $\Lambda_c$  de la manière suivante :

$$x_i = \cos\left(i \frac{\pi}{N}\right) \quad i = 0, \dots, N \quad (5.2)$$

La vitesse sera donc exprimée en utilisant les  $N + 1$  points de la grille de collocation  $\Lambda_c$ , tandis que la pression ne sera exprimée qu'en fonction des  $N - 1$  points intérieurs de cette même grille. De cette manière la vitesse et la pression ne sont pas du même espace, même si la grille de collocation utilisée est la même. Les opérateurs de dérivation de chacun de ces champs seront donc exprimés différemment. Les opérateurs sont construits à partir d'une base d'interpolation de Lagrange.

## I.2 Expression de l'opérateur de dérivée première dans $\mathbb{P}_N$

L'espace vectoriel  $\mathbb{P}_N(\Lambda)$  est muni de la base de dimension  $N + 1$   $\{h_i; i = 0, \dots, N\}$  où  $h_i$  est le polynôme d'interpolation de Lagrange de degré  $N$  associé au point de collocation  $x_i$ . Ainsi on a :

$$h_i(x_j) = \delta_{ij} \quad \forall j = 0, \dots, N \quad (5.3)$$

où le polynôme d'interpolation de Lagrange s'écrit :

$$h_i(x) = \prod_{j=0, j \neq i}^N \frac{x - x_j}{x_i - x_j} = \frac{(-1)^{i+1} (1 - x^2) T'_N(x)}{\bar{c}_i N^2 (x - x_i)} \quad \forall i = 0, \dots, N \quad (5.4)$$

$T_N(x) = \cos(N \arccos(x))$  est le polynôme de Tchebychev de degré  $N$ , et

$$\bar{c}_i = \begin{cases} 1/2 & i = 0, N \\ 1 & \text{sinon} \end{cases} \quad (5.5)$$

Ces polynômes d'interpolation sont représentés sur la figure 5.1 pour  $N = 10$ .

De cette manière un élément  $u$  de  $\mathbb{P}_N(\Lambda)$  peut s'écrire,

$$u(x) = \sum_{j=0}^N u(x_j) h_j(x) \quad (5.6)$$

et admet pour dérivée première, le polynôme

$$u'(x) = \sum_{j=0}^N u(x_j) h'_j(x) \quad (5.7)$$

La valeur en un point de collocation  $x_i \in \Lambda_c$  s'écrit donc :

$$u'(x_i) = \sum_{j=0}^N u(x_j) h'_j(x_i) = \sum_{j=0}^N \mathcal{D}_{ij}^N u(x_j) \quad (5.8)$$

où les coefficients de la matrice de dérivation sont donnés par  $\mathcal{D}_{ij}^N = h'_j(x_i)$ . De manière plus précise ces coefficients sont égaux à :

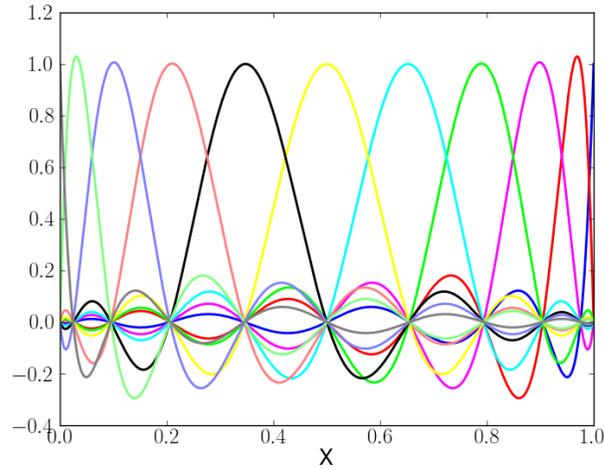


FIG. 5.1 – Fonction d'interpolation dans dans  $\mathbb{P}_N$ .

$$\mathcal{D}_{ij}^N = \begin{cases} -\frac{(1) \bar{c}_i}{2} \frac{(-1)^{i+j}}{\bar{c}_j \sin(\frac{\pi(i+j)}{2N}) \sin(\frac{\pi(i-j)}{2N})} & i \neq j \\ -\frac{\cos \frac{\pi j}{N}}{2 \sin^2 \frac{\pi i}{N}} & 1 \leq i = j \leq N - 1 \\ \frac{2N^2 + 1}{6} & i = j = 0 \\ -\frac{2N^2 + 1}{6} & i = j = N \end{cases} \quad (5.9)$$

où les formules trigonométriques suivantes ont été utilisées :

$$\begin{cases} x_i - x_j = -2 \sin(\frac{\pi(i+j)}{2N}) \sin(\frac{\pi(i-j)}{2N}) \\ 1 - x_i^2 = -\sin^2 \frac{\pi i}{N} \end{cases} \quad (5.10)$$

**Remarque**

La matrice de dérivation  $\mathcal{D}^N$  n'est valable que pour  $x_i \in ]-1, 1[$ . Si l'on est sur un intervalle du type  $X \in ]a, b[$  il faut utiliser la transformation affine suivante :

$$\begin{cases} X_i = \alpha x_i + \beta & \text{pour } i = 0, \dots, N \\ \mathcal{D}_X^N = \alpha \mathcal{D}^N \end{cases} \quad (5.11)$$

où  $\alpha$  et  $\beta$  sont les coefficients de la transformation de  $x_i$  en  $X_i$  définis par

$$\alpha = \frac{b-a}{2} \quad \beta = \frac{b+a}{2} \quad (5.12)$$

### I.3 Expression de l'opérateur de dérivée première dans $\mathbb{P}_{N-2}$

L'espace vectoriel  $\mathbb{P}_{N-2}(\Lambda)$  est muni de la base de dimension  $N-1$   $\{\tilde{h}_i; i = 1, \dots, N-1\}$  où  $\tilde{h}_i$  est le polynôme d'interpolation de Lagrange de degré  $N-2$  associé au point de collocation  $x_i$ . Ainsi :

$$\tilde{h}_i(x_j) = \delta_{ij} \quad \forall j = 1, \dots, N-1 \quad (5.13)$$

où le polynôme d'interpolation de Lagrange s'écrit :

$$\tilde{h}_i(x) = \prod_{j=1, j \neq i}^{N-1} \frac{x - x_j}{x_i - x_j} = \frac{1 - x_i^2}{1 - x^2} h_i(x) \quad \forall i = 1, \dots, N-1 \quad (5.14)$$

Ainsi, tout élément de  $\mathbb{P}_{N-2}(\Lambda)$  s'écrit

$$p(x) = \sum_{j=1}^{N-1} p(x_j) \tilde{h}_j(x) \quad (5.15)$$

et sa dérivée première aux points de collocations  $x_i$  internes du domaine a pour expression :

$$p'(x_i) = \sum_{j=1}^{N-1} p(x_j) \tilde{h}'_j(x_i) \quad (5.16)$$

avec

$$\tilde{h}'_j(x_i) = \frac{1 - x_j^2}{1 - x_i^2} h'_j(x_i) + \frac{2x_i}{1 - x_i^2} \delta_{ij} \quad \forall i = 1, \dots, N-1 \quad (5.17)$$

On considère dans la suite que  $h'_j(x_i)$  correspond à la  $(i, j)^{eme}$  coordonnée de la matrice  $\tilde{\mathcal{D}}^N$ . Ainsi

$$\tilde{\mathcal{D}}_{ij}^N = \tilde{h}'_j(x_i) \quad (5.18)$$

et la dérivée d'un élément de  $\mathbb{P}_{N-2}(\Lambda)$  s'écrit

$$p'(x_i) = \sum_{j=1}^{N-1} \tilde{\mathcal{D}}_{ij}^N p(x_j) \quad (5.19)$$

En se servant des coefficients de la dérivée première dans  $\mathbb{P}_{N-2}(\Lambda)$  détaillée en (5.9), on obtient l'expression des coefficients de  $\tilde{\mathcal{D}}_{ij}^N$  :

$$\tilde{\mathcal{D}}_{ij}^N = \begin{cases} -\frac{1 \sin^2 \frac{\pi j}{N}}{2 \sin^2 \frac{\pi i}{N}} \frac{(-1)^{i+j}}{\sin(\frac{\pi(i+j)}{2N}) \sin(\frac{\pi(i-j)}{2N})} & i \neq j \\ \frac{3 \cos \frac{\pi j}{N}}{2 \sin^2 \frac{\pi i}{N}} & 1 \leq i = j \leq N - 1 \end{cases} \quad (5.20)$$

### Remarque

De la même manière que dans l'espace  $\mathbb{P}_{\mathbf{N}}$ , la matrice dérivation pour  $X \in ]a, b[$  sera pondérée par le coefficient de transformation affine  $\alpha$  défini dans la remarque précédente

$$\tilde{\mathcal{D}}_X^N = \alpha \tilde{\mathcal{D}}^N \quad (5.21)$$

## I.4 Expression des opérateurs bidimensionnels

Afin de donner l'expression des opérateurs en deux dimensions dans le domaine  $\Omega = ]-1, 1[ \times ]-1, 1[$  (en pratique il suffit de multiplier les matrices de dérivation de chacun des espaces par le coefficient  $\alpha$  de la transformation affine associée pour retrouver leurs expressions sur un intervalle quelconque  $\Omega_q = ]a, b[ \times ]c, d[$ ), nous allons utiliser les opérateurs unidimensionnels définis sur  $\Lambda = ]-1, 1[$ . Pour ce faire, définissons l'espace vectoriel des polynômes définis sur  $\Omega$  et de degré au plus  $N$  dans la direction  $x$ , et de degré au plus  $M$  dans la direction  $y$  de la manière suivante :

$$\mathbb{P}_{\mathbf{N},\mathbf{M}}(\Omega) = \mathbb{P}_{\mathbf{N}}(\Lambda) \otimes \mathbb{P}_{\mathbf{M}}(\Lambda) \quad (5.22)$$

Dans la suite nous noterons  $u_N$  un élément de  $\mathbb{P}_{\mathbf{N},\mathbf{M}}(\Omega)$ . De la même manière nous pouvons définir l'espace vectoriel des polynômes définis sur  $\Omega$  et de degré au plus  $N-2$  dans la direction  $x$ , et de degré au plus  $M-2$  dans la direction  $y$  :

$$\mathbb{P}_{\mathbf{N}-2,\mathbf{M}-2}(\Omega) = \mathbb{P}_{\mathbf{N}-2}(\Lambda) \otimes \mathbb{P}_{\mathbf{M}-2}(\Lambda) \quad (5.23)$$

L'espace vectoriel  $\mathbb{P}_{\mathbf{N},\mathbf{M}}(\Omega)$ , de dimension  $(N+1) \times (M+1)$  est muni de la base  $\{(h_i, f_j); i = 0, \dots, N; j = 0, \dots, M\}$  où  $h_i$  est le polynôme d'interpolation de Lagrange défini dans (5.4), et  $f_j$  est le polynôme de Lagrange caractéristique du point de collocation  $y_j$  de telle sorte que  $f_j(y_i) = \delta_{ij}$  et :

$$f_j(y) = \prod_{i=0, i \neq j}^M \frac{y - y_i}{y_j - y_i} = \frac{(-1)^{i+1} (1 - y^2) T'_M(y)}{\bar{c}_i M^2 (y - y_j)} \quad \forall j = 0, \dots, M \quad (5.24)$$

De cette manière un élément  $u_N$  de  $\mathbb{P}_{\mathbf{N},\mathbf{M}}(\Omega)$  au point de collocation  $(x_i, y_j)$  s'écrira

$$u_N(x_i, y_j) = \sum_{k=0}^N \sum_{l=0}^M u_N(x_k, y_l) h_k(x_i) f_l(y_j) \quad (5.25)$$

On peut donc définir les dérivées premières du polynôme  $u_N$  de la manière suivante<sup>2</sup> :

$$\frac{\partial u_N}{\partial x}(x_i, y_j) = \sum_{k=0}^N \sum_{l=0}^M u_N(x_k, y_l) h'_k(x_i) f_l(y_j) = \sum_{k=0}^N \mathcal{D}_{ik}^N u_N(x_k, y_j) \quad (5.26)$$

et

$$\frac{\partial u_N}{\partial y}(x_i, y_j) = \sum_{k=0}^N \sum_{l=0}^M u_N(x_k, y_l) h_k(x_i) f'_l(y_j) = \sum_{l=0}^M \mathcal{D}_{jl}^M u_N(x_i, y_l) \quad (5.27)$$

où les coefficients de la matrice  $\mathcal{D}_{ik}^N$  sont définis en (5.9), et les coefficients de la matrices de dérivation dans la direction  $y$ ,  $\mathcal{D}_{jl}^M$ , sont construits de la même manière que ce qui à été fait dans la partie I.2. Les dérivées seconde au même point s'expriment de la manière suivante :

$$\frac{\partial^2 u_N}{\partial x^2}(x_i, y_j) = \sum_{k=0}^N \sum_{l=0}^M u_N(x_k, y_l) h''_k(x_i) f_l(y_j) = \sum_{k=0}^N \mathcal{D}_{ik}^{N,2} u_N(x_k, y_j) \quad (5.28)$$

et

$$\frac{\partial^2 u_N}{\partial y^2}(x_i, y_j) = \sum_{k=0}^N \sum_{l=0}^M u_N(x_k, y_l) h_k(x_i) f''_l(y_j) = \sum_{l=0}^M \mathcal{D}_{jl}^{M,2} u_N(x_i, y_l) \quad (5.29)$$

où les coefficients des matrices de dérivation seconde dans chacune des directions s'écrivent

$$\mathcal{D}_{ij}^{N,2} = \sum_{m=0}^N \mathcal{D}_{im}^N \mathcal{D}_{mj}^N \quad \mathcal{D}_{ij}^{M,2} = \sum_{m=0}^M \mathcal{D}_{im}^M \mathcal{D}_{mj}^M \quad (5.30)$$

## II Méthode $\mathbb{P}_N - \mathbb{P}_{N-2}$ associée au schéma de prédiction de Van-Kan

### II.1 Rappel de l'algorithme de Van-Kan

On rappelle que l'algorithme de Van-Kan est divisé en deux étapes :

- L' étape de prédiction :

$$\begin{cases} \frac{\tilde{\mathbf{u}}^{k+1}}{\delta t} - \frac{1}{2Re} \nabla^2 \tilde{\mathbf{u}}^{k+1} = \hat{\mathbf{f}}^{k+1} \\ \tilde{\mathbf{u}}^{k+1}|_{\Gamma} = \mathbf{g}_D \end{cases} \quad (5.31)$$

où

$$\hat{\mathbf{f}}^{k+1} = \frac{\mathbf{u}^k}{\delta t} + \frac{1}{2Re} \nabla^2 \mathbf{u}^k - \nabla p^k - \frac{3}{2} (\mathbf{u}^k \cdot \nabla) \mathbf{u}^k + \frac{1}{2} (\mathbf{u}^{k-1} \cdot \nabla) \mathbf{u}^{k-1} + (\mathbf{f}^{k+1}) \quad (5.32)$$

---

<sup>2</sup>en tenant compte du fait que  $f_l(y_j) = \delta_{lj}$  et  $h_k(x_i) = \delta_{ki}$

- L'étape de projection :

$$\begin{cases} \frac{1}{\delta t}(\mathbf{u}^{k+1} - \tilde{\mathbf{u}}^{k+1}) + \nabla p^{k+1} = 0 \\ \nabla \cdot \mathbf{u}^{k+1} = 0 \\ \mathbf{u}^{k+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (5.33)$$

où  $\delta t$  est le pas de temps de la discrétisation et l'indice  $k$  indique que les fonctions sont évaluées au temps  $t^k = k\delta t$  pour  $k = 0, 1, \dots$

Par la suite dans un souci de clarté, les indices temporels seront omis. Posons de plus,

$$\tilde{\mathbf{u}}(x, y) = (\tilde{u}(x, y), \tilde{v}(x, y)) \quad \hat{\mathbf{f}}(x, y) = (\hat{f}_x(x, y), \hat{f}_y(x, y)). \quad (5.34)$$

## II .2 Etape de prédiction

Le problème (5.31) peut donc être décomposé en deux équations de diffusion instationnaire avec conditions limites de type Dirichlet pour chacune des composantes de la vitesse intermédiaire :

$$\begin{cases} \tilde{u} - \nu \delta t \left( \frac{\partial^2 \tilde{u}}{\partial x^2} + \frac{\partial^2 \tilde{u}}{\partial y^2} \right) = \hat{f}_x \\ \tilde{u}|_{\Gamma} = 0 \end{cases} \quad (5.35)$$

et

$$\begin{cases} \tilde{v} - \nu \delta t \left( \frac{\partial^2 \tilde{v}}{\partial x^2} + \frac{\partial^2 \tilde{v}}{\partial y^2} \right) = \hat{f}_y \\ \tilde{v}|_{\Gamma} = 0 \end{cases} \quad (5.36)$$

Introduisons l'espace polynomial d'approximation de la vitesse :

$$X_{N,M}(\Omega) = \{u_N \in \mathbb{P}_{N,M}(\Omega) / \forall (x_c, y_c) \in \Gamma_c, u(x_c, y_c) = 0\}$$

L'équation de quantité de mouvement pour la variable  $\tilde{\mathbf{u}}_N$  en projection selon  $\mathbf{x}$  s'écrit :

Trouver  $\tilde{u}_N \in X_{N,M}$  tel que, pour  $i = 1, \dots, N-1$ , pour  $j = 1, \dots, M-1$  :

$$\frac{\tilde{u}_N(x_i, y_j)}{\delta t} - \frac{1}{2Re} \left( \frac{\partial^2 \tilde{u}_N}{\partial x^2}(x_i, y_j) + \frac{\partial^2 \tilde{u}_N}{\partial y^2}(x_i, y_j) \right) = \hat{f}_x(x_i, y_j) \quad (5.37)$$

Ce problème peut être réécrit en utilisant la discrétisation spectrale des opérateurs définis précédemment. On obtient donc le système linéaire suivant à résoudre :

$$\frac{\tilde{u}_N(x_i, y_j)}{\delta t} - \frac{1}{2Re} \left( \sum_{k=1}^{N-1} \mathcal{D}_{ik}^{N,2} \tilde{u}_N(x_k, y_j) + \sum_{l=1}^{M-1} \mathcal{D}_{jl}^{M,2} \tilde{u}_N(x_i, y_l) \right) = \hat{f}_x(x_i, y_j) \quad (5.38)$$

$$+\nu \left( \mathcal{D}_{i0}^{N,2} \tilde{u}_N(x_0, y_j) + \mathcal{D}_{iN}^{N,2} \tilde{u}_N(x_N, y_j) + \mathcal{D}_{j0}^{M,2} \tilde{u}_N(x_i, y_0) + \mathcal{D}_{jM}^{M,2} \tilde{u}_N(x_i, y_M) \right)$$

De même, l'équation de quantité de mouvement pour la variable  $\tilde{\mathbf{u}}_N$  en projection selon  $\mathbf{y}$  s'écrit :

Trouver  $\tilde{v}_N \in X_{N,M}$  tel que, pour  $i = 1, \dots, N-1$ , pour  $j = 1, \dots, M-1$  :

$$\frac{\tilde{v}_N(x_i, y_j)}{\delta t} - \frac{1}{2Re} \left( \frac{\partial^2 \tilde{v}_N}{\partial x^2}(x_i, y_j) + \frac{\partial^2 \tilde{v}_N}{\partial y^2}(x_i, y_j) \right) = \hat{f}_y(x_i, y_j) \quad (5.39)$$

qui peut être réécrit

$$\frac{\tilde{v}_N(x_i, y_j)}{\delta t} - \frac{1}{2Re} \left( \sum_{k=1}^{N-1} \mathcal{D}_{ik}^{N,2} \tilde{v}_N(x_k, y_j) + \sum_{l=1}^{M-1} \mathcal{D}_{jl}^{M,2} \tilde{v}_N(x_i, y_l) \right) = \hat{f}_y(x_i, y_j) \quad (5.40)$$

$$+\nu \left( \mathcal{D}_{i0}^{N,2} \tilde{v}_N(x_0, y_j) + \mathcal{D}_{iN}^{N,2} \tilde{v}_N(x_N, y_j) + \mathcal{D}_{j0}^{M,2} \tilde{v}_N(x_i, y_0) + \mathcal{D}_{jM}^{M,2} \tilde{v}_N(x_i, y_M) \right)$$

où les termes impliquant les conditions limites ont été passés dans le second membre.

