Some advances in PGD-based Model Reduction for High order PDEs, Complex Geometries and Solution of the Unsteady Navier-Stokes Equations

Guang Tao Xu

▶ **To cite this version:**

Guang Tao Xu. Some advances in PGD-based Model Reduction for High order PDEs, Complex Geometries and Solution of the Unsteady Navier-Stokes Equations . Fluid mechanics [physics.class-ph]. Ecole Centrale de Nantes (ECN), 2014. English. NNT : . tel-01207252

HAL Id: tel-01207252

https://hal.science/tel-01207252v1

Submitted on 30 Sep 2015

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 Centrale de Nantes

Ecole Doctorale

Sciences Pour l'Ingénieur, Géosciences, Architecture

Année 2013-2014 N° **B.U :**

Thèse de Doctorat

Spècialitè: Mécanique des milieux fluides

Présenté et soutenue publiquement par

Guangtao XU

Le 19 Mai 2014

à L'Ecole Centrale de Nantes

Des avancées dans la réduction de modèle de type PGD pour les EDPs d'ordre élevé, le traitement des géométries complexes et la résolution des équations de Navier-Stokes instationnaires

JURY

Rapporteurs:

Cyrille ALLERY	Maître de conférences(HDR) - Université de La Rochelle
Mejdi AZAIEZ	Professeur - Institut Polytechnique de Bordeaux

Examinateurs:

Francisco CHINESTA	Professeur - Ecole Centrale de Nantes
Christophe CORRE	Professeur - Institut National Polytechnique de Grenoble
Adrien LEYGUE	Chargé de recherche CNRS - Ecole Centrale de Nantes
Michel VISONNEAU	Directeur de recherche CNRS - Ecole Centrale de Nantes

---

Directeur de thèse : Michel VISONNEAU N°**ED**
Laboratoire: Hydrodynamics, Energetics, Atmospheric Environment - Ecole Centrale de Nantes

Co-directeur : Francisco CHINESTA
Laboratoire: Institut de Recherche en Génie Civil et Mécanique - Ecole Centrale de Nantes

Ecole Centrale de Nantes

Ecole Doctorale

Sciences Pour l'Ingénieur, Géosciences, Architecture

Année 2013-2014 N° **B.U :**

Doctoral Thesis

Subject: Mécanique des milieux fluides

written by

Guangtao XU

defenced on May 19 2014

at Ecole Centrale Nantes

Some advances in PGD-based Model Reduction for High order PDEs, Complex Geometries and Solution of the Unsteady Navier-Stokes Equations

JURY

Reviewers:

Cyrille ALLERY	Maître de conférences(HDR) - Université de La Rochelle
Mejdi AZAIEZ	Professeur - Institut Polytechnique de Bordeaux

Examinators:

Francisco CHINESTA	Professeur - Ecole Centrale de Nantes
Christophe CORRE	Professeur - Institut National Polytechnique de Grenoble
Adrien LEYGUE	Chargé de recherche CNRS - Ecole Centrale de Nantes
Michel VISONNEAU	Directeur de recherche CNRS - Ecole Centrale de Nantes

Thesis supervisor : Michel VISONNEAU N°**ED**
Laboratory: Hydrodynamics, Energetics, Atmospheric Environment - Ecole Centrale Nantes

Co-supervisor : Francisco CHINESTA
Laboratory: Institut de Recherche en Génie Civil et Mécanique - Ecole Centrale Nantes

Résumé

L'objectif principal de ce travail est de proposer une nouvelle approche de simulation basée sur une Méthode de réduction du modèle (MOR) utilisant une décomposition PGD. Dans ce travail, cette approche est d'abord utilisée pour résoudre des équations aux dérivées partielles d'ordre élevé avec un exemple numérique pour les équations aux dérivées partielles du quatrième ordre sur le problème de la cavité entraînée. Ensuite un changement de coordonnées pour transformer le domaine physique complexe en un domaine de calcul simple est étudié, ce qui conduit à étendre la méthode PGD au traitement de certaines géométries complexes. Divers exemples numériques pour différents types de domaines géométriques sont ainsi traités avec l'approche PGD.

Enfin, une séparation espace-temps est proposée pour résoudre les équations de Navier-Stokes instationnaires à l'aide d'une approche PGD. Cette décomposition est basée sur le choix de modes temporels communs pour la vitesse et la pression, ce qui conduit à une décomposition basée sur des modes spatiaux satisfaisant individuellement la condition d'incompressibilité. L'adaptation d'une formulation volumes finis à cette décomposition PGD est présentée et validée sur de premiers exemples analytiques ou académiques pour les équations de Stokes ou Navier-Stokes instationnaires. Une importante réduction des temps calculs est observée sur les premiers exemples traités.

Mots clés:

Réduction de modèles; PGD; géométrie complexe; EDP d'ordre élevé; Navier-Stokes instationnaire, solveur ISIS-CFD.

Abstract

The main purpose of this work is to describe a simulation method for the use of a PGD-based Model reduction Method (MOR) for solving high order partial differential equations. First, the PGD method is used for solving fourth order PDEs and the algorithm is illustrated on a lid-driven cavity problem. Transformations of coordinates for changing the complex physical domain into the simple computational domain are also studied, which lead to extend the spatial PGD method to complex geometry domains. Some numerical examples for different kinds of domain are treated to illustrate the potentialities of this methodology.

Finally, a PGD-based space-time separation is introduced to solve the unsteady Stokes or Navier-Stokes equations. This decomposition makes use of common temporal modes for both velocity and pressure, which lead to velocity spatial modes satisfying individually the incompressibility condition. The adaptation and implementation of a PGD approach into a general purpose finite volume framework is described and illustrated on several analytic and academic flow examples. A large reduction of the computational cost is observed on most of the treated examples.

Keywords:

Reduced Order Models; PGD method; Complex geometry; high order PDEs; Unsteady Navier-Stokes Equations; ISIS-CFD solver.

Contents

List of figures **ix**

General Introduction **1**

I PGD for the High order PDEs, Complex Geometry problem **5**

1 High Order PDEs and Complex Geometry problem **7**

1.1 High Order PDEs 7

1.2 Complex Geometry domain 9

1.3 Model Reduction Method 11

1.4 Scope and Outline for Part I 12

2 PGD for High order PDEs Problem **15**

2.1 Introduction 16

2.1.1 Harmonic problem and Biharmonic problem 16

2.1.2 Chebyshev polynomials & Chebyshev method 17

2.1.3 Numerical results 20

2.1.4 2D incompressible Flow problem 23

2.2 PGD formulation for High order PDEs 25

2.2.1 Coupling PGD and Chebyshev method 25

2.2.2 Pseudo-Spectral Chebyshev method 28
2.3 Numerical results 30
2.3.1 2D Laplace problem 30
2.3.2 4th order PDE problem 31
2.3.3 Chebyshev Polynomials for Boundary Value Problem 35
2.3.4 PGD for the Non-Homogeneous BC Biharmonic Problem . . . 47
2.3.5 PGD method for the Lid driven cavity problem 48
2.4 Discussion and Conclusion 52

3 PGD for the Decomposition of Complex Geometry 55

3.1 Introduction 55
3.1.1 Coordinate transform for Complex Geometry 56
3.1.2 FD method in the (θ, r) variables 58
3.1.3 Examples for the different kinds of domain 59
3.2 PGD applied to a complex geometry problem 63
3.2.1 General weak form for the new formulation 63
3.3 Numerical examples 68
3.3.1 Manufactured problem 69
3.3.2 Circular Domain with Uniform Source Term 71
3.3.3 Circular Domain with Non-uniform Source Term 71
3.3.4 Ellipsoidal Domain 74
3.3.5 Star Domain 77
3.4 Discussion and Conclusion 79

II PGD for Resolving the Unsteady Navier-Stokes Equations 81

4 Resolving the Unsteady Navier-Stokes Equations 83

4.1 Fluid flow and Navier-Stokes Equations 83

4.2 Numerical Method for the Navier-Stokes Equation 85

4.3 Model Reduction Method 88

4.4 Scope and Outline of Part II 89

5 The ISIS-CFD flow solver 91

5.1 Introduction 91

5.2 The ISIS-CFD flow solver 92

5.2.1 Reynolds Averaged Navier-Stokes Equations 92

5.2.2 Discretization of the momentum equation 92

5.2.3 Velocity-pressure coupling algorithm 94

5.2.4 The linear solvers 96

5.3 Numerical results 97

5.3.1 Steady Stokes for Lid driven cavity 97

5.3.2 Analytical solution for Steady Stokes 101

5.4 Discussion and Conclusion 104

6 PGD for Resolving the Unsteady Navier-Stokes Equations 105

6.1 Introduction 106

6.2 Singular Value Decomposition (SVD) for ISIS-CFD solution 106

6.2.1 Singular Value Decomposition (SVD) 106

6.2.2 Numerical example for SVD of ISIS-CFD solution 107

6.3 The PGD algorithm illustrated on an unsteady convection-diffusion equation 116

6.3.1 Computing the function $S(x)$ . . . 117
6.3.2 Computing the function $R(t)$ . . . 117
6.4 Comparison between an incremental approach and a separated decomposition . . . 118
6.4.1 In terms of computational effort . . . 118
6.4.2 In terms of memory allocation . . . 119
6.5 PGD formulation for Resolving the Unsteady Navier-Stokes equations 119
6.5.1 Introduction . . . 119
6.5.2 Unsteady Stokes Equations . . . 120
6.5.3 PGD Generalization to the unsteady 2D Stokes equations for incompressible flows . . . 120
6.5.4 A pressure equation formulation to solve the PGD formulation of Navier-Stokes equations for incompressible flows . . . 124
6.6 Treatment of non-linearities for the Navier-Stokes equations . . . 127
6.6.1 Linearization of the non-linear terms in the Navier-Stokes . . . 129
6.6.2 PGD formulation for the linearization . . . 130
6.6.3 Simplifying more the linearization . . . 137
6.7 Discussion and Conclusion . . . 139

7 Application of PGD for solving Unsteady Navier-Stokes equations 141

7.1 Introduction . . . 141
7.2 Analytical flow problems . . . 142
7.2.1 Unsteady Diffusion problem . . . 142
7.2.2 Analytical Stokes problem . . . 145
7.2.3 Burgers problem . . . 149
7.3 Real flow problem . . . 153
7.3.1 Stokes flow in a lid-driven cavity . . . 153

7.3.2 Navier-Stokes flow in a lid-driven cavity 157
7.3.3 2D Couette flow 161
7.4 Discussion and Conclusion 163

Conclusions and Perspectives **165**

Bibliography **169**

List of Figures

2.1 Graphic of the Chebyshev polynomials for $k = 0, \cdots, 6$ 18

2.2 Laplace solution and Convergence 21

2.3 Exact solution and Chebyshev solution for 4th order problem 22

2.4 Error between the exact solution and Chebyshev solution 22

2.5 The contour for the 2D Stokes problem 23

2.6 Singular values and finite sum representation for the flow problem . . 24

2.7 Error between the finite sum with 10 terms and the solution 24

2.8 Result for laplace problem 31

2.9 PGD modes and PGD convergence 31

2.10 Error for PGD and FD solution compare with exact solution 32

2.11 $L-2$ Error for PGD and Exact solution VS PGD modes 32

2.12 Exact solution and PGD solution 33

2.13 PGD mode and PGD convergence 33

2.14 Error between the PGD solution and exact solution 34

2.15 CPU time and L^2 error compare between the 2-D Chebyshev method and PGD method 35

2.16 Test case 1 37

2.17 Test Case 2 38

2.18 Flow problem 39

2.19 Result from Chebyshev Polynomials for Boundary Value Problem . . 42
2.20 Geometry and Boundary Condition for the 2D problem 43
2.21 The solution from chebyshev method 45
2.22 Souce term 46
2.23 Error for the Lid driven cavity problem 49
2.24 Contour by PGD and the Chebyshev method 50
2.25 PGD modes and PGD convergence 50
2.26 Error for the Lid driven cavity problem 51
2.27 Contour by PGD and the Chebyshev method 51
2.28 PGD convergence 52

3.1 Complex Geometry domain 56
3.2 Solution for unit Circle domain 60
3.3 Solution for Slanted Ellipse domain 60
3.4 Mesh for the star domain 61
3.5 $a(\theta), b(\theta), c(\theta), d(\theta)$ for the star domain 61
3.6 FE mesh and FE solution 62
3.7 FD solution and FD-FE comparison 62
3.8 Solution for the square domain 62
3.9 Solution for the manufactured problem 70
3.10 Error for the manufactured problem 70
3.11 PGD convergence 71
3.12 PGD solution for the circular domain 72
3.13 PGD error for the circular domain 72
3.14 PGD convergence for the circular domain 72
3.15 Solution for the problem with Non-uniform source 73

3.16 PGD error for the problem with Non-uniform Source 73

3.17 PGD error as a function of the number of modes and Error of the converged PGD solution (the discretization for (θ, r) have the relation $N_{theta} = 6N_r$) for the the problem with Non-uniform Source 74

3.18 Geometry for the Ellipsoidal domain 75

3.19 Solution for the Ellipsoidal domain 75

3.20 PGD error for the Ellipsoidal domain 76

3.21 PGD Convergence and Error of the converged PGD for different meshes for the Ellipsoidal domain 76

3.22 Comparison of CPU time between PGD and FEM 76

3.23 Geometry for the Star Domain 77

3.24 Solution for the Star Domain 78

3.25 PGD error for the Star Domain compare with the FEM solution . . . 78

3.26 PGD convergence and PGD-FEM error for different meshes for the star domain 79

3.27 FEM-PGD CPU time comparison for the star-like domain. 79

4.1 Flow around submarine 84

4.2 Flow generated by an earthquake 84

4.3 The flow from a faucet 85

4.4 Comparison of the PGD and FEM based 3D discretizations 89

4.5 CPU time for the PGD and standard solvers 89

5.1 Geometry and boundary conditions for the lid driven cavity 98

5.2 Convergence analysis 99

5.3 FV and FE solutions for Lid-driven cavity 99

5.4 FEM and FVM solutions for the lid-driven cavity (streamlines) 100

5.5 Residual vs iteration for 4 cases 101

5.6 Residual vs number of nodes for 4 cases . . . 102
5.7 Evolution of the CPU time and number of iterations to converge for various grids . . . 102
5.8 L-2 norm of error for the velocities and pressure . . . 103

6.1 Geometry and boundary conditions . . . 108
6.2 3rd mode for u . . . 109
6.3 4th mode for u . . . 110
6.4 Eigenvalue for U . . . 110
6.5 Error for U (logplot) . . . 110
6.6 1st-3rd modes for v . . . 112
6.7 4th mode for v . . . 113
6.8 Eigenvalue for V . . . 113
6.9 Error for V (logplot) . . . 113
6.10 1st-3rd modes for P . . . 114
6.11 4th mode for Pressure . . . 115
6.12 Eigenvalue for Pressure . . . 115
6.13 Error for Pressure (logplot) . . . 115

7.1 PGD solution (top) & error with analytical solution (bottom) at 0.01s 143
7.2 PGD solution (top) & error with analytical solution (bottom) at 0.2s 143
7.3 PGD solution (top) & error with analytical solution (bottom) at 0.9s 144
7.4 Space and time discretization effects for the diffusion problem . . . 144
7.5 Convergence for the diffusion problem . . . 145
7.6 Geometry and boundary conditions for the Stokes analytical flow . . 146
7.7 PGD solution (top) & exact solution (bottom) at 0.01s . . . 147
7.8 PGD solution (top) & exact solution (bottom) at 0.2s . . . 147

7.9 PGD solution (top) & exact solution (bottom) at 0.9s 148

7.10 Space and time discretization effect for the analytical Stokes problem 148

7.11 Convergence for the analytical Stokes problem 149

7.12 Geometry and boundaries for Burgers equation 149

7.13 PGD solution for Burgers equation at 0.01s 151

7.14 PGD solution for Burgers equation at 0.2s 151

7.15 PGD solution for Burgers equation at 0.9s 152

7.16 Error vs PGD modes for Burgers equation 152

7.17 Space and time discretization effects for Burgers equation 153

7.18 Geometry and boundary conditions for the lid-driven cavity flow . . . 154

7.19 PGD solution at 0.01s for the lid-driven cavity problem 155

7.20 PGD solution at 0.2s for the lid-driven cavity problem 156

7.21 PGD solution at 0.9s for the lid-driven cavity problem 156

7.22 Convergence for the lid-driven cavity problem 157

7.23 CPU time comparison for the lid-driven cavity problem 157

7.24 PGD solution with 20 modes and the ISIS-CFD solution for the lid-driven cavity problem by Navier-Stokes (continuous line: ISIS-CFD solution, dashed line: ISIS-CFD-PGD solution) 158

7.25 PGD solution with 50 modes and the ISIS-CFD solution for the lid-driven cavity problem by Navier-Stokes ((continuous line: ISIS-CFD solution, dashed line: ISIS-CFD-PGD solution) 159

7.26 PGD convergence vs modes number 160

7.27 CPU comparison for the lid-driven cavity 160

7.28 Geometry for the Couette flow problem 161

7.29 PGD solution with 20 modes and the ISIS-CFD solution (continuous line is ISIS-CFD solution and dashed line is ISIS-CFD-PGD solution) 162

7.30 PGD solution with 32 modes and the ISIS-CFD solution(continuous line is ISIS-CFD solution and dashed line is ISIS-CFD-PGD solution) 162

7.31 $L-2$ norm of the difference for U, p vs modes number 163

7.32 PGD convergence vs modes number 163

General Introduction

Model reduction Method is the calculation of the problem in the science and the engineering for the complex geometry problem and with their thermo mechanical behavior. Many of them have never been solved because of the computational complexity caused by such problems. This method can be very fast for these problem, which are real time solving for some problem.

Along with the ROM-POD method[Kerschen et al., 2005, Liberge, 2010] and the reduced-basis element method [Rozza et al., 2008], Proper general decomposition (PGD) method is one kind of Model reduction Method (MOR) which is a kind of very efficient method to solve the multi phase problem, but in the case of PGD the construction of the representation takes into account the nature of the problem directly. The general form of a PGD separated representation of a function u of N variables is $u(x_1, ..., x_N) = \sum_{m=1}^{M} u_m^1(x_1) \times ... \times u_m^N(x_N)$, M being the order of the approximation. This method is based on the space-time separated idea, but this idea is not a new proposal. In fact, it was proposed by Pierre ladeveze as an ingredient of new powerful non-linear non-incremental LATIN solver in the 80s. In the LATIN method, radial approximation are used as a approach which could be seen as a variation of the PGD with a space-time separation (see example as [Ladevèze, 1999, Ladevèze et al., 2010, Ladevèze and Nouy, 2003, Cremonesi et al., 2013]). And PGD now has been used to solve many kinds of problem in multi-dimensional spaces: quantum chemistry, kinetic theory description of complex fluid and chemical master equations, etc...The interested reader can refer to [Chinesta et al., 2011] for a recent review in the context of computational rheology.

Many problems governed by multi-harmonic equations, such as thin-plate bending and Stokes flow problems which involve biharmonic equations. And many real engineering problem are concerned with the problem in the complex geometry. But no work has been done by the PGD based Model reduction Method (MOR) for high order PDEs problem, even fewer works has been done for the problem with complex geometry which is always concerned by the real engineering problem. In this work, the PGD method for solving the high order PDEs problem and the problem with

complex geometry domain are proposed in the first part.

In the Chapter 1, the state of the art for the high order PDEs problem and the problem in complex geometry domain is presented, and the PGD based MOR will introduced and the progress for this kind of MOR will be presented.

Secondly, after the common method -Chebyshev method for solving the high order PDEs is introduced, the PGD method for solving the high order PDEs is presented in the Chapter 2, the numerical example for the fourth order PDEs and Lid driven cavity problem will be detailed.

At last of the first part, transformation of coordinates for changing the complex domain problem into the simple compute domain will be studied in the Chapter 3, then the PGD method for the complex geometry problem is proposed in the same chapter, the numerical examples for different kinds of domain problem are resolved by the PGD method.

In the other part, because of the nonlinear terms, convective terms and the pressure gradient term, it is very difficulty for solving the Navier-Stokes equations, more transient resolutions need a small enough time step to ensure the stability and convergence of numerical schemes. so it is desirable to have a sufficient density of points to describe the interface of discontinuity. There were a lot of problems related to the very large calculations [Montagnier et al., 2013]. Efficient solver are needed for the simulation. Although the PGD based Model reduction Method (MOR) has been done in some work for resolving the Navier-stokes problem, but never been done with the real kind of Navier-stokes equations. So, in the second part, we will use PGD method for Resolving the Unsteady Navier-Stokes Equations, apply the model reduction techniques based on the method PGD (Proper Generalized Decomposition) for non-incremental discretization of the Navier Stokes equations.

The Chapter 4 will make a point on the vast state of art on Navier-Stokes Equation and numerical methods for solving the Navier-stokes Equations, and also the Proper Generalized Decomposition Methods for solving the flow problem will be stated.

The Chapter 5 will present the basic theory for the ISIS-CFD solver [Deng et al., 2001, Queutey and Visonneau, 2007] and a numerical example will be presented for solving the steady stokes in the lid driven cavity problem.

The Chapter 6 will firstly use the SVD for representing the unsteady flow by the priori solution from the ISIS-CFD solver of the 2D unsteady flow in lid-driven cavity. Then we will give the theories about the PGD method coupling with the ISIS-CFD solver for the Unsteady Navier-Stokes Equations without the nonlinear

term; the same temporal function for the velocity and pressure for the decomposition is used, this strategy was used for avoiding to change the incompressible problem into the compressible problem. The finite volume formulation for the PGD method for solving the unsteady Reynolds Averaged Navier-Stokes Equation will also be given in this chapter.

The numerical examples for using the proposed PGD method to the flow problems governed by STOKES and Navier-stokes equations will be given in the Chapter 7.

At last, the general conclusions and the perspectives of this work will be given.

Part I

PGD for the High order PDEs, Complex Geometry problem

Chapter 1

High Order PDEs and Complex Geometry problem

1.1 High Order PDEs 7
1.2 Complex Geometry domain 9
1.3 Model Reduction Method 11
1.4 Scope and Outline for Part I 12

1.1 High Order PDEs

Partial differential equations (PDEs) defined on surfaces embedded in R^3 arise in a wide range of applications [Greer et al., 2006], including fluid dynamics, biology (e.g., fluids on the lungs), materials science (e.g., ice formation), electromagnetism, image processing (e.g., images on manifolds and inverse problems such as EEG), computer graphics (e.g., water flowing on a surface), computer aided geometric design (e.g., special curves on surfaces), and pattern formation. Alternating Direction Implicit (ADI) schemes are constructed for the solution of two-dimensional higher-order linear and nonlinear diffusion equations, particularly including the fourth-order thin film equation for surface tension driven fluid flows [Witelski and Bowen, 2003]. In [Mai-Duy and Tanner, 2005], the unsymmetric indirect RBF collocation method is extended to solve high-order PDEs directly, and the method is verified successfully through the solution of thin-plate bending and viscous flow problems which are governed by biharmonic equations.

Consider problems governed by multi-harmonic equations, such as thin-plate

bending and Stokes flow problems involving biharmonic equations. The Biharmonic problem [Erturk and Dursun, 2007] which is a 4-th order PDEs problem has been raised in many research filed, such as in elasticity problem which dealing with the transverse displacements of plates [Li et al., 2011] and shells and in fluid problem which the governing equation of the stokes flow [Montlaur et al., 2008] is the biharmonic equation. In the thin plate theory, the biharmonic equation can be represent the clamped plate under the external load [Li et al., 2011]. As we know, the boundary and load condition is very complex, so it's very difficulty to get the analytical solution. Therefore, many studies are focus on the numerical method to make the biharmonic problem to practical engineering.

To solve these problems, new variables are usually introduced in order to transform the multi-harmonic equations into the coupled sets of harmonic equations from which the conventional low-order methods of discretization such as the boundary element methods (BEMs) [Mai-Duy et al., 2006], finite difference methods (FDMs) or finite element methods (FEMs)[Gudi et al., 2008] can be applied for obtaining a numerical solution [Mai-Duy and Tanner, 2005], but the researchers are interested in using the spectral method, such us Variational iteration method [Ali and Raslan, 2007], Domain Decomposition Method [Avudainayagam and Vani, 2000, Shang and He, 2009], finite volume method[Wang, 2004], Fast multiple method [Gumerov and Duraiswami, 2006] and fundamental solutions method [Marin and Lesnic, 2005].