## II.3 Etape de correction

### a) Discrétisation

En utilisant les notations définies dans la partie précédente, l'étape de projection peut s'écrire, pour chacune des composantes, de la manière suivante :

$$u + \delta t \frac{\partial p}{\partial x} = \tilde{u} \quad (5.41)$$

$$v + \delta t \frac{\partial p}{\partial y} = \tilde{v} \quad (5.42)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (5.43)$$

Ce système est muni de conditions limites portant sur la vitesse normale aux frontière du domaine, soit

$$\mathbf{u} \cdot \mathbf{n}|_{\Gamma} = 0 \quad (5.44)$$

Le système composé des équations (5.41), (5.42) et (5.43) sera discrétisé en utilisant une approximation spectrale  $\mathbb{P}_{\mathbf{N}} - \mathbb{P}_{\mathbf{N}-2}$ . La pression étant définie sur les points intérieurs au domaine, la contrainte de continuité ne sera vérifiée de façon forte qu'en ces points. Une seconde

approximation introduite par Azaïez et al. dans [Azaïez et al., 1994] consiste à satisfaire la contrainte de continuité de façon faible sur tous les points du domaine, y compris sur les frontières.

Les équations (5.41), (5.42) et (5.43) sont discrétisées de la manière suivante :

$$u_N(x_i, y_j) + \delta t \frac{\partial p_N}{\partial x}(x_i, y_j) = \tilde{u}_N(x_i, y_j) \quad (5.45)$$

$$v_N(x_i, y_j) + \delta t \frac{\partial p_N}{\partial y}(x_i, y_j) = \tilde{v}_N(x_i, y_j) \quad (5.46)$$

$$\frac{\partial u_N}{\partial x}(x_i, y_j) + \frac{\partial v_N}{\partial y}(x_i, y_j) = 0 \quad (5.47)$$

auxquels il faut ajouter les conditions aux limites suivantes :

$$u_N(x_0, y_j) = u_N(x_N, y_j) = 0 \quad \text{pour } j = 1, \dots, M-1 \quad (5.48)$$

$$u_N(x_i, y_0) = u_N(x_i, y_M) = 0 \quad \text{pour } i = 1, \dots, N-1 \quad (5.49)$$

Le problème peut être réécrit en utilisant les opérateurs de dérivation spectraux définis dans la partie I.3. Les équations (5.45), (5.46) et (5.47) s'écrivent donc pour  $i = 1, \dots, N-1$  et  $j = 1, \dots, M-1$

$$u_N(x_i, y_j) + \delta t \sum_{k=1}^{N-1} \tilde{\mathcal{D}}_{ik}^N p_N(x_k, y_j) = \tilde{u}_N(x_i, y_j) \quad (5.50)$$

$$v_N(x_i, y_j) + \delta t \sum_{l=1}^{M-1} \tilde{\mathcal{D}}_{jl}^M p_N(x_i, y_l) = \tilde{v}_N(x_i, y_j) \quad (5.51)$$

$$\sum_{k=1}^{N-1} \mathcal{D}_{ik}^N u_N(x_k, y_j) + \sum_{l=1}^{M-1} \mathcal{D}_{jl}^M v_N(x_i, y_l) = \mathcal{S}_{ij} \quad (5.52)$$

avec

$$\mathcal{S}_{ij} = -\mathcal{D}_{i0}^N u_N(x_0, y_j) - \mathcal{D}_{iN}^N u_N(x_N, y_j) - \mathcal{D}_{j0}^M v_N(x_i, y_0) - \mathcal{D}_{jM}^M v_N(x_i, y_M) \quad (5.53)$$

### b) Algorithme d'Uzawa

Le système d'équations (5.50), (5.51) et (5.52) s'écrit sous forme matricielle :

$$U + \delta t \tilde{D}^N P = \tilde{U} \quad (5.54)$$

$$V + \delta t P {}^t \tilde{D}^M = \tilde{V} \quad (5.55)$$

$$D^N U + V {}^t D^M = S \quad (5.56)$$

où

- $U$  (resp  $\tilde{U}$ ), est la matrice contenant les valeurs de la première composante de vitesse (resp de vitesse intermédiaire) aux points de collocation interne :

$$U = [u_N(x_i, y_j)]_{1 \leq i \leq N-1; 1 \leq j \leq M-1} \quad \tilde{U} = [\tilde{u}_N(x_i, y_j)]_{1 \leq i \leq N-1; 1 \leq j \leq M-1}$$

- $V$  (resp  $\tilde{V}$ ), est la matrice contenant les valeurs de la seconde composante de vitesse (resp de vitesse intermédiaire) aux points de collocation interne :

$$V = [v_N(x_i, y_j)]_{1 \leq i \leq N-1; 1 \leq j \leq M-1} \quad \tilde{V} = [\tilde{v}_N(x_i, y_j)]_{1 \leq i \leq N-1; 1 \leq j \leq M-1}$$

- $P$ , la matrice contenant les valeurs de la pression aux points de collocation interne :

$$P = [p_N(x_i, y_j)]_{1 \leq i \leq N-1; 1 \leq j \leq M-1},$$

- $D^N$  (resp  $\tilde{D}^N$ ), la matrice de dérivée première en vitesse (resp en pression) dans la direction  $x$  :

$$D^N = [\mathcal{D}_{ij}^N]_{1 \leq i \leq N-1; 1 \leq j \leq N-1} \quad \tilde{D}^N = [\tilde{\mathcal{D}}_{ij}^N]_{1 \leq i \leq N-1; 1 \leq j \leq N-1}$$

- $D^M$  (resp  $\tilde{D}^M$ ), la matrice de dérivée première en vitesse (resp en pression) dans la direction  $y$  :

$$D^M = [\mathcal{D}_{ij}^M]_{1 \leq i \leq M-1; 1 \leq j \leq M-1} \quad \tilde{D}^M = [\tilde{\mathcal{D}}_{ij}^M]_{1 \leq i \leq M-1; 1 \leq j \leq M-1}$$

- $S$ , la matrice telle que :

$$S = [\mathcal{S}_{ij}]_{1 \leq i \leq N-1; 1 \leq j \leq M-1},$$

Le système composé des équations (5.54), (5.55) et (5.56) est découplé en utilisant un algorithme de type Uzawa : L'équation (5.54) s'écrit :

$$U = \tilde{U} - \delta t \tilde{D}^N P \quad (5.57)$$

De même l'équation (5.55) s'écrit :

$$V = \tilde{V} - \delta t P {}^t \tilde{D}^M \quad (5.58)$$

En injectant ces formes de  $U$  et  $V$  dans l'équation (5.56), on obtient une équation de type Poisson également appelée "pseudo-Poisson" pour la pression :

$$\left( D^N \tilde{D}^N \right) P + P {}^t \left( D^M \tilde{D}^M \right) = \frac{1}{\delta t} \left( D^N \tilde{U} + \tilde{V} {}^t D^M - S \right) \quad (5.59)$$

Une fois la pression calculée, les champs de vitesses sont donnés par les équations (5.57) et (5.58). L'opérateur en pression  $D^N \tilde{D}^N$  est appelé opérateur d'Uzawa. Il est composé du produit d'un opérateur de dérivation dans  $\mathbb{P}_{N-2}$  avec un opérateur de dérivation de  $\mathbb{P}_N$  muni de conditions de Dirichlet. Ce n'est pas l'opérateur usuel de dérivée seconde muni de conditions de Neumann. En effet lors d'une résolution classique la résolution du problème en pression revient à résoudre une équation de poisson avec des conditions aux limites de type Neumann.

Or cette condition est artificielle car la pression physique ne vérifie pas nécessairement cette condition [Temam, 1977, Temam, 1991]. En conséquence la formulation classique détruit la précision en temps de la pression et affecte la stabilité des schémas. De plus l'équation de continuité n'est pas satisfaite sur les bords du domaine. Dans le cas présent, les conditions aux limites sur la pression ont disparues du fait de la discrétisation de la pression sur un espace de dimension  $(N - 1) \times (M - 1)$  alors que la vitesse est discrétisée sur un espace de dimension  $(N + 1) \times (M + 1)$ .

### III Formulation PGD du problème

Dans cette partie les différentes étapes décrites précédemment vont être reprise de manière algébrique afin de construire les systèmes linéaires utiles à la résolution des équations de Navier-Stokes à l'aide d'une formulation spectrale  $\mathbb{P}_N - \mathbb{P}_{N-2}$  associée à un schéma de prédiction-correction.

#### III .1 Etape de prédiction

Le système d'équations (5.35) et (5.36) peut s'écrire sous la forme tensorielle suivante,

$$\begin{aligned} [\mathbb{I}_x \otimes \mathbb{I}_y - \nu \delta t (D^{N,2} \otimes \mathbb{I}_y + \mathbb{I}_x \otimes D^{M,2})] \tilde{u} &= \hat{f}_x \\ [\mathbb{I}_x \otimes \mathbb{I}_y - \nu \delta t (D^{N,2} \otimes \mathbb{I}_y + \mathbb{I}_x \otimes D^{M,2})] \tilde{v} &= \hat{f}_y \end{aligned} \quad (5.60)$$

où on a noté

- $D^{N,2}$ , la matrice de dérivée seconde en vitesse dans la direction x de dimension  $(N - 1) \times (N - 1)$  :

$$D^{N,2} = \left[ \mathcal{D}_{ij}^{N,2} \right]_{1 \leq i \leq N-1; 1 \leq j \leq N-1},$$

- $D^{M,2}$ , la matrice de dérivée seconde en vitesse dans la direction y de dimension  $(M - 1) \times (M - 1)$  :

$$D^{M,2} = \left[ \mathcal{D}_{ij}^{M,2} \right]_{1 \leq i \leq M-1; 1 \leq j \leq M-1},$$

- $\mathbb{I}_x$ , la matrice identité de dimension  $(N - 1) \times (N - 1)$
- $\mathbb{I}_y$ , la matrice identité de dimension  $(M - 1) \times (M - 1)$

Les systèmes linéaires (5.60) peuvent être résolus par une méthode classique de résolution type gradient-conjugué ou bi-gradient-conjugué. Il est important de noter que quelques sous matrices de l'opérateur permettant de calculer  $\tilde{u}$  et  $\tilde{v}$  sont pleines (en particulier les matrices de dérivations), ce qui n'est pas le cas en volumes finis. En effet les matrices 1D  $D^{N,2}$  et  $D^{M,2}$  sont totalement pleines et non pas tridiagonales comme leurs équivalentes en volumes finis. En revanche, la taille de ces matrices peut être moindre pour une même qualité de représentation. L'intérêt de la formulation en produit tensoriel dans la méthode PGD sur ce type de problème est de ne résoudre que des problèmes 1D sur des matrices de tailles réduites plutôt que le problème standard directement.

Dans ce cas la PGD consistera à résoudre les équations (5.35) et (5.36) en recherchant les solutions sous forme séparée. Ici encore nous nous limiterons à une décomposition uniquement spatiale. L'équation (5.60) a exactement la forme requise pour être traitée par méthode PGD. En effet le fait d'avoir réécrit l'opérateur de l'étape de prédiction comme somme de produits tensoriels d'opérateurs de chacune des directions permet d'y appliquer la PGD aisément. La résolution de (5.60) par PGD consiste donc à rechercher les solutions  $\tilde{u}_N(x_i, y_j)$  et  $\tilde{v}_N(x_i, y_j)$  sous la forme suivante :

$$\tilde{u}_N(x_i, y_j) = \sum_{l=1}^n \alpha_{\tilde{u}}^l \mathbf{F}_{\tilde{u}}^l(x_i) \mathbf{G}_{\tilde{u}}^l(y_j) \quad \tilde{v}_N(x_i, y_j) = \sum_{l=1}^n \alpha_{\tilde{v}}^l \mathbf{F}_{\tilde{v}}^l(x_i) \mathbf{G}_{\tilde{v}}^l(y_j) \quad (5.61)$$

En introduisant cette forme de la solution dans (5.60), et en tenant compte des propriétés de la représentation tensorielle, on obtient pour  $\tilde{u}$  :

$$\sum_{k=1}^3 \sum_{l=1}^n \alpha_{\tilde{u}}^l \left( A_x^k \mathbf{F}_{\tilde{u}}^l \otimes A_y^k \mathbf{G}_{\tilde{u}}^l \right) = \sum_{l=1}^m \mathbf{f}_x^l \otimes \mathbf{f}_y^l d\Omega \quad (5.62)$$

où les opérateurs  $A_x^k$  (respectivement  $A_y^k$ ) sont des matrices carrées de dimension  $(N-1) \times (N-1)$  (respectivement  $(M-1) \times (M-1)$ ) et correspondent à :

$$\begin{aligned} A_x^1 &= \mathbb{I}_x & A_x^2 &= D^{N,2} & A_x^3 &= -\nu \delta t \mathbb{I}_x \\ A_y^1 &= \mathbb{I}_y & A_y^2 &= -\nu \delta t \mathbb{I}_y & A_y^3 &= D^{M,2} \end{aligned} \quad (5.63)$$

Les seconds membres  $\mathbf{f}_x^l$  et  $\mathbf{f}_y^l$  sont construits à partir du second membre général de l'équation de Navier-Stokes défini par l'équation (5.32) en tenant compte du fait que l'on dispose des champs de vitesse et de pression sous forme séparée aux pas de temps précédents.

Une fois les opérateurs de la PGD identifiés la méthode se décompose exactement de la même manière que dans ce qui a été énoncé . La résolution PGD de  $\tilde{v}$  est effectuée avec les mêmes opérateurs que ceux définis pour  $\tilde{u}$ , seul les seconds membres varieront.

### III .2 Etape de correction

Le système d'équations (5.54),(5.55) et (5.56) peut être réécrit de la manière suivante :

$$u_k + \delta t \left( \tilde{D}^N \otimes \mathbb{I}_y \right) p_k = \tilde{u}_k \quad (5.64)$$

$$v_k + \delta t \left( \mathbb{I}_x \otimes \tilde{D}^M \right) p_k = \tilde{v}_k \quad (5.65)$$

$$\left( D^N \otimes \mathbb{I}_y \right) u_k + \left( \mathbb{I}_x \otimes D^M \right) v_k = 0 \quad (5.66)$$

L'algorithme d'Uzawa consiste donc à extraire les vitesses  $u_k$  et  $v_k$  des équations (5.64) et (5.65) afin de les réinjecter dans l'équation (5.66), cela donne

$$(D^N \otimes \mathbb{I}_y) \left[ \tilde{u}_k - \delta t \left( \tilde{D}^N \otimes \mathbb{I}_y \right) p_k \right] + (\mathbb{I}_x \otimes D^M) \left[ \tilde{v}_k - \delta t \left( \mathbb{I}_x \otimes \tilde{D}^M \right) p_k \right] = 0 \quad (5.67)$$

ou encore

$$\left[ (D^N \otimes \mathbb{I}_y) \left( \tilde{D}^N \otimes \mathbb{I}_y \right) + (\mathbb{I}_x \otimes D^M) \left( \mathbb{I}_x \otimes \tilde{D}^M \right) \right] p_k = \frac{1}{\delta t} \left[ (D^N \otimes \mathbb{I}_y) \tilde{u}_k + (\mathbb{I}_x \otimes D^M) \tilde{v}_k \right] \quad (5.68)$$

enfin, en tenant compte des propriétés du produit tensoriel, on obtient

$$\left[ \left( D^N \tilde{D}^N \otimes \mathbb{I}_y \right) + \left( \mathbb{I}_x \otimes D^M \tilde{D}^M \right) \right] p_k = \frac{1}{\delta t} \left[ (D^N \otimes \mathbb{I}_y) \tilde{u}_k + (\mathbb{I}_x \otimes D^M) \tilde{v}_k \right] \quad (5.69)$$

Ici la PGD consiste à résoudre l'équation (5.69) en recherchant la solution sous forme de séparation de variable. Cette équation a exactement la forme requise pour être traitée par méthode PGD. La solution  $p_N(x_i, y_j)$  est donc recherchée sous la forme :

$$p_N(x_i, y_j) = \sum_{l=1}^n \alpha_p^l \mathbf{F}_p^l(x_i) \mathbf{G}_p^l(y_j) \quad (5.70)$$

Le problème à résoudre peut donc être réécrit

$$\sum_{k=1}^2 \sum_{l=1}^n \alpha_p^l \left( A_x^k \mathbf{F}_p^l \otimes A_y^k \mathbf{G}_p^l \right) = \sum_{l=1}^m \mathbf{f}_x^l \otimes \mathbf{f}_y^l d\Omega \quad (5.71)$$

où les opérateurs  $A_x^k$  (respectivement  $A_y^k$ ) différents de ceux définis dans la section précédente pour l'étape de prédiction, sont des matrices carrées de dimension  $(N-1) \times (N-1)$  (respectivement  $(M-1) \times (M-1)$ ) définis par :

$$\begin{aligned} A_x^1 &= D^N \tilde{D}^N & A_x^2 &= \mathbb{I}_x \\ A_y^1 &= \mathbb{I}_y & A_y^2 &= D^M \tilde{D}^M \end{aligned} \quad (5.72)$$

Une fois les différents opérateurs identifiés, la méthode PGD peut être appliquée afin de calculer la pression à l'intérieur du domaine. Par suite les vitesses  $u_N$  et  $v_N$  sous forme séparée sont corrigés en appliquant les équations (5.64) et (5.65).

## IV Résultats

Dans cette partie, plusieurs problèmes vont être résolus en utilisant la méthode PGD associée à une discrétisation spectrale type Tchebychev. Tout d'abord le cas de l'équation de diffusion stationnaire sera traité. Nous nous intéresserons ensuite à la résolution d'un problème de Darcy par un schéma d'Uzawa qui correspond à l'étape de correction de l'algorithme de Van-Kahn. Enfin les équations de Navier-Stokes seront traitées à travers le problème de 'Taylor-Green' et de la cavité entraînée 2D.

### IV .1 Equation de diffusion

Les opérateurs PGD de cette équation peuvent être construits de manière très simple en se basant sur ce qui à été fait dans le cas d'une équation de diffusion instationnaire (voir II .2).

Les deux problèmes traités dans cette partie sont les mêmes que ceux du chapitre 2 section I .2. Soit le problème 1 suivant,

Chercher  $T^1(x, y) \in \Omega = ]0, 1[ \times ]0, 1[$  tel que

$$\nabla^2 T^1(x, y) = -8\pi^2 \cos(2\pi x) \sin(2\pi y) \quad (5.73)$$

et le problème 2 suivant,

Chercher  $T^2(x, y) \in \Omega = ]0, 1[ \times ]0, 1[$  tel que

$$\nabla^2 T^2(x, y) = 12(x^2 y^4 + x^4 y^2) - 8\pi^2 \cos(2\pi x) \sin(2\pi y) \quad (5.74)$$

Les solutions analytiques de ces problèmes sont données par :

$$T_{ana}^1(x, y) = \cos(2\pi x) \sin(2\pi y) \quad (5.75)$$

$$T_{ana}^2(x, y) = x^4 y^4 + \cos(2\pi x) \sin(2\pi y)$$

Dans les deux cas, les conditions aux limites ont été choisies de type Dirichlet et seront imposées de manière à satisfaire la solution analytique.

Définissons l'erreur entre la solution exacte et la solution calculée numériquement de la manière suivante :

$$E(|\Phi_N - \Phi_{ana}|) = \frac{\|\Phi_N - \Phi_{ana}\|_{L^2(\Omega)}^N}{\|\Phi_N\|_{L^2(\Omega)}^N} \quad (5.76)$$

où

$$\|\Phi_N\|_{L^2(\Omega)}^N = \frac{1}{N-1} \left( \sum_{i,j=1}^{N-1} \Phi_N(x_i, y_j)^2 \right)^{\frac{1}{2}} \quad (5.77)$$

Dans ce cas  $\Phi$  représente  $T^1$  ou  $T^2$ . La figure 5.2 indique les erreurs commises pour le problème 1 et pour le problème 2 en échelle 'log-log' pour chacun des solveurs standards et PGD. On constate que le solveur PGD est aussi précis que le solveur standard et qu'à partir de 20 nœuds dans chaque direction, l'erreur est proche du zéros machine. De plus on a bien une convergence exponentielle en fonction du nombre de nœuds, ce qui est caractéristique des discrétisations spectrales.

Nous n'avons pas indiqué les temps de calcul nécessaires à la résolution de ces problèmes car la différence entre la résolution PGD et à l'aide d'un solveur standard est trop importante. En effet pour le problème 1, avec un maillage de taille  $64 \times 64$  il faut 300 secondes au solveur

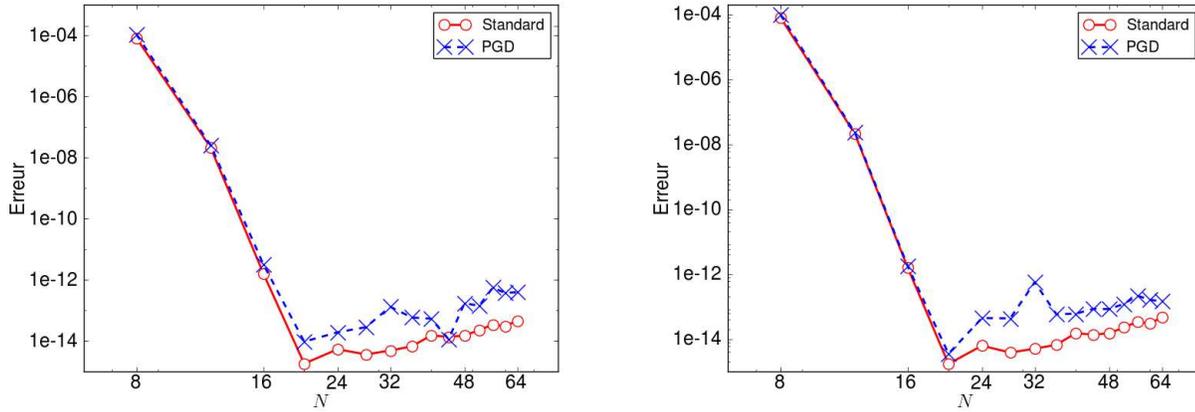


FIG. 5.2 – Erreur pour le problème 1 (gauche) et pour le problème 2 (droite) par rapport à la solution analytique en fonction du nombre de nœuds dans chaque direction.

standard pour converger alors que la PGD converge en moins d'un dixième de seconde. En effet pour la résolution du problème 1, une seule fonction PGD suffit pour avoir une erreur équivalente à celle du solveur standard. Pour le problème 2, la PGD nécessite une cinquantaine de fonctions pour converger mais là encore les gains en temps de calculs sont très importants. Dans ce cas, avec 64 nœuds dans chaque directions le modèle standard nécessite également 300 secondes pour converger, alors que la PGD converge en moins de 3 secondes.

## IV .2 Problème correspondant à l'étape de correction

Ce problème a été discrétisé à l'aide d'un schéma de projection de Tchebychev basé sur une discrétisation  $\mathbb{P}_N - \mathbb{P}_{N-2}$ . Dans cette partie, nous testerons la résolution du problème associé à l'étape de correction<sup>3</sup> du schéma de Van-Kan, à savoir les équations (5.64), (5.65) et (5.66). Nous allons traiter un problème identique à celui traité par Botella dans [Botella, 1996], où les solutions analytiques suivantes sont données par :

$$u_{ana}(x, y) = -4\pi \sin(4\pi x) \cos(4\pi y),$$

$$v_{ana}(x, y) = 4\pi \cos(4\pi x) \sin(4\pi y),$$

$$p_{ana}(x, y) = \cos(4\pi x) \cos(4\pi y)$$

De cette manière le terme source ainsi que les conditions aux limites sont parfaitement connues. On a donc :

$$\tilde{u}_N = u_N + \frac{\partial p}{\partial x} = -8\pi \sin(4\pi x) \cos(4\pi y) \quad \text{et} \quad \tilde{v}_N = v_N + \frac{\partial p}{\partial y} = 0 \quad (5.78)$$

<sup>3</sup>Ce problème correspond au problème de Darcy.

En injectant ces vitesses intermédiaires dans le problème (5.69), ce problème peut être résolu par PGD en utilisant les opérateurs définis dans (5.72).

Le problème est donc résolu à l'aide de la méthode PGD et à l'aide d'un solveur standard (ici le bigradient conjugué ne fournissant plus des résultats satisfaisants en précision a été remplacé par un solveur 'LU'). Les simulations ont été effectuée pour un nombre de nœuds similaire dans les directions  $x$  et  $y$  variant de  $N = 8$  à  $N = 64$ .

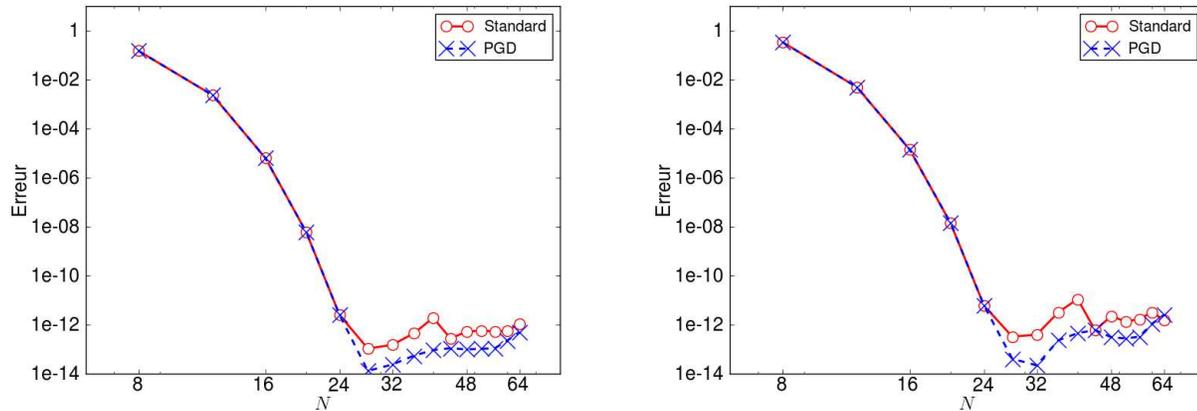


FIG. 5.3 – Erreur sur la vitesse horizontale (gauche) et sur la pression (droite) par rapport à la solution analytique en fonction du nombre de nœuds dans chaque direction.

Les erreurs (définies par l'équation (5.76)) sur la vitesse horizontale et sur la pression en échelle 'log-log' pour le solveur standard et le solveur PGD sont représentées sur la figure 5.3. On constate que le solveur PGD est aussi précis que le solveur standard et qu'à partir d'un nombre de nœuds dans chaque direction égal à 24, l'erreur n'est plus significative. Là encore, on note une convergence exponentielle en fonction du nombre de nœuds. Ces résultats sont similaires à ceux obtenus par Botella dans [Botella, 1996]. L'erreur sur la vitesse verticale n'a pas été représentée ici car elle est identique à celle de la vitesse horizontale.

Les temps de calcul nécessaires à la résolution du problème de correction pour les solveurs standards et PGD sont représentés sur la figure 5.4. On constate ici que les temps de calculs sont relativement similaires pour les deux solveurs étudiés. En effet pour un maillage de taille  $64 \times 64$  le temps de calcul avec le solveur standard est de 2322 secondes alors que le temps de calcul avec le solveur séparé est de 1929 secondes.

L'étape de correction par un algorithme d'Uzawa est donc validée sur ce cas modèle. Nous allons maintenant revenir à notre problème de Navier-Stokes instationnaire en démarrant par le cas test de Taylor-Green traité dans la partie IV .1.

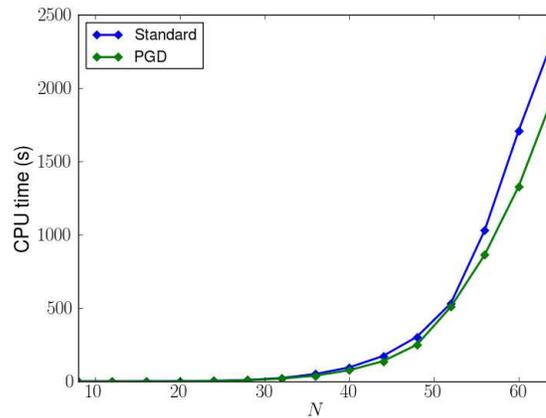


FIG. 5.4 – Temps de calcul pour les solveurs standards et PGD en fonction du nombre de nœuds dans chaque direction.

### IV .3 Problème de Taylor-Green

On rappelle que le problème de Taylor-Green (voir IV .1) consiste à résoudre les équations de Navier-Stokes dans un domaine  $\Omega = [0.25; 1.25] \times [0.5; 1.5]$  et qui admet les solutions analytiques suivantes :

$$u_{ana}(x, y) = -\cos(2\pi x) \sin(2\pi y) e^{-8\pi^2 \nu t},$$

$$v_{ana}(x, y) = \sin(2\pi x) \cos(2\pi y) e^{-8\pi^2 \nu t},$$

$$p_{ana}(x, y) = -\frac{1}{4} [\cos(4\pi x) + \cos(4\pi y)] e^{-16\pi^2 \nu t}$$

Les conditions aux limites sont définies de manière à correspondre à la solution analytique sur les bords.

Les simulations ont été effectuées par méthode PGD et à l'aide d'un solveur standard. Le maillage est choisi de taille  $20 \times 20$  et le pas de temps varie de  $\delta t = 10^{-2}$  à  $\delta t = 10^{-4}$ . Les erreurs sont calculées pour un temps de simulation égal à 1 seconde. L'erreur commise par rapport à la solution analytique définie par l'équation (5.76) en fonction du pas de temps en échelle log-log est tracée sur la figure (5.5) pour la vitesse horizontale et pour la pression.

Cette figure montre que dans le cas de la résolution standard, l'erreur en vitesse et en pression admet un comportement spectral. En effet, l'erreur décroît exponentiellement avec le pas de temps. Concernant l'erreur obtenue par résolution PGD, la figure (5.5) montre que le comportement de l'erreur en vitesse est très proche de celui obtenu par une résolution standard. Concernant la pression, pour des pas de temps égaux à  $10^{-2}$ ,  $5 \cdot 10^{-3}$ ,  $10^{-3}$  et  $5 \cdot 10^{-4}$  l'erreur obtenue par résolution PGD a un comportement similaire à celle obtenue par résolution standard. Par contre, pour le plus petit pas de temps étudié,  $\delta t = 10^{-4}$ , l'erreur obtenue par PGD continue à décroître mais de manière exponentielle.

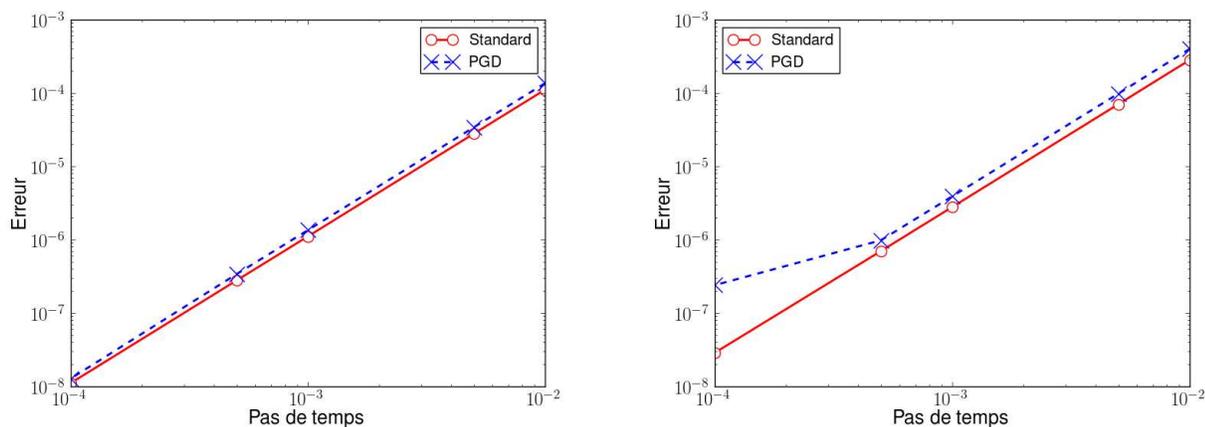


FIG. 5.5 – Erreur en vitesse (droite) et en pression (gauche) par rapport à la solution analytique en fonction du pas de temps pour un temps de simulation de 1 seconde.

#### IV .4 Ecoulement dans une cavité 2D

Les équations de Navier-Stokes vont maintenant être résolues pour le cas test de la cavité entraînée 2D isotherme déjà traité auparavant (voir chapitre 3 section IV .2). On rappelle la géométrie du problème ainsi que les conditions aux limites sur la figure 5.6.

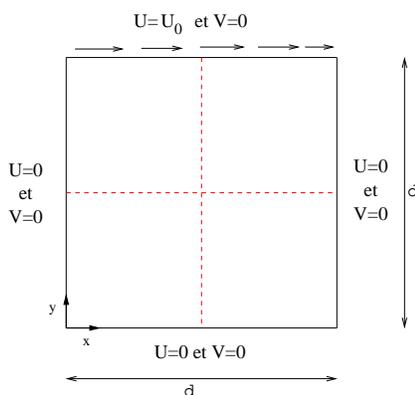


FIG. 5.6 – Geometrie de la cavité entraînée 2D.

Les simulations sont effectuées uniquement dans le cas où la solution converge vers une solution stationnaire. Ici nous avons choisi les cas où  $Re = 100$  et  $Re = 1000$ . Le pas de temps des simulations a été fixé à  $\delta t = 10^{-3}$  et le critère de convergence du problème défini par l'équation (3.40) a été fixé à  $\epsilon_u = 10^{-10}$  et  $\epsilon_p = 10^{-6}$ . Enfin les simulations ont été effectuées avec la méthode PGD ainsi qu'avec un solveur classique (bi-gradient conjugué). Les simulations ont été effectuées avec un maillage de taille  $20 \times 20$  dans le cas où  $Re = 100$  et avec un maillage de taille  $26 \times 26$  dans le cas où  $Re = 1000$ .

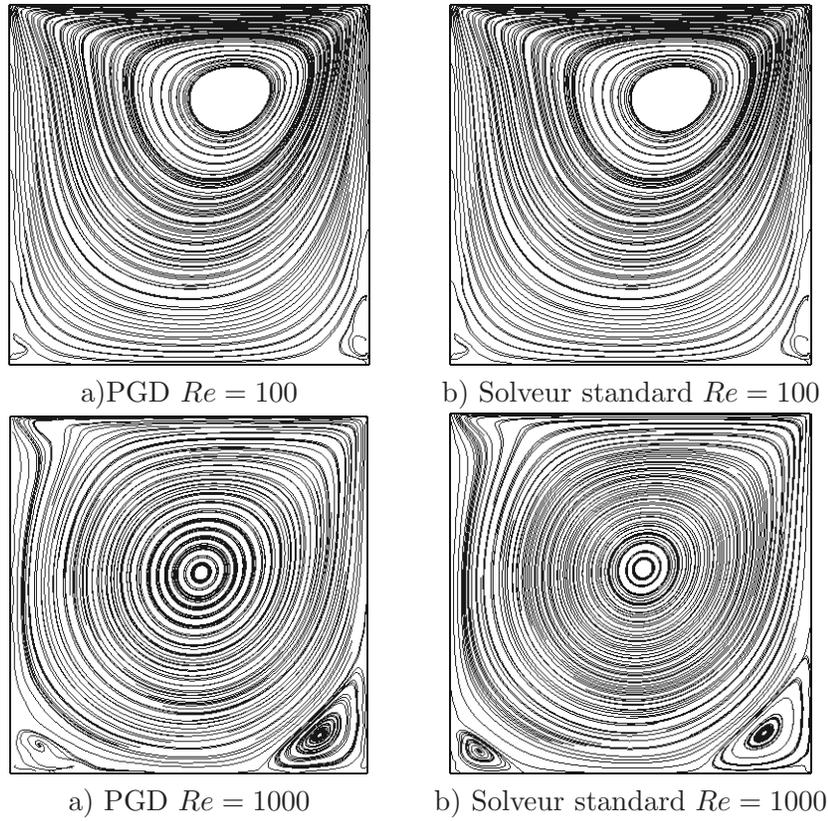


FIG. 5.7 – Lignes de courant calculées par PGD et avec le solveur standard avec  $N_h = 20$  pour  $Re = 100$  et avec  $N_h = 26$  pour  $Re = 1000$ .

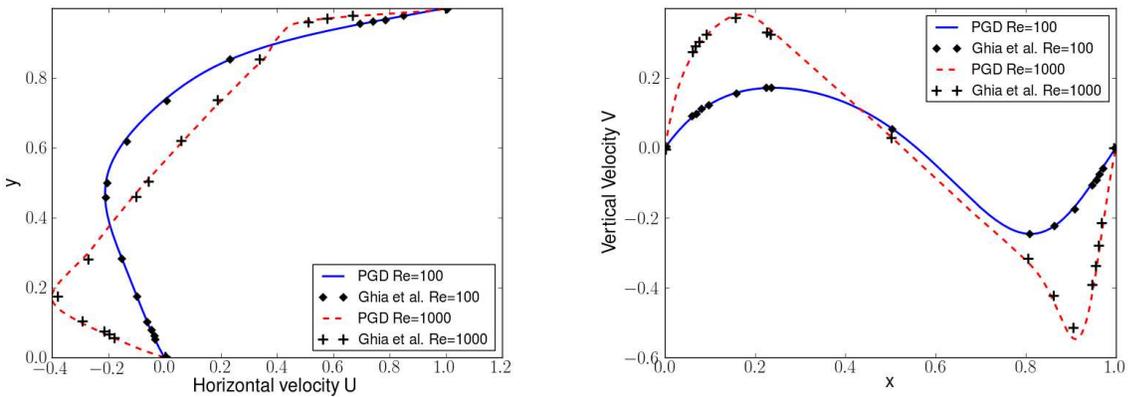


FIG. 5.8 – Comparaison des vitesses horizontales à  $x=0.5$  (à gauche) et des vitesses verticales à  $y=0.5$  (à droite) avec les résultats de Ghia. and al. avec  $N_h = 20$  pour  $Re = 100$  et avec  $N_h = 26$  pour  $Re = 1000$ .