Due to their bigger accuracy when compared to Finite Differences (FD) and Finite Elements (FE) methods, the rate of convergence of spectral approximations depends only on the smoothness of the solution, yielding the ability to achieve high precision with a small number of data. The spectral method has been popularly used in the computation of continuous mechanics problems. The expression spectral methods has different meanings for several sub-areas of Mathematics, like Functional Analysis and Signal Processing, Spectral methods has the meaning of a high accuracy numerical method to solve Partial Differential Equations. The numerical solution is expressed as a finite expansion of some set of basis functions. When the PDE is written in terms of the coefficients of this expansion, the method is known as a Galerkin spectral method. Spectral collocation methods, also known as pseudo spectral methods, is another subclass of spectral methods which are similar to Finite Differences methods due to direct use of a set of grid points, which are called "collocation points". A third class are the Tau spectral methods. These methods are similar to the Galerkin spectral methods, however the expanding basis is not obliged to satisfy the boundary conditions, requiring extra equations. In a spectral collocation method based on integrated chebyshev polynomials for the solution of first and second kind of biharmonic problems is analyzed [Mai-Duy and Tanner, 2007].

This kind of method has been successfully used to solve the harmonic and biharmonic problem in [Chantasiriwan, 2006].

The Chebyshev spectral collocation method [Martinez and Esperança, 2007, Li et al., 2008] can be described in the following way. An approximation based on Chebyshev polynomials to the variable u is first introduced. The set of collocation equations is then generated. The equation system consists of two parts. The first part is formed by making the associated residual, e.g.$(\triangle^2 u - f)$, equal to zero at the collocation points; while the second part is obtained by forcing the boundary conditions e.g., u and $\frac{\partial u}{\partial x}$, to be satisfied at the boundary collocation points. These tasks need to be conducted in an appropriate manner.

1.2 Complex Geometry domain

In the science and engineering problems, many real problems are relevant with the complex geometry domain, for example Shape Optimization [Wang, 2012, Yoon and Sigmund, 2008], not as simple as the rectangular or square domain which can be easily applied the Finite Difference Method for example when the numerical simulation were needed to be done; so for the problem on irregular domains, it is still challenging to solve the partial diffusion equation efficiently.

Currently, the FDM were used for the problem in irregular domain, but the Finite Difference approximation need to be modified at grids near the boundary. Due to the difficulties in handling the finite difference approximation close to a curved boundary, most the finite difference methods are restricted on regular (rectangular and circular) domains. [Lai, 2001] proposed a simple second-order finite difference treatment of polar coordinate singularity for Poisson equation on a disk; [Chen et al., 2008] has proposed a fast FDM for biharmonic equations on irregular domain. [Weibin and Xionghua, 2009] proposed a meshless method: Chebyshev tau matrix method (CTMM) for Poisson-type equations on irregular domains. The Fast Fourier Transform (FFT) was used to transform from the physical space to the spectral space efficiently and the matrix technique was used to represent the differentiation. [Shao et al., 2012] reported the Chebyshev tau meshless method based on the integration differentiation (CTMMID) for numerically solving Biharmonic-type equations on irregularly shaped domains with complex boundary conditions. An integral collocation approach based on Chebyshev polynomials for numerically solving biharmonic equations further developed for the case of irregularly shaped domains in [Mai-Duy and Tran-Cong, 2009]. [McCorquodale et al., 2004] presented a method for solving Poissons equation with Dirichlet boundary conditions on an

three-dimensional irregular bounded region. A pervasive embedded boundary domain specification for the data connected with the numerical integration of conservation-law PDEs was studied by [Day et al., 1998]. [Li et al., 2009] studied a generalized approach for solving the complex, stationary, or moving geometries with Dirichlet, Neumann, and Robin boundary conditions Partial Differential Equations. [Mayo, 1984] proposed an the embedded boundary integral method (EBI) for the Poisson's and the Biharmonic Equations on Irregular Regions, and extended this method to the stokes flow in complex geometry [Biros et al., 2004]. The method for solving fourth order PDEs on surfaces of arbitrary geometry by finite difference schemes on a Cartesian grid was studied by [Greer et al., 2006]. Meshless method were used for solving coupled radiative and conductive problem for the complex geometries in [Sakami et al., 1996, Wang et al., 2010, Sadat et al., 2012].

In the fluid mechanics, one of the main challenges for the Navier-stokes solver is the geometric flexibility, the unstructured grids is not always adopted as the filetering procedure involved in LES and is associated with increased computing cost [Bui, 2000]; simulation results for a variety of flows to show that robust, accurate solutions are now obtained at high Reynolds numbers in very complex geometries in [Mahesh et al., 2004]; In [Münster et al., 2012] explain the details of how the fictitious boundary method (FBM) can be used to simulate flows with complex geometries that are hard to describe analytically, also explained how complex geometry can be easily used in the Finite element-fictitious boundary methods (FEM-FBM) for context. Spectral methods are known to be well suited for laminar or transitional flows in simple geometries (Cartesian, cylindrical, spherical...), [Baur et al., 2009] described some routes allowing to address more complex flows, especially turbulent flows in complex geometries. Solving the time-dependent fluid equations as opposed to the time-averaged equations (as done in most solvers) is significantly more stable because the solution is integrated forward in time from a valid solution. However, this approach is often prohibitive and requires typically an order of magnitude or more increase in computing, as well as robust models for any unresolved scales (subgrid stress models or turbulence models) [Duggleby et al., 2011]. Therefore, the first challenge in reducing the time for a CFD solution is reducing the simulation time [Cant, 2002]. The second challenge in shortening the total time of CFD is meshing. In all CFD, mesh quality plays a significant role in improving both solution convergence as well as accuracy [Lohner, 2007].

Nowadays GMSH is 3D finite element grid generator with a build-in CAD engine and post-processor [Geuzaine and Remacle, 2009], the mesh generator by GMSH is well used in many computed code. DistMesh is a simple MATLAB code for generation of unstructured triangular and tetrahedral meshes [Persson and Strangt, 2004], but the main disadvantage are slow execution and the possibility of nontermination. It is

still challenge for finding a fast and easy handling method for generate the mesh for simulation, also the mew method for reducing the simulation time are needed.

1.3 Model Reduction Method

As model reduction is a mathematical theory to find a low-dimensional approximation for a system of ordinary differential equations, model reduction is a topic which receives growing attention, both in the mathematics community and the various application areas. It can be used to reduce the computational effort. It is also has been used to solve the fluid problem, Balanced truncation model reduction methods used to analysis the stokes equation in [Stykel, 2006], the important properties of this method are that the regularity and stability is preserved in the reduced order system and there is a priori bound on the approximation error [Vierendeels et al., 2007] developed reduced order models for strongly coupled fluid-structure interaction problem. This method was used for the fluid and the structural solver that are built up during the coupling iterations. The method also can be implemented very easily. [Buffat et al., 2011] presented a spectral projection method for incompressible flow simulation based on an orthogonal decomposition of the velocity into two solenoid vector fields and to apply it for the problem of boundary layer bypass transition in a plane channel configuration. One can see the other model reduction method such as ROM-POD method in [Kerschen et al., 2005, Allery et al., 2005, Burkardt et al., 2006, Liberge, 2010, Allery et al., 2011]and the reduced-basis element method in [Rozza et al., 2008] and the reference therein.

Another possible strategy able to compute the partial differential equations is proper general decomposition (PGD) method which is a kind of very efficient method to solve the multi phase problem. This method is based on the space-time separated idea, but this idea is not a new proposal. In fact, it was proposed by Pierre ladeveze as an ingredient of new powerful non-liner non-incremental LATIN solver in the 80s. In the LATIN method, radial approximation are used as a approach which could be seen as a variation of the PGD with a space-time separation (see example as [Ladevèze, 1999, Ladevèze et al., 2010, Ladevèze and Nouy, 2003, Cremonesi et al., 2013]). The functions depending on space and the ones depending on time were also unknown, and were computed by a suitable technique. [Ammar et al., 2006, Ammar et al., 2007] proposed the Proper general decomposition (PGD), and they used this decomposition solved many kind of problem in multidimensional spaces, quantum chemistry, kinetic theory description of complex fluide, chemical master equation $\cdots$. A detail review work can be found in [Chinesta et al., 2010b, Chinesta et al., 2013]. In the modeling of polymeric liquids,

which need to solve the Fokker-Planck equation, based on PGD, the case of non-homogeneous flows was analyzed [Mokdad et al., 2010, Pruliere et al., 2008]. The models arising from quantum chemistry, since the equations that govern the electronic distribution, the Schrodiger equation are defined in spaces whose dimensionality scales with the number of elementary particles involved in the quantum system, these models are redoubtable, by using PGD, in many system the real issue could be circumvented [Ammar et al., 2008, Ammar and Chinesta, 2008]. Analyzed the non-liner models by an incremental linearization and a Newton linearization [Ammar et al., 2010d], also in [Ammar et al., 2010a, Pruliere et al., 2010] gave the tensor notation for PGD, in [Ammar et al., 2010b] the enforcement of Non-homogeneous boundary conditions and the treatment of complex geometries were involved; and it was also applied for solving multi-physics models arising in the composites manufacturing processes, where are coupled with the non-linear thermal and thermo-mechanical behaviors [Prulière et al., 2010] . And PGD was also applied to the kinetic theory description of complex fluid. Related to parametric deterministic models and models in particular were involved in non-Newtonian fluids [Ammar et al., 2010c]. The direct solution of Fokker-planck equation for complex fluids in configuration spaces of high dimension was given in [Chinesta et al., 2011]. PGD also can be coupled with FEM [Ammar et al., 2010d] and BEM [Bonithon et al., 2011] for solving the partial different equation (PDE), the non-incremental compute strategy which used by PGD method demonstrate that significant CPU time savings are expected and alleviate the storage needs.

PGD with a spectral collocation method to solve transfer equations as well as NavierStokes equations has been done in [Dumon et al., 2013], but nothing has been done with the high order PDEs problems. There were only few PGD works concerned with the complex geometry problem, see example [González et al., 2010, Ghnatios et al., 2012, Ammar et al., 2014, Chady Ghnatios, 2012], more work in the complex geometry problem are need to be done.

1.4 Scope and Outline for Part I

The objective of this part is the application of model reduction techniques based on the method PGD (Proper Generalized Decomposition) for solving the high PDEs problem and the problem in complex geometry domain. The oeuvre is developed in the following way:

- The Chapter 2 will introduce the Chebyshev method for solving the fourth-order PDEs which can be used for the flow streamline. The PGD method was coupling

with the spectral discretization method for the fourth order PDE. And using the PGD for solving the high order PDEs problem, especially for the 2D lid driven cavity flow problem.

- The Chapter 3 will change the equation in the complex geometry problem into the rectangular compute domain by using the coordinate transformation; then we apply PGD to the problem in the complex geometric domain, and several examples with different kind of shape function are applied by the proposed PGD method.

Chapter 2

PGD for High order PDEs Problem

2.1 Introduction 16
2.1.1 Harmonic problem and Biharmonic problem 16
2.1.2 Chebyshev polynomials & Chebyshev method 17
2.1.3 Numerical results 20
2.1.4 2D incompressible Flow problem 23
2.2 PGD formulation for High order PDEs 25
2.2.1 Coupling PGD and Chebyshev method 25
2.2.2 Pseudo-Spectral Chebyshev method 28
2.3 Numerical results 30
2.3.1 2D Laplace problem 30
2.3.2 4th order PDE problem 31
2.3.3 Chebyshev Polynomials for Boundary Value Problem 35
2.3.4 PGD for the Non-Homogeneous BC Biharmonic Problem 47
2.3.5 PGD method for the Lid driven cavity problem 48
2.4 Discussion and Conclusion 52

Equations of the harmonic and biharmonic types are two kinds of equation frequently encountered in engineering problems: the biharmonic equation with a given Boundary Condition can be used to describe the stream function for incompressible flow problems for example. Concerning solution strategies, the Pseudo-Spectral Chebyshev method which is based on integrated Chebyshev polynomials for the solution of the problem has been popularly used in the solution of continuous mechanics problems. In this chapter, we propose a technique that combines the use

of Pseudo-Spectral Chebyshev method with the Proper Generalized Decomposition (PGD) that allows space time separated representation of the unknown field within a non incremental integration scheme. We have used the PGD to separate the unknown field between the different coordinates, then Pseudo-Spectral Chebyshev method is used to solve the resulting one dimension problems. We proposed the technique to solve the harmonic problem and the fourth order PDE–biharmonic problem. Specific strategies for dealing with non-homogeneous boundary conditions have been implemented to solve the lid-driven cavity problem in the stream function formulation.

This chapter is thus organized as follows:

First, we give some introduction about the Chebyshev method and describe the harmonic problem and biharmonic problems which are used in this chapter. A 2D Chebyshev method is used for the flow problem and the SVD is used for studying the separability of the 2D solution. In Section 2 we couple the PGD method and the Chebyshev method for the harmonic and biharmonic problem, we considered Pseudo-Spectral Chebyshev method for solving the different one dimensional problems. Section 3 is concerned with applications and numerical examples using the method. A significant part of the section is dedicated to the discussion of different techniques for the imposition of non-homogeneous boundary conditions within the Chebyshev-PGD framework. Finally, Chebyshev-PGD method was applied to the Non-Homogeneous Biharmonic Problem and the proposed technique was used to solve the lid-driven cavity problem in the stream function formulation.

2.1 Introduction

2.1.1 Harmonic problem and Biharmonic problem

2.1.1.1 2D Harmonic problem

The harmonic problem, found in the Laplace equation and Poisson equation, is overwhelmingly present in PDEs describing physical problems and is often a standard benchmark to test the program code.

We consider the problem:

$$\Delta u = h(x, y) \quad in \quad \Omega \tag{2.1}$$

with boundary condition:

$$u = g(x, y) \quad on \quad \partial\Omega \tag{2.2}$$

where Δ is the Laplace operator, g is a given function, Ω is a bounded domain in the plane, and $\partial\Omega$ is the boundary of the domain.

2.1.1.2 2D Biharmonic problem

The Biharmonic problem [Erturk and Dursun, 2007] has been raised in many research fields, such as in elasticity problem which deal with the transverse displacements of plates [Li et al., 2011] and shells as well as in fluid problem where the governing equation for the stokes flow in stream function formulation [Montlaur et al., 2008] is the biharmonic equation. In the thin plate theory, the biharmonic equation can model a clamped plate under some external load [Li et al., 2011].

Here, we consider the Biharmonic problem which follows:

$$\triangle^2 u = f(x,y) \quad in \quad \Omega \tag{2.3}$$

$$u = 0 \; and \; \frac{\partial u}{\partial n} = 0 \quad on \quad \partial\Omega \tag{2.4}$$

where

$$\triangle^2 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4} \tag{2.5}$$

and f is a given function.

2.1.2 Chebyshev polynomials & Chebyshev method

The Chebyshev Spectral Collocation method for a differential problem [Martinez and Esperança, 2007, Li et al., 2008] can be described in the following way. An approximation based on Chebyshev polynomials of the unknown function u is first introduced. The set of collocation equations is then generated. The equation system consists of two parts. The first part is formed by making the associated residual, e.g.$(\triangle^2 u - f)$, equal to zero at the collocation points, while the second part is obtained by enforcing the boundary conditions e.g., and $\frac{\partial u}{\partial x}$, to be satisfied at the boundary collocation points. These tasks need to be conducted in an appropriate manner and they will be presented in detail.

It is convenient for the Chebyshev method to select the interpolation points in the interval $[-1, 1]$

$$x_j = cos\frac{\pi j}{N} (j = 0 \cdots N) \tag{2.6}$$

The Chebyshev polynomial of the first kind $T_k(x)$ with the polynomial of degree k defined for $x \in [-1, 1]$ by

$$T_k(x) = \cos(k \cos^{-1} x)) \tag{2.7}$$

which

$$-1 \leq T_k(x) \leq 1 \tag{2.8}$$

It is straightforward to deduce:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \end{aligned} \tag{2.9}$$

the Chebyshev polynomials for $k = 0, \cdots, 6$ are represented in Figure (2.1).

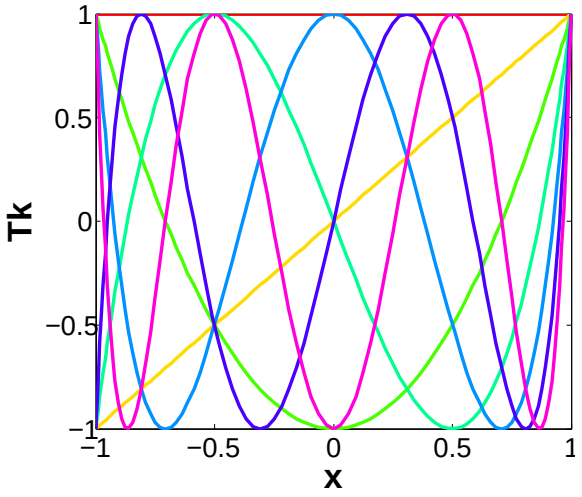


Figure 2.1: Graphic of the Chebyshev polynomials for $k = 0, \cdots, 6$

2.1.2.1 Chebyshev method for the biharmonic equation

We just describe here the Chebyshev method for the non-homogeneous BC problem for the biharmonic equation. For a two-dimensional incompressible flow, the dimensionless vorticity transport for the stokes equation can be expressed as

$$\triangle^2 \Psi = 0 \tag{2.10}$$

where

$$\triangle^2 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4} \tag{2.11}$$

and Ψ is the stream function.

This is only one fourth-order PDE for the stream function needs to be solved, namely the bi-harmonic equation. The velocity is obtained from the Lagrange stream function through:

$$\begin{aligned} u = &\quad \frac{\partial \Psi}{\partial y} \quad \Gamma \in \partial\Omega \\ v = &\quad -\frac{\partial \Psi}{\partial x} \quad \Gamma \in \partial\Omega \end{aligned} \tag{2.12}$$

In the domain $\Omega \in [-1, 1]^2$, we consider the following collocation points:

$$\begin{aligned} x_i = &\quad cos\frac{\pi i}{M}(i = 0, \cdots, M) \\ y_j = &\quad cos\frac{\pi j}{N}(i = 0, \cdots, N) \end{aligned} \tag{2.13}$$

where M and N are the nodes numbers in x and y direction, respectively.

We suppose that we can write the stream function for each point (x_i, y_j) as:

$$\Psi(x_i, y_j) = \sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} T_i(x_k) T_j(y_l) \tag{2.14}$$

Putting Eq.(2.14) into Eq.(2.10) yields

$$\begin{aligned} &\triangle^2 \Psi(x_k, y_l) = \textstyle\sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} T_i^{''''}(x_k) T_j(y_l) \\ &+2 \textstyle\sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} T_i^{''}(x_k) T_j^{''}(y_l) + \sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} T_i(x_k) T_j^{''''}(y_l) = 0 \end{aligned} \tag{2.15}$$

As the Chebyshev polynomials have the following property:

$$T_i(x_j) = \delta_{ij} \tag{2.16}$$

with δ defined as:

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \tag{2.17}$$

One can rewrite Eq.(2.15) as:

$$\triangle^2 \Psi(x_k, y_l) = \sum_{i=0}^{N} a_{il} T_i^{''''}(x_k) \delta_{jl} + 2 \sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} T_i^{''}(x_k) T_j^{''}(y_l) + \sum_{j=0}^{M} a_{kj} \delta_{ik} T_j^{''''}(y_l) = 0. \tag{2.18}$$

By introducing the following notation:

$$T_i^p(x_k) = (D_p)_{ik} \tag{2.19}$$

we can rewrite equation (2.18) as

$$\triangle^2\Psi(x_k,y_l)=\sum_{i=0}^{N}a_{il}(D_4)_{ik}+2\sum_{i=0}^{N}\sum_{j=0}^{M}a_{ij}(D_2)_{ik}(D_2)_{jl}+\sum_{j=0}^{M}a_{kj}(D_4)_{jl}=0 \tag{2.20}$$

where $(D_i)_{kl}$ is the Kuibyshev Matrix, which is Chebyshev collocation derivative matrix[Martinez and Esperança, 2007], this matrix is given by the following expression:

$$(D_1)_{ik}=\begin{cases}\frac{c_i(-1)^{i+k}}{c_k(x_i-x_k)} & i\neq k\\ -\frac{x_k}{2(1-x_k^2)} & 1\leq i=k\leq N-1\\ \frac{2N^2+1}{6} & i=k=0\\ -\frac{2N^2+1}{6} & i=k=N\end{cases} \tag{2.21}$$

with

$$\begin{aligned}c_0=c_N=2,\\ c_j=1(1\leq j\leq N-1)\end{aligned} \tag{2.22}$$

This expression is easily found in the literature of the spectral methods [Martinez and Esperança, 2007, Mai-Duy and Tanner, 2007]. The Chebyshev collocation derivative matrix at another order $(D_p)_{ik}$ can be obtained analytically using an explicit expression (see [Mai-Duy and Tran-Cong, 2009]) or by the following relation:

$$D_p=(D_1)^p. \tag{2.23}$$

For a specific PDE, the above discrete system has to be solved with the proper boundary conditions enforced.

2.1.3 Numerical results

2.1.3.1 2D Laplace problem

The Laplace problem is a second-order harmonic equation, simpler the fourth order Biharmonic problem, it is therefore being used here as the test problem for the Chebyshev method. The governing equation for the Harmonic equation is:

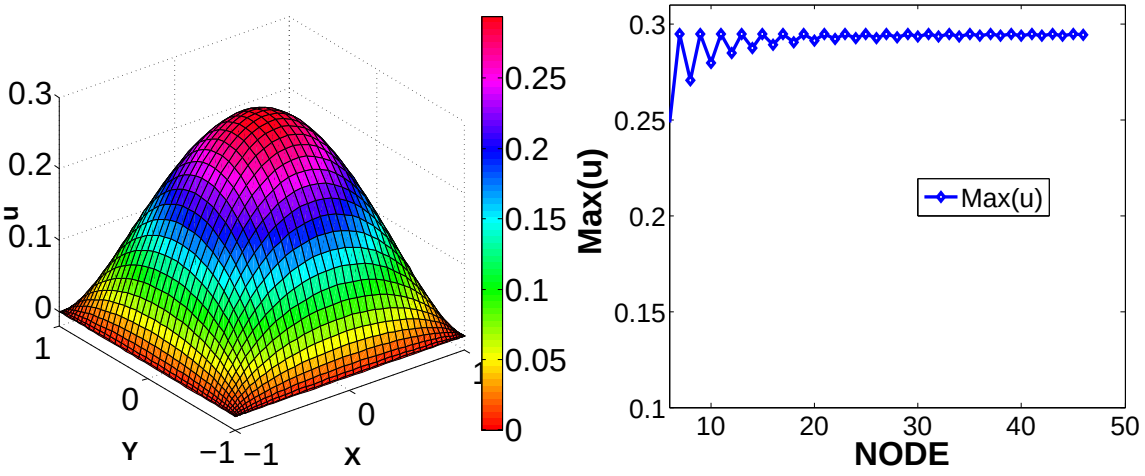
$$\triangle u(x,y)=f \qquad -1\leq(x,y)\leq 1 \tag{2.24}$$

$$f = -1 \tag{2.25}$$

and the boundary condition is

$$u = 0 \tag{2.26}$$

The solution for the discretization as (45×45) of this problem using the Chebyshev collocation is presented in Fig.(2.2(a)). To illustrate the convergence of the method, the maximum of u for different node numbers in each direction of is given in Fig.(2.2(b)), we can see that the solution converges indeed very quickly.



(a) Solution for laplace problem

(b) Convergence for Laplace problem

Figure 2.2: Laplace solution and Convergence

2.1.3.2 Simply-supported square-thin-plate

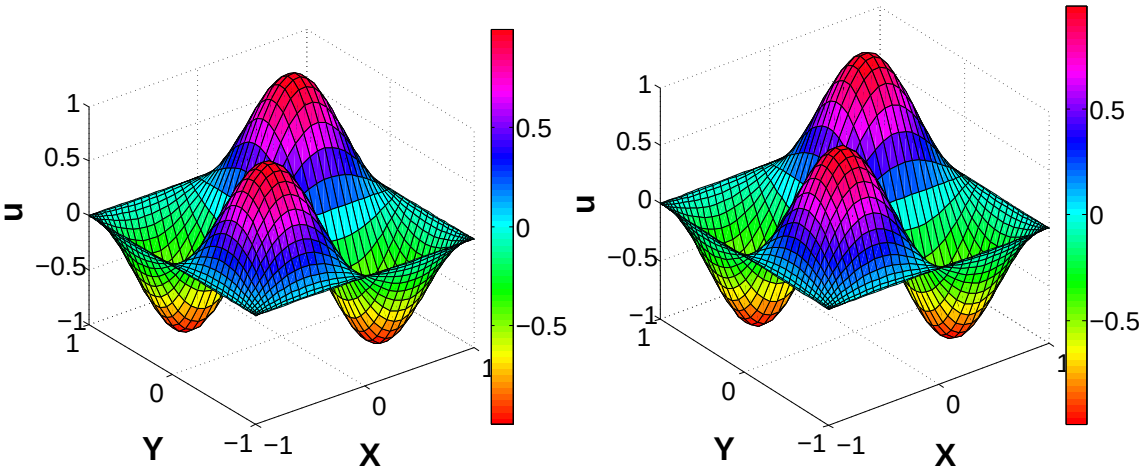
Now, the proposed method was used for the 4-th order problem, here a model of a simply-supported square-thin-plate was considered under the action of a distributed loading of the form $f(x, y)$. The equation and boundary conditions for the simply-supported plate [Mai-Duy and Tanner, 2007] are:

$$\begin{gathered} \triangle^2 u(x,y) = f(x,y) \\ u = g, \triangle u = h \\ \Omega = [-1,1]^2 \\ f(x,y) = 4\pi^4 sin(\pi x) sin(\pi y) \\ g = 0 \\ h = 0 \end{gathered} \tag{2.27}$$

This problem can be solved analytically and the exact solution for the deflection is:

$$u(x, y) = sin(\pi x)sin(\pi y) \tag{2.28}$$

which is shown in Figure(2.3(a)), the solution of the same problem for the discretization of (41×41) calculated by the Chebyshev method is also shown in Figure(2.3(b)), the error between the numerical solution and the exact solution are shown in Figure(2.4). Although the error is very small enough. We can see that the Chebyshev method is an efficient way for solving high order PDE problems.



(a) Exact solution for 4th order problem

(b) Solution by chebyshev method

Figure 2.3: Exact solution and Chebyshev solution for 4th order problem

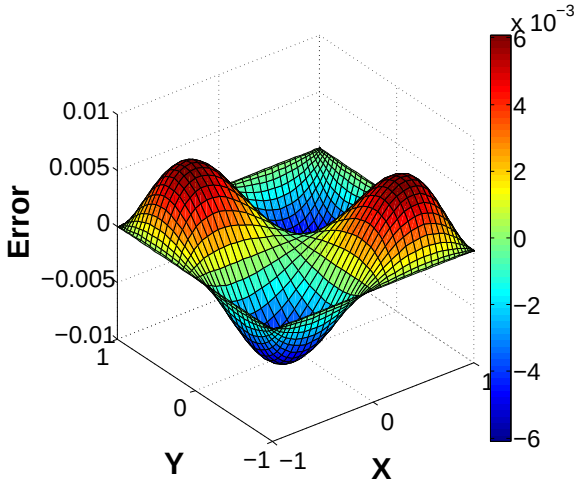


Figure 2.4: Error between the exact solution and Chebyshev solution

2.1.4 2D incompressible Flow problem

2.1.4.1 Chebyshev solution for the 2D incompressible Flow problem

Now, let us consider the flow problem which we mentioned before: the Lagrange stream function form for the flow in a square cavity.

We consider the following set of boundary conditions for the bottom, the up side, the left side and the right side,respectively:

$$\begin{aligned} \Psi &= C_1, \quad \frac{\partial \Psi}{\partial y} = 0 \\ \Psi &= C_1, \quad \frac{\partial \Psi}{\partial y} = C_2 \\ \Psi &= C_1, \quad \frac{\partial \Psi}{\partial x} = 0 \\ \Psi &= C_1, \quad \frac{\partial \Psi}{\partial x} = 0 \end{aligned} \tag{2.29}$$

the parameter C_1 and C_2 for boundary condition in Eq.(2.29) are

$$\begin{aligned} C_1 &= 0 \\ C_2 &= x^2 - 1 \end{aligned} \tag{2.30}$$

by using Chebyshev method, we obtain the solution for the discretization of (40×40) which is displayed in contour form in Fig.(2.5)

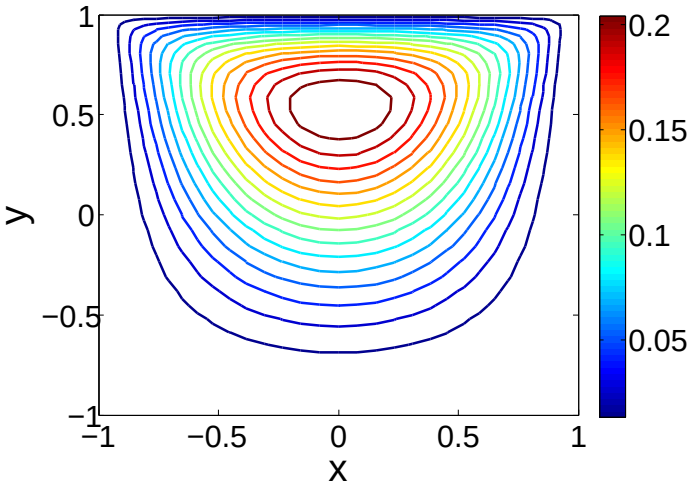


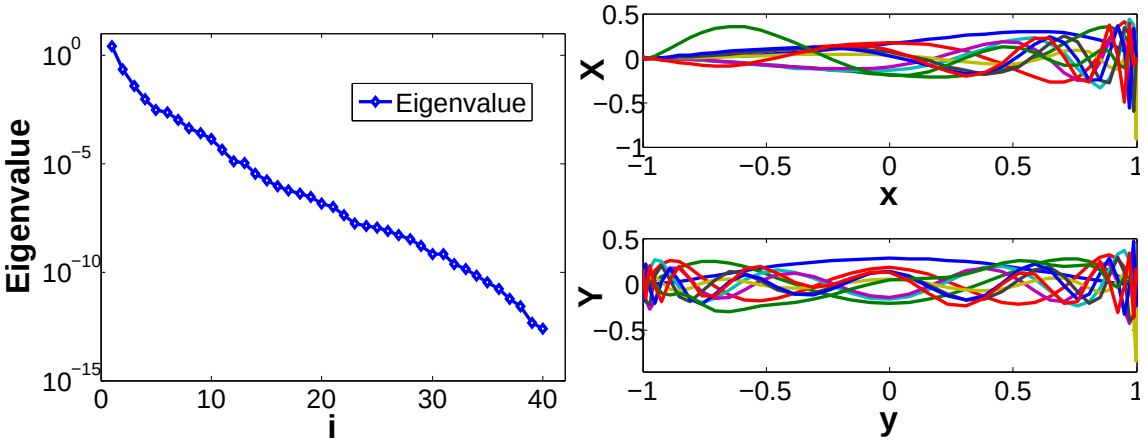
Figure 2.5: The contour for the 2D Stokes problem

2.1.4.2 Separability of the flow problem in stream function formulation

By applying the Singular Value Decomposition algorithm, we can write the computed stream function as the following finite sum:

$$\Psi(x,y) = \sum_{i=1}^{n} \alpha_i X(x) * Y_i(y) \tag{2.31}$$

In Figure(2.6(a)), we show the singular values α_i for the flow problem.



(a) Eigenvalue for the flow problem

(b) POD for represent the flow problem

Figure 2.6: Singular values and finite sum representation for the flow problem

By truncating the sum to the first 10 terms (which are shown in the Figure(2.6(b)); we can already represent the stream function with a small error, as shown in Figure(2.7).

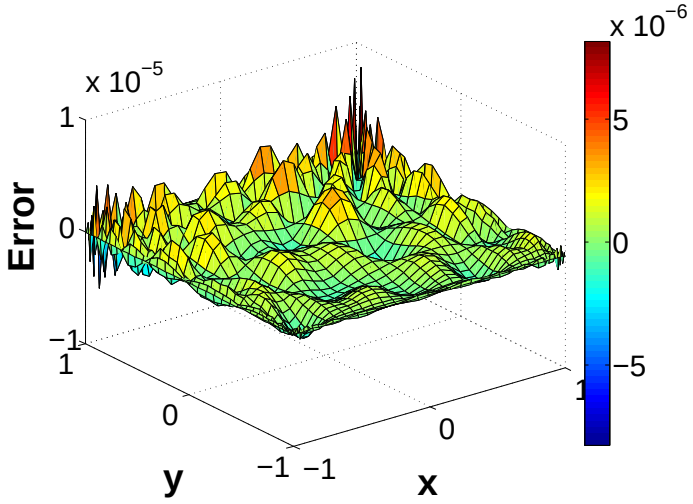


Figure 2.7: Error between the finite sum with 10 terms and the solution

2.2 PGD formulation for High order PDEs

2.2.1 Coupling PGD and Chebyshev method

In this section, we illustrate the coupling between the PGD separated representation framework and the proposed Chebyshev method on the harmonic problem and the biharmonic problem.

The aim of the method is to compute N couples of functions $(X_i(x), Y_i(y)),\ i = 1, \cdots, N$ such that $X_i(x)\ i = 1, \cdots, N$ and $Y_i(y)\ i = 1, \cdots, N$ are defined in 1D domains. The 2D solution for the problem (1) and (3) is sought as:

$$u(x, y) = \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \tag{2.32}$$

The weak form of problem (2.1) and (2.3) writes as:

Find $u(x, y)$, in an appropriate functional space, verifying the boundary conditions (2.2) and (2.4), respectively, for harmonic problem and the biharmonic problem:

$$\int_{\Omega_x} \int_{\Omega_y} u^*(x, y)(\triangle u(x, y) - f(x, y))\ dx \cdot dy = 0 \tag{2.33}$$

for the harmonic problem; and

$$\int_{\Omega_x} \int_{\Omega_y} u^*(x, y)(\triangle^2 u(x, y) - f(x, y))\ dx \cdot dy = 0 \tag{2.34}$$

for the biharmonic problem, all the functions $u^*(x, y)$ being also in an appropriate functional space.

We now compute the functions involved in the separated representation. We suppose that the set of functional couples $(X_i(x), Y_i(y)),\ i = 1, \cdots, n$ with $1 \leq n < N$ are already known (they have been previously computed) and at the present iteration we search the enrichment couple $(R(x), S(y))$ by applying an alternating directions fixed-point algorithm which after convergence will constitute the next functional couple (X_{n+1}, Y_{n+1}). Hence at the present iteration, $n + 1$, we assume the separated representation

$$u(x, y) \approx \sum_{i=1}^{n} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y) \tag{2.35}$$

The weighting function $u^*(x, y)$ is then assumed as

$$u^*(x, y) = R^*(x) \cdot S(y) + R(x) \cdot S^*(y) \tag{2.36}$$

As the Eq.(2.33) and the Eq.(2.34) are similar, we give the details of the PGD formulation of the biharmonic problem only, the harmonic problem be derived easily.

After introducing the trial and test function into the weak form, it results:

$$\begin{gathered}\int_\Omega \left[R^*(x)\cdot S(y)+R(x)\cdot S^*(y)\right]\\ \left[\frac{\partial^4 R(x)}{\partial x^4}\cdot S(y)+2\frac{\partial^2 R(x)}{\partial x^2}\cdot\frac{\partial^2 S(y)}{\partial y^2}+R(x)\cdot\frac{\partial^4 S(y)}{\partial y^4}\right]dx\cdot dy\\ =\int_\Omega \left[R^*(x)\cdot S(y)+R(x)\cdot S^*(y)\right]\cdot\left[f(x,y)-\right.\\ \left.\sum_{i=1}^n\left(\frac{\partial^4 X_i(x)}{\partial x^4}\cdot Y_i(y)+2\frac{\partial^2 X_i(x)}{\partial x^2}\cdot\frac{\partial^2 Y_i(y)}{\partial y^2}+X_i(x)\frac{\partial^4 Y_i(y)}{\partial y^4}\right)\right]dx\cdot dy\end{gathered} \tag{2.37}$$

First, we suppose that $R(x)$ is known, implying that $R^*(x)=0$. Thus, equation(2.37) reads

$$\begin{gathered}\int_{\Omega_y} S^*(y)[\alpha_{Rx}S(y)+2\beta_{Rx}\frac{\partial^2 S(y)}{\partial y^2}+\gamma_{Rx}\frac{\partial^4 S(y)}{\partial y^4}]dy=\\ \int_{\Omega_y} S^*(y)[\eta_{Rx}(y)-\sum_{i=1}^n(\alpha^i_{Rx}Y_i(y)+2\beta^i_{Rx}\frac{\partial^2 Y_i(y)}{\partial y^2}+\gamma^i_{Rx}\frac{\partial^4 Y_i(y)}{\partial y^4})]dy\end{gathered} \tag{2.38}$$

where:

$$\begin{aligned}\alpha_{Rx}&=\int_{\Omega_x} R(x)\frac{\partial R^4(x)}{\partial x^4}dx\\ \beta_{Rx}&=\int_{\Omega_x} R(x)\frac{\partial R^2(x)}{\partial x^2}dx\\ \gamma_{Rx}&=\int_{\Omega_x} R(x)\ R(x)dx\\ \alpha^i_{Rx}&=\int_{\Omega_x} R(x)\frac{\partial X_i^4(x)}{\partial x^4}dx\\ \beta^i_{Rx}&=\int_{\Omega_x} R(x)\frac{\partial X_i^2(x)}{\partial x^2}dx\\ \gamma^i_{Rx}&=\int_{\Omega_x} R(x)\ X_i(x)dx\\ \eta_{Rx}(y)&=\int_{\Omega_x} R(x)\ f(x,y)dx\end{aligned} \tag{2.39}$$

As the weak formulation is satisfied for all $S^*(y)$, we can come back to its associated strong form:

$$\begin{gathered}\alpha_{Rx}S(y)+2\beta_{Rx}\frac{\partial^2 S(y)}{\partial y^2}+\gamma_{Rx}\frac{\partial^4 S(y)}{\partial y^4}=\\ \eta_{Rx}(y)-\sum_{i=1}^n(\alpha^i_{Rx}Y_i(y)+2\beta^i_{Rx}\frac{\partial^2 Y_i(y)}{\partial y^2}+\gamma^i_{Rx}\frac{\partial^4 Y_i(y)}{\partial y^4})\end{gathered} \tag{2.40}$$

This fourth order equation will be solved by using a Pseudo-Spectral Chebyshev method.

Now, from the function $S(y)$ just computed, we search $R(x)$. In this case, $S(y)$ being known, $S^*(y)$ vanishes and Eq.(2.37) reads:

$$\begin{gathered}\int_{\Omega_x} R^*(x)[\alpha_{Sy}\frac{\partial^4 R(x)}{\partial x^4} + 2\beta_{Sy}\frac{\partial^2 R(x)}{\partial x^2} + \gamma_{Sy}R(x)]dx = \\ \int_{\Omega_x} R^*(x)\eta_{_{Sy}}(x)dx - \\ \int_{\Omega_x} R^*(x)[\sum_{i=1}^{n}(\alpha^i_{Sy}\frac{\partial^4 X_i(x)}{\partial x^4} + 2\beta^i_{Sy}\frac{\partial^2 X_i(x)}{\partial x^2} + \gamma^i_{Sy}X_i(x))]dx\end{gathered} \tag{2.41}$$

where:

$$\begin{gathered}\alpha_{Sy} = \int_{\Omega_y} S(y)S(y)dy \\ \beta_{Sy} = \int_{\Omega_y} S(y)\frac{\partial S^2(y)}{\partial y^2}dy \\ \gamma_{Sy} = \int_{\Omega_y} S(y)\frac{\partial S^4(y)}{\partial y^4}dy \\ \alpha^i_{Sy} = \int_{\Omega_y} S(y)Y_i(y)dy \\ \beta^i_{Sy} = \int_{\Omega_y} S(y)\frac{\partial Y_i^2(y)}{\partial y^2}dy \\ \gamma^i_{Sy} = \int_{\Omega_y} S(y)\frac{\partial Y_i^4(y)}{\partial y^4}dy \\ \eta_{_{Sy}}(x) = \int_{\Omega_y} S(y)f(x,y)dy\end{gathered} \tag{2.42}$$

whose strong form reads:

$$\begin{gathered}\alpha_{Sy}\frac{\partial^4 R(x)}{\partial x^4} + 2\beta_{Sy}\frac{\partial^2 R(x)}{\partial x^2} + \gamma_{Sy}R(x) = \\ \eta_{_{Sy}}(x) - \sum_{i=1}^{n}(\alpha^i_{Sy}\frac{\partial^4 X_i(x)}{\partial x^4} + 2\beta^i_{Sy}\frac{\partial^2 X_i(x)}{\partial x^2} + \gamma^i_{Sy}X_i(x))\end{gathered} \tag{2.43}$$

that will be solved again by using a Pseudo-Spectral Chebyshev method.

These two steps are repeated until convergence to a fixed point. If we denote the functions $R(x)$ at the present and previous fixed point iterations as $R^p(x)$ and $R^{p-1}(x)$, respectively, and the same for the function $S(y)$, $S^p(y)$ and $S^{p-1}(y)$, the fixed point convergence criterion at the present iteration can be defined from:

$$e = \int_{\Omega_x \times \Omega_y} (R^p(x) \cdot S^p(y) - R^{p-1}(x) \cdot S^{p-1}(y))^2 dx \cdot dy \leq \varepsilon \tag{2.44}$$

where ε is a small enough parameter.

After convergence, we can define the next functional couple as: $X_{n+1} = R$ and $Y_{n+1} = S$.

The enrichment procedure must continue until reaching the convergence, that can be evaluated from the PGD enrichment error E:

$$E = \frac{\|\Delta^2 u - f(x,y)\|}{\|f(x,y)\|} \leq \tilde{\varepsilon} \tag{2.45}$$

with $\tilde{\varepsilon}$ another small enough parameter.

For the harmonic problem, it is a little simpler, so that after applying the same procedure, after doing the mathematics, Eq.(2.37) becomes:

$$\begin{aligned}&\int_\Omega \left[R^*(x)\cdot S(y)+R(x)\cdot S^*(y)\right]\left[\frac{\partial^2 R(x)}{\partial x^2}\cdot S(y)+R(x)\cdot\frac{\partial^2 S(y)}{\partial y^2}\right]dx\cdot dy\\&=\int_\Omega \left[R^*(x)\cdot S(y)+R(x)\cdot S^*(y)\right]\cdot[f(x,y)-\\&\sum_{i=1}^n(\frac{\partial^2 X_i(x)}{\partial x^2}\cdot Y_i(y)+X_i(x)\frac{\partial^2 Y_i(y)}{\partial y^2})\Big]\,dx\cdot dy\end{aligned}\tag{2.46}$$

and also Eq.(2.47)-Eq.(2.49) and Eq.(2.50) simplify into the following formulation:

$$\begin{aligned}&\int_{\Omega_y} S^*(y)[\beta_{Rx}S(y)+\gamma_{Rx}\frac{\partial^2 S(y)}{\partial y^2}]dy=\\&\int_{\Omega_y} S^*(y)[\eta_{Rx}(y)-\sum_{i=1}^n(\beta^i_{Rx}Y_i(y)+\gamma^i_{Rx}\frac{\partial^2 Y_i(y)}{\partial y^2})]dy\end{aligned}\tag{2.47}$$

$$\beta_{Rx}S(y)+\gamma_{Rx}\frac{\partial^2 S(y)}{\partial y^2}=\eta_{Rx}(y)-\sum_{i=1}^n(\beta^i_{Rx}Y_i(y)+\gamma^i_{Rx}\frac{\partial^2 Y_i(y)}{\partial y^2})\tag{2.48}$$

$$\begin{aligned}&\int_{\Omega_x} R^*(x)[\alpha_{Sy}\frac{\partial^2 R(x)}{\partial x^2}+\beta_{Sy}R(x)]dx=\\&\int_{\Omega_x} R^*(x)\eta_{Sy}(x)dx-\int_{\Omega_x} R^*(x)[\sum_{i=1}^n(\alpha^i_{Sy}\frac{\partial^2 X_i(x)}{\partial x^2}+\beta^i_{Sy}X_i(x))]dx\end{aligned}\tag{2.49}$$

$$\alpha_{Sy}\frac{\partial^2 R(x)}{\partial x^2}+\beta_{Sy}R(x)=\eta_{Sy}(x)-\sum_{i=1}^n(\alpha^i_{Sy}\frac{\partial^2 X_i(x)}{\partial x^2}+\beta^i_{Sy}X_i(x))\tag{2.50}$$

the parameters in Eq.(2.47)-(2.49) and Eq.(2.50) are the same as the parameters for the biharmonic problem, so these parameters can be chosen from the Eq.(2.39) and Eq.(2.42); and the strong formulation Eq.(2.48) and Eq.(2.50) can be solved by using a pseudo-spectral Chebyshev method.

2.2.2 Pseudo-Spectral Chebyshev method

We assume the general form of a 1D fourth order differential boundary value problem for the bi-harmonic problem:

$$a\frac{d^4u}{dx^4}+b\frac{d^2u}{dx^2}+cu=g(x)\tag{2.51}$$

and a 1D 2nd ODE for the harmonic problem:

$$b\frac{d^2u}{dx^2}+cu=g(x)\tag{2.52}$$

The unknown function $u(x)$ is approximated in $\Omega_x =]-1,1[$ from:

$$u(x) = \sum_{i=1}^{i=M} \alpha_i \cdot T_i(x) \tag{2.53}$$

where M denotes the number of nodes considered on Ω_x, whose coordinates are given by

$$x_i = \cos\left(\frac{(i-1)\cdot\pi}{M-1}\right), \quad i = 1, \cdots, M \tag{2.54}$$

The interpolations $T_i(x)$ verify the Kronecker delta property, i.e. $T_i(x_k) = \delta_{ik}$.

At each node k, $3 \leq k \leq M-2$ (the remaining 4 nodes will be used for enforcing the boundary conditions) the discrete equations Eq.(2.51)and Eq.(2.52) writes as :

$$a \cdot \sum_{i=1}^{i=M} \alpha_i \cdot \frac{dT_i^4}{dx^4}|_{x_k} + b \cdot \sum_{i=1}^{i=M} \alpha_i \cdot \frac{dT_i^2}{dx^2}|_{x_k} + c \cdot \alpha_k = f(x_k) \tag{2.55}$$

and

$$b \cdot \sum_{i=1}^{i=M} \alpha_i \cdot \frac{dT_i^2}{dx^2}|_{x_k} + c \cdot \alpha_k = f(x_k) \tag{2.56}$$

When we assume that the first modes of the separated representation verify the boundary conditions (2.2) and (2.4), the functions $R(x)$ and $S(y)$ are subjected to homogeneous Dirichlet and Neumann conditions. Thus, we should enforce $u(x_1) = u(x_M) = 0$ and $\frac{du}{dx}|_{x_1} = \frac{du}{dx}|_{x_M} = 0$ for the bi-harmonic problem and $u(x_1) = u(x_M) = 0$ for harmonic problem. These conditions results in:

$$\begin{cases} \alpha_1 = 0 \\ \sum_{i=1}^{i=M} \alpha_i \cdot \frac{dT_i(x)}{dx}|_{x_1} = 0 \\ \alpha_M = 0 \\ \sum_{i=1}^{i=M} \alpha_i \cdot \frac{dT_i(x)}{dx}|_{x_M} = 0 \end{cases} \tag{2.57}$$

for the bi-harmonic problem, and

$$\begin{cases} \alpha_1 = 0 \\ \alpha_M = 0 \end{cases} \tag{2.58}$$

for the harmonic problem.

2.3 Numerical results

In this section, we present some numerical results from the application of the above strategy (PGD–Chebyshev pseudo-spectral method) on some test cases. Strategies for enforcing the Non-homogeneous boundary conditions are discussed.

2.3.1 2D Laplace problem

First, we use the proposed technique for calculating the Laplace problem.

The governing equation reads:

$$\triangle u(x,y) = f(x,y) \qquad -1 \leq (x,y) \leq 1 \tag{2.59}$$

where

$$f(x,y) = -1 \tag{2.60}$$

and the boundary condition is

$$u = 0 \quad on \quad \partial\Omega \tag{2.61}$$

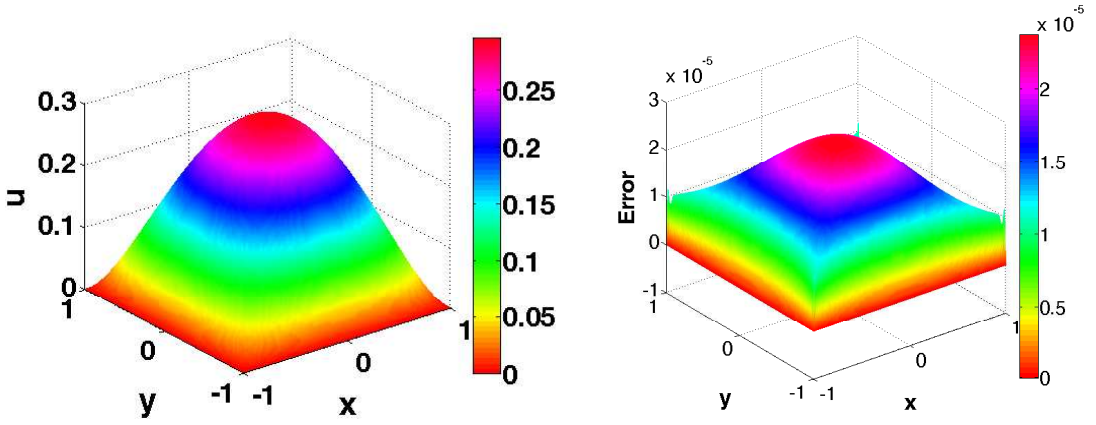
The solution computed by Chebyshev-PGD method for discretization of (100×100) with only 6 modes is shown in Fig.(2.8(a)). The error between the method proposed in this work with a 2D Finite difference method for the same discretization is given in Fig.(2.8(b)). We see that the method rapidly achieves a high level of accuracy.