La figure 5.7 représente les lignes de courant<sup>4</sup> calculées avec la méthode PGD et avec le modèle standard pour  $Re = 100$  et  $Re = 1000$ . On constate que pour les deux nombres de Reynolds considérés les lignes de courant calculées par PGD sont très proches de celles obtenues avec le solveur standard avec des allures de vortex très similaires à ce qui a déjà été obtenu dans le chapitre 3 section IV .2. Pour confirmer cela, la figure 5.8 montre que les vitesses horizontales en  $x = 0.5$  et les vitesses verticales en  $y = 0.5$  obtenues par PGD correspondent à celles issues de la littérature (Ghia et al.[Ghia et al., 1982])

Les champs de pression obtenus par PGD et avec le solveur standard sont également représentés sur la figure (5.9) pour  $Re = 100$  et  $Re = 1000$ . Là encore, on constate une bonne correspondance entre le calcul standard et le calcul PGD.

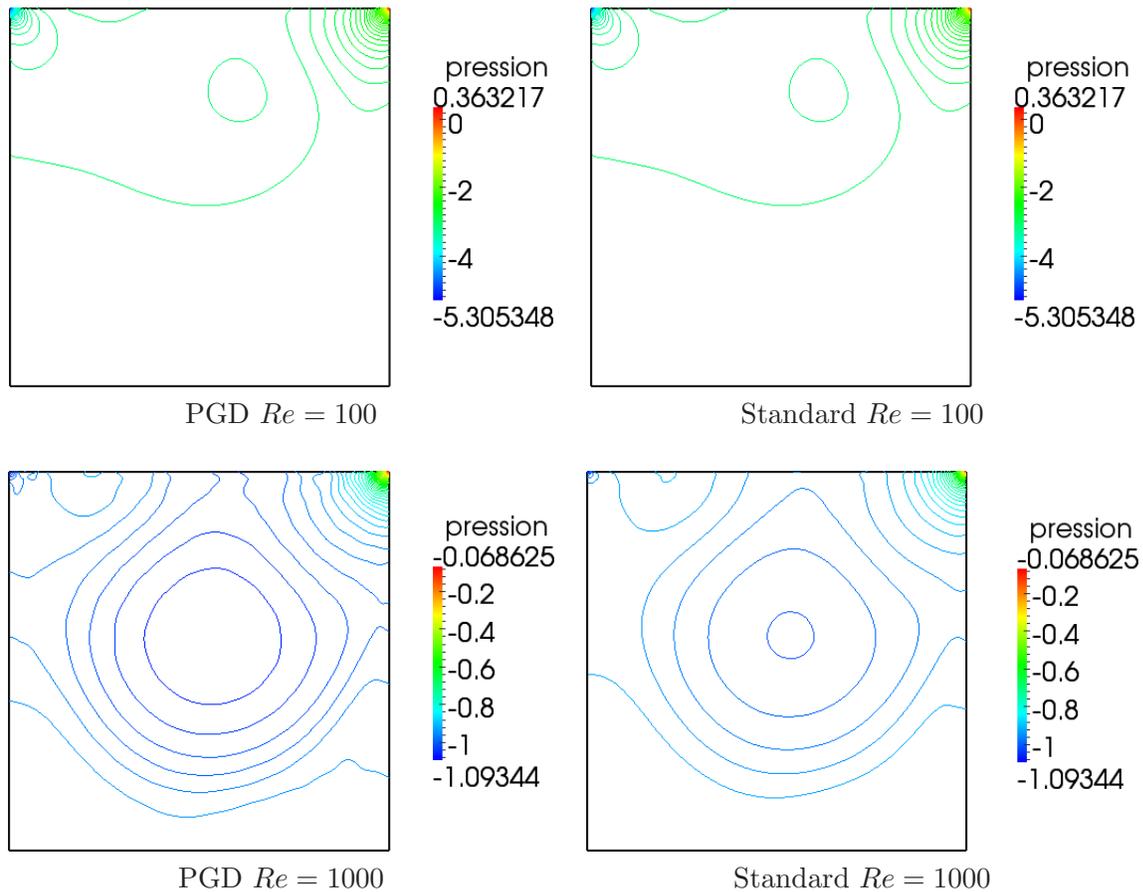


FIG. 5.9 – Pression calculée par PGD et par le solveur standard pour  $Re = 100$  ( $N_h = 20$ ) et  $Re = 1000$  ( $N_h = 26$ ).

<sup>4</sup>Il est à noter que pour pouvoir tracer les résultats suivants de manière correcte nous avons effectué une interpolation des résultats obtenus sur un maillage régulier de taille  $200 \times 200$  en utilisant l'équation (5.6) pour le champ de vitesse et l'équation (5.15) pour la pression.

La norme du résidu associé à l'équation vérifiée par  $\tilde{u}$  (resp.  $\tilde{v}$  et  $\tilde{p}$ ) en fonction du nombre de fonctions gardées est tracée sur la figure 5.10 (respectivement 5.11 et 5.12) pour  $Re = 100$  et  $Re = 1000$ . Pour  $Re = 100$  on observe que la courbe représentant la norme du résidu pour les fonctions PGD est identique aux courbes représentant la norme des résidus de chacune des SVD jusqu'à une valeur de  $10^{-5}$ . A ce moment, la courbe PGD décroît plus lentement jusqu'à une norme du résidu proche du zéro machine en approximativement cent dix fonctions alors que les SVD décroissent subitement vers le zéro machine en vingt fonctions exactement. On remarque que le nombre de fonctions SVD nécessaires à une convergence proche du zéro machine est égal au nombre de nœuds dans chaque direction<sup>5</sup>. Cette information est confirmée à  $Re = 1000$ .

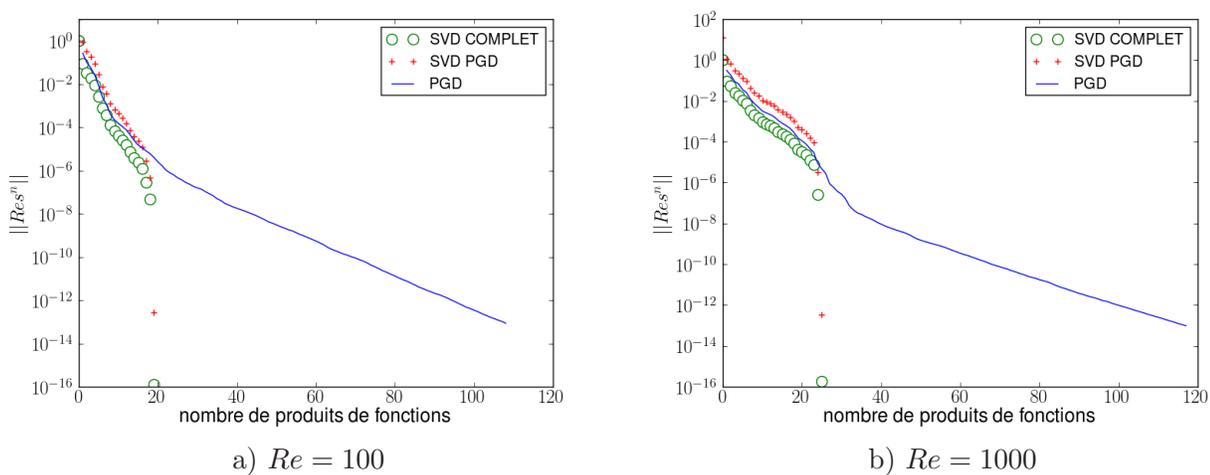


FIG. 5.10 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la vitesse intermédiaire horizontale  $\tilde{u}$  et pour  $Re = 100$  et  $Re = 1000$ .

Les normes du résidu pour la pression de correction à  $Re = 100$  et  $Re = 1000$  sont tracées sur la figure 5.12. Dans ce cas, la PGD n'est pas du tout optimale. En effet, comme pour les vitesses intermédiaires, il suffit d'un nombre de fonction équivalent au nombre de nœuds dans chaque direction pour que la courbe SVD se rapproche du zéro machine alors que la PGD semble avoir beaucoup de difficulté à atteindre cette même erreur. Dans le cas  $Re = 1000$ , on observe un palier aux alentours de  $10^{-2}$  pour les six cents premières itérations, puis, à partir de cette valeur l'erreur chute brusquement. Pour la résolution de la pression, nous nous retrouvons dans la pire des situations, celle où le nombre de fonctions PGD nécessaires à la résolution du problème est quasiment égal au nombre de noeuds du maillage.

<sup>5</sup>Dans le cas de la SVD sur les résultats finaux obtenus par PGD, la norme du résidu ne décroît pas au delà d'une certaine valeur proche du zéro machine. Cette valeur correspond en fait à la norme du résidu de la solution obtenue par PGD

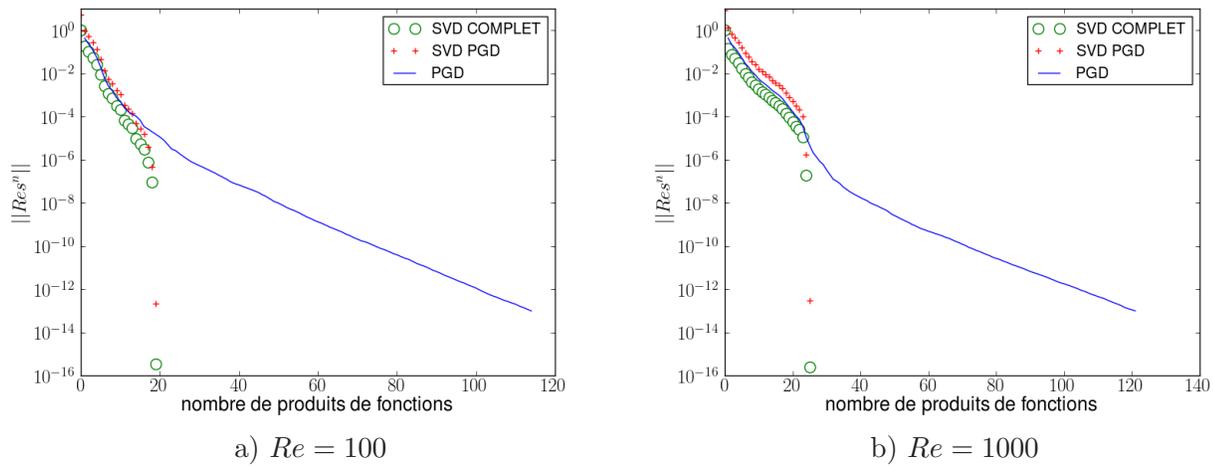


FIG. 5.11 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la vitesse intermédiaire verticale  $\tilde{v}$  et pour  $Re = 100$  et  $Re = 1000$ .

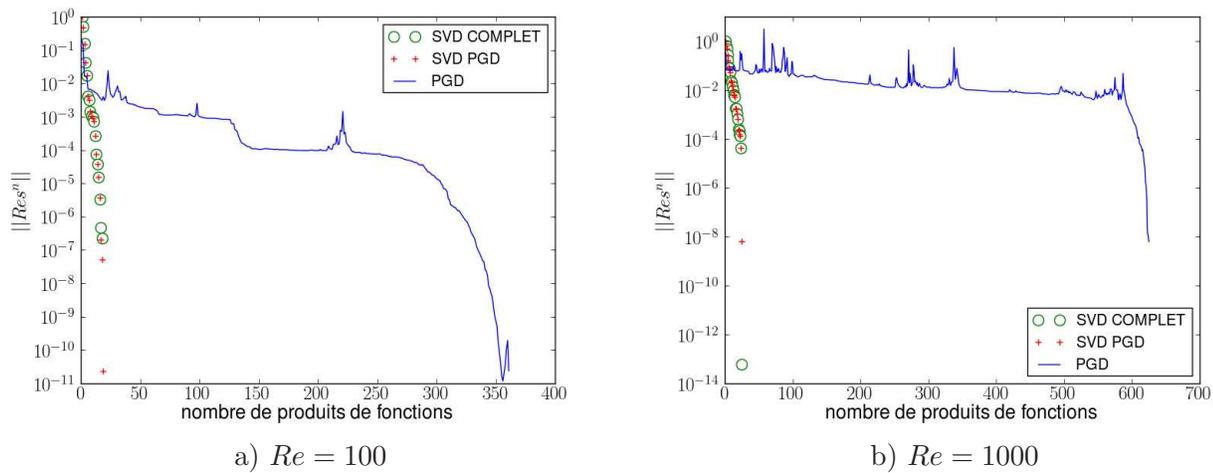


FIG. 5.12 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la pression de correction  $\tilde{p}$  et pour  $Re = 100$  et  $Re = 1000$ .

Au vu de ces résultats, il est évident que la PGD ne peut être plus rapide que le solveur standard. C'est pourquoi, nous n'avons pas tracé de courbes représentant le temps de calcul par méthode PGD et avec le solveur standard en fonction du nombre de nœuds du maillage, car dans le cas où  $Re = 1000$  et avec 26 nœuds dans chaque direction par exemple, le solveur standard est beaucoup plus rapide que la PGD.

## V Conclusions du chapitre

Dans cette partie, nous avons appliqué une discrétisation spectrale de type  $\mathbb{P}_N - \mathbb{P}_{N-2}$  afin de résoudre les équations de Navier-Stokes. Nous avons montré, qu'en terme d'écriture des problèmes, la formulation spectrale se prêtait particulièrement au formalisme PGD. En effet la formulation spectrale, à l'instar de la PGD consiste à écrire les problèmes multidimensionnels et à résoudre les problèmes assemblés. La PGD, quant à elle, n'a pas besoin de rassembler les opérateurs et résout directement les problèmes 1D de manière successive. Le cas d'une équation de diffusion stationnaire a été traitée et a montré la puissance de la PGD pour ce type de problème. En effet pour une précision similaire à celle obtenue avec un solveur standard, le gain de temps est de l'ordre de 100 sur un maillage de taille  $64 \times 64$  lorsque la solution n'est pas décomposable en un seul produit de fonction. La méthode de résolution utilisée implique la résolution d'une étape de correction par un algorithme d'Uzawa. Cette étape de correction a donc été traitée sur un cas où l'on dispose d'une solution analytique. Cela nous a permis de retrouver les résultats de la littérature, c'est à dire une convergence spectrale en fonction de la taille du maillage, malheureusement dans ce cas le gain de temps observé est très faible, ce qui semble venir du très mauvais conditionnement des matrices de résolution 1D de l'opérateur d'Uzawa. Nous avons ensuite résolu le problème dit de 'Taylor-Green', qui est un problème de Navier-Stokes où l'on dispose d'une solution analytique. Dans ce cas l'on retrouve également les résultats de la littérature, c'est à dire une convergence spectrale en fonction de la taille du pas de temps. Enfin le cas de la cavité entraînée a été traité pour des nombres de Reynolds égaux à  $Re = 100$  et  $Re = 1000$ . Dans ce cas, on observe que les simulations réalisées avec PGD et avec le solveur standard fournissent les mêmes résultats et sont en accords avec ceux fournis par la littérature.

Cette étude constitue une première approche vers une discrétisation spectrale de la PGD pour la résolution des équations de Navier-Stokes. Les résultats obtenus sont satisfaisants au point de vue de la précision, en revanche, les gains de temps ne sont pas du tout à la hauteur de ce que l'on aurait espéré. En effet la résolution PGD du problème de pression, nécessitant un nombre de fonctions quasiment équivalent au nombre de nœuds du maillage, il est clair que lorsque nous souhaiterons augmenter la finesse du maillage, les temps de calculs deviendront très rapidement prohibitifs. Une des pistes envisageables serait de préconditionner ce problème. Une extension au cas 3D serait également intéressante.



# Conclusion et perspectives

Motivés par la résolution rapide des équations de Navier-Stokes, nous avons présenté dans ce manuscrit les résultats obtenus à l'aide de la méthode de réduction PGD. Cette méthode, qui consiste à rechercher la solution d'une EDP de façon itérative sous forme séparée, est une méthode de réduction (ROM) a priori qui ne nécessite donc pas d'informations préalable sur l'écoulement. Elle a déjà été appliquée dans de nombreux domaines (mécanique du solide, problèmes stochastiques, problèmes multidimensionnels, ...) mais jamais pour simuler des écoulements fluides, ce qui constitue l'originalité de cette étude.

Dans ce travail, la méthode a tout d'abord été utilisée pour résoudre des équations modèles disposant d'une solution analytique. L'équation de diffusion stationnaire 2D et 3D, l'équation de diffusion instationnaire 2D et les équations de Burgers et Stokes ont ainsi été traitées. Dans le cas de problèmes instationnaires, la solution a été recherchée à chaque pas de temps sous forme de décomposition sur les variables d'espace uniquement, sauf pour l'équation de diffusion instationnaire 2D où une décomposition sur les variable d'espace et de temps a été mise en œuvre. Nous avons montré que dans tous ces cas la méthode PGD permet de retrouver les solutions analytiques avec une précision équivalente au modèle standard. Nous avons également mis en évidence la supériorité de la PGD par rapport au modèle standard en terme de temps de calcul. En effet, dans le cas des équations de diffusion 2D ou 3D, lorsque la PGD a été effectuée sur les variables d'espace uniquement, la PGD s'est montrée beaucoup plus rapide que le solveur standard (plusieurs dizaine de fois). Le gain de temps est encore plus significatif lorsque le temps est intégré dans la séparation (de l'ordre de la centaine). Le caractère non linéaire d'un problème ainsi que le couplage vitesse pression ont été traités à travers la résolution de l'équation de Burgers et de Stokes respectivement. Dans ces cas aussi nous avons constaté des gains de temps significatifs (de l'ordre de cinq pour l'équation de Burgers et de trente cinq pour l'équation de Stokes).

La résolution des équations de Navier-Stokes par méthode PGD, qui est l'objectif de ce travail, a tout d'abord été faite en utilisant un schéma de prédiction correction sur un maillage décalé en formulation volumes finis. La résolution du problème de Kim et Moin, dont une solution analytique est connue, a permis de mettre en avant la capacité de la méthode à retrouver des résultats aussi précis que ceux du modèle standard avec un gain de temps de l'ordre de la vingtaine par rapport au modèle standard. La méthode a ensuite été appliquée à la résolution de l'écoulement dans une cavité entraînée 2D stationnaire et instationnaire. Dans ce cas, la méthode PGD est une dizaine de fois plus rapide que le solveur standard avec une précision comparable. Une comparaison du nombre de fonctions PGD, et du nombre de fonctions SVD effectuée sur le champ de vitesse et la pression, a mis en évidence le caractère optimal de la méthode jusqu'à une certaine valeur de la norme du résidu associé à l'équation discrétisée. Au delà de cette valeur, le PGD cesse d'être optimale et nécessite plus de fonctions que la SVD pour une même norme du résidu. Chose étonnante, la valeur du seuil d'optimalité de la PGD diminue lorsque le nombre de Reynolds augmente. De plus il a été observé que la résolution de la pression nécessite plus de fonctions PGD que la résolution de la vitesse. Il est à noter que ces résultats ont fait l'objet de deux publications [Dumon et al., 2011, Dumon et al., 2010].

Dans le cas de la résolution de l'écoulement dans une cavité entraînée 3D stationnaire, le champ de vitesse obtenu par PGD est identique à celui obtenu par le solveur standard pour les deux nombres de Reynolds considérés ( $Re = 100$  et  $Re = 1000$ ). Par contre, pour un nombre

de Reynolds égal à 1000, la pression calculée par PGD et par le solveur standard est différente. Cependant celle-ci n'affecte pas le champ de vitesse. De plus aucun gain de temps notable n'est observé. La résolution des équations de Navier-Stokes 2D anisotherme avec et sans transport de masse a également été traitée. Là encore, la PGD permet de retrouver les résultats du modèle standard et de la littérature tout en observant des gains de temps significatifs (de l'ordre de la dizaine).