The different PGD modes (enrichments in the solution) for this problem are also plotted in Fig.(2.9(a)). The $L-2error$ between PGD solution and the FD solution vs the number of PGD modes is shown in Fig.(2.9(a)), where the L^2Error in the Fig.(2.9(b)) is used:

$$Error = \frac{\int_\Omega (U^n_{PGD} - U_{ref})^2 d\Omega}{\int_\Omega U^2_{ref} d\Omega} \tag{2.62}$$

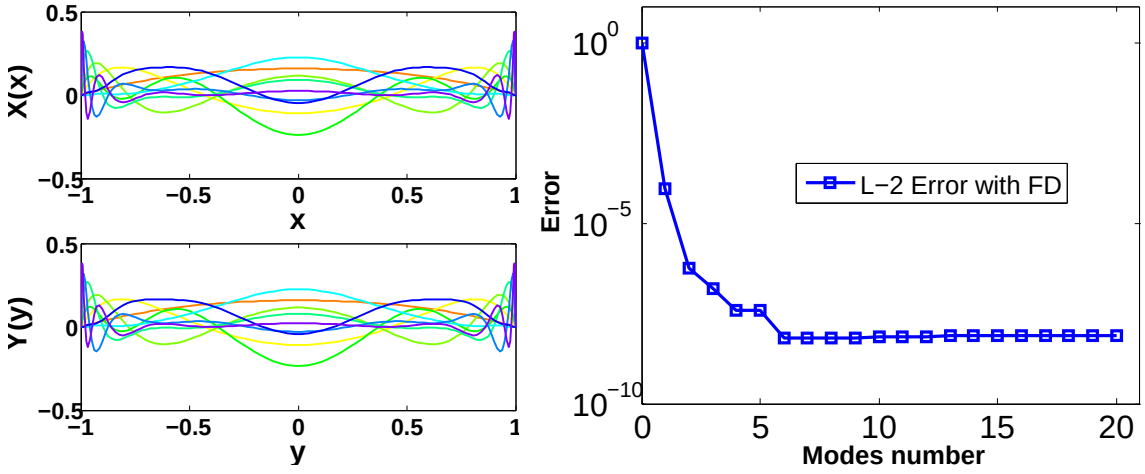
we see that the PGD method for the Laplace problem is converged very fast.

This problem have an analytical solution, when we compare the PGD solution and the Finite Difference solution with the analytical solution with the same grids number (100×100), we can see that the PGD method have a higher accuracy than the Finite Difference method solution from the Figure(2.10). And the PGD convergence is given in the Figure(2.11).



(a) PGD solution (b) PGD Error with FD solution

Figure 2.8: Result for laplace problem



(a) PGD modes (b) PGD convergence

Figure 2.9: PGD modes and PGD convergence

2.3.2 4th order PDE problem

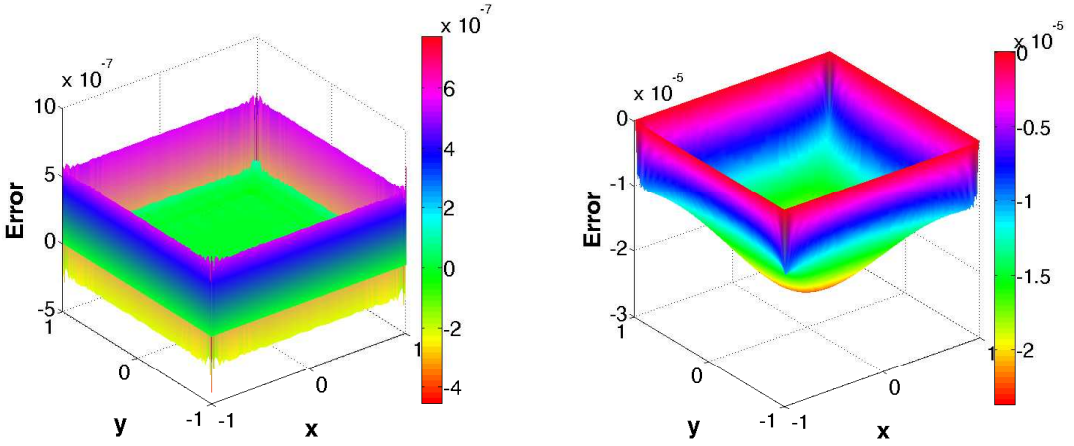
Now, we consider the 2-D bi-harmonic problem in([Kirby and Yosibash, 2003]), which corresponds to a model for the deflection of a plate:

$$\triangle^2 u = f(x,y) \quad in \quad \Omega = [-1,1]^2 \tag{2.63}$$

where

$$f(x,y) = -8\pi^4[cos(2\pi^2 x)sin^2(\pi y) + sin^2(\pi x)cos(2\pi y) - cos(2\pi x)cos(2\pi y)] \tag{2.64}$$

$$u = 0 \; and \; \frac{\partial u}{\partial n} = 0 \quad on \quad \partial\Omega \tag{2.65}$$



(a) PGD Error with Exact solution

(b) FD Error with Exact solution

Figure 2.10: Error for PGD and FD solution compare with exact solution

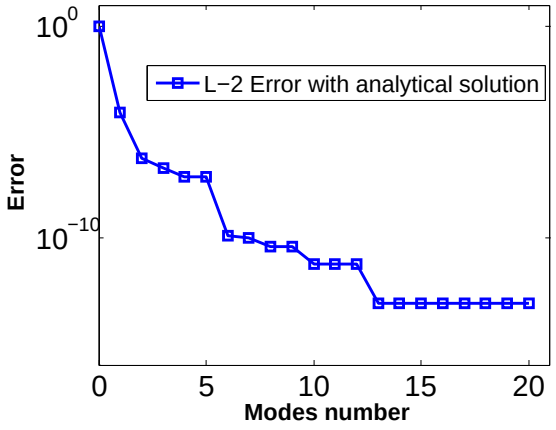
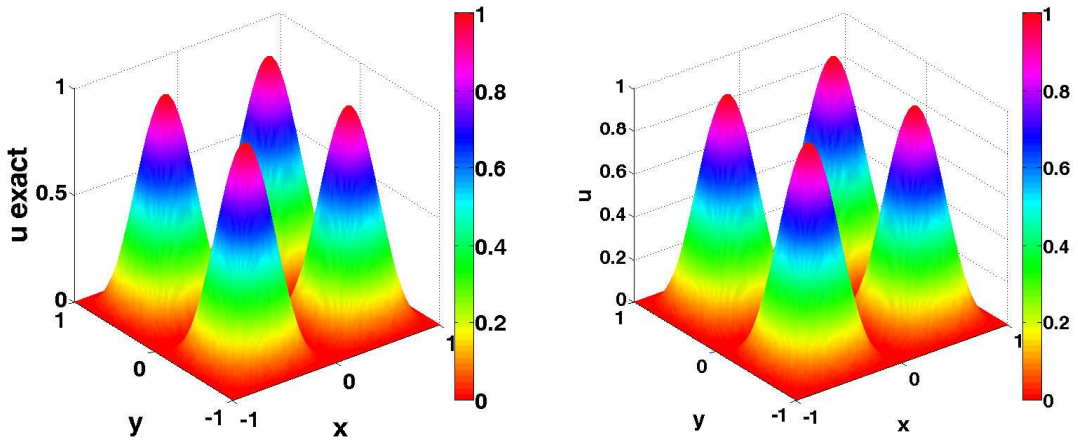


Figure 2.11: $L-2$ Error for PGD and Exact solution VS PGD modes

This problem can be solved analytically and the exact solution for the deflection is given by:

$$u(x,y) = sin^2(\pi x)sin^2(\pi y) \tag{2.66}$$

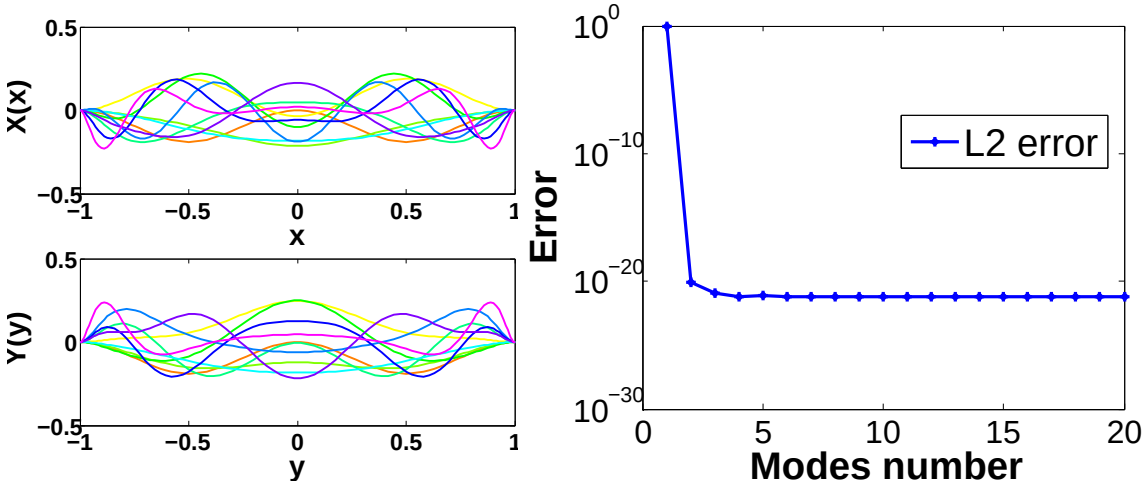
The exact solution and the solution by the PGD method for discretization of (100×100) are shown in Figure(2.12).



(a) Exact solution (b) PGD solution

Figure 2.12: Exact solution and PGD solution

The PGD modes are shown in Figure(2.13(a)). The convergence of the method with the number of PGD modes is illustrated in Figure(2.13(b)).Where the L^2 norm Error (as Eq.(2.62)) is used for evaluate the PGD convergence; we can see that the PGD solution converges very fast, as we can see from the exact solution, it is only one modes needed for the solution. And the error between the solution by PGD and the exact solution is shown in Figure(2.14).



(a) PGD mode (b) PGD converge

Figure 2.13: PGD mode and PGD convergence

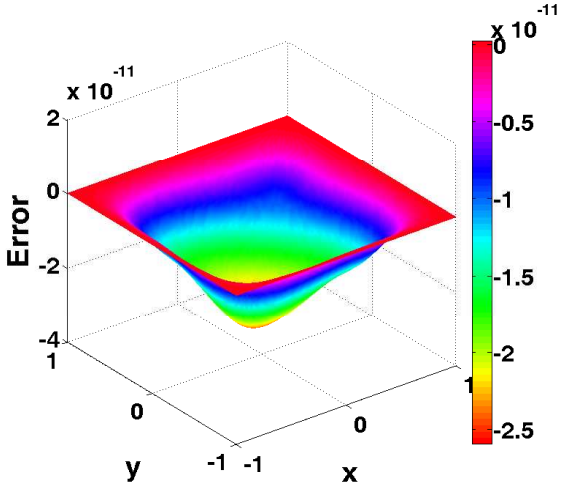


Figure 2.14: Error between the PGD solution and exact solution

This method can be used for compute the deformation of the rectangular plate in the domain of $(0, a) \times (0, b)$, the governing equation as:

$$\begin{aligned} \nabla^4 u = -\frac{q}{D} \quad & in\Omega \\ u = 0 \quad & on\partial\Omega \\ \frac{\partial^2 u}{\partial n^2} = 0 \quad & on\partial\Omega \end{aligned} \tag{2.67}$$

with $q(x, y) = q_0 sin\frac{\pi x}{a} sin\frac{\pi y}{b}$. One of the boundary condition is the second derivative not the same as the boundary for the analytical problem; By using the variable changement relation:

$$\begin{aligned} x &= \frac{\xi+1}{2} a \\ y &= \frac{\eta+1}{2} b \end{aligned} \tag{2.68}$$

we can change the problem in $(0, a) \times (0, b)$ to $](-1, 1) \times (-1, 1)[$, and then use the PGD method very easily to compute this problem, here, we don't show the results for this problem, we just distribute the cpu time and L^2 error(as Eq.(2.62)) compare between the 2-D Chebyshev method and PGD method in the Figure(2.15). And the $L-2$ error between the two methods is decreased as the increase of grids node. Base on the computation of the same PC, we can see that the PGD method is faster than the Chebyshev method. and when we increase the discretization more than $90 * 90$, the PC will out of memory. But the PGD can do the problem with hundreds without problem.

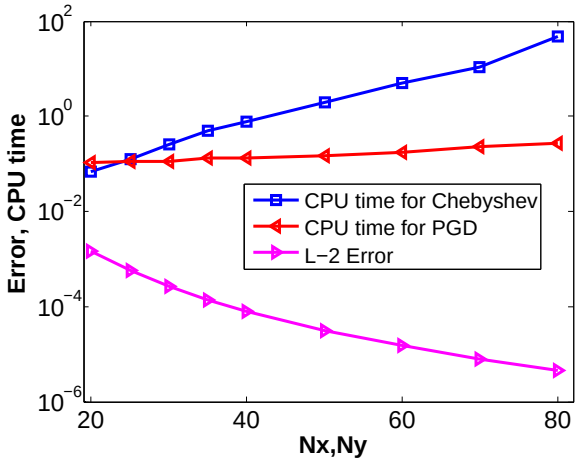


Figure 2.15: CPU time and L^2 error compare between the 2-D Chebyshev method and PGD method

2.3.3 Chebyshev Polynomials for Boundary Value Problem

In this subsection, we will detail three kinds of methodologies for enforcing the boundary conditions which will be used in dealing with the non-homogeneous Boundary Condition Biharmonic Problem modeling the flow in a square driven cavity.

2.3.3.1 Method 1

Here, we propose the strategy to represent the boundary value using the chebyshev polynomials as follows, the function $f(x, y)$ can be expressed as

$$f(x, y) = \sum \sum \alpha_{ij} T_i(x) T_j(y) \tag{2.69}$$

where $T_i(x)$ and $T_j(y)$ are the Chebyshev polynomials of the first kind, which have the relation.

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{2.70}$$

The first few Chebyshev polynomials $T_i(x)$ and $T_j(y)$ are

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ T_5(x) &= 16x^5 - 20x^3 + 5x \\ T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1 \end{aligned} \tag{2.71}$$

and

$$\begin{aligned} T_0(y) &= 1 \\ T_1(y) &= y \\ T_2(y) &= 2y^2 - 1 \\ T_3(y) &= 4y^3 - 3y \\ T_4(y) &= 8y^4 - 8y^2 + 1 \\ T_5(y) &= 16y^5 - 20y^3 + 5y \\ T_6(y) &= 32y^6 - 48y^4 + 18y^2 - 1 \end{aligned} \tag{2.72}$$

Using these polynomials require the identification of 49 (7^2) coefficients.

The boundary condition of the problem is

$$\begin{aligned} f(x,y) &= f1 \quad x = bound, y = bound \\ \tfrac{\partial f(x,y)}{\partial n} &= f2 \quad x = bound, y = bound \end{aligned} \tag{2.73}$$

we let $f(x,y)$ satisfy the boundary condition, there are $4*7-4=24$ equation for $f(x,y)$ at the boundary value, and $6*4=24$ equation for the $\frac{\partial f(x,y)}{\partial n}$ at the boundary nodes(not the same nodes as for the value $f(x,y)$). 48 equations for 49 coefficients, in the matrix form is

$$Ax = B \tag{2.74}$$

Here, A is a $48*49$ rectangle matrix have relation with the polynomials. x is a $49*1$ vector for the coefficients α_{ij}. The matlab function x=linsolve(A,B) was used for solving the problem.

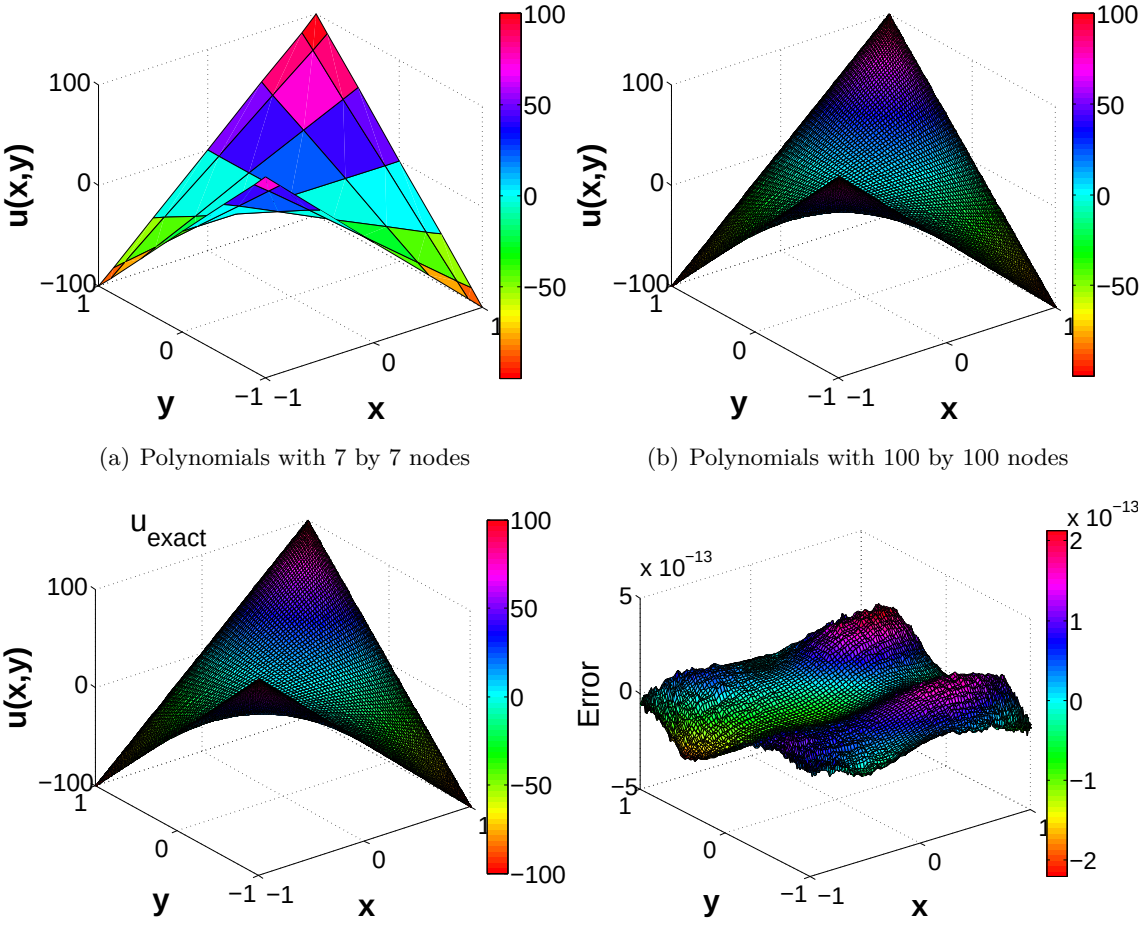
Then through Eq.(2.69) for every discrete nodes in the domain, we get the function value $f(x,y)$ in the domain.

Test 1: $f(x,y) = 100xy$

Suppose that, we don't know the function

$$f(x,y) = 100xy \tag{2.75}$$

and we just know the value of $f(x,y)$ and the derivative value $\frac{\partial f(x,y)}{\partial n}$ at the boundary of the domain. In Fig.(2.16(a)) and Fig.(2.16(b)) we show the result obtained by applying the above method for $7 * 7$ nodes and $100 * 100$ nodes. Fig.(2.16(b)) shows the exact result for the same test problem, while the error between the exact result and the $100 * 100$ case is plot in Fig.(2.16(d)). The error is very small. For this problem, there is only one coefficient α_{ij}, which is exactly 100 before xy. This is why the error is very small.



(a) Polynomials with 7 by 7 nodes

(b) Polynomials with 100 by 100 nodes

(c) Exact solution $f(x,y) = 100xy$

(d) error between the exact and the 100 by 100 case

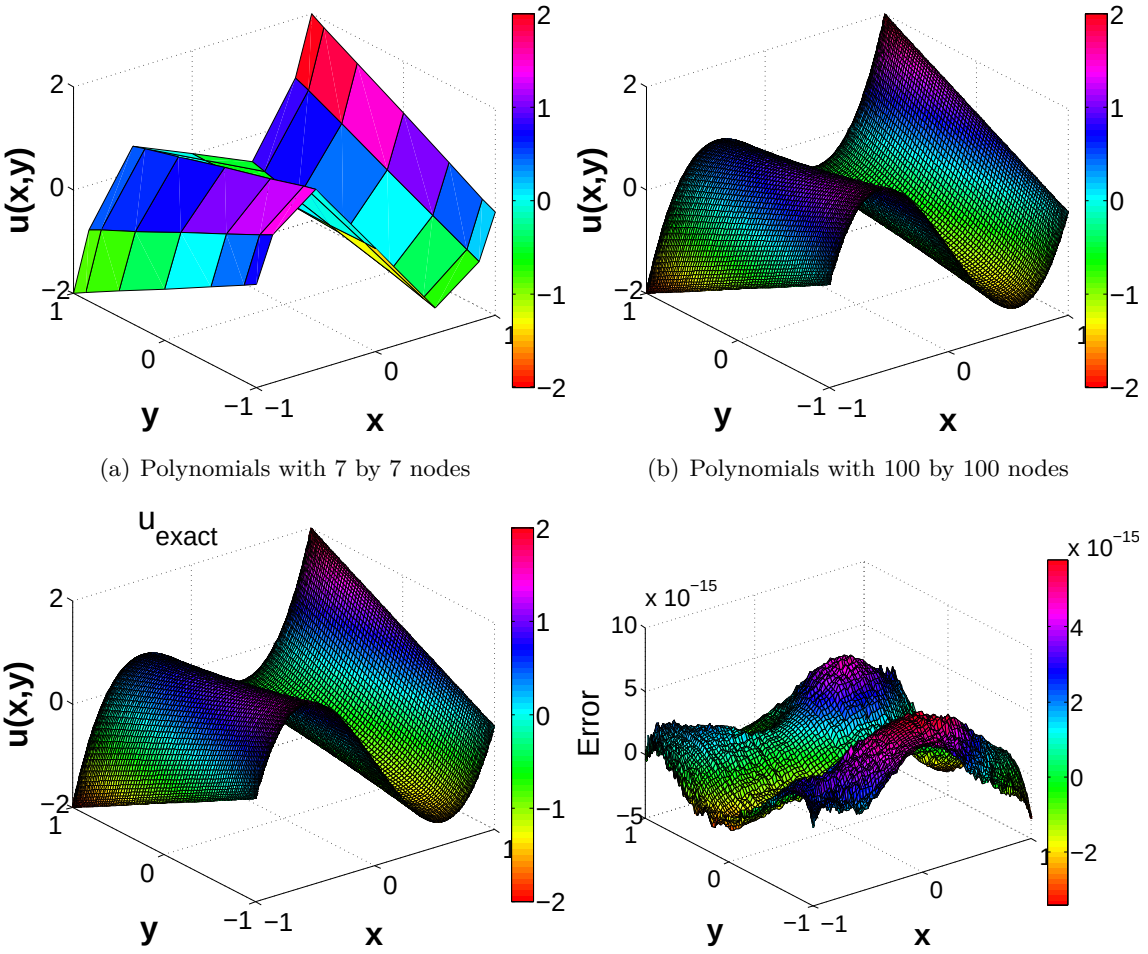
Figure 2.16: Test case 1

Test 2: $f(x,y) = 4x^3 - 3x + xy$

The test function changed to

$$f(x,y) = 4x^3 - 3x + xy; \tag{2.76}$$

In Fig.(2.17(a)) and Fig.(2.17(b)) we show the results of the method applied to Eq.(2.76) for $7 * 7$ nodes and $100 * 100$ nodes. Fig.(2.17(c)) gives the exact result for this problem while the error between the exact result and the 100 by 100 case result is plot in Fig.(2.17(d)). We can draw similar conclusions.



(a) Polynomials with 7 by 7 nodes

(b) Polynomials with 100 by 100 nodes

(c) Exact solution: $f(x,y) = 4 * x^3 - 3 * x + x * y$

(d) error between the exact solution and the 100 by 100 case

Figure 2.17: Test Case 2

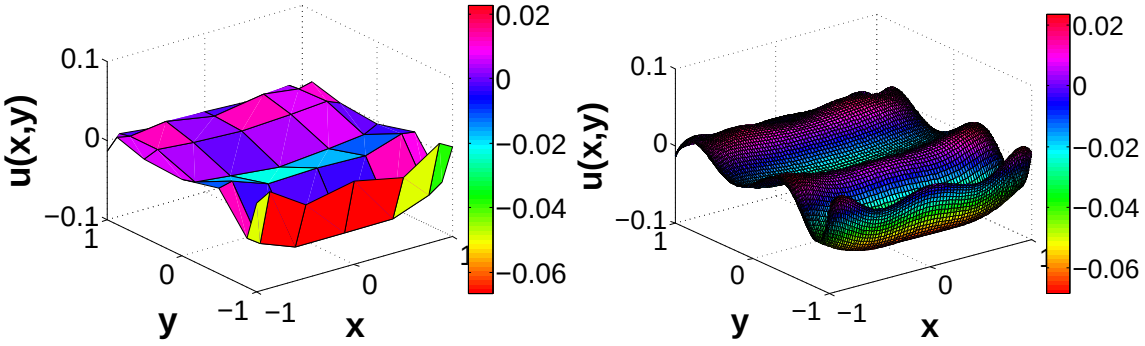
Flow problem

Now, let us use this strategy the flow problem of a lid driven cavity, we just know the value $f(x,y)$ and the derivative value $\frac{\partial f(x,y)}{\partial n}$ at the boundary of the domain. The

domain is $(x, y) \in]-1, 1[$. The boundary condition for the problem is

$$\begin{aligned} f(x,y) &= 0 \quad x = \pm 1 \\ f(x,y) &= 0 \quad y = \pm 1 \\ \frac{\partial f(x,y)}{\partial y} &= 0 \quad x = \pm 1 \\ \frac{\partial f(x,y)}{\partial x} &= 1 \quad y = 1 \\ \frac{\partial f(x,y)}{\partial y} &= 0 \quad y = -1 \end{aligned} \tag{2.77}$$

the result of the flow problem is plot in Fig.2.18(a) and Fig.(2.18(b)). It is obvious that the method fails here to correctly represent the boundary values and derivatives.



(a) Polynomial with 7 by 7 nodes

(b) Polynomials with 100 by 100 nodes

Figure 2.18: Flow problem

Short discussion

The proposed strategy can represent the problem for the test 1 and test 2, which are particular as they can be exactly can be expressed by the Polynomials. The test 1 is exactly the polynomials $100T_1(x)T_1(y)$; and the test 2 is $T_3(x)T_0(y) + T_1(x)T_1(y)$.

Nevertheless this method can not be used to express the boundary value for the flow problem. Because the boundary value for the flow is known at sampling points only and cannot a priori be represented by the chosen polynomials. Moreover the system to solve to compute the Chebyshev polynomials describing the boundary conditions is rectangular and admits more than one solution.

2.3.3.2 Method 2

1.1D problem

Let us suppose that we can discretize the domain as:

$$x_i = cos(\frac{\pi i}{N}) \quad (i = 0, \cdots N) \tag{2.78}$$

where x_i are associated with the $N-$th order Chebyshev Polynomials $T_n(x)$.

Suppose then that a function $f(x)$ can be written as the sum of the Chebyshev polynomials[Mai-Duy and Tanner, 2007].

$$f(x) = \sum_{i=1}^{n} \alpha_i T_i(x) \tag{2.79}$$

The $n-$th order derivative of $f(x)$ is as follow:

$$\frac{df^n(x)}{dx^n} = \sum_{i=1}^{n} \alpha_i D_i^{(n)}(x) \tag{2.80}$$

where $D^n(x)$ is the $n-$th Chebyshev differentiation Matrix.

Knowing $n+1$ boundary conditions (values and derivatives) of the function the function can be determined easily through Eq.(2.79) and Eq.(2.80).

Example for 1D problem

Test 1

Suppose the boundary values of a function are known as follows, use Eq.(2.79) and Eq.(2.80) to build the linear system for the problem, and we can get the value at the discrete points.

$$\begin{aligned} f(-1) &= 0 \\ f(1) &= 0 \\ \tfrac{df(-1)}{dx} &= 0 \\ \tfrac{df(1)}{dx} &= 1 \end{aligned} \tag{2.81}$$

This problem has the exact solution

$$f(x) = \frac{1}{4}(x^3 + x^2 - x - 1) \tag{2.82}$$

The result of the problem and the exact function is described in Fig.(2.19(a)).

Test 2

If the boundary value of a function $f(x)$ is like this:

$$\begin{aligned} f(-1) &= 1 \\ \frac{df(-1)}{dx} &= -2 \\ \frac{df^2(-1)}{dx^2} &= 2 \\ \frac{df^3(-1)}{dx^3} &= -6 \end{aligned} \tag{2.83}$$

and the exact function $f(x)$ is:

$$f(x) = -x^3 - 2x^2 - 3x - 1 \tag{2.84}$$

The result of the problem and the exact function is plotted in Fig.(2.19(b)).

Test 3

If the boundary value of a function $f(x)$ is like this:

$$\begin{aligned} f(-1) &= -3 \\ \frac{df(-1)}{dx} &= 2 \\ \frac{df^2(-1)}{dx^2} &= -2 \\ \frac{df^3(1)}{dx^3} &= 6 \end{aligned} \tag{2.85}$$

and the exact function $f(x)$ is:

$$f(x) = x^3 + 2x^2 + 3x - 1 \tag{2.86}$$

The result of the problem and the exact function is shown in Fig.(2.19(c)).

Test 4

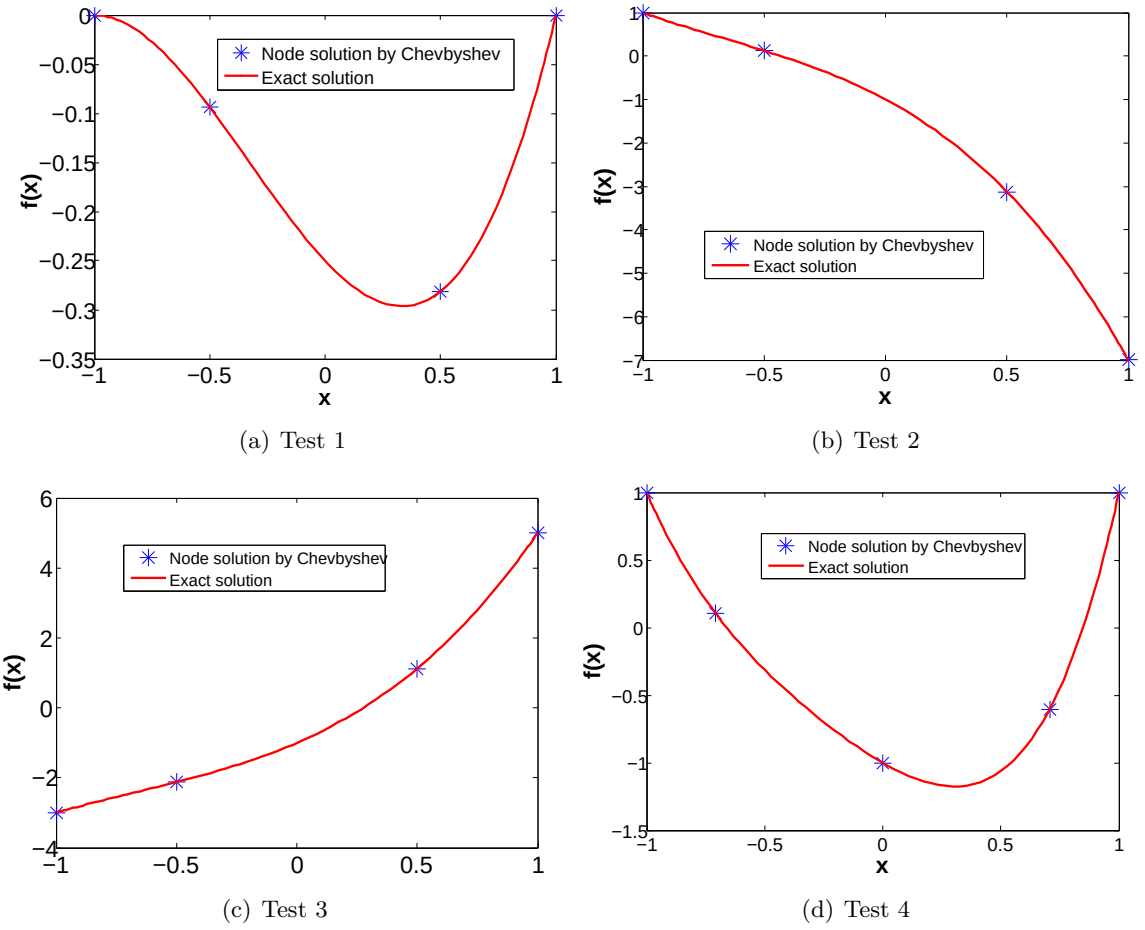
And also we can use this method to solve a high order differential problem.

$$\begin{aligned} f(-1) &= 1 \\ \frac{df(-1)}{dx} &= -4 \\ \frac{df^2(-1)}{dx^2} &= 8 \\ \frac{df^3(-1)}{dx^3} &= -18 \\ \frac{df^4(-1)}{dx^4} &= 24 \end{aligned} \tag{2.87}$$

and the exact solution is

$$f(x) = x^4 + x^3 + x^2 - x - 1 \tag{2.88}$$

we give the result in the Fig.(2.19(d)).



(a) Test 1

(b) Test 2

(c) Test 3

(d) Test 4

Figure 2.19: Result from Chebyshev Polynomials for Boundary Value Problem

2.2D problem

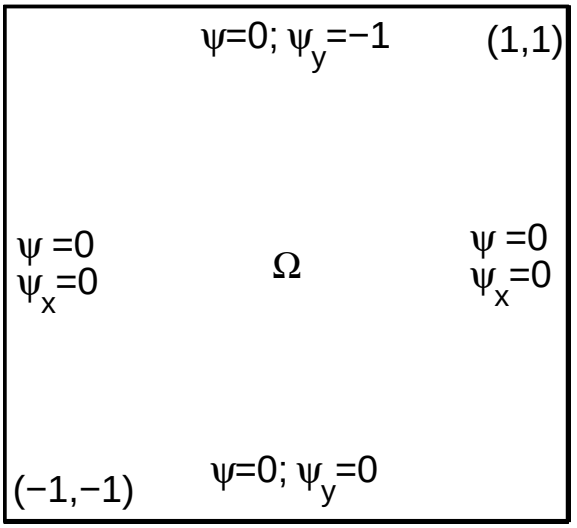


Figure 2.20: Geometry and Boundary Condition for the 2D problem

We assume here that there is a function $\psi(x, y)$ in a 2D domain shown in Figure(2.20), and the boundary conditions, respectively, for the bottom side, the top side, the left side and the right side are as follow:

$$\begin{aligned} &\psi = 0, \quad \tfrac{\partial \psi}{\partial y} = 0, \quad y = -1 \\ &\psi = 0, \quad \tfrac{\partial \psi}{\partial y} = -1, \quad y = 1 \\ &\psi = 0, \quad \tfrac{\partial \psi}{\partial x} = 0, \quad x = -1 \\ &\psi = 0, \quad \tfrac{\partial \psi}{\partial x} = 0, \quad x = 1 \end{aligned} \tag{2.89}$$

The function $\psi(x, y)$ is identified using chebyshev polynomials using the following method:

$$\psi(x, y) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} T_i(x) T_j(y) \alpha_{ij} \tag{2.90}$$

Suppose there are n discrete collocation Chebyshev points in the x, y direction, respectively, there are therefore n^2 unknown coefficients, so n^2 equations are need to get the unknown coefficient, and the number of the boundary condition is $4n - 4 + 4(n-2)$; if $n^2 = 4n - 4 + 4(n-2)$, the problem could be solved. Actually when $n = 6$, the problem can be solved.

When one looks back the boundary condition, there were two type boundary condition, the first one is,

$$\psi(x_{bound}, y_{bound}) = \sum_{i=1}^{6} \sum_{j=1}^{6} T_i(x) T_j(y) \alpha_{ij} \tag{2.91}$$

By using the relationship $T_i(x_k) = \delta_{ik}$ and the characteristic of the δ function, the Eq.(2.89) for the first type boundary condition is simplified as:

$$\begin{aligned} \psi(x_{bound=K}, y_{bound=L}) = \quad & \alpha_{1L}, L = 1 \cdots 6 \\ & \alpha_{6L}, L = 1 \cdots 6 \\ & \alpha_{K1}, K = 2 \cdots 5 \\ & \alpha_{K6}, K = 2 \cdots 5 \end{aligned} \tag{2.92}$$

for the first kind of boundary condition, there are 20 equation for 20 unknown coefficients,

And there are another 16 unknown coefficients determined by the second type boundary condition $\frac{\partial \psi}{\partial n}$, such as:

$$\frac{\partial \psi(x,y)}{\partial y} = \sum_{j=2}^{5} T_j'(y_L) \alpha_{(kj)} = \sum_{j=2}^{5} D_{Lj}^{(1)} \alpha_{(Kj)} = 0, (K = 2 \cdots 5, y_L = -1) \tag{2.93}$$

We use the same for the other 3 boundary conditions.

$$\begin{aligned} \tfrac{\partial \psi(x,y)}{\partial y} &= \textstyle\sum_{j=2}^{5} T_j'(y_L) \alpha_{(kj)} = \sum_{j=2}^{5} D_{Lj}^{(1)} \alpha_{(Kj)} = -1, (K = 2 \cdots 5, y_L = 1) \\ \tfrac{\partial \psi(x,y)}{\partial x} &= \textstyle\sum_{i=2}^{5} T_i'(x_K) \alpha_{(il)} = \sum_{i=2}^{5} D_{Ki}^{(1)} \alpha_{(iL)} = 0, (L = 2 \cdots 5, x_K = -1) \\ \tfrac{\partial \psi(x,y)}{\partial x} &= \textstyle\sum_{i=2}^{5} T_i'(x_K) \alpha_{(il)} = \sum_{i=2}^{5} D_{Ki}^{(1)} \alpha_{(iL)} = 0, (L = 2 \cdots 5, x_K = 1) \end{aligned} \tag{2.94}$$

So, there are 36 equation for the 36 coefficients, which can be written as a system as follow:

$$L\alpha = \beta \tag{2.95}$$

The system (see Eq.(2.95)) which is singular is solved by the Moore-Penrose pseudo inverse of matrix L, by $Pinv(L) * \beta$, the result is shown in the Figure(2.21), and when we check the $norm(L * \alpha - \beta)$ is not 0, but 0.7876. So the solution is not the exact solution, but an approximate representation.

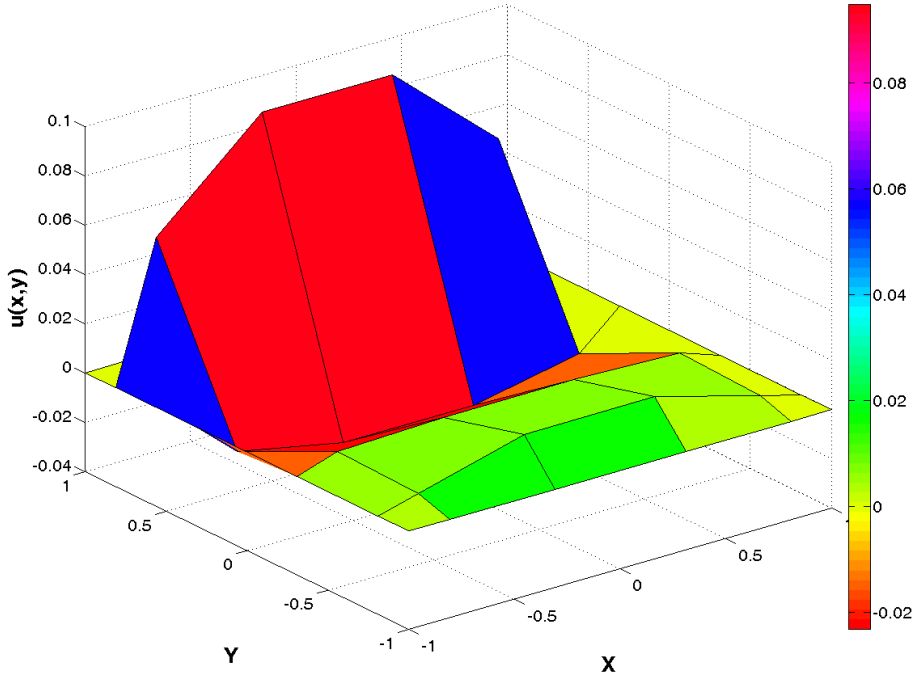


Figure 2.21: The solution from chebyshev method

2.3.3.3 Method 3

Suppose that we want to describe the boundary value problem for the following 2D Biharmonic problem:

$$\Delta^2 u = f \ in \qquad \Omega = (-1,1)^2 \tag{2.96}$$

with the following boundary conditions:

$$u = f_1; \qquad \frac{\partial u}{\partial n} = f_2 : \ on \qquad \Gamma \in \partial\Omega \tag{2.97}$$

We use the Spectral Collocation Chebyshev Method for the Eq.(2.96), and for the boundary condition Eq.(2.97), we multiply a penalization parameter as follow:

$$\begin{aligned} \lambda u &= \lambda f_1 \\ \lambda \tfrac{\partial u}{\partial n} &= \lambda f_2 \end{aligned} \tag{2.98}$$

where λ is the penalization parameter.

The system consisting of the discrete biharmonic equation and the new boundary condition, can be written as

$$A * u = g \tag{2.99}$$

when the penalization parameter λ is very large, the system of the equation means that the influence from the equation can be neglected, and the BC for the problem is the important part for the problem. Actually, the matrix A is singular, and it can not be solved directly, it can be solved by the Moore-Penrose pseudo inverse of matrix $A^T * A$, and the solution can be obtained from,

$$u = Pinv(A^T * A) * (A^T * g) \tag{2.100}$$

The penalization parameter λ and the discretization for the domain are determined by a trial and error method.

Here, we take the flow problem which used in 2.3.3.1, and the boundary condition as

$$\begin{aligned}
\psi(x,y) &= 0 && x = \pm 1 \\
\psi(x,y) &= 0 && y = \pm 1 \\
\tfrac{\partial \psi(x,y)}{\partial y} &= 0 && x = \pm 1 \\
\tfrac{\partial \psi(x,y)}{\partial x} &= x^2 - 1 && y = 1 \\
\tfrac{\partial \psi(x,y)}{\partial y} &= 0 && y = -1
\end{aligned} \tag{2.101}$$

By using the method in this part, the $\psi(x,y)$ function concerning the boundary condition is shown in Figure(2.22).

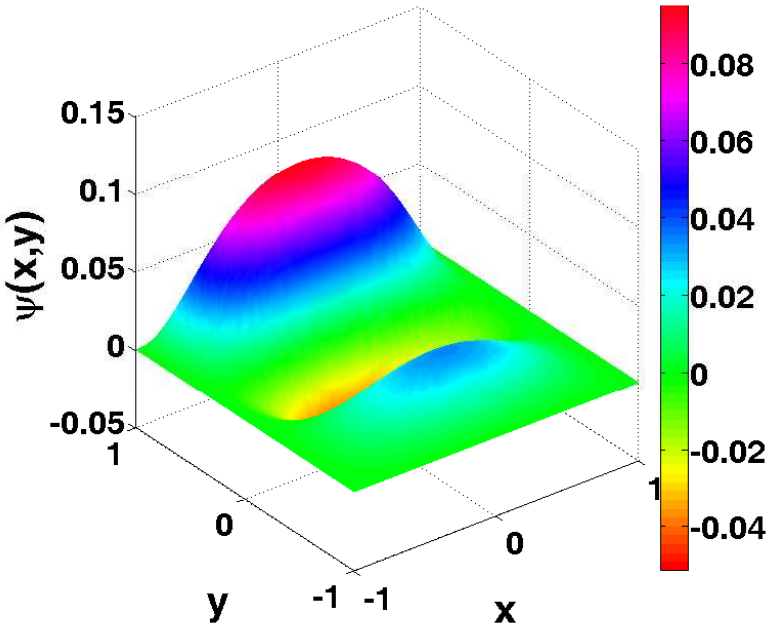


Figure 2.22: Souce term

This method can be used to express the boundary value for the flow problem, but it is very expensive, because it is solve a whole system for the flow problem.

2.3.4 PGD for the Non-Homogeneous BC Biharmonic Problem

In the subsection(2.3), we used the PGD-Chebyshev method proposed in this chapter for some examples with homogeneous BC, but did not address problems with non-homogeneous boundary conditions. in this subsection, we study the Non-Homogeneous boundary condition Biharmonic Problem.

2.3.4.1 Strategy 1

Here, we consider the Biharmonic problem Eq.(2.96) with the boundary condition Eq.(2.97).

Suppose the solution of the Biharmonic problem Eq.(2.96) with the boundary condition Eq.(2.97) can be write as:

$$u = \psi + u_1 \tag{2.102}$$

where function ψ in Ω satisfy the Eq.(2.97), then we only have a homogeneous BC problem need to be solved with the unknown u_1:

$$\Delta^2 u_1 = f - \Delta^2 \psi \ in \ \Omega \tag{2.103}$$

$$u_1 = \frac{\partial u_1}{\partial n} = 0 \ \ on \qquad \Gamma \in \partial\Omega \tag{2.104}$$

which is can be solve by chebyshev-PGD method.

Based on this idea, the Non-Homogeneous BC Biharmonic Problem can be changed into the homogeneous BC problem, but the boundary value problem need to be decided, that is why we study the BC problem in the last subsection. From the solution of last subsection, method 3 can be used for determined the boundary value problem of the biharmonic problem.

2.3.4.2 Strategy 2

There is another possibility for using PGD solving the Non-Homogeneous BC Biharmonic problem. For the 2D Biharmonic problem Eq.(2.96) with the boundary

condition by Eq.(2.97). By using penalization method, instead of solving the Eq.(2.96) with the boundary condition Eq.(2.97), we will solve a new equation:

$$\begin{aligned} &\int_\Omega u^*(\Delta^2 u - f) + \lambda \int_\Gamma u^*(\tfrac{\partial u}{\partial n} - f_2) = 0 \\ &u|_\Gamma = f_1 \end{aligned} \tag{2.105}$$

where λ is a penalization parameter, which is the weak from of the PGD method for solving the Non-Homogeneous boundary condition Biharmonic problem. Here, we do not detail the Mathematics formulation for using PGD method for this equation as it is the similar form in the subsection(2.2.1).

2.3.5 PGD method for the Lid driven cavity problem

Lid-driven cavity problem has been investigated by a lot of methods, such as [Montlaur et al., 2010, Montlaur et al., 2008, Bruneau and Saad, 2006, Gupta and Kalita, 2005], the lid of a square cavity is moving in the positive direction at a positive, constant velocity while the other walls of the cavity are at rest. Since the flow is viscous, the moving lid will cause a motion in the fluid. Stream function for 2D incompressible fluids can be used to transform the Navier-Stokes Equations into the fourth-order nonlinear partial differential equation with the biharmonic operator as a principal part, many author researched the stream function for the fluid [Montlaur et al., 2010, Kim et al., 2007, Dubois et al., 2003, Kalita and Gupta, 2009]; and in [Dumon et al., 2011] the PGD also has been used to simulate the lid-driven cavity problem.

This cavity flow possesses corner singularities which result in non-smoothness of the solution in some regions. In case of avoiding the singularity at the corner for the problem, the boundary condition is taken as

$$\begin{aligned} u(x,y) &= 0 && x = \pm 1 \\ u(x,y) &= 0 && y = \pm 1 \\ \tfrac{\partial u(x,y)}{\partial y} &= 0 && x = \pm 1 \\ \tfrac{\partial u(x,y)}{\partial x} &= x^2 - 1 && y = 1 \\ \tfrac{\partial u(x,y)}{\partial y} &= 0 && y = -1 \end{aligned} \tag{2.106}$$

in the square domain $]-1,1[\times]-1,1[$, where u is the stream function for the flow in the Lid-driven cavity problem.

2.3.5.1 PGD numerical solution by Strategy 1

Firstly, we used the first strategy for the problem, which need a function to describe the boundary value problem. By using the method 3 in the subsection(2.3.3) for the boundary value problem, the function is shown in the Figure(2.22).