Enfin, nous avons effectué une première approche de la résolution des équations de transfert par méthode PGD en formulation spectrale. Sur les différents problèmes traités, la PGD a montré une précision de calcul aussi bonne que le solveur standard. Un gain de temps a été observé pour le cas de l'équation de Poisson. Par contre, concernant le problème de Darcy ou les équations de Navier-Stokes aucun gain de temps n'a été observé. Dans le cas de la cavité entraînée la PGD est même beaucoup moins rapide que le solveur standard. Une comparaison du nombre de fonctions PGD et SVD dans ce dernier cas a mis en évidence que pour l'équation en pression, il fallait autant de fonctions que de nœuds pour obtenir une norme du résidu satisfaisante pour une méthode spectrale.

En résumé, il ressort que la méthode PGD permet de résoudre les équations de Navier-Stokes 2D isothermes et anisothermes correctement avec un gain de temps significatif. Par contre des améliorations doivent être apportées dans le cas 3D et dans la formulation spectrale. Une première amélioration pourrait être de traiter différemment le schéma de prédiction-correction. En effet pour le moment la vitesse intermédiaire est recherchée sous forme séparée dans un premier temps, et dans un deuxième temps, la pression de correction est calculée à partir de la vitesse intermédiaire. Une alternative consisterait à effectuer des enrichissements simultanés sur la vitesse et la pression. De cette manière chaque produit de fonctions (pour la vitesse intermédiaire et pour la pression de correction) vérifierait le problème de prédiction-correction indépendamment. Un deuxième point important est le fait que l'estimateur d'erreur utilisé basé sur la minimisation de la norme du résidu n'est peut être pas toujours le choix le plus adapté. Cette piste serait également intéressante à explorer. Une autre perspective, au vu des résultats obtenus sur l'équation de diffusion instationnaire 2D, serait d'intégrer le temps dans la séparation. Enfin, étendre la PGD à la simulation de problème d'interaction fluide structure peut être envisagé.



## Annexe A

# PGD et Formulation $\psi - \omega$ des équations de Navier-Stokes

## I La formulation ligne de courant-vorticité

La formulation ligne de courant-vorticité consiste à traiter le couplage vitesse-pression en passant des variables dites primitives, qui sont la vitesse et la pression, aux variables vorticité et fonction de courants. Ici nous ne nous intéresserons qu'au cas 2D. La méthode permettant d'effectuer ce changement de variable sera détaillée dans ce cadre.

Les équations de Navier-Stokes sous forme adimensionnelle en formulation vitesse-pression s'écrivent :

$$\nabla \cdot \mathbf{u} = 0 \quad (\text{A.1})$$

$$\frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re} \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \mathbf{f} \text{ dans } \Omega \times [0, T] \quad (\text{A.2})$$

$$\mathbf{u} \quad (t = 0) = \mathbf{u}_0 \quad (\text{A.3})$$

$$\mathbf{u} = \mathbf{g}_D \text{ sur } \Gamma \quad (\text{A.4})$$

où  $\mathbf{u} (u, v)$  et  $p$  sont respectivement le champ de vitesse et la pression,  $\mathbf{f}$  est une force connue,  $Re$  est le nombre de Reynolds,  $\Omega$  un ouvert de  $\mathbb{R}^2$ ,  $\Gamma$  la frontière de ce domaine.

La relation d'incompressibilité (A.1) implique l'existence d'une fonction de courant  $\psi$  telle que :

$$\mathbf{u} \equiv (u, v) = \left( \frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \quad (\text{A.5})$$

Lorsque l'opérateur rotationnel ( $\nabla \times$ ) est appliqué à chaque membre de l'équation de quantité de mouvement (A.2), et que les relations suivantes sont utilisées :

$$\nabla \times \mathbf{u} = -\nabla^2 \psi \cdot \mathbf{z} \quad (\text{A.6})$$

et

$$\nabla \times (\mathbf{u} \cdot \nabla) \mathbf{u} = (\nabla \cdot \mathbf{u})(\nabla \times \mathbf{u}) + (\mathbf{u} \cdot \nabla)(\nabla \times \mathbf{u}) \quad (\text{A.7})$$

il vient la formulation suivante :

$$\frac{\partial \nabla^2 \psi}{\partial t} - \frac{1}{Re} \nabla^2 (\nabla^2 \psi) + (\mathbf{u} \cdot \nabla) \nabla^2 \psi = -\nabla \times \mathbf{f} \quad (\text{A.8})$$

$$\frac{\partial \psi}{\partial y} = \mathbf{g}_D \cdot \mathbf{x}, \quad \frac{\partial \psi}{\partial x} = -\mathbf{g}_D \cdot \mathbf{y} \quad \text{sur } \partial \Omega \quad (\text{A.9})$$

Toutefois la solution  $\psi$  de ce problème est définie à une constante près et l'équation aux dérivées partielles de degré quatre est en général difficile à résoudre. La séparation de cette équation constitue donc une bonne alternative pour rendre ce problème traitable numériquement. Pour ce faire, il est possible d'introduire la vorticité  $\omega = (\nabla \times \mathbf{u}) \cdot \mathbf{z}$  qui est une notion classique en mécanique des fluides. En injectant l'expression de la vorticité dans (A.9) la formulation des équations de Navier-Stokes en variables  $(\psi, \omega)$  est alors obtenue :

$$\begin{cases} \frac{\partial \omega}{\partial t} - \frac{1}{Re} \nabla^2 \omega + (\mathbf{u} \cdot \nabla) \omega = -\nabla \times \mathbf{f} & \text{dans } \Omega \\ \omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} & \text{sur } \partial \Omega \end{cases} \quad (\text{A.10})$$

et

$$\begin{cases} \nabla^2 \psi = -\omega & \text{dans } \Omega \\ \frac{\partial \psi}{\partial y} = \mathbf{g}_D \cdot \mathbf{x}, \quad \frac{\partial \psi}{\partial x} = -\mathbf{g}_D \cdot \mathbf{y} & \text{sur } \partial\Omega \end{cases} \quad (\text{A.11})$$

Résoudre les équations de Navier-Stokes en formulation ligne de courant vorticité revient donc à résoudre successivement les problèmes (A.10) et (A.11). Ainsi en tenant compte de la discrétisation temporelle, l'algorithme de résolution pour cette formulation est :

- 1) Les couples  $(\omega^n, \psi^n)$  et  $(\omega^{n-1}, \psi^{n-1})$  sont supposés connus (et donc les vitesses  $\mathbf{u}^n$  et  $\mathbf{u}^{n-1}$ ).
- 2) L'équation associée à la vorticité  $\omega^{n+1}$  est résolue, où le terme diffusif a été discrétisé par un schéma de Crank-Nicholson, et le terme convectif à été discrétisé de manière totalement explicite à l'aide d'un schéma d'Adams-Bashforth :

$$\begin{cases} \frac{\omega^{n+1}}{\delta t} - \frac{1}{2Re} \nabla^2 \omega^{n+1} = \frac{\omega^n}{\delta t} + \frac{1}{2Re} \nabla^2 \omega^n - \nabla \times \mathbf{f} - \frac{3}{2}(\mathbf{u}^n \cdot \nabla) \omega^n + \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla) \omega^{n-1} \\ \omega^{n+1} = \frac{\partial v^n}{\partial x} - \frac{\partial u^n}{\partial y} & \text{sur } \partial\Omega \end{cases} \quad (\text{A.12})$$

- 3) L'équation associée à la fonction de courants  $\psi^{n+1}$  est alors résolue sous la forme suivante :

$$\begin{cases} \nabla^2 \psi^{n+1} = -\omega^{n+1} \\ \frac{\partial \psi^{n+1}}{\partial y} = \mathbf{g}_D^{n+1} \cdot \vec{\mathbf{x}}, \quad \frac{\partial \psi^{n+1}}{\partial x} = -\mathbf{g}_D^{n+1} \cdot \vec{\mathbf{y}} & \text{sur } \partial\Omega \end{cases} \quad (\text{A.13})$$

- 4) Les variables de vitesses sont remise à jour grâce à la définition de la fonction de courant  $\psi$  :

$$\mathbf{u}^{n+1} = (u^{n+1}, v^{n+1}) = \left( \frac{\partial \psi^{n+1}}{\partial y}, -\frac{\partial \psi^{n+1}}{\partial x} \right) \quad (\text{A.14})$$

## II Résolution par PGD

La mise en place de la PGD va maintenant être brièvement explicitée dans le cas de la formulation ligne de courant-vorticité. La PGD se fait uniquement sur les variables d'espace à chaque instant  $n$ . Au pas de temps  $n+1$ , le problème (A.12), qui est une équation de diffusion instationnaire dont les opérateurs PGD sont les mêmes que ceux définis dans la partie III.1, est résolu en cherchant  $\omega^{n+1}$  sous la forme séparée suivante :

$$\omega^{n+1}(x, y) = \sum_{i=1}^{n_\omega^{n+1}} \alpha_\omega^{i, n+1} F_\omega^{i, n+1}(x) G_\omega^{i, n+1}(y)$$

De même, le problème (A.13), qui est une équation de diffusion stationnaire dont les opérateurs PGD ont également été définis dans la partie I, est résolu en cherchant  $\psi^{n+1}$  sous la forme séparée suivante :

$$\psi^{n+1}(x, y) = \sum_{i=1}^{n_\psi^{n+1}} \alpha_\psi^{i^{n+1}} F_\psi^{i^{n+1}}(x) G_\psi^{i^{n+1}}(y)$$

Le champ de vitesse est alors remis à jour sous forme séparée à l'aide de la définition de la fonction de courant, soit :

$$\begin{cases} u^{n+1}(x, y) = \sum_{i=1}^{n_\psi^{n+1}} \alpha_\psi^{i^{n+1}} \frac{dF_\psi^{i^{n+1}}(x)}{dx} G_\psi^{i^{n+1}}(y) \\ v^{n+1}(x, y) = \sum_{i=1}^{n_\psi^{n+1}} \alpha_\psi^{i^{n+1}} F_\psi^{i^{n+1}}(x) \frac{dG_\psi^{i^{n+1}}(y)}{dy} \end{cases}$$

### III Écoulement dans une cavité entraînée 2D

#### III .1 Régime stationnaire ( $Re = 100$ et $Re = 1000$ )

Comme mentionné dans la section précédente (partie IV .2), l'écoulement dans une cavité entraînée 2D à  $Re = 100$  et  $Re = 1000$  est stationnaire. Les simulations ont donc été effectuées pour un pas de temps égal à  $\delta t = 10^{-3}$ . Le problème est considéré convergé lorsque les deux critères suivant sont vérifiés simultanément :

$$\frac{\|\omega^k - \omega^{k-1}\|}{\|\omega^k\|} \leq \epsilon_\omega \quad ; \quad \frac{\|\psi^k - \psi^{k-1}\|}{\|\psi^k\|} \leq \epsilon_{psi} \quad (\text{A.15})$$

Dans la suite les critères d'arrêts ont été fixés à  $\epsilon_\omega = 10^{-10}$  et  $\epsilon_\psi = 10^{-10}$ .

Les figures (A.1), (A.2) et (A.3) sur lesquelles ont été tracées respectivement les lignes de courant, les vitesses au centre de la cavité et la vorticit e pour  $Re = 100$  et  $Re = 1000$  montrent que les r esultats obtenus par la m ethod e PGD sont identiques  a ceux obtenus par la m ethod e standard et  a ceux de de Ghia et al. [Ghia et al., 1982].

La Figure (A.4) compare les temps de simulations en fonction du nombre de nœuds de discr etisation spatiale  $N_h$  dans chaque direction. Ce nombre de nœuds varie de  $N_h = 50$   a  $N_h = 500$ . On constate qu' a partir de 300 nœuds dans chaque direction les simulations effectu ees  a l'aide de la m ethod e PGD sont plus rapide que celles effectu ees avec le solveur standard. De plus, pour un maillage de taille  $500 \times 500$ , les temps de calculs sont divis es par deux avec la m ethod e PGD pour  $Re = 100$  et par quatre pour  $Re = 1000$ .

La figure (A.5) compare le nombre de fonctions PGD  a celui d'une SVD sur la vorticit e obtenue par la m ethod e standard pour chaque pas de temps. Dans le cas o u  $Re = 100$ , le nombre de fonctions PGD n ecessaire au calcul des lignes de courants est de l'ordre de la cinquantaine alors que pour une SVD, il est de l'ordre de la trentaine. Par contre pour un nombre de Reynolds  egal  a 1000 le nombre de fonctions PGD est de l'ordre de la soixantaine pour un nombre de fonctions SVD de l'ordre de la quarantaine. Lorsque le nombre de Reynolds augmente le nombre de fonctions PGD et SVD augmente  egalement, pourtant l' ecart entre les deux semble diminuer. Ces conclusions restent valables pour la r esolution de la vorticit e.

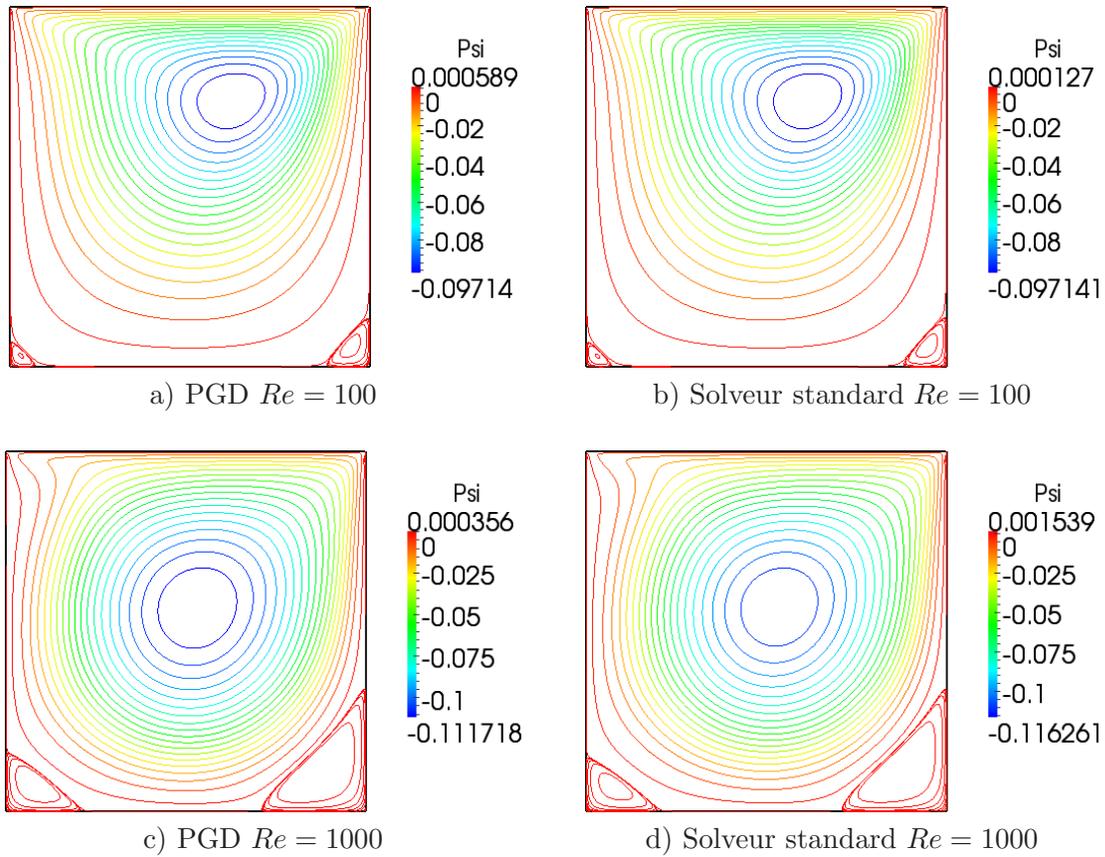


FIG. A.1 – Lignes de courant calculées par PGD et avec le solveur standard avec  $N_h = 200$  et pour  $Re = 100$  et  $Re = 1000$ .

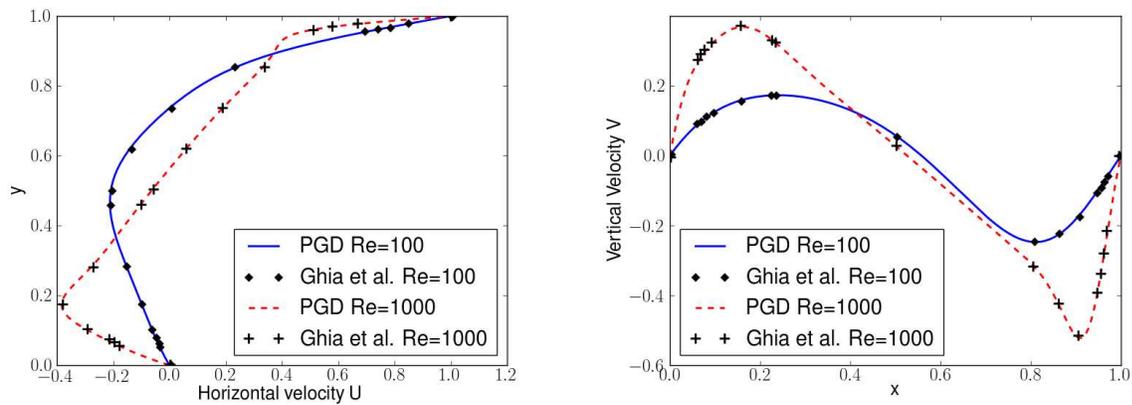


FIG. A.2 – Comparaison des vitesses horizontales à  $x=0.5$  (à gauche) et des vitesses verticales à  $y=0.5$  (à droite) avec les résultats de Ghia. and al. pour  $Re = 100$  et  $Re = 1000$ .

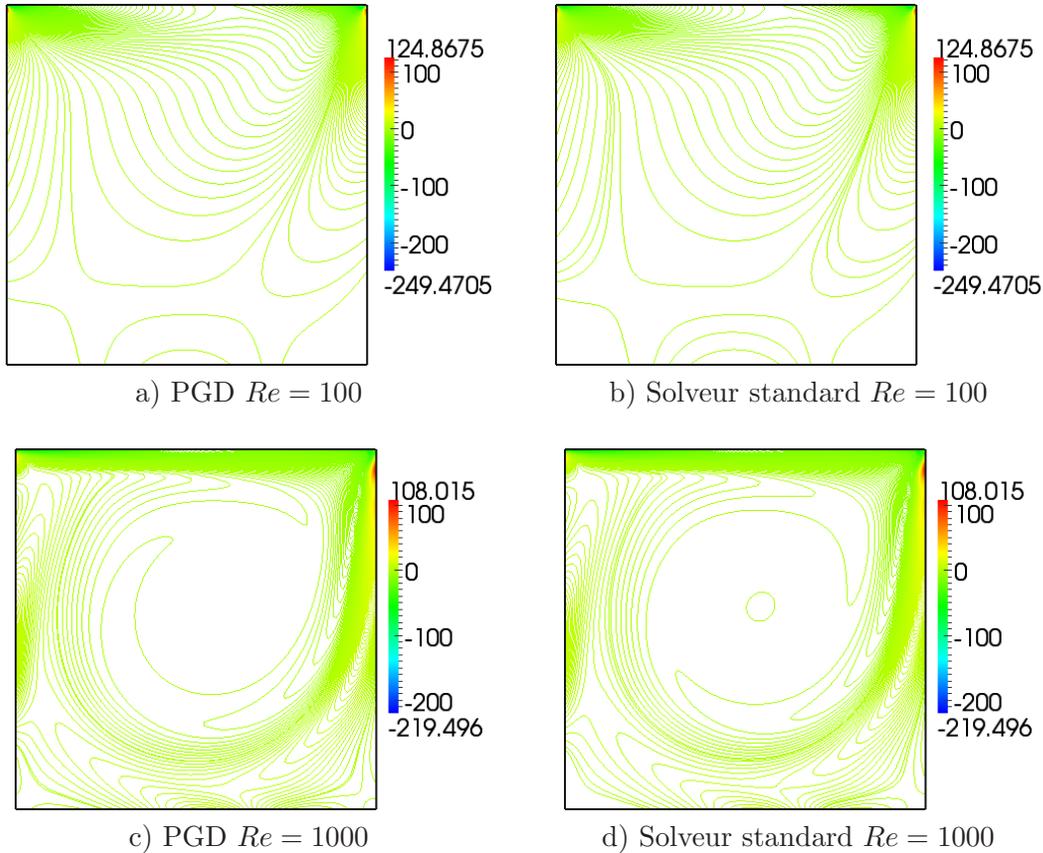


FIG. A.3 – Isocontours de vorticité calculées par PGD et avec le solveur standard avec  $N_h = 200$  et pour  $Re = 100$  et  $Re = 1000$ .