By using the PGD-Checyshev method, we get the solution and the reference solution which is obtained from the 2D-Chebyshev method, the error between the two method is shown in Figure(2.26).

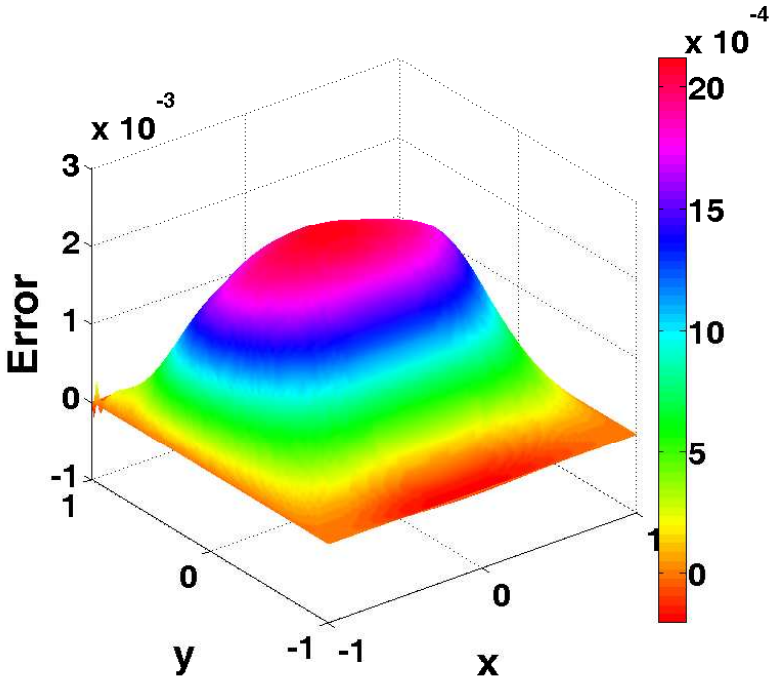
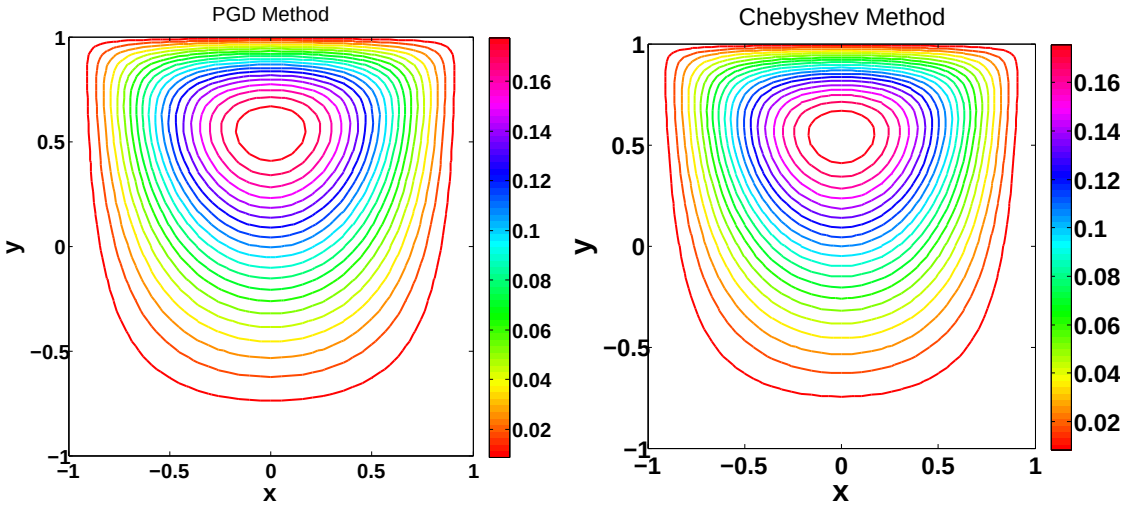


Figure 2.23: Error for the Lid driven cavity problem

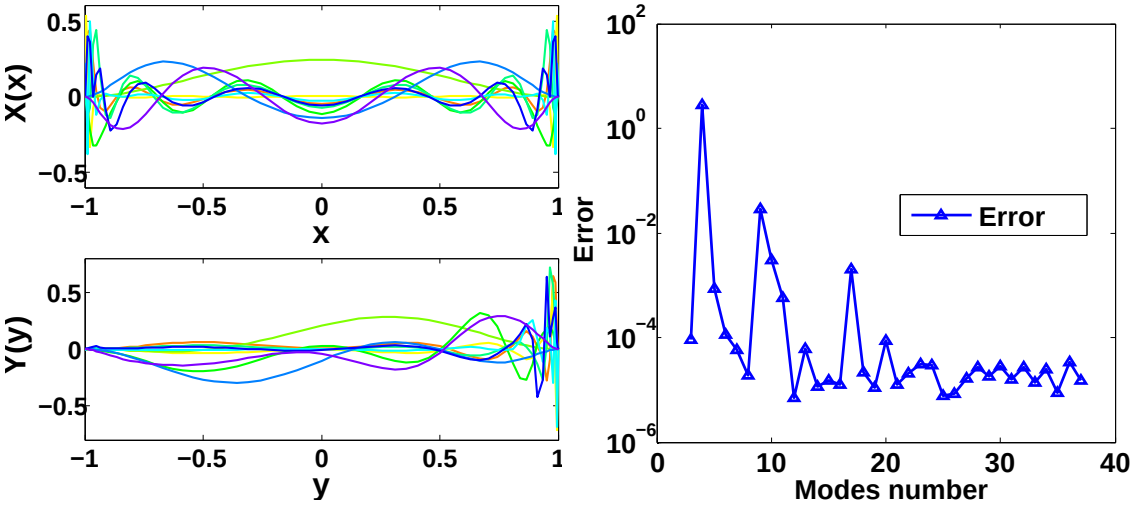
Contour plots for the two solutions are shown in Fig.(2.27(a)) (PGD-Chebyshev) and Fig.(2.27(b)) (2D-Chebyshev).



(a) Contour by PGD (b) Contour by Chebyshev

Figure 2.24: Contour by PGD and the Chebyshev method

The dominant PGD modes and the error as a function of the number of modes are shown in Fig.(2.25(a)) and Fig.(2.25(b)) respectively. The error is evaluated as $Error = \left\| U^n_{PGD} - U^{n-1}_{PGD} \right\|$. Besides the draw back of the CPU time is very high, which is not the drawback of PGD method, it's the disadvantage for finding the function for describing the non-homogeneous term; we can see that this method converge very slow.



(a) PGD modes (b) Error vs modes

Figure 2.25: PGD modes and PGD convergence

2.3.5.2 PGD numerical solution by Strategy 2

We used the second strategy for using the PGD method for solving the lid driven cavity problem.

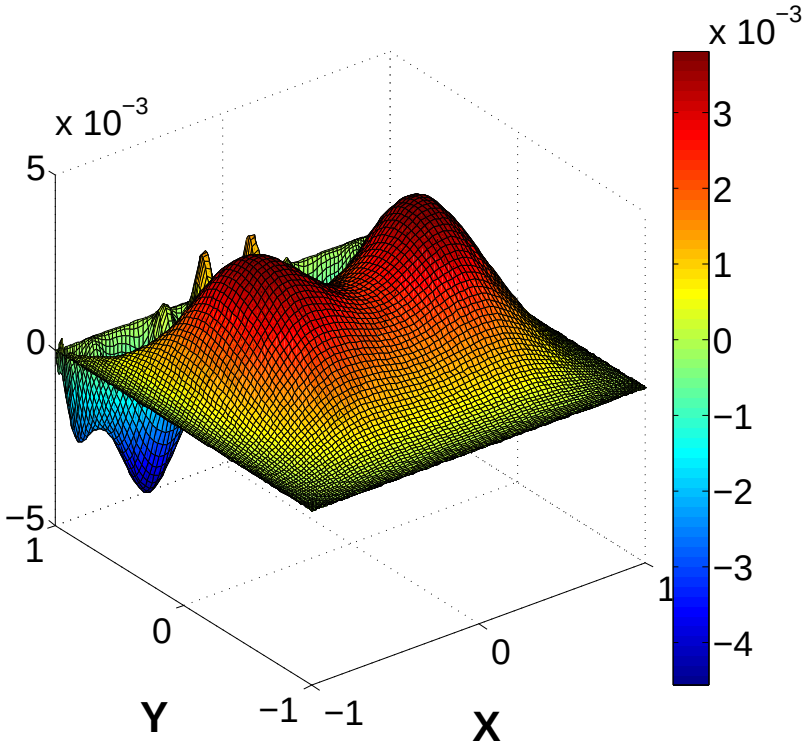
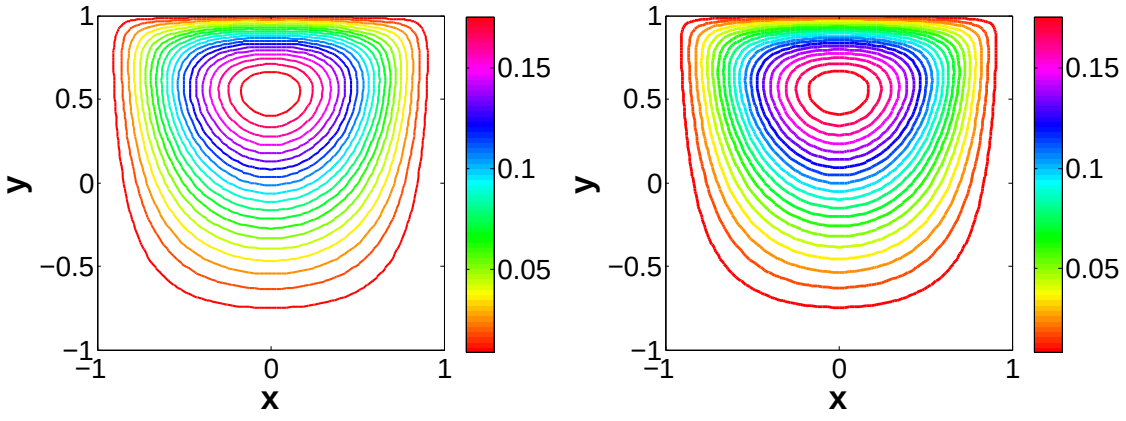


Figure 2.26: Error for the Lid driven cavity problem

By using the PGD method, contour plots for the PGD solution and the 2D chebyshev Method solution are shown in Fig.(2.27(a)) (PGD-Chebyshev) and Fig.(2.27(b)) (2D-Chebyshev).



(a) Contour by PGD

(b) Contour by Chebyshev

Figure 2.27: Contour by PGD and the Chebyshev method

The L^2 norm error as a function of the number of modes are shown in Fig.(2.28), where the L^2Error (as Eq.(2.62)) is used for evaluate the PGD convergence.

As we can see, the PGD solution is converged after 8 modes, even it is not as fast as the Simply-supported square-thin-plate problem, but it is fast enough.

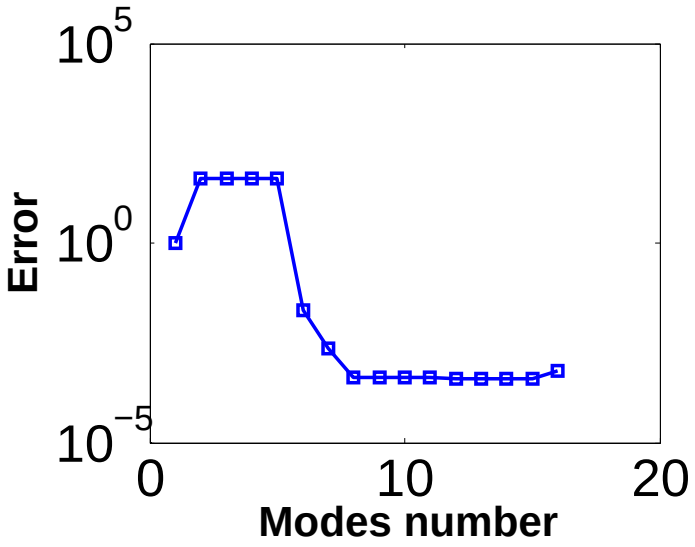


Figure 2.28: PGD convergence

2.4 Discussion and Conclusion

In this Chapter, we studied the PGD method coupled with and the Chebyshev method for the high order PDEs problem, first the Chebyshev method was used for the 2D harmonic equation and the 4-th order biharmonic equation, by applying the SVD to the 4th order PDE solution computed with the Chebyshev method we checked for the separability of the solution. A PGD formulation coupled to a Chebyshev pseudo-spectral collocation method was proposed for solving the harmonic problem and the biharmonic equation. In the cas of the biharmonic equation which is a high order partial differential equations; different strategies for dealing with the Non-homogeneous Boundary Condition were proposed. The resulting method was applied to the Lid-driven cavity problem in stream function formulation.

The results seem indicate that PGD and spectral techniques can be efficiently combined. In general, even if the computed code is not optimized from a computing time point of view, the use of the proposed technique should alleviate the storage needs, in comparison with the multidimensional (2D, 3D) Chebyshev spectral collocation strategies. Significant CPU time savings are expected, as were noticed. A difficult point concerns the enforcement of the boundary conditions. The Chebyshev-PGD can

be used to give an approximate representation for the simple boundary value problem of Biharmonic problem; that is, the creation of the first modes of the separated representation in order to account for all the boundary conditions known on the whole domain boundary.

Chapter 3

PGD for the Decomposition of Complex Geometry

3.1 Introduction 55
3.1.1 Coordinate transform for Complex Geometry 56
3.1.2 FD method in the (θ, r) variables 58
3.1.3 Examples for the different kinds of domain 59
3.2 PGD applied to a complex geometry problem 63
3.2.1 General weak form for the new formulation 63
3.3 Numerical examples 68
3.3.1 Manufactured problem 69
3.3.2 Circular Domain with Uniform Source Term 71
3.3.3 Circular Domain with Non-uniform Source Term 71
3.3.4 Ellipsoidal Domain 74
3.3.5 Star Domain 77
3.4 Discussion and Conclusion 79

3.1 Introduction

In order to take advantage of the separation of variables, a major difficulty lies in the separated representation of geometrically complex domains, possibly parametrized, other than cubes or hyper-cubes. A simple strategy based on the use of R-functions has been explored by [González et al., 2010]. In the Thesis of Ghnatios

[Chady Ghnatios, 2012] a parametric transformation of the domain of the following type is explored:

$$\begin{aligned} x &= r + \lambda_1 \cdot r \cdot s \\ y &= s + \lambda_2 \cdot r \cdot s \end{aligned} \tag{3.1}$$

Efficient developments for obtaining the solution for models with parametrized complex geometries are found in [Ghnatios et al., 2012, Ammar et al., 2014] . Most of the cases where PGD has been used are problems formulated in a cartesian coordinate system and little work has been done on polar and curvilinear coordinate systems.

It is the purpose of this Chapter to propose an efficient Proper Generalized Decomposition approach for computing the separated solution of 2D problems defined on a class of complex domains. Through the introduction of a curvilinear coordinate we transform the irregular simulation domain into a rectangular computational domain.

3.1.1 Coordinate transform for Complex Geometry

We consider a complex domain Ω such as the one depicted in Fig.(3.1). The domain Ω of $\mathbb{R}^2$ is such that it contains the origin and any segment between the origin and a point on its boundary belongs to Ω. In a polar coordinate system, the boundary is described by a function $R(\theta)$, for $\theta \in]0, 2\pi[$.

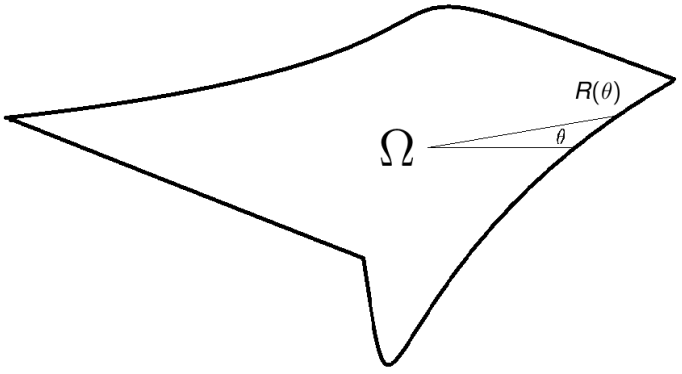


Figure 3.1: Complex Geometry domain

For the domain Ω, we introduce the system of parametric coordinates (r, θ) as follows:

$$\begin{aligned} x &= rR(\theta)cos(\theta) \\ y &= rR(\theta)sin(\theta) \end{aligned} \tag{3.2}$$

Consider a partial differential equation in the domain Ω, relative to the variables x and y, such that its coefficients are in separated form. Using the proposed change of variables (3.2), we can obtain a new formulation, which is still separable in the rectangular domain $(\theta, r) \in]1, 2\pi[\times]0, 1[$.

In order to simplify the notations, we define:

$$\begin{aligned} \mu(\theta) &= R(\theta)cos(\theta) \\ \nu(\theta) &= R(\theta)sin(\theta) \end{aligned} \tag{3.3}$$

To express derivatives such as $\frac{\partial u}{\partial x}$ using $\frac{\partial u}{\partial r}$ and $\frac{\partial u}{\partial \theta}$, we proceed as follows:

$$\begin{aligned} \frac{\partial u}{\partial r} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial r} &= \mu(\theta)\frac{\partial u}{\partial x} + \nu(\theta)\frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial \theta} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial r} &= r\mu^{'}(\theta)\frac{\partial u}{\partial x} + r\nu^{'}(\theta)\frac{\partial u}{\partial y} \end{aligned} \tag{3.4}$$

after noticing that $\mu\nu^{'} - \mu^{'}\nu = R^2$, we have,

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{\nu^{'}}{R^2}\frac{\partial u}{\partial r} - \frac{1}{r}\frac{\nu}{R^2}\frac{\partial u}{\partial \theta} \\ \frac{\partial u}{\partial y} &= -\frac{\mu^{'}}{R^2}\frac{\partial u}{\partial r} + \frac{1}{r}\frac{\mu}{R^2}\frac{\partial u}{\partial \theta} \end{aligned} \tag{3.5}$$

We notice that the coefficients of $\frac{\partial u}{\partial r}$ and $\frac{\partial u}{\partial \theta}$ have a separated form with respect to (θ, r).

Furthermore, from Eq.(3.5), we can calculate $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$, $\frac{\partial^2 u}{\partial x \partial y}$ and so on.

After some calculation, we can express the Laplace operator (in (x, y)) as:

$$\begin{aligned} \Delta u &= \frac{\partial^2 u}{\partial r^2}\left(\frac{1}{R^2} + \frac{(R^{'})^2}{R^4}\right) \\ &+ \frac{\partial u}{\partial r}\left(-\frac{R^{''}}{rR^3} + \frac{2(R^{'})^2}{rR^4} + \frac{1}{rR^2}\right) \\ &+ \frac{\partial^2 u}{\partial r \partial \theta}\left(\frac{-2R^{'}}{rR^3}\right) \\ &+ \frac{\partial^2 u}{\partial \theta^2}\left(\frac{1}{r^2R^2}\right) \end{aligned} \tag{3.6}$$

To simplify the equations, we define:

$$\begin{aligned} a(\theta) &= \frac{1}{R^2} + \frac{(R^{'})^2}{R^4} \\ b(\theta) &= -\frac{R^{''}}{R^3} + \frac{2(R^{'})^2}{R^4} + \frac{1}{R^2} \\ c(\theta) &= \frac{-2R^{'}}{R^3} \\ d(\theta) &= \frac{1}{R^2} \end{aligned} \tag{3.7}$$

Equation(3.6) then becomes:

$$\Delta u = a(\theta)\frac{\partial^2 u}{\partial r^2} + b(\theta)\frac{1}{r}\frac{\partial u}{\partial r} + c(\theta)\frac{1}{r}\frac{\partial^2 u}{\partial r \partial \theta} + d(\theta)\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = f(r,\theta) \tag{3.8}$$

When the domain Ω is the unit disk, i.e. $R(\theta) = 1$, Eq.(3.6) becomes:

$$\Delta u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} \tag{3.9}$$

which is the well known Laplace operator in polar coordinates.

It is of course possible to apply the same approach to more complex differential problems.

3.1.2 FD method in the (θ, r) variables

In order to solve the problem in the (θ, r) formulation we use second order Finite Difference method. In this section, we briefly detail the enforcement of boundary conditions. The boundary condition on a M by N FD grid are enforced as following:

- Periodic BC in the θ direction

  For the θ direction, we discretize $\frac{\partial u}{\partial \theta}$ and $\frac{\partial u^2}{\partial \theta^2}$ on the boundary nodes as:

$$\frac{\partial u^2_{i,1}}{\partial \theta^2} = \frac{u_{i,0} - 2u_{i,1} + u_{i,2}}{(\Delta\theta)^2} = \frac{u_{i,M} - 2u_{i,1} + u_{i,2}}{(\Delta\theta)^2} \tag{3.10}$$

$$\frac{\partial u^2_{i,M}}{\partial \theta^2} = \frac{u_{i,M-1} - 2u_{i,M} + u_{i,M+1}}{(\Delta\theta)^2} = \frac{u_{i,M-1} - 2u_{i,M} + u_{i,1}}{(\Delta\theta)^2} \tag{3.11}$$

  where $u_{i,1}$ is the point corresponding to $\theta = 0$ and $u_{i,M}$ is the point corresponding to $\theta = 2\pi$.

- BC in the r direction

In the r direction, for the outer points, homogeneous Dirichlet boundary condition are considered:

$$u(N, j) = 0 \tag{3.12}$$

For $r = 0$, we follow the idea presented in [Bruno-alfonso et al., 2012], which is to enforce,

$$\int_0^{2\pi} \frac{\partial u}{\partial r}(0^+, \theta) d\theta = 0 \tag{3.13}$$

3.1.3 Examples for the different kinds of domain

In this subsection, we present the solution of a model problem in different domains, solved either in (θ, r) using FD or in (x, y) using the FE method to illustrate the equivalence of the two formulations and the flexibility of the proposed change of variable.

The Poisson equation writes:

$$-\Delta u = 1 \tag{3.14}$$

with homogenous boundary condition

$$u = 0 \tag{3.15}$$

3.1.3.1 Unite Circle domain

The domain is characterized by the function $R(\theta)$ as:

$$R(\theta) = 1, \theta \in]0, 2\pi[\tag{3.16}$$

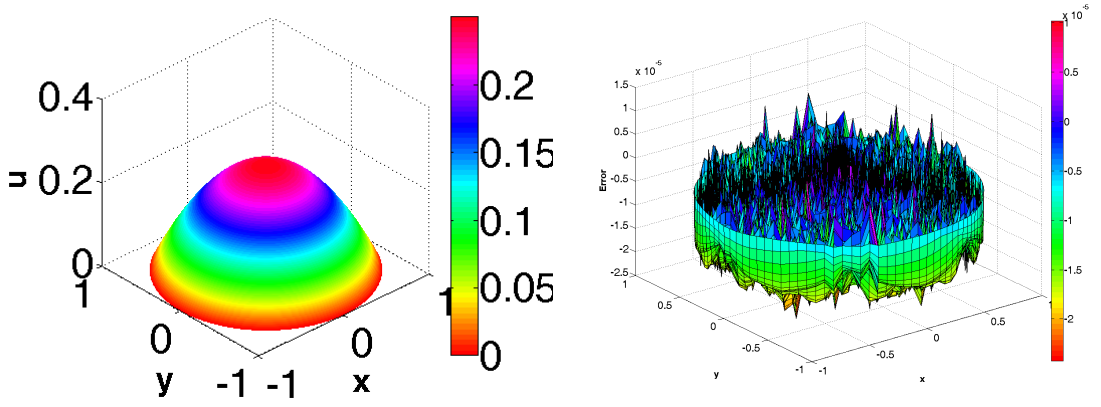
which is a unit circle domain.

The FD solution (discretization for (θ, r) is (1000×100)) and the difference with the FE solution (discretization degrees of freedom is 20281) are shown in Fig.(3.2). The order of the difference is 10^{-5}.