L'évolution de la norme du résidu par rapport au nombre de fonctions PGD et au nombre de fonctions issu des deux SVD (une sur les résultats finaux PGD et l'autre sur les résultats du solveur standard) a été tracée à convergence pour les variables de vorticité et de lignes de courants sur les figures A.6 et A.7. Concernant la fonction lignes de courant, pour  $Re = 100$  et  $Re = 1000$ , la PGD semble être optimale jusqu'à une valeur de la norme du résidu égale à  $10^{-2}$ . A partir de cette valeur de la norme du résidu, la décroissance du résidu en fonction du nombre de fonctions est plus lente pour la PGD que pour les deux SVD. En effet pour atteindre une norme du résidu inférieure à  $10^{-5}$  pour  $Re = 100$ , la PGD nécessite 80 fonctions contre une vingtaine pour les SVD. De même pour atteindre une norme du résidu inférieure à  $10^{-4}$  à  $Re = 1000$  la PGD nécessite plus de 70 fonctions contre une trentaine pour les SVD. Dans le cas de la vorticité, à  $Re = 100$  la courbe obtenue par PGD est identique à celles obtenues par SVD jusqu'à une norme du résidu égale à  $10^{-2}$  alors que pour  $Re = 1000$  cette valeur est de  $10^{-4}$ . Une fois ces valeurs seuil dépassées la PGD cesse d'être optimale. Il semblerait que le fait d'augmenter le nombre de Reynolds ait tendance à rendre la méthode PGD optimale jusqu'à une plus grande précision pour le calcul de la vorticité.

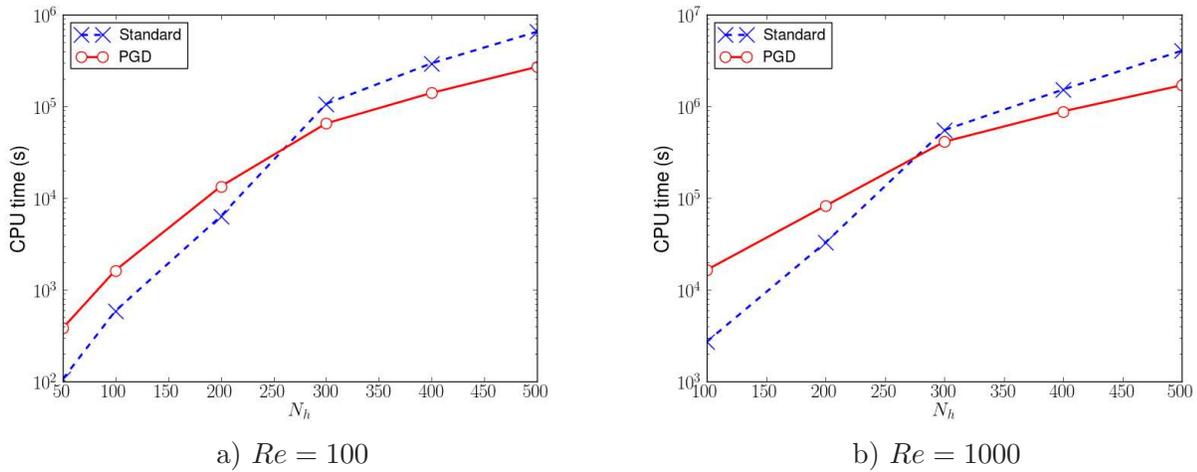


FIG. A.4 – Comparaison des temps de calcul entre la méthode PGD et la méthode standard pour la résolution de la cavité entraînée en formulation ligne de courant-vorticité pour  $Re = 100$  et  $Re = 1000$ .

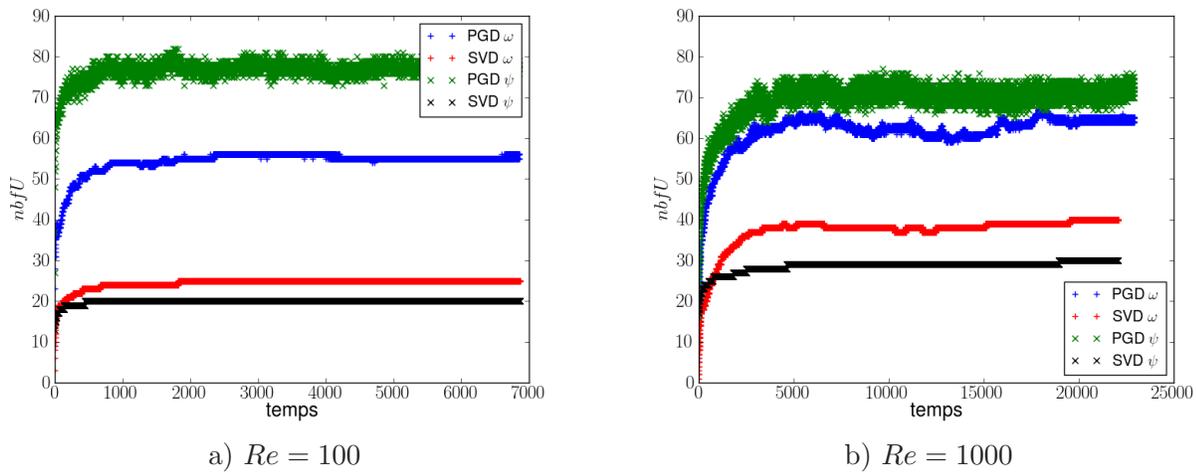


FIG. A.5 – Nombre de fonctions obtenues par PGD et par SVD sur les résultats de la méthode standard pour la fonction ligne de courant et la vorticité à  $Re = 100$  et  $Re = 1000$  ( $N_h = 200$ ).

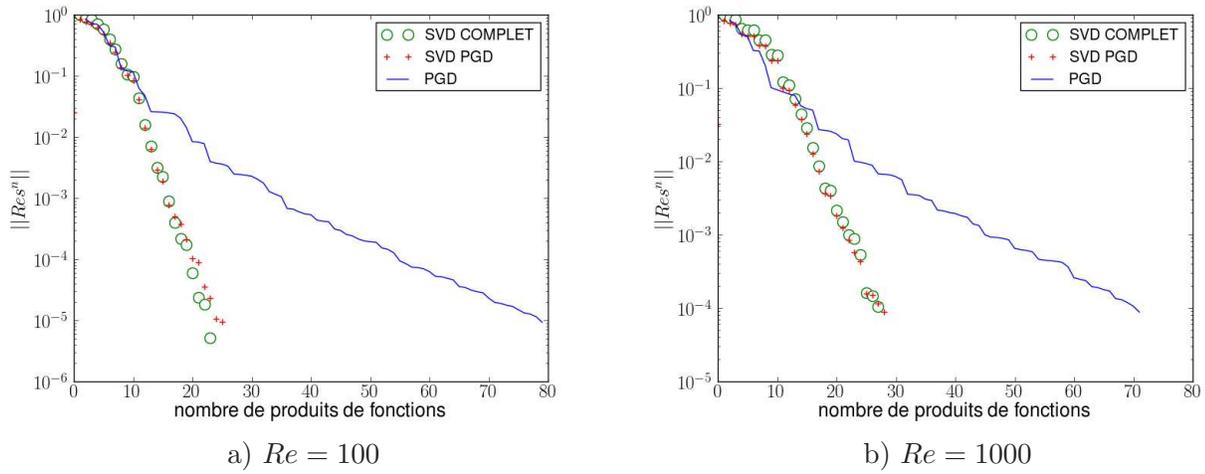


FIG. A.6 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la fonction lignes de courants  $\psi$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 200$ ).

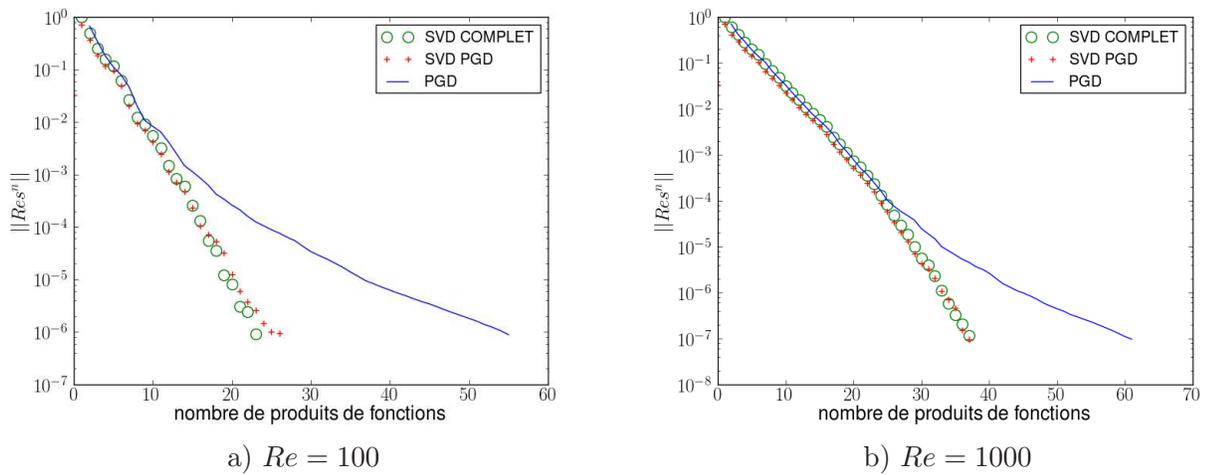


FIG. A.7 – Norme du résidu en fonction du nombre de fonctions pour chacune des décomposition PGD et SVD pour la vorticité  $\omega$  et pour  $Re = 100$  et  $Re = 1000$  ( $N_h = 200$ ).

### III .2 Régime périodique ( $Re = 10.000$ )

Comme mentionné dans le chapitre 3 section IV .2, à  $Re = 10000$ , l'écoulement est périodique. La période observée à partir des simulations PGD et avec le modèle standard est de  $T = 1.5s$ . Cette période est légèrement différente (erreur de 10%) de celle obtenue par Bruneau et Saad. [Bruneau et Saad, 2006]. Comme les résultats PGD et les résultats du modèle complet sont similaires, cette différence ne peut être imputée à la méthode PGD. Elle est très certainement due aux schémas de discrétisation utilisés.

Les lignes de courants sont représentées sur les figures (A.8) et (A.9) pour chacun des solveurs étudiés. Une très bonne correspondance est observée entre les résultats obtenus par méthode PGD et ceux obtenus à l'aide du solveur standard. De plus, ces lignes de courants ont un comportement similaires à celles obtenus par Bruneau et al. toujours dans [Bruneau et Saad, 2006]. Dans ce cas, le temps de calcul nécessaire à la résolution PGD est du même ordre de grandeur que le temps de calcul nécessaire à la résolution par solveur standard.

L'évolution de la norme des résidus associés aux équations vérifiées par  $\psi$  et  $\omega$  est tracée à  $T = 0$  sur la figure A.10. Dans ce cas une très bonne correspondance entre les résultats PGD et SVD pour la vorticit  est not e, ce qui implique qu'   $Re = 10000$ , le calcul de la vorticit  est r alis  de mani re optimal. Concernant le calcul de la fonction ligne de courant, l' volution de la norme des r sidus est globalement identique   celle des cas o   $Re = 100$  et  $Re = 1000$ .

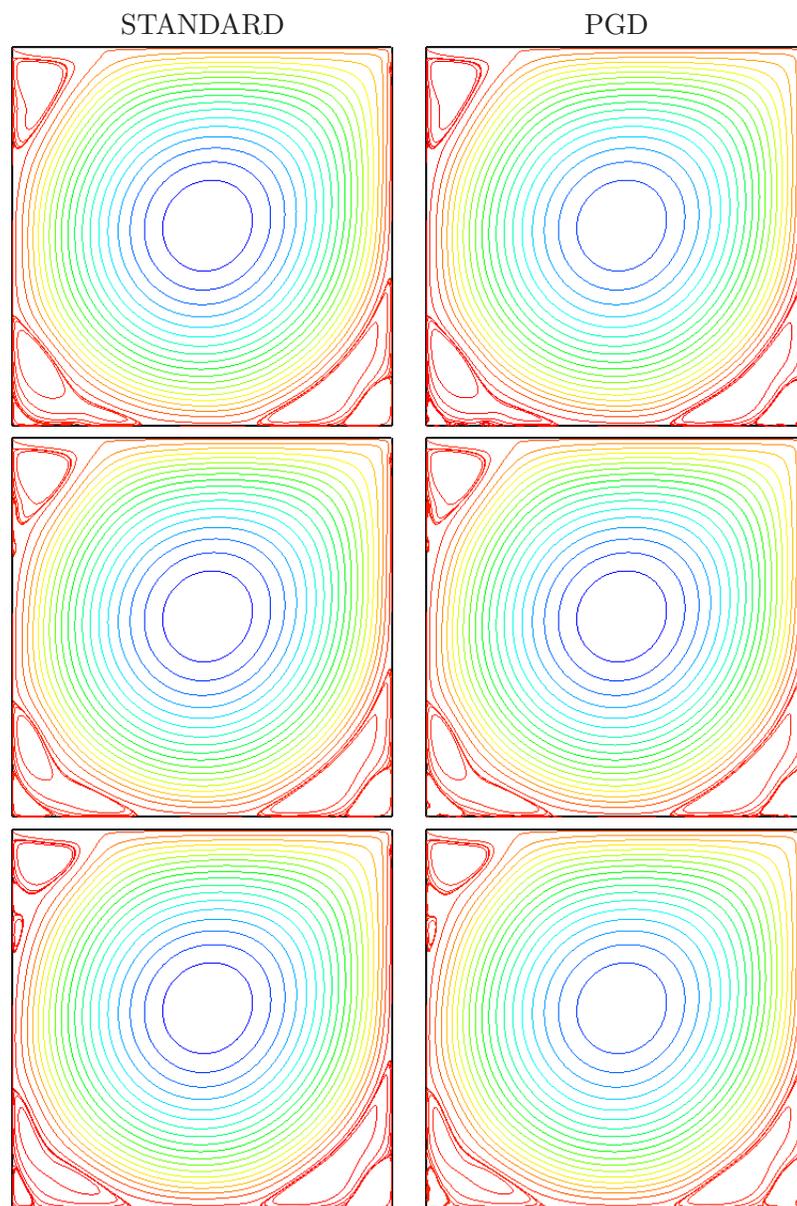


FIG. A.8 – Evolution des lignes de courant durant une période pour  $Re = 10000$  avec un maillage  $250 \times 250$ . De haut en bas pour chacun des solveurs (standard et PGD), les temps  $t = 0$ ,  $t = 0.2T$  et  $t = 0.4T$  sont représentés.

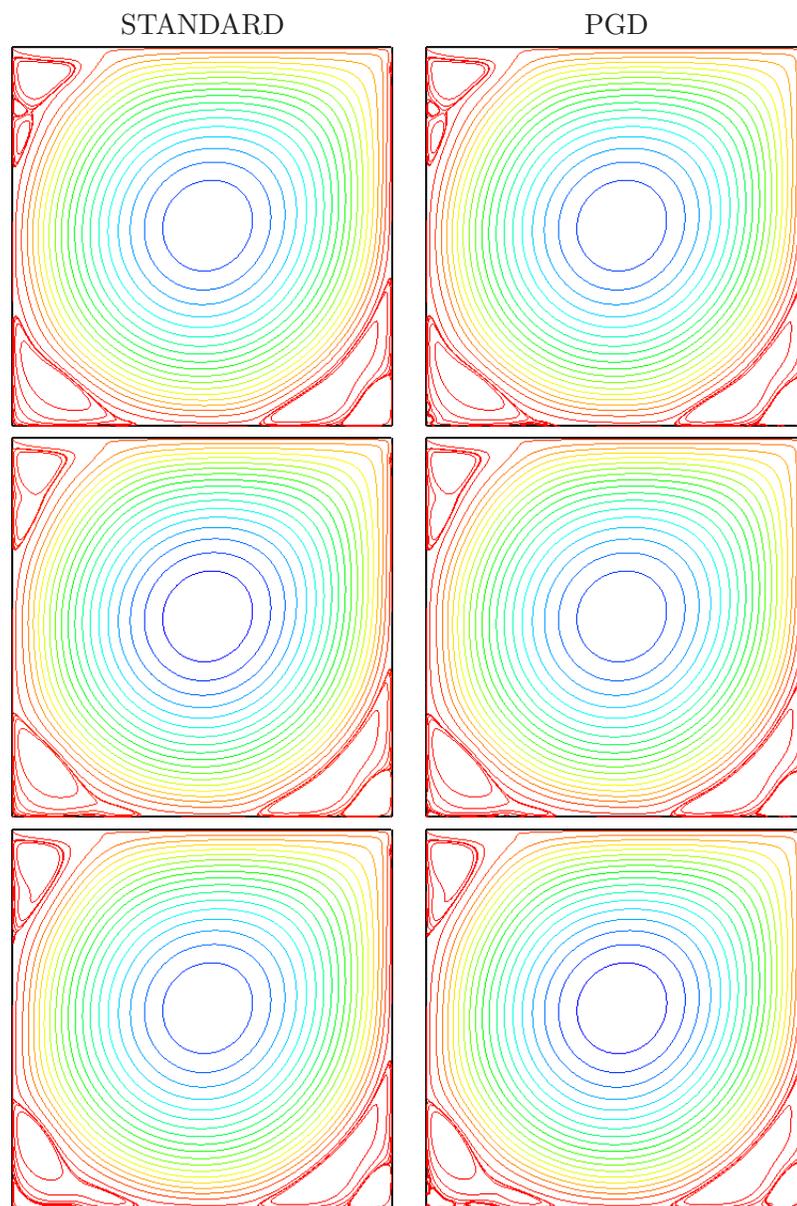


FIG. A.9 – Evolution des lignes de courant durant une période pour  $Re = 10000$  avec un maillage  $250 \times 250$ . De haut en bas pour chacun des solveurs (standard et PGD), les temps  $t = 0.6T$ ,  $t = 0.8T$  et  $t = T$  sont représentés.

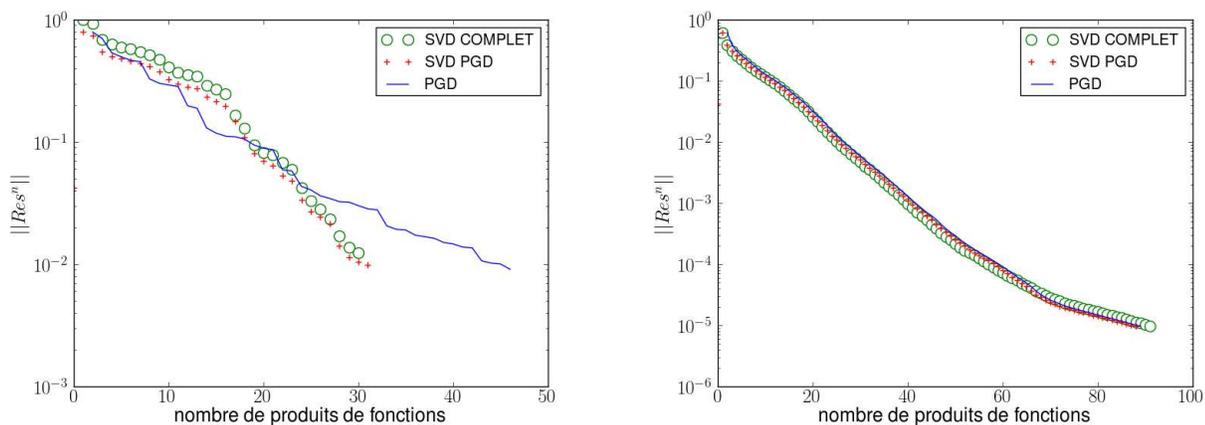


FIG. A.10 – Norme du résidu en fonction du nombre de fonctions de chacune des décomposition PGD, SVD sur le résultat PGD et SVD sur le résultats du modèle complet pour la fonction lignes de courants  $\psi$  (gauche) et la vorticité  $\omega$  (droite) et pour  $Re = 10000$  ( $N_h = 250$ ).

## IV Conclusions

Dans cette section, la PGD a été appliquée à la résolution des équations de Navier-Stokes avec une formulation ligne de courant-vorticité pour un écoulement dans une cavité entraînée 2D isotherme. Dans le cas stationnaire ( $Re = 100$  et  $Re = 1000$ ), les résultats obtenus par PGD sont en accords avec ceux obtenus par le modèle complet ainsi qu’avec les résultats de la littérature. De plus, la PGD permet d’obtenir un gain de temps (gain de temps d’un facteur deux pour  $Re = 100$  et quatre pour  $Re = 1000$  pour  $N_h = 500$ ) à partir de 300 nœuds dans chaque direction. Concernant le cas instationnaire, la PGD permet de retrouver une période similaire à celle obtenue dans la littérature et un comportement des lignes de courants pendant cette même période comparable aux résultats du modèle complet. Cependant un gain de temps peu significatif a été observé.

## Annexe B

# PGD appliquée à l'équation de Poisson 2D stationnaire

Afin d'illustrer la méthode PGD, nous allons traiter l'exemple simple d'une équation de Poisson stationnaire en deux dimensions.

Le problème de Poisson 2D, ou équation de diffusion stationnaire 2D, peut s'écrire de la manière suivante :

$$\text{Chercher } T(x,y) \text{ tel que } \begin{cases} \nabla^2 T = f & \text{dans } \Omega = ]-L, L[ \times ]-L, L[ \\ + \text{conditions limites} \end{cases} \quad (\text{B.1})$$

On suppose que le second membre  $f(x, y)$  peut s'écrire sous forme séparée, soit :

$$f(x, y) = \sum_{i=1}^{n_f} f_x^i f_y^i \quad (\text{B.2})$$

Concrètement, cette séparation de la représentation du second membre n'est pas nécessaire mais cela permet de garder le caractère séparé du problème. Ici, nous allons expliciter la méthode PGD dans le cas d'une résolution à l'aide de la méthode des volumes finis et d'une formulation PGD progressive pondérée. La méthode sera appliquée dans le contexte continu. La solution  $T$  est recherchée sous la forme séparée suivante :

$$T(x, y) \approx T_m(x, y) = \sum_{i=1}^m \alpha^i F^i(x) G^i(y) \quad (\text{B.3})$$

En injectant la forme séparée de  $T_m$  dans le problème (B.1), et après intégration sur l'espace<sup>1</sup>, on obtient :

$$\int_{\Omega} \nabla^2 \left( \sum_{i=1}^m \alpha^i F^i(x) G^i(y) \right) d\Omega = \int_{\Omega} \sum_{i=1}^{n_f} f_x^i f_y^i d\Omega + \int_{\Omega} Res^m d\Omega \quad (\text{B.4})$$

que l'on peut aussi écrire,

$$\sum_{i=1}^m \alpha^i \int_{\Omega} \left[ \frac{d^2 F^i(x)}{dx^2} G^i(y) + F^i(x) \frac{d^2 G^i(y)}{dy^2} \right] d\Omega = \sum_{i=1}^{n_f} \int_{\Omega} f_x^i f_y^i d\Omega + \int_{\Omega} Res^m d\Omega \quad (\text{B.5})$$

Ainsi on a

$$\begin{aligned} \sum_{i=1}^m \alpha^i & \left[ \int_X \frac{d^2 F^i(x)}{dx^2} dx \int_Y G^i(y) dy + \int_X F^i(x) dx \int_Y \frac{d^2 G^i(y)}{dy^2} dy \right] \\ & = \sum_{i=1}^{n_f} \int_X f_x^i dx \int_Y f_y^i dy + \int_{\Omega} Res^m d\Omega \end{aligned} \quad (\text{B.6})$$

où la notation suivante a été utilisée,

$$\int_X \cdot dx = \int_{x=-L}^{x=+L} \cdot dx \quad \text{et} \quad \int_Y \cdot dy = \int_{y=-L}^{y=+L} \cdot dy \quad (\text{B.7})$$

<sup>1</sup>Dans la suite du document, la formulation volume finis est utilisée pour discrétiser les différents problèmes. C'est pourquoi nous avons fait apparaître une intégration sur l'espace.

Nous allons maintenant détailler les trois étapes de résolution de l'algorithme PGD progressive pondérée, nous sommes à l'étape  $m$ , et la solution  $T_{m-1}(x, y)$  de l'étape précédente définie par l'équation (B.3) est connue, on cherche à calculer la solution au rang  $m$ ,  $T_m(x, y)$ .

## I Etape d'enrichissement

A cette étape, la solution n'est pas assez précise, elle est donc enrichie en lui ajoutant un produit de fonctions, c'est à dire que la solution  $T_m(x, y)$  sera recherchée sous la forme suivante :

$$T_m(x, y) = \sum_{i=1}^{m-1} \alpha^i F^i(x) G^i(y) + R(x) S(y) \quad (\text{B.8})$$

où les inconnues sont les deux fonctions  $R$  et  $S$ . Le problème (B.1) s'écrit donc :

$$\begin{aligned} & \int_{\Omega} \left[ \frac{d^2 R(x)}{dx^2} S(y) + R(x) \frac{d^2 S(y)}{dy^2} \right] d\Omega \\ &= - \sum_{i=1}^{m-1} \alpha^i \int_{\Omega} \left[ \frac{d^2 F^i(x)}{dx^2} G^i(y) + F^i(x) \frac{d^2 G^i(y)}{dy^2} \right] d\Omega + \int_{\Omega} \left( \sum_{i=1}^{n_f} f_x^i f_y^i \right) d\Omega + \int_{\Omega} Res^{m-1} d\Omega \end{aligned} \quad (\text{B.9})$$

Comme décrit dans la partie précédente, ce système non-linéaire sera résolu par la méthode du point fixe, en imposant  $(Res^{m-1}, S)_Y = 0$  pour le calcul de  $R$ , et  $(Res^{m-1}, R)_X = 0$  pour le calcul de  $S$ . La notation suivante est adoptée pour la définition des produits scalaires :

$$(f, g)_X = \int_{x=-L}^{x=+L} f g dx \quad (f, g)_Y = \int_{y=-L}^{y=+L} f g dy \quad (\text{B.10})$$

Ainsi en projection sur  $S$ , l'équation (B.9) s'écrit :

$$\int_X \frac{d^2 R(x)}{dx^2} dx \underbrace{(S, S)_Y}_{a \in \mathbb{R}} + \int_X R(x) dx \underbrace{\left( \frac{d^2 S}{dy^2}, S \right)_Y}_{b \in \mathbb{R}} = S_X \quad (\text{B.11})$$

De même, en projection sur  $R$ , l'équation (B.9) s'écrit :

$$\underbrace{\left( \frac{d^2 R}{dx^2}, R \right)_X}_{a \in \mathbb{R}} \int_Y S(y) dy + \underbrace{(R, R)_X}_{b \in \mathbb{R}} \int_Y \frac{d^2 S(y)}{dy^2} dy = S_Y \quad (\text{B.12})$$

Les second membres précédents s'écrivent de la manière suivante :

$$\begin{aligned} S_X &= - \sum_{i=1}^{m-1} \alpha^i \left[ (G^i, S)_Y \int_X \frac{d^2 F^i}{dx^2} dx + \left( \frac{d^2 G^i}{dy^2}, S \right)_Y \int_X F^i dx \right] + (f_y^i, S)_Y \sum_{i=1}^{n_f} \int_X f_x^i dx \\ S_Y &= - \sum_{i=1}^{m-1} \alpha^i \left[ \left( \frac{d^2 F^i}{dx^2}, R \right)_X \int_Y G^i dy + (F^i, R)_X \int_Y \frac{d^2 G^i}{dy^2} dy \right] + \sum_{i=1}^{n_f} (f_x^i, R)_X \int_Y f_y^i dy \end{aligned}$$

Les équations B.11 et B.12 sont résolues avec la méthode du point fixe. A convergence de ce point fixe, les nouvelles fonctions  $F^n$  et  $G^n$  sont prises égales à la norme de  $R$  et de  $S$ .

## II Etape de projection

Cette étape consiste à calculer les  $m$  coefficients  $\alpha_i$  correspondant à l'ensemble des produits de fonctions  $\{F^i G^i\}_{1 \leq i \leq m}$ . Pour cela nous allons imposer que  $(Res^m, F^j G^j)_\Omega = 0$  pour  $1 \leq j \leq m$ . En projetant l'équation (B.4) selon les  $F^j(x)G^j(y)$  on obtient, le système linéaire suivant de taille  $m \times m$  à résoudre :

$$\mathbb{M}_{ij} \alpha^i = b_j \quad (\text{B.13})$$

où

$$\mathbb{M}_{ij} = \left( \frac{d^2 F^i}{dx^2}, F^j \right)_X (G^i, G^j)_Y + (F^i, F^j)_X \left( \frac{d^2 G^i}{dy^2}, G^j \right)_Y \quad (\text{B.14})$$

et

$$b_j = - \sum_{i=1}^{n_f} (f_x^i, F^j)_X (f_y^i, G^j)_Y \quad (\text{B.15})$$

Une fois les  $m$  coefficients  $\alpha^i$  calculés il est possible de définir la solution d'ordre  $m$  de la manière suivante

$$T_m(x, y) = \sum_{i=1}^m \alpha^i F^i(x) G^i(y) \quad (\text{B.16})$$

## III Etape d'estimation de la convergence

Le résidu est donc maintenant d'ordre  $m$  et est calculé de la manière suivante :

$$Res^m = \nabla^2 T_m - f(x, y) \quad (\text{B.17})$$

soit

$$Res^m = \nabla^2 \sum_{i=1}^m \alpha^i F^i(x) G^i(y) - \sum_{i=1}^{n_f} f_x^i f_y^i \quad (\text{B.18})$$

Si la norme  $L^2$  de ce résidu est inférieure à une erreur  $\epsilon$  fixée par l'utilisateur, alors on estime que l'algorithme a convergé, et donc que la solution peut s'écrire comme dans l'équation (B.16), sinon il faut enrichir notre base de fonctions avec un nouveau produit de fonctions.

# Bibliographie

- [Al-Amiri et al., 2007] Al-Amiri, A. M., Khanafer, K. M., et Pop, I. (2007). Numerical simulation of combined thermal and mass transport in a square lid-driven cavity. *International Journal of Thermal Sciences*, 46(7) :662 – 671.
- [Alleborn et al., 1999] Alleborn, N., Raszillier, H., et Durst, F. (1999). Lid-driven cavity with heat and mass transport. *International Journal of Heat and Mass Transfer*, 42(5) :833 – 853.
- [Allery et al., 2008] Allery, C., Beghein, C., et Hamdouni, A. (2008). On investigation of particle dispersion by a pod approach. *International Applied Mechanics*, 44 :110–119. 10.1007/s10778-008-0025-2.
- [Allery et al., 2005] Allery, C., Béghein, C., et Hamdouni, A. (2005). Applying proper orthogonal decomposition to the computation of particle dispersion in a two-dimensional ventilated cavity. *Communications in Nonlinear Science and Numerical Simulation*, 10(8) :907 – 920.
- [Allery et al., 2004] Allery, C., Guerin, S., Hamdouni, A., et Sakout, A. (2004). Experimental and numerical pod study of the coanda effect used to reduce self-sustained tones. *Mechanics Research Communications*, 31(1) :105 – 120.
- [Allery et al., 2011] Allery, C., Hamdouni, A., Ryckelynck, D., et Verdon, N. (2011). A priori reduction method for solving the two-dimensional burgers' equations. *Applied Mathematics and Computation*, 217(15) :6671 – 6679.
- [Ammar et al., ] Ammar, A., Chinesta, F., et Cueto, E. Coupling finite elements and separated representation. *International Journal of Multiscale Computational Engineering*, *In press*.
- [Ammar et al., 2010a] Ammar, A., Chinesta, F., Diez, P., et Huerta, A. (2010a). An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28) :1872–1880.
- [Ammar et al., 2010b] Ammar, A., Chinesta, F., Diez, P., et Huerta, A. (2010b). An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199 :1872–1880.
- [Ammar et al., 2010c] Ammar, A., Chinesta, F., et Falco, A. (2010c). On the Convergence of a Greedy Rank-One Update Algorithm for a Class of Linear Systems. *Archives Of Computational Methods In Engineering*, 17(4) :473–486.
- [Ammar et al., 2006a] Ammar, A., Mokdad, B., Chinesta, F., et Keunings, R. (2006a). A new family of solvers for some classes of multidimensionnal partial differential equations

- encountered in kinetic theory modeling of complex fluids. *J. of Non Newtonian Fluid Mechanics*, 139 :153–176.
- [Ammar et al., 2007] Ammar, A., Mokdad, B., Chinesta, F., et Keunings, R. (2007). A new family of solvers for some classes of multidimensionnal partial differential equations encountered in kinetic theory modeling of complex fluids. part ii : Transient simulation using space-time separated representations. *J. of Non Newtonian Fluid Mechanics*, 144 :98–121.
- [Ammar et al., 2010d] Ammar, A., Normandin, M., et Chinesta, F. (2010d). Solving parametric complex fluids models in rheometric flows. *Journal of Non-Newtonian Fluid Mechanics*, 165(23-24) :1588–1601.
- [Ammar et al., 2010e] Ammar, A., Normandin, M., Daim, F., Gonzalez, D., Cueto, E., et Chinesta, F. (2010e). Non incremental strategies based on separated representations : Applications in computational rheology. *Communications in Mathematical Sciences*, 8(3, Sp. Iss. SI) :671–695.
- [Ammar et al., 2006b] Ammar, A., Ryckelynck, D., Chinesta, F., et Keunings, R. (2006b). On the reduction of kinetic theory models related to finitely extensible dumbbells. *Journal Non-Newtonian Fluid Mech*, 134.
- [Antoulas, 2005a] Antoulas, A. (2005a). A new results on on passivity preserving model reduction. *Systems and Control Letters*, 54.
- [Antoulas, 2005b] Antoulas, A. (2005b). An overview of approximation methods for large-scale dynamical systems. *Annual reviews in Control*, 29.
- [Aubry et al., 1988] Aubry, N., Holmes, P., Lumley, J., et Stone, E. (1988). The dynamics of coherent structures in the wall regions of a turbulent wall layer. *Journal of Fluid Mechanics*, 192 :115–173.
- [Aubry et al., 1993] Aubry, N., Lian, W., et Titi, E. (1993). Preserving symmetries in the proper orthogonal decomposition. *SIAM J.Sci.Comput*, 14 :483–505.
- [Azaïez, 1990] Azaïez, M. (1990). *Calcul de la pression dans le problème de Stokes pour des fluides visqueux incompressibles par une méthode spectrale de collocation*. thèse de Doctorat, Paris-sud.
- [Azaiez et al., 1994] Azaiez, M., Fikri, A., et Labrosse, G. (1994). A unique grid spectral solver of the nd cartesian unsteady strokes system. illustrative numerical results. *Finite Elements in Analysis and Design*, 16(3-4) :247 – 260.
- [Basak et al., 2010] Basak, T., Roy, S., Singh, S. K., et Pop, I. (2010). Analysis of mixed convection in a lid-driven porous square cavity with linearly heated side wall(s). *International Journal of Heat and Mass Transfer*, 53(9-10) :1819 – 1840.
- [Bennacer et Gobin, 1996a] Bennacer, R. et Gobin, D. (1996a). Cooperating thermosolutal convection in enclosures–i. scale analysis and mass transfer. *International Journal of Heat and Mass Transfer*, 39(13) :2671 – 2681.
- [Bennacer et Gobin, 1996b] Bennacer, R. et Gobin, D. (1996b). Cooperating thermosolutal convection in enclosures–ii. heat transfer and flow structure. *International Journal of Heat and Mass Transfer*, 39(13) :2683 – 2697.

- [Bergooz et Titi, 1993] Bergooz, G. et Titi, E. (1993). Galerkin projections and the proper orthogonal decomposition for equivariant equations. *Phys. Lett, A* 174 :94–102.
- [Berkooz et al., 1993] Berkooz, G., Holmes, P., et Lumley, J. (1993). The proper orthogonal decomposition in turbulent flows. *Ann.Rev.Fluid.Mech*, 25 :539–575.
- [Bernardi et Maday, 1988] Bernardi, C. et Maday, Y. (1988). A collocation method over staggered grids for the stokes problem. *Int. J. Num. Meth. Fluids*, 8 :537 – 557.
- [Bonithon et al., 2011] Bonithon, G., Joyot, P., Chinesta, F., et Villon, P. (2011). Non-incremental boundary element discretization of parabolic models based on the use of the proper generalized decompositions. *Engineering Analysis with Boundary Elements*, 35(1) :2 – 17.
- [Botella, 1996] Botella, O. (1996). Résolution des équations de navier-stokes par des schémas de projection tchebychev. *Rapport de Recherche INRIA*, 3018.
- [Brezzi, 1974] Brezzi, F. (1974). On the existences, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *R.A.I.R.O, R.2*, pages 129–151.
- [Bruneau et Saad, 2006] Bruneau, C.-H. et Saad, M. (2006). The 2d lid-driven cavity problem revisited. *Computers and Fluids*, 35(3) :326 – 348.
- [Burkardt et al., 2006] Burkardt, J., Gunzburger, M., et Lee, H. (2006). Pod and cvt-based reduced order modeling of navier-stokes flows. *Comput. Methods Appl. Mech. Engrg*, 196 :337–355.
- [Caignot et al., 2010] Caignot, A., Ladevèze, P., et Neron, D. (2010). Virtual testing for the prediction of damping in joints. *Engineering Computations*, 27 :621 –644.
- [Canuto et al., 1986] Canuto, C., Bernadi, C., et Maday, y. (1986). Generalized inf-sup conditions for chebychev approximations of the navier-stokes equations. *Technical Report, No. 86-61, ICASE*.
- [Canuto et al., 1988] Canuto, C., Hussaini, M. Y., et Quarteroni, A. (1988). Spectral methods in fluid dynamics. *Springer-Verlag, New York*.
- [Champaney et al., 2008] Champaney, L., Boucard, P., et Guinard, S. (2008). Adaptive multi-analysis strategy for contact problems with friction. *Computational Mechanics*, 42(2) :305–315.
- [Champaney et al., 1999] Champaney, L., Cognard, J. Y., et Ladèveze, P. (1999). Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions. *Computers and Structures*, 73(1-5) :249 – 266.
- [Cheng et Liu, 2010] Cheng, T. et Liu, W.-H. (2010). Effect of temperature gradient orientation on the characteristics of mixed convection flow in a lid-driven square cavity. *Computers and Fluids*, 39(6) :965 – 978.
- [Chinesta et al., 2010a] Chinesta, F., Ammar, A., et Cueto, E. (2010a). Proper generalized decomposition of multiscale models. *International Journal of Numerical Methods in Engineering*, 83(8-9) :1114–1132.
- [Chinesta et al., 2010b] Chinesta, F., Ammar, A., et Cueto, E. (2010b). Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, 83(8-9, Sp. Iss. SI) :1114–1132.