3.1.3.2 Slanted Ellipse domain

We change the geometry function $R(\theta)$ into

$$R(\theta) = \sqrt{\frac{4}{(2*cos(\theta+\frac{\pi}{4})).^2+(1*sin(\theta+\frac{\pi}{4})).^2}}, \theta \in]0, 2\pi[\tag{3.17}$$

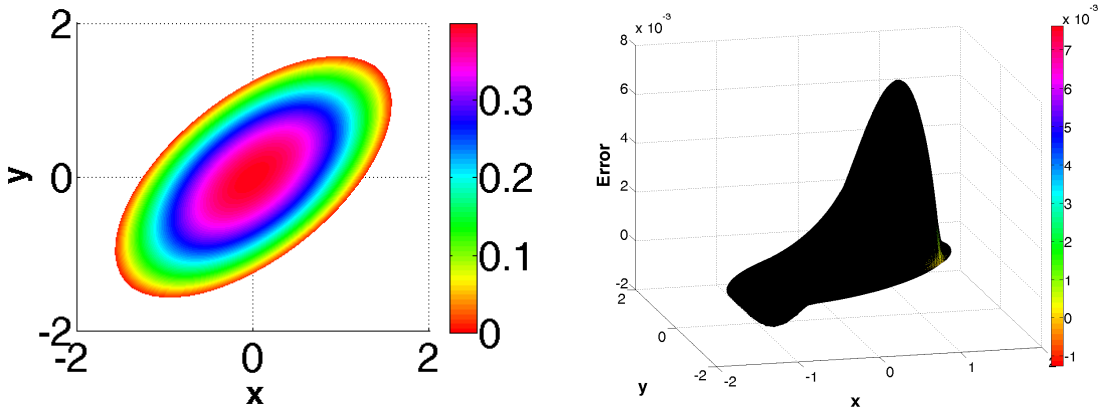


(a) FD solution (b) Difference with FE solution

Figure 3.2: Solution for unit Circle domain

which describes a slanted ellipse domain.

The FD solution (discretization for (θ, r) is (1000×100)) and the difference with the FE solution (discretization degrees of freedom is 20281) are shown in Fig.(3.3). The order of error is 10^{-3}.



(a) FD solution (b) Difference with FE solution

Figure 3.3: Solution for Slanted Ellipse domain

3.1.3.3 Star domain

Here, we consider the domain depicted in Fig.(3.4). The grid shown in Fig.(3.4) actually is the (θ, r) FD grid in the (x, y) domain. The function $R(\theta)$ is defined as:

$$R(\theta) = 0.6 + 0.25sin(5\theta), \theta \in]0, 2\pi[\tag{3.18}$$

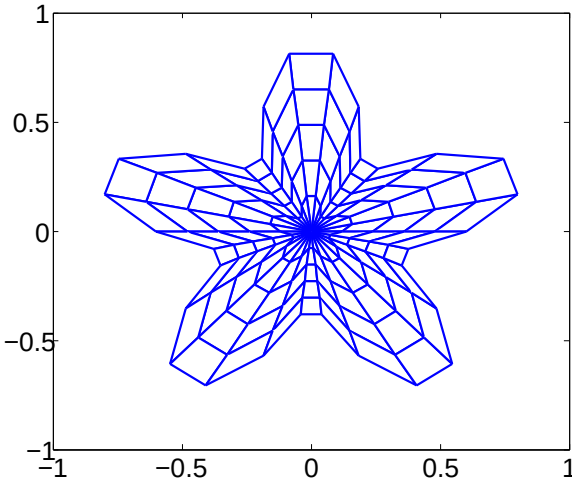


Figure 3.4: Mesh for the star domain

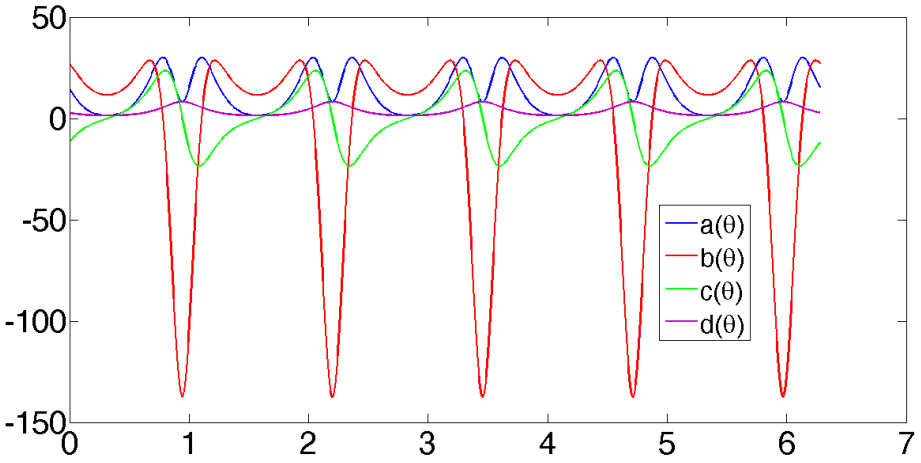


Figure 3.5: $a(\theta), b(\theta), c(\theta), d(\theta)$ for the star domain

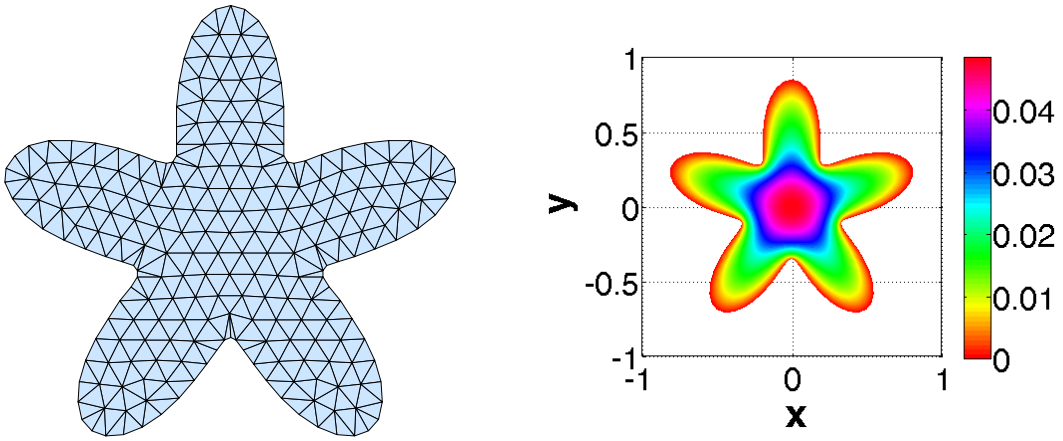
The functions $a(\theta), b(\theta), c(\theta), d(\theta)$ (cf. Eq(3.7)) are shown in Fig.(3.5).

The FE mesh and the FE solution (discretization degrees of freedom is 20281)of this problem is given in Fig.(3.6), while the FD solution (discretization for (θ, r) is (1000×100)) is given in Fig.(3.7).

3.1.3.4 Square domain

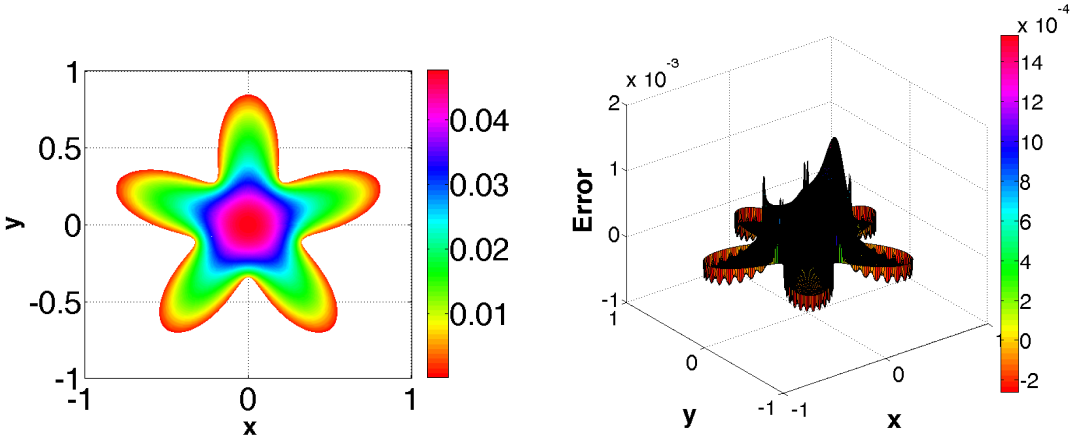
The proposed change of variable can even be applied to an already square domain.

The difference between the FD method (discretization for (θ, r) is (360×300)) in (θ, r), and the FE solution (discretization degrees of freedom is 1906) in (x, y) is shown in Fig.(3.8). The order of the difference is 10^{-3}.



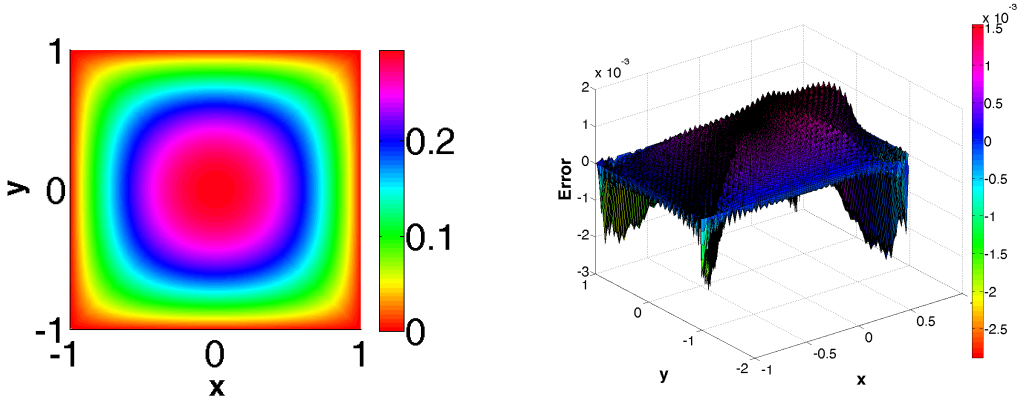
(a) FE mesh (b) FE solution $N = 56610$

Figure 3.6: FE mesh and FE solution



(a) FD(N_r110, N_θ500) (b) Difference between FD and FE solutions

Figure 3.7: FD solution and FD-FE comparison



(a) FD solution (b) Difference with FE solution

Figure 3.8: Solution for the square domain

3.2 PGD applied to a complex geometry problem

In the last section, we used a Finite Difference method for computing the solution in the new formulation in the rectangular computational domain. It is however obvious that for a very fine mesh, the computing time will rapidly increase. In the section, we therefore detail the PGD method for computing the solution of the new formulation in a separated way.

3.2.1 General weak form for the new formulation

Here, we consider the following problem:

$$-\Delta u(\theta, r) = f(\theta, r) \tag{3.19}$$

The weak form of the problem writes:

$$\begin{aligned} &\int_{\Omega_x}\int_{\Omega_y} u^*(x,y)(-\Delta u(x,y) - f(x,y))\ dxdy = \\ &\int_{\Omega_r}\int_{\Omega_\theta} u^*(\theta,r)(-\Delta u(\theta,r) - f(\theta,r))J\ d\theta dr = 0 \end{aligned} \tag{3.20}$$

where the Jacobian J originates from the change of variables (3.2), and is given by:

$$J(\theta, r) = \begin{Vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{Vmatrix} = rR^2(\theta) \tag{3.21}$$

An integral in the Cartesian Coordinate system is transformed into an integral in the Curvilinear Coordinate system as:

$$\int\int_A dxdy = \int\int_B rR^2(\theta) d\theta dr \tag{3.22}$$

So the weak form for the problem at hand becomes:

$$\int_{\Omega_r}\int_{\Omega_\theta} u^*(\theta,r)[-\Delta u(\theta,r) - f(\theta,r)]rR^2(\theta)\ d\theta dr = 0 \tag{3.23}$$

Introduced the expression for the Laplace operator(3.6) into Eq.(3.23) yields:

$$\begin{aligned} \int_{\Omega_r}\int_{\Omega_\theta} U^* \quad &\Big[-\tfrac{\partial^2 u}{\partial r^2} r(1 + \tfrac{(R')^2}{R^2}) - \tfrac{\partial u}{\partial r}(-\tfrac{R''}{R} + \tfrac{2(R')^2}{R^2} + 1) \\ &- \tfrac{\partial^2 u}{\partial r \partial \theta}(\tfrac{-2R'}{R}) - \tfrac{\partial^2 u}{\partial \theta^2}(\tfrac{1}{r}) - f(\theta, r)\Big] rR^2(\theta) d\theta dr = 0 \end{aligned} \tag{3.24}$$

After integration by parts,

$$\begin{aligned} &-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial^2 u}{\partial r^2} r(1+\frac{(R^{'})^2}{R^2}) + \frac{\partial u}{\partial r}(1+\frac{(R^{'})^2}{R^2})\right] d\theta dr = \\ &\int_{\Omega_r}\int_{\Omega_\theta} \quad (1+\frac{(R^{'})^2}{R^2})\frac{\partial U^*}{\partial r} r \frac{\partial u}{\partial r} d\theta dr \end{aligned} \tag{3.25}$$

$$\begin{aligned} &-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial u}{\partial r}(-\frac{(R^{''})}{R} + \frac{(R^{'})^2}{R^2}) + \frac{\partial^2 u}{\partial r \partial \theta}(\frac{-R^{'}}{R}))\right] d\theta dr = \\ &-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial u}{\partial r}\frac{\partial(-\frac{R^{'}}{R})}{\partial \theta} + \frac{\partial^2 u}{\partial r \partial \theta}(\frac{-R^{'}}{R}))\right] d\theta dr = \\ &-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial}{\partial \theta}(-\frac{\partial u}{\partial r}\frac{R^{'}}{R})\right] d\theta dr = \\ &\int_{\Omega_r}\int_{\Omega_\theta} \quad \left[\frac{\partial U^*}{\partial \theta}(-\frac{\partial u}{\partial r}\frac{R^{'}}{R})\right] d\theta dr \end{aligned} \tag{3.26}$$

$$-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial^2 u}{\partial r \partial \theta}(\frac{-R^{'}}{R})\right] d\theta dr = \int_{\Omega_r}\int_{\Omega_\theta} \quad \left[\frac{\partial U^*}{\partial r}\frac{-R^{'}}{R}\frac{\partial u}{\partial \theta}\right] d\theta dr \tag{3.27}$$

$$-\int_{\Omega_r}\int_{\Omega_\theta} \quad U^*\left[\frac{\partial^2 u}{\partial \theta^2}\frac{1}{r}\right] d\theta dr = \int_{\Omega_r}\int_{\Omega_\theta} \quad \left[\frac{\partial U^*}{\partial \theta}\frac{1}{r}\frac{\partial u}{\partial \theta}\right] d\theta dr \tag{3.28}$$

After some rearranging the terms, the weak form Eq.(3.24) becomes:

$$\begin{aligned} &\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\frac{(R^{'})^2}{R^2})\frac{\partial U^*}{\partial r} r \frac{\partial u}{\partial r}\right] d\theta dr + \int_{\Omega_r}\int_{\Omega_\theta}\left[\frac{\partial U^*}{\partial \theta}(-\frac{\partial u}{\partial r}\frac{R^{'}}{R})\right] d\theta dr \\ &\int_{\Omega_r}\int_{\Omega_\theta}\left[\frac{\partial U^*}{\partial r}\frac{-R^{'}}{R}\frac{\partial u}{\partial \theta}\right] d\theta dr + \int_{\Omega_r}\int_{\Omega_\theta}\left[\frac{\partial U^*}{\partial \theta}\frac{1}{r}\frac{\partial u}{\partial \theta}\right] dr d\theta = \\ &\int_{\Omega_r}\int_{\Omega_\theta}\left[U^* f(\theta, r) r R(\theta)\right] d\theta dr \end{aligned} \tag{3.29}$$

We can see from the Eq.(3.29), that the first term and the 4th term are intrinsically symmetric matrix, while the second and the third terms together also provide a symmetric contribution.

Suppose that we have the n-terms PGD solution u^n at the $n-th$ iteration as

$$u^n(\theta, r) = \sum_{i=1}^{N} R_i(r) \bullet \Theta_i(\theta) \tag{3.30}$$

The $n+1$-th solution is sought as:

$$u^{n+1}(\theta, r) = \sum_{i=1}^{N} R_i(r) \bullet \Theta_i(\theta) + R(r)\Theta(\theta) \tag{3.31}$$

The test function $u^*(\theta, r)$ is then assumed as

$$u^*(\theta, r) = \Theta^*(\theta) \cdot R(r) + \Theta(\theta) \cdot R^*(r) \tag{3.32}$$

After introducing the test function (3.32) and the $(n+1)^{-th}$ solution (3.31) into the weak form (3.29) we get:

$$\begin{aligned}
&\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\tfrac{(R^{'})^2}{R^2})\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)+\Theta(\theta)R^*(r)]r\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)+R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)+\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)+R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)+\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial\theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)+R(r)\Theta(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)+\Theta(\theta)R^*(r)]\tfrac{1}{r}\tfrac{\partial}{\partial\theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)+R(r)\Theta(\theta)]\right]d\theta dr\\
&=\int_{\Omega_r}\int_{\Omega_\theta}\left[[\Theta^*(\theta)R(r)+\Theta(\theta)R^*(r)]f(\theta,r)rR(\theta)\right]d\theta dr
\end{aligned} \tag{3.33}$$

- Computing $\Theta(\theta)$ from $R(r)$

  Since $R(r)$ is known from the previous iteration, the test function reads:

$$u(\theta, r) = \Theta^*(\theta)R(r) \tag{3.34}$$

  Eq.(3.33) reduces to:

$$\begin{aligned}
&\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\tfrac{(R^{'})^2}{R^2})\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)]r\tfrac{\partial}{\partial r}[R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial r}[R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial\theta}[R(r)\Theta(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)]\tfrac{1}{r}\tfrac{\partial}{\partial\theta}[R(r)\Theta(\theta)]\right]d\theta dr\\
&=\int_{\Omega_r}\int_{\Omega_\theta}\left[[\Theta^*(\theta)R(r)]f(\theta,r)rR^2(\theta)\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\tfrac{(R^{'})^2}{R^2})\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)]r\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial r}[\Theta^*(\theta)R(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial\theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial\theta}[\Theta^*(\theta)R(r)]\tfrac{1}{r}\tfrac{\partial}{\partial\theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr
\end{aligned} \tag{3.35}$$

As an efficient implementation requires a separated representation of the source term, we assume that:

$$f(\theta, r) = \sum_{i=1}^{i=NN} F_i(r).G_i(\theta) \tag{3.36}$$

Such decomposition that can be performed using the SVD algorithm.

$$\begin{aligned}
&\int_{\Omega_\theta} \left[(1 + \tfrac{(R^{'})^2}{R^2}) \Theta^*(\theta) \alpha_\theta \Theta(\theta) \right] d\theta - \int_{\Omega_\theta} \left[\tfrac{\partial \Theta^*(\theta)}{\partial \theta} \tfrac{R^{'}}{R} \beta_\theta \Theta(\theta) \right] d\theta \\
&- \int_{\Omega_\theta} \left[\Theta^*(\theta) \tfrac{R^{'}}{R} \beta_\theta \tfrac{\partial \Theta(\theta)}{\partial \theta} \right] d\theta + \int_{\Omega_\theta} \left[\tfrac{\partial \Theta^*(\theta)}{\partial \theta} \gamma_\theta \tfrac{\partial \Theta(\theta)}{\partial \theta} \right] d\theta \\
&= \int_{\Omega_\theta} \left[\Theta^*(\theta) \Re_i(\theta) R^2(\theta) \right] d\theta \\
&- \int_{\Omega_\theta} \left[(1 + \tfrac{(R^{'})^2}{R^2}) \Theta^*(\theta) [\textstyle\sum_{i=1}^{N} \alpha_{i\theta} \Theta_i(\theta)] \right] d\theta + \int_{\Omega_\theta} \left[\tfrac{\partial \Theta^*(\theta)}{\partial \theta} \tfrac{R^{'}}{R} [\textstyle\sum_{i=1}^{N} \beta_{i\theta} \Theta_i(\theta)] \right] d\theta \\
&+ \int_{\Omega_\theta} \left[\Theta^*(\theta) \tfrac{R^{'}}{R} [\textstyle\sum_{i=1}^{N} B_{i\theta} \tfrac{\partial \Theta_i(\theta)}{\partial \theta}] \right] d\theta - \int_{\Omega_\theta} \left[\tfrac{\partial \Theta^*(\theta)}{\partial \theta} [\textstyle\sum_{i=1}^{N} \gamma_{i\theta} \tfrac{\partial \Theta_i(\theta)}{\partial \theta}] \right] d\theta
\end{aligned} \tag{3.37}$$

The scalars, $\alpha_\theta, \beta_\theta$ are defined as:

$$\begin{aligned}
\alpha_\theta &= \textstyle\int_a^b \tfrac{\partial R(r)}{\partial r} r \tfrac{\partial R(r)}{\partial r} dr \\
\beta_\theta &= \textstyle\int_a^b R(r) \tfrac{\partial R(r)}{\partial r} dr \\
\gamma_\theta &= \textstyle\int_a^b \tfrac{1}{r} R^2(r) dr \\
\Re_i(\theta) &= \textstyle\int_a^b R(r) f(\theta, r) r dr \\
\alpha_{i\theta} &= \textstyle\int_a^b \tfrac{\partial R(r)}{\partial r} r \tfrac{\partial R_i(r)}{\partial r} dr \\
\beta_{i\theta} &= \textstyle\int_a^b R(r) \tfrac{\partial R_i(r)}{\partial r} dr \\
B_{i\theta} &= \textstyle\int_a^b \tfrac{\partial R(r)}{\partial r} R_i(r) dr \\
\gamma_{i\theta} &= \textstyle\int_a^b \tfrac{1}{r} R(r) R_i(r) dr
\end{aligned} \tag{3.38}$$

The above equations can be solved by using any standard technique, as for example a 1D finite element discretization with a periodic boundary condition.

- Computing $R(r)$ from $\Theta(\theta)$

  Assuming that the just computed function $\Theta(\theta)$ is known, the test function is written as:

$$u(\theta, r) = \Theta(\theta) R^*(r) \tag{3.39}$$

and the weak form becomes:

$$
\begin{aligned}
&\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\tfrac{(R^{'})^2}{R^2})\tfrac{\partial}{\partial r}[\Theta(\theta)R^*(r)]r\tfrac{\partial}{\partial r}[R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial \theta}[\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial r}[R(r)\Theta(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial r}[\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial \theta}[R(r)\Theta(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial \theta}[\Theta(\theta)R^*(r)]\tfrac{1}{r}\tfrac{\partial}{\partial \theta}[R(r)\Theta(\theta)]\right]d\theta dr\\
&=\int_{\Omega_r}\int_{\Omega_\theta}\left[[\Theta(\theta)R^*(r)]f(\theta,r)rR(\theta)\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[(1+\tfrac{(R^{'})^2}{R^2})\tfrac{\partial}{\partial r}[\Theta(\theta)R^*(r)]r\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial \theta}[\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&+\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial r}[\Theta(\theta)R^*(r)]\tfrac{R^{'}}{R}\tfrac{\partial}{\partial \theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr\\
&-\int_{\Omega_r}\int_{\Omega_\theta}\left[\tfrac{\partial}{\partial \theta}[\Theta(\theta)R^*(r)]\tfrac{1}{r}\tfrac{\partial}{\partial \theta}[\textstyle\sum_{i=1}^{N}R_i(r)\Theta_i(\theta)]\right]d\theta dr
\end{aligned}
\tag{3.40}
$$

this equation can be reduced as follow,

$$
\begin{aligned}
&\int_{\Omega_r}\left[\tfrac{\partial R^*(r)}{\partial r}r\alpha_r\tfrac{\partial R(r)}{\partial r}\right]dr-\int_{\Omega_r}\left[R^*(r)\beta_r\tfrac{\partial R(r)}{\partial r}\right]dr\\
&-\int_{\Omega_r}\left[\tfrac{\partial R^*(r)}{\partial r}\beta_r R(r)\right]dr+\int_{\Omega_r}\left[R^*(r)\tfrac{1}{r}\gamma_r R(r)\right]dr\\
&=\int_{\Omega_r}\left[R^*(r)r\Re_i r\right]dr\\
&-\int_{\Omega_r}\left[\tfrac{\partial R^*(r)}{\partial r}r[\textstyle\sum_{i=1}^{N}\tfrac{\partial R_i(r)}{\partial r}\alpha_{ir}]\right]dr+\int_{\Omega_r}\left[R^*(r)[\textstyle\sum_{i=1}^{N}\tfrac{\partial R_i(r)}{\partial r}\beta_{ir}]\right]dr\\
&+\int_{\Omega_r}\left[\tfrac{\partial R^*(r)}{\partial r}[\textstyle\sum_{i=1}^{N}R_i(r)B_{ir}]\right]dr-\int_{\Omega_r}\left[R^*(r)\tfrac{1}{r}[\textstyle\sum_{i=1}^{N}R_i(r)\gamma_{ir}]\right]dr
\end{aligned}
\tag{3.41}
$$

where:

$$\begin{aligned}
\alpha_r &= \int_0^{2\pi}(1+\tfrac{(R^{'})^2}{R^2})\Theta(\theta)\Theta(\theta)d\theta \\
\beta_r &= \int_0^{2\pi}\tfrac{\partial\Theta(\theta)}{\partial\theta}\tfrac{R^{'}}{R}\Theta(\theta)d\theta \\
\gamma_r &= \int_0^{2\pi}\tfrac{\partial\Theta(\theta)}{\partial\theta}\tfrac{\partial\Theta(\theta)}{\partial\theta}d\theta \\
\Re_i(r) &= \int_0^{2\pi}\Theta(\theta)f(\theta,r)R^2(\theta)d\theta \\
\alpha_{ir} &= \int_0^{2\pi}(1+\tfrac{(R^{'})^2}{R^2})\Theta(\theta)\Theta_i(\theta)d\theta \\
\beta_{ir} &= \int_0^{2\pi}\tfrac{\partial\Theta(\theta)}{\partial\theta}\tfrac{R^{'}}{R}\Theta_i(\theta)d\theta \\
B_{ir} &= \int_0^{2\pi}\tfrac{\partial\Theta_i(\theta)}{\partial\theta}\tfrac{R^{'}}{R}\Theta(\theta)d\theta \\
\gamma_{ir} &= \int_0^{2\pi}\tfrac{\partial\Theta(\theta)}{\partial\theta}\tfrac{\partial\Theta_i(\theta)}{\partial\theta}d\theta
\end{aligned} \tag{3.42}$$

This is also a 1D problem which can be solved by using any standard technique, as for example the 1D Finite Element method used in this work.