- [Chinesta et al., 2010c] Chinesta, F., Ammar, A., et Cueto, E. (2010c). Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models. *Archives of Computational methods in Engineering*, 17(4) :327–350.
- [Chinesta et al., 2008] Chinesta, F., Ammar, A., et Joyot, P. (2008). The nanometric and micrometric scales of the structure and mechanics of materials revisited : An introduction to the challenges of fully deterministic numerical descriptions. *International Journal for Multiscale Computational Engineering*, 6/3 :191–213.
- [Chorin, 1968] Chorin, A. (1968). Numerical solution of the navier-stokes equations. *Math. Comput.*, 22 :745–762.
- [Chudanov et al., 1995] Chudanov, V. V., Popkov, A. G., Churbanov, A. G., Vabishchevich, P. N., et Makarov, M. M. (1995). Operator-splitting schemes for the stream function-vorticity formulation. *Computers and Fluids*, 24(7) :771 – 786.
- [Couplet et al., 2005] Couplet, M., Basdevant, C., et Sagaut, P. (2005). Calibrated reduced-order (pod) galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207 :192–220.
- [Cresta et al., 2007] Cresta, P., Allix, O., Rey, C., et Guinard, S. (2007). Nonlinear localization strategies for domain decomposition methods : Application to post-buckling analyses. *Computer Methods in Applied Mechanics and Engineering*, 196(8) :1436 – 1446.
- [Ding et al., 2006] Ding, H., Shu, C., Yeo, K., et Xu, D. (2006). Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadric differential quadrature method. *Computer Methods in Applied Mechanics and Engineering*, 195(7-8) :516 – 533.
- [Dowell et al., 1997] Dowell, E., Hall, K., et Romanowski, M. (1997). Eigenmode analysis in unsteady aerodynamics : Reduced order models. *Applied Mechanics reviews.*, 50.
- [Du et al., 1999] Du, Q., Faber, V., et Gunzburger, M. (1999). Centroidal voronoi tessellations : applications and algorithms. *SIAM Rev.*, 41 :637–676.
- [Du et Gunzburger, 2002a] Du, Q. et Gunzburger, M. (2002a). Grid generation and optimization based on centroidal voronoi tessellations. *Appl. Math. Comput.*, 133 :591–607.
- [Du et Gunzburger, 2002b] Du, Q. et Gunzburger, M. (2002b). Model reduction by proper orthogonal decomposition coupled with centroidal voronoi tessellations : applications and algorithms. Proc.Fluids Engineering Division Summer Meeting, FEDSM2002-31051, ASME.
- [Dumon et al., 2010] Dumon, A., Allery, C., et Ammar, A. (2010). Proper generalized decomposition method for incompressible flows in stream-vorticity formulation. *EJCM - Computational methods for coupled fluid and structural problems*, 19(5-6-7) :591 – 617.
- [Dumon et al., 2011] Dumon, A., Allery, C., et Ammar, A. (2011). Proper general decomposition (pgd) for the resolution of navier-stokes equations. *Journal of Computational Physics*, 230(4) :1387 – 1407.
- [Faure, 2003] Faure, S. (2003). Colocated finite volume schemes for fluid flows. *PhD thesis, University of Paris sud.*

- [Gadoin et al., 2001] Gadoin, E., Le Quéré, P., et Daube, O. (2001). A general methodology for investigating flow instabilities in complex geometries : application to natural convective in enclosures. *International Journal for Numerical methods in fluids.*, 37.
- [Ghia et al., 1982] Ghia, U., Ghia, K., et Shin, C. (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *J. of Computational physics*, 48 :387 – 411.
- [Goda, 1979] Goda, K. (1979). A multi-step technique with implicit difference schemes for calculating two or three dimensional cavity flows. *J.Comput.Phys.*, 30 :76–95.
- [Gonzalez et al., 2010] Gonzalez, D., Ammar, A., Chinesta, F., et Cueto, E. (2010). Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81(5) :637–659.
- [Guermond et al., 2006] Guermond, J., Mineev, P., et Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47) :6011 – 6045.
- [Gugercin et Antoulas, 2006] Gugercin, S. et Antoulas, A. (2006). Model reduction of large scale systems by least squares. *Linear Algebra and its Applications*, 415.
- [Guidault et al., 2008] Guidault, P.-A., Allix, O., Champaney, L., et Cornuault, C. (2008). A multiscale extended finite element method for crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 197(5) :381 – 399. Enriched Simulation Methods and Related Topics.
- [Haldenwang, 1984] Haldenwang, P. (Décembre 1984). *Thèse d'état, Université de Provence.*
- [Hall et al., 1995] Hall, K., Florea, R., et Lanzkron, P. (1995). A reduced order model of unsteady flows in turbomachinery. *Transactions-ASME : Journal of Turbomachinery.*, 117.
- [Holmes et al., 1996] Holmes, P., Lumley, J., et Berkooz, G. (1996). Turbulence, coherent structures, dynamical systems and symmetry. Cambridge University, Cambridge.
- [Holmes et al., 1997] Holmes, P., Lumley, J., Berkooz, G., attingky, J., et Wittenberg, R. (1997). Low-dimensionals models of coherent structures in turbulence. *Phys. Rep*, 287 :337–384.
- [Ilak et Rowley, 2006] Ilak, M. et Rowley, C. (2006). Reduced-order modeling of channel flow using traveling pod and balanced pod. *IN. Collectio of Technical Papers - 3rd AIAA Flow Control Conference*, 2.
- [Ito et Ravindran, 1998] Ito, K. et Ravindran, S. (1998). A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics.*, 143.
- [Iwatsu et Hyun, 1995] Iwatsu, R. et Hyun, J. M. (1995). Three-dimensional driven-cavity flows with a vertical temperature gradient. *International Journal of Heat and Mass Transfer*, 38(18) :3319 – 3328.
- [Ju et al., 2002] Ju, L., Du, Q., et Gunzburger, M. (2002). Probabilistics methods for centroidal voronoi tessellations and their parallel implementations. *J. Parallel Comput.*, 28.
- [Kerfriden et al., 2010] Kerfriden, P., Gosselet, P., Adhikari, S., et Bordas, S. (2010). Bridging proper orthogonal decomposition methods and augmented newton-krylov algorithms :

- an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, In Press, Accepted Manuscript :-.
- [Khanafar et al., 2007] Khanafar, K. M., Al-Amiri, A. M., et Pop, I. (2007). Numerical simulation of unsteady mixed convection in a driven cavity using an externally excited sliding lid. *European Journal of Mechanics - B/Fluids*, 26(5) :669 – 687.
- [Kim et al., 2007] Kim, Y., Kim, D. W., Jun, S., et Lee, J. H. (2007). Meshfree point collocation method for the stream-vorticity formulation of 2d incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34) :3095 – 3109.
- [Ladevèze, 1999] Ladevèze, P. (1999). New approaches and non-incremental methods of calculation. *Nonlinear Computational Structural Mechanics*, Springer Verlag.
- [Ladevèze et Nouy, 2003] Ladevèze, P. et Nouy, A. (2003). On a multiscale computational strategy with time and space homogenization for structural mechanics. *Compu. Methods Appl. Mech. Engrg*, 192.
- [Lamari et al., 2010] Lamari, H., Ammar, A., Cartraud, P., Legrain, G., Chinesta, F., et Jacquemin, F. (2010). Routes for Efficient Computational Homogenization of Nonlinear Materials Using the Proper Generalized Decompositions. *Archives of Computational methods in Engineering*, 17(4) :373–391.
- [Liberge et Hamdouni, 2010] Liberge, E. et Hamdouni, A. (2010). Reduced order modelling method via proper orthogonal decomposition (pod) for flow around an oscillating cylinder. *Journal of Fluids and Structures*, 26(2) :292 – 311.
- [Lo et al., 2007] Lo, D., Young, D., Murugesan, K., Tsai, C., et Gou, M. (2007). Velocity-vorticity formulation for 3d natural convection in an inclined cavity by dq method. *International Journal of Heat and Mass Transfer*, 50(3-4) :479 – 491.
- [Lumley, 1967] Lumley, J. (1967). The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation in A.M. Yaglom Taransky*, pages 166–178.
- [Maday et al., 1993] Maday, Y., Meiron, D., Patera, A. T., et Rønquist, E. M. (1993). Analysis of iterative methods for the steady and unsteady stokes problem : Application to spectral element discretizations. *SIAM Journal on Scientific Computing*, 14(2) :310–337.
- [Maday et al., 1987] Maday, Y., Patera, A., et Rønquist, M. (1987). A well-posed optimal spectral element approximation for the stokes problem. *Technical report, ICASE, Hampton, VA*,.
- [Maday et al., 1992] Maday, Y., Patera, A. T., et Rønquist, E. M. (1992). the  $\mathbb{P}_N - \mathbb{P}_{N-2}$  method for the approximation of the stokes problem. *Laboratoire d'Analyse Numérique, Paris VI*, 11.
- [Mohamad et Viskanta, 1995] Mohamad, A. A. et Viskanta, R. (1995). Flow and heat transfer in a lid-driven cavity filled with a stably stratified fluid. *Applied Mathematical Modelling*, 19(8) :465 – 472.
- [Montigny-Rannou et Morchoisne, 1987] Montigny-Rannou, F. et Morchoisne, Y. (1987). A spectral method with staggered grid for incompressible navier-stokes equations. *Int. J. Num. Meth. Fluids*, 7 :175 – 189.

- [Moore, 1981] Moore, B. (1981). Principal component analysis in linear systems-controllability, observation, and model reduction. *IEEE Transactions and Automatic Control AC*, 26.
- [Morchoisne, 1983] Morchoisne, Y. (1983). Résolution des équations de navier-stokes par une méthode spectrale de sous-domaines. *3 ème Cong. Int. Meth. Num/ Ing., Paris, Ed P.Lascaux*, pages 275 – 281.
- [Noor et al., 2009] Noor, D. Z., Kanna, P. R., et Chern, M.-J. (2009). Flow and heat transfer in a driven square cavity with double-sided oscillating lids in anti-phase. *International Journal of Heat and Mass Transfer*, 52(13-14) :3009 – 3023.
- [Nouy, 2007] Nouy, A. (2007). A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Comput.Meth.Appl.Mech.Engrg*, 196 :4251–4537.
- [Nouy, 2008] Nouy, A. (2008). Generalized spectral decomposition method for solving stochastic finite element equations : invariant subspace problem and dedicated algorithms. *Comput.Meth.Appl.Mech.Engrg*, 197 :4718–4736.
- [Nouy, 2010] Nouy, A. (2010). A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24) :1603 – 1626.
- [Nouy et Ladevèze, 2004] Nouy, A. et Ladevèze, P. (2004). Multiscale computational strategy with time and space homogenization : a radial-type approximation technique for solving micro problems. *Int. J. Multiscale Comput. Engrg*, 170.
- [Nouy et Le Maître, 2007] Nouy, A. et Le Maître, O. (2007). Generalized spectral decompositions for stochastic nonlinear problems. *J.Comput.Phys*, 8 :283–288.
- [Ouertatani et al., 2009] Ouertatani, N., Cheikh, N. B., Beya, B. B., Lili, T., et Campo, A. (2009). Mixed convection in a double lid-driven cubic cavity. *International Journal of Thermal Sciences*, 48(7) :1265 – 1272.
- [Oztop et Dagtekin, 2004] Oztop, H. F. et Dagtekin, I. (2004). Mixed convection in two-sided lid-driven differentially heated square cavity. *International Journal of Heat and Mass Transfer*, 47(8-9) :1761 – 1769.
- [Oztop et al., 2009] Oztop, H. F., Zhao, Z., et Yu, B. (2009). Fluid flow due to combined convection in lid-driven enclosure having a circular body. *International Journal of Heat and Fluid Flow*, 30(5) :886 – 901. The 3rd International Conference on Heat Transfer and Fluid Flow in Microscale, The 3rd International Conference on Heat Transfer and Fluid Flow in Microscale.
- [Park et Lee, 1997] Park, H. et Lee, W. (1997). An efficient method of solving the navier-stokes equations for flow controls. *Phys. Rep*, 287 :337–384.
- [Peric et al., 1988] Peric, M., Kessler, R., et Scheuerer, G. (1988). Comparison of finite-volume numerical methods with staggered and collocated grids. *Computers and Fluids*, 16(4) :389 – 403.
- [Peterson, 1989] Peterson, J. (1989). The reduced basis method for incompressible viscous flow calculation. *SIAM J. Sci. Stat. Comput.*, 10.

- [Piller et Stalio, 2004] Piller, M. et Stalio, E. (2004). Finite-volume compact schemes on staggered grids. *Journal of Computational Physics*, 197(1) :299 – 340.
- [Pruliere et al., 2010] Pruliere, E., Chinesta, F., et Ammar, A. (2010). On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. *Mathematics and Computers in Simulation*, 81(4) :791 – 810.
- [Quarteroni et al., 2000] Quarteroni, A., Saleri, F., et Veneziani, A. (2000). Factorization methods for the numerical approximation of navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 188(1-3) :505 – 526.
- [Ramsak et al., 2005] Ramsak, M., Skerget, L., Hribersek, M., et Zunic, Z. (2005). A multidomain boundary element method for unsteady laminar flow using stream function-vorticity equations. *Engineering Analysis with Boundary Elements*, 29(1) :1 – 14.
- [Ravnik et al., 2009] Ravnik, J., Skerget, L., et Zunic, Z. (2009). Combined single domain and subdomain bem for 3d laminar viscous flow. *Engineering Analysis with Boundary Elements*, 33(3) :420 – 424.
- [Rempfer, 2000] Rempfer, D. (2000). On low-dimensional galerkin models for fluid flow. *Theoret. Comput. Fluid Dynamics*, 14 :75–88.
- [Rønquist, 1990] Rønquist, E. M. (1990). *Optimal spectral element methods for the unsteady 3-dimensional incompressible Navier-Stokes equations*. thèse de Doctorat, M.I.T, Cambridge, Mass.
- [Rowley, 2005] Rowley, C. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *Internantional Journal of Bifurcation and Chaos in Applied Sciences and Engineering.*, 15.
- [Ryckelynck, 2002] Ryckelynck, D. (2002). Reduction a priori de modèles thermomécaniques. *Comptes Rendus Mécanique*, 330.
- [Ryckelynck, 2005] Ryckelynck, D. (2005). A priori hyperreduction method : an adaptative approach. *Journal of Computational Physics*, 202.
- [Ryckelynck et al., 2005] Ryckelynck, D., Hermanns, L., Chinesta, F., et Alarcon, E. (2005). An efficient a priori model reduction for boundary element models. *Engineering Analysis with Boundary Elements*, 29.
- [Ryckelynck et Benziane, 2010] Ryckelynck, D. et Benziane, D. M. (2010). Multi-level a priori hyper-reduction of mechanical models involving internal variables. *Computer Methods in Applied Mechanics and Engineering*, 199(17-20) :1134–1142.
- [Sengupta et al., 2009] Sengupta, T., Vijay, V., et Bhaumik, S. (2009). Further improvement and analysis of ccd scheme : Dissipation discretization and de-aliasing properties. *Journal of Computational Physics*, 228(17) :6150 – 6168.
- [Shi et Khodadadi, 2003] Shi, X. et Khodadadi, J. (2003). Transient evolution to periodic fluid flow and heat transfer in a lid-driven cavity due to an oscillating thin fin. In Bathe, K., editor, *Computational Fluid and Solid Mechanics 2003*, pages 1131 – 1134. Elsevier Science Ltd, Oxford.

- [Shi et Khodadadi, 2005] Shi, X. et Khodadadi, J. (2005). Periodic state of fluid flow and heat transfer in a lid-driven cavity due to an oscillating thin fin. *International Journal of Heat and Mass Transfer*, 48(25-26) :5323 – 5337.
- [Sirovich, 1987] Sirovich, L. (1987). Turbulence and the dynamics of coherent structures, part 1 : Coherent structures. part 2 : Symmetries and transformations. part 3 : Dynamics and scaling. *Quarterly of Applied Mechanics*, 45 :561–590.
- [Stykel, 2006] Stykel, T. (2006). Balanced truncation model reduction for semidiscretized stokes equation. *Linear Algebra and its Applications*, 415.
- [Sun et al., 2011] Sun, C., Yu, B., Oztop, H. F., Wang, Y., et Wei, J. (2011). Control of mixed convection in lid-driven enclosures using conductive triangular fins. *International Journal of Heat and Mass Transfer*, 54(4) :894 – 909.
- [Temam, 1969] Temam, R. (1969). Sur l’approximation de la solution des equations de navier-stokes par la méthode des pas fractionnaires ii. *Arch.Ration.Mech.Anal.*, 33 :377–385.
- [Temam, 1977] Temam, R. (1977). Navier-stokes equations. *North-Holland, Amsterdam*.
- [Temam, 1991] Temam, R. (1991). Remark on the pressure boundary condition for the projection-method. *Theoret. Comput. Fluid Dynamics*, 3 :181– 184.
- [Timmermans et al., 1996] Timmermans, L., Mineev, P., et Van de Voose, F. (1996). An approximate projection scheme for incompressible flow usong spectral elements. *Int.J.Numer.Methods Fluids*, 22 :673–688.
- [Trevisan et Bejan, 1990] Trevisan, O. V. et Bejan, A. (1990). Combined heat and mass transfer by natural convection in a porous medium. volume 20 of *Advances in Heat Transfer*, pages 315 – 352. Elsevier.
- [Tuckermann, 1989] Tuckermann, L. (1989). Divergence-free velocity field in non-periodic geometries. *J. Comp. Phys.*, 80 :403 – 441.
- [van Kahn, 1986] van Kahn, J. (1986). A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J.Sci.Stat.Comput.*, 3(7) :870–891.
- [Verdon, 2007] Verdon, N. (2007). *Un système dynamique d’ordre réduit basé sur une approche APR-POD pour l’étude de l’interaction écoulement turbulent-particules*. thèse de Doctorat, LEPTIAB, Université de La Rochelle.
- [Verdon et al., 2011] Verdon, N., Allery, C., Béghein, C., Hamdouni, A., et Ryckelynck, D. (2011). Reduced-order modelling for solving linear and non-linear equations. *International Journal for Numerical Methods in Biomedical Engineering*, 27(1) :43–58.
- [Verdon et al., 2006] Verdon, N., Allery, C., Ryckelynck, D., et Hamdouni, A. (2006). An adaptive rom approach for solving transfer equations. *European Journal of Computational Mechanics*, 15(5) :589–605.
- [Violeau et al., 2009] Violeau, D., Ladevèze, P., et Lubineau, G. (2009). Micromodel-based simulations for laminated composites. *Composites Science and Technology*, 69(9) :1364 – 1371.
- [Weaver et Viskanta, 1991a] Weaver, J. et Viskanta, R. (1991a). Natural convection due to horizontal temperature and concentration gradients–1. variable thermophysical property effects. *International Journal of Heat and Mass Transfer*, 34(12) :3107 – 3120.

- [Weaver et Viskanta, 1991b] Weaver, J. et Viskanta, R. (1991b). Natural convection due to horizontal temperature and concentration gradients—2. species interdiffusion, solet and dufour effects. *International Journal of Heat and Mass Transfer*, 34(12) :3121 – 3133.
- [Zienkiewicz et Taylor, 1991] Zienkiewicz, O. et Taylor, R. (1991). In *The Finite Element Method, volume 2*. Mc Graw-Hill Book Company, fourth edition.