These two steps are repeated until convergence to a fixed point. If we denote the function $R(r)$ at the present and previous iteration as $R^p(x)$ and $R^{p-1}(r)$, respectively, and the same for the function $S(\theta)$, $S^p(\theta)$ and $S^{p-1}(\theta)$, the convergence criterion can be defined as:

$$e = \int_{\Omega_r\times\Omega_\theta}(R^p(r)\cdot S^p(\theta) - R^{p-1}(r)\cdot S^{p-1}(\theta))^2 Jdr\cdot d\theta \leq \varepsilon \tag{3.43}$$

where ε is a small enough parameter.

We can then define the next functional couple: $R_{n+1} = R$ and $\Theta_{n+1} = S$.

3.3 Numerical examples

In this section we present some results obtained by applying the above method to different domains. In order to avoid any singularity for $r = 0$, we extended the proposed change of variables to account for a small hole in the center of the domain. In the following examples, the computational domain in (θ, r) reduces to $(\theta, r) \in]0, 2\pi[\times]0.1, 1[$. For all the examples the following boundary conditions are considered:

$$\begin{aligned} U_{(r=0.1,\theta)} &= 0 \\ U_{(r=1,\theta)} &= 0 \\ U_{(r,\theta=0)} &= U_{(r,\theta=2\pi)} \end{aligned} \tag{3.44}$$

The error of the PGD solution is defined with respect to the FE solution on the same mesh (FEM solution in (θ, r)):

$$Error = U_{FEM} - U \tag{3.45}$$

The convergence of the PGD solution to the FE solution is assessed using the following indicator

$$E^n = \frac{\int_\Omega (U_{FEM} - U^n_{PGD}) d\Omega}{\int_\Omega U_{FEM} d\Omega} \tag{3.46}$$

where U^n is the PGD solution truncated to the first n enrichments.

3.3.1 Manufactured problem

First, we consider the following manufactured problem:

$$-\Delta u = f \tag{3.47}$$

on a disk domain for $0.1 \leq \sqrt{x^2 + y^2} \leq 1$. In polar coordinates this equation becomes:

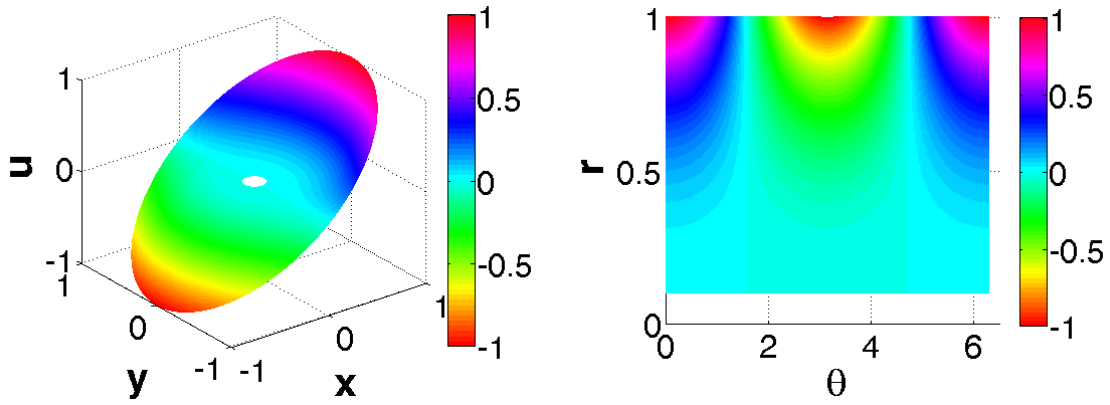
$$-\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right) = f \tag{3.48}$$

The source term f is chosen such that exact solution of this problem is:

$$u = r^3 cos(\theta) \tag{3.49}$$

This problem was solved by the PGD method (discretization in (θ, r) as (201×201)) and the solution is shown in Fig.(3.9) for the two coordinate systems. The error between the PGD solution and the analytical solution is shown in Fig.(3.10(a)) in the original (x, y) domain and in Fig.(3.10(b)) in the (θ, r) computational domain.

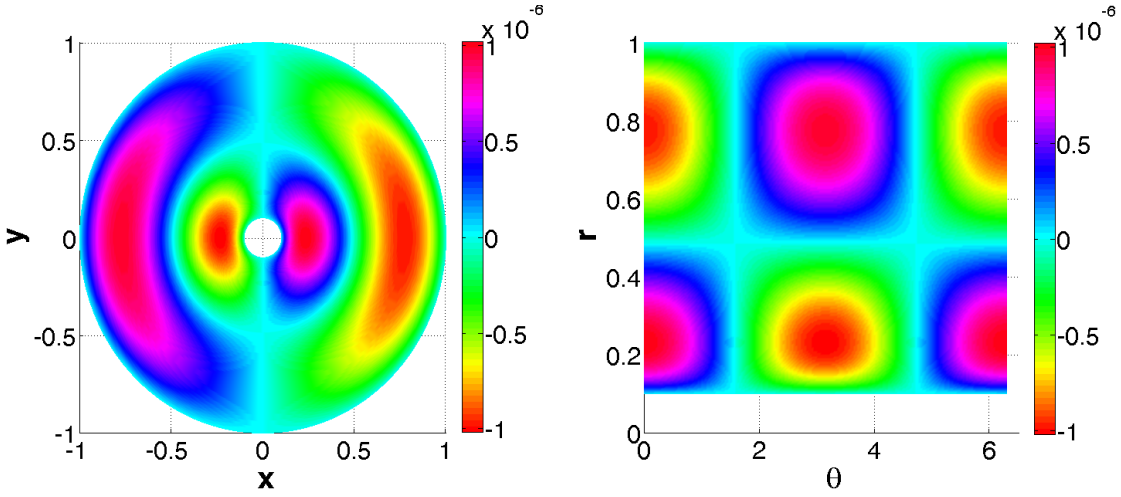
The PGD converges in one enrichment steps to the FEM solution. This is illustrated in Fig.(3.11). This could be expected since the solution (6.62) of this problem has a one-term separated representation.



(a) (x, y) Cartesian Coordinate

(b) (θ, r) Curvilinear Coordinates

Figure 3.9: Solution for the manufactured problem



(a) Error in (x, y)

(b) Error in (θ, r)

Figure 3.10: Error for the manufactured problem

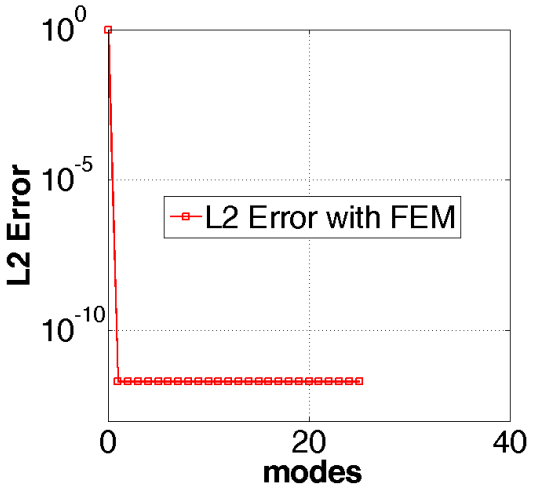


Figure 3.11: PGD convergence

3.3.2 Circular Domain with Uniform Source Term

We consider then the following problem:

$$-\Delta u = f \tag{3.50}$$

in the Disk domain defined by $0.1 \leq \sqrt{x^2+y^2} \leq 1$, and for a source term $f = 1$. The boundary condition is

$$u = 0 \quad \text{for } \sqrt{x^2+y^2} = 0.1 \text{ and } \sqrt{x^2+y^2} = 1 \tag{3.51}$$

The PGD solution (discretization in (θ, r) as (201×201)) is shown in Fig.(3.12). The error between PGD solution and FE solution, given in Fig.(3.13), is very small, almost the machine zero. As shown in Fig.(3.14), one mode only is needed to converge the solution since it is just a function of r.

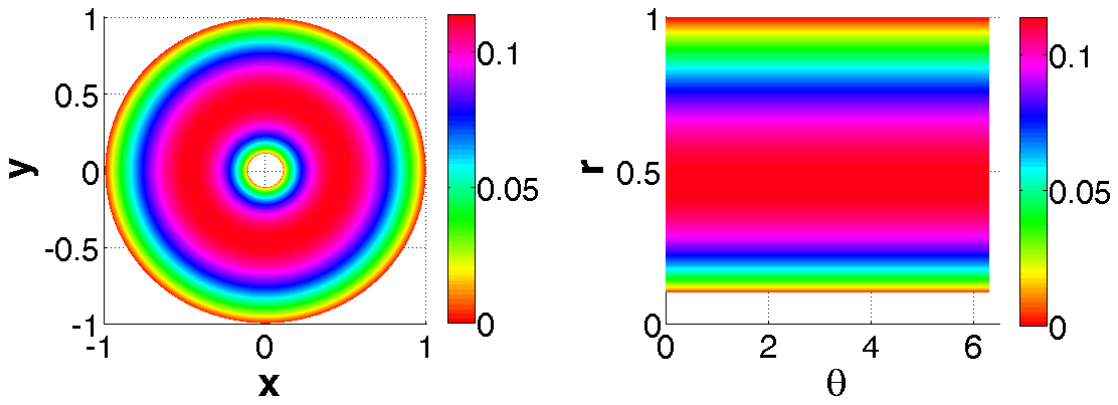
3.3.3 Circular Domain with Non-uniform Source Term

We now test the method on a problem where we expect a richer separated solution:

$$-\Delta u = f \tag{3.52}$$

in a Disk domain defined by $0.1 \leq \sqrt{x^2+y^2} \leq 1$, and the source term f is given as:

$$f = \begin{cases} 40 & \text{for } -0.7 \leq x \leq -0.3 \text{ and } -0.2 \leq y \leq 0.2 \\ 2 & \text{otherwise} \end{cases} \tag{3.53}$$



(a) Solution in (x, y) (b) Solution in (θ, r)

Figure 3.12: PGD solution for the circular domain

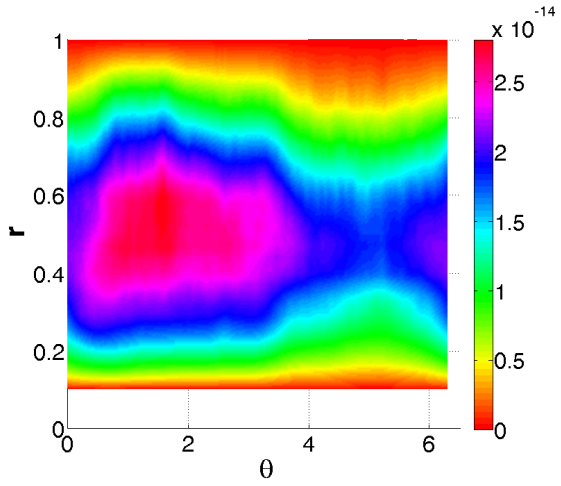


Figure 3.13: PGD error for the circular domain

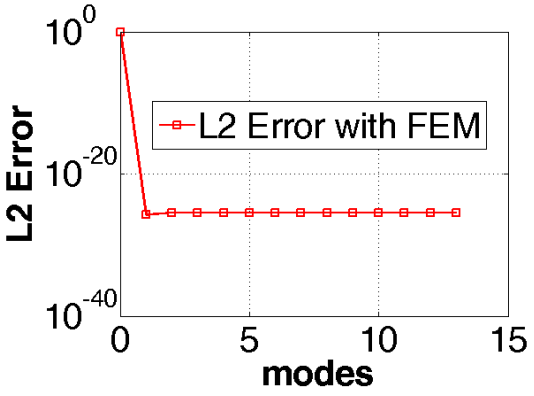


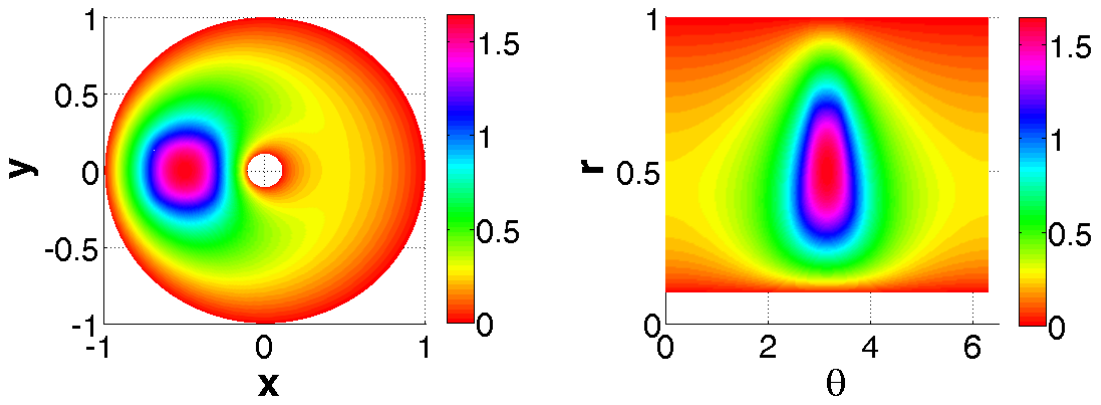
Figure 3.14: PGD convergence for the circular domain

The boundary condition is

$$u = 0 \quad \text{for } \sqrt{x^2 + y^2} = 0.1 \text{ and } \sqrt{x^2 + y^2} = 1 \tag{3.54}$$

This problem was solved by PGD (discretization in (θ, r) as (201×201)) in (θ, r) and the solution is given in Fig.(3.15). The error between the PGD solution and FE solution is given in Fig.(3.16). The PGD convergence is shown in Fig.(3.17(a)).

The mesh dependency of the difference between the converged PGD solution and the FE solution is shown in Fig.(3.17(b)).



(a) Solution in (x, y) (b) Solution in (θ, r)

Figure 3.15: Solution for the problem with Non-uniform source

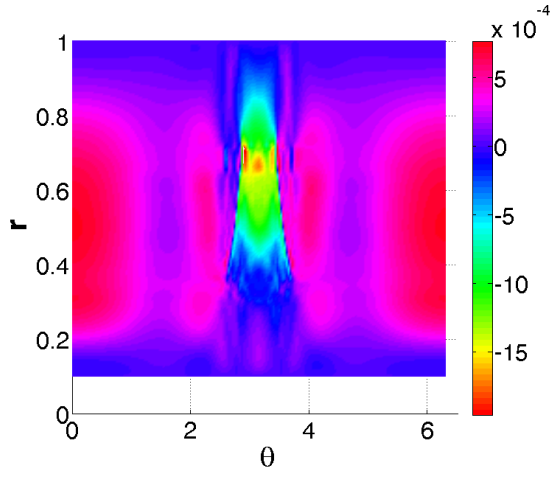
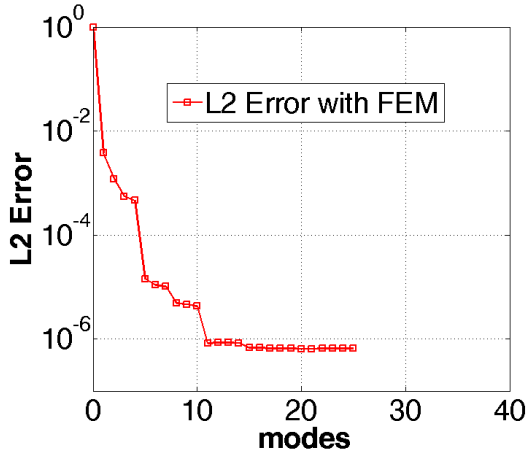
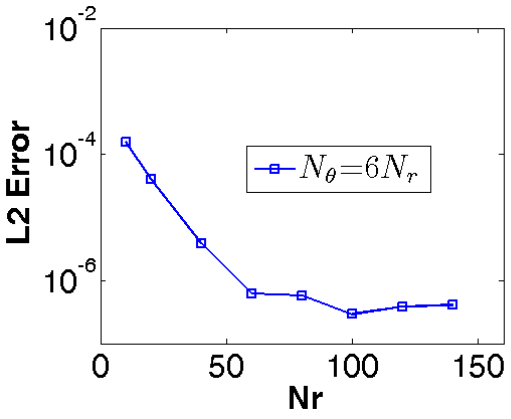


Figure 3.16: PGD error for the problem with Non-uniform Source



(a) PGD error as a function of the number of modes



(b) Error of the converged PGD solution)

Figure 3.17: PGD error as a function of the number of modes and Error of the converged PGD solution (the discretization for (θ, r) have the relation $N_{theta} = 6N_r$) for the the problem with Non-uniform Source

3.3.4 Ellipsoidal Domain

We now investigate more complex domains by considering more sophisticated functions $R(\theta)$. The PDE remains the same:

$$-\Delta u = f \tag{3.55}$$

but we consider an ellipsoidal domain which is shown in Fig.(3.18), the boundary of the domain Γ_1 and Γ_1 is defined by

$$R(\theta) = \sqrt{\frac{4}{4cos^2(\theta + \frac{\pi}{4}) + sin^2(\theta + \frac{\pi}{4})}}, \theta \in]0, 2\pi[, \tag{3.56}$$

and the source term f is given as follow,

$$f = \begin{cases} 40 & \text{for } -0.5 \leq x \leq -0.3 \text{ and } -0.2 \leq y \leq 0.2 \\ 2 & \text{for the other domain} \end{cases} \tag{3.57}$$

The boundary condition is still homogeneous:

$$u = 0 \quad \text{for } \Gamma_1 \text{ and } \Gamma_2 \tag{3.58}$$

The PGD solution (discretization in (θ, r) as (201×201)) in cartesian and

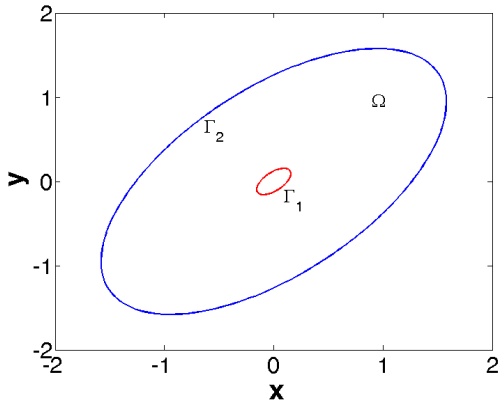
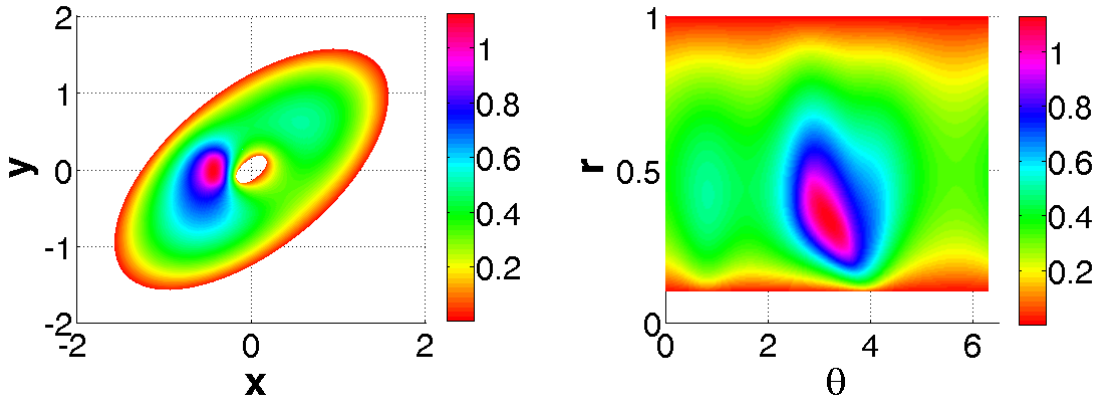


Figure 3.18: Geometry for the Ellipsoidal domain

curvilinear coordinates is given in Fig.(3.19). The difference between 30-modes PGD solution and FE solution is shown in Fig.(3.20).

The PGD convergence is shown in Fig.(3.21(a)). As expected for this kind of problems, we observe a monotonic convergence of the PGD. The error as a function of the discretization is shown in Fig.(3.21(b)). This shows that the PGD solution is actually closer to the FEM solution for finer meshes. The comparison of the CPU time for the homemade FEM and PGD methods is shown in Fig.(3.22). From this figure, we clearly see that the PGD method is much faster than FEM for finer and finer meshes. For the finer meshes the PGD can be several order of magnitudes faster.



(a) Solution in (x, y)

(b) Solution in (θ, r)

Figure 3.19: Solution for the Ellipsoidal domain

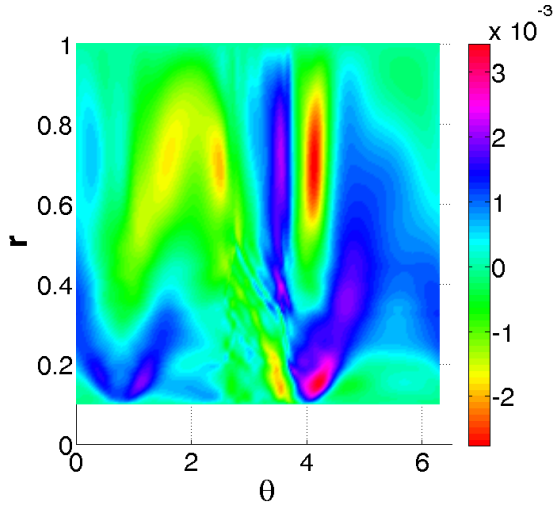
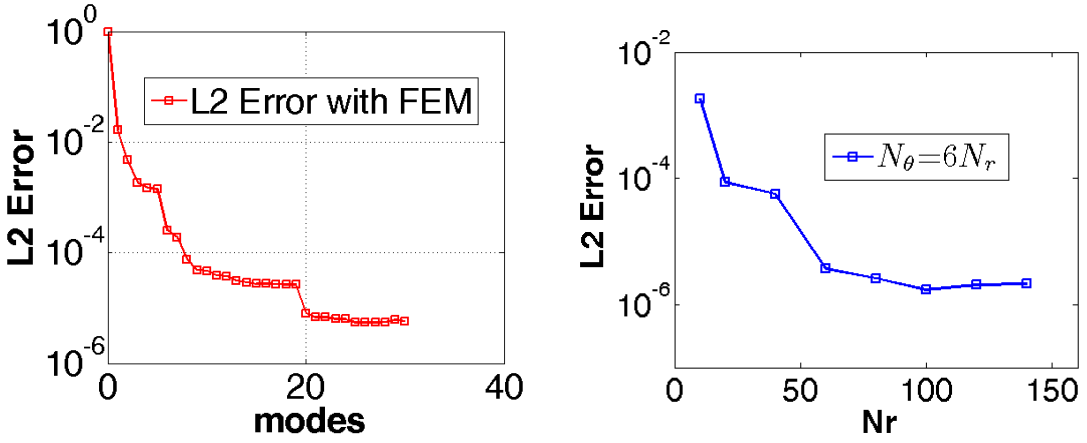


Figure 3.20: PGD error for the Ellipsoidal domain



(a) PGD Convergence

(b) Error for different meshes

Figure 3.21: PGD Convergence and Error of the converged PGD for different meshes for the Ellipsoidal domain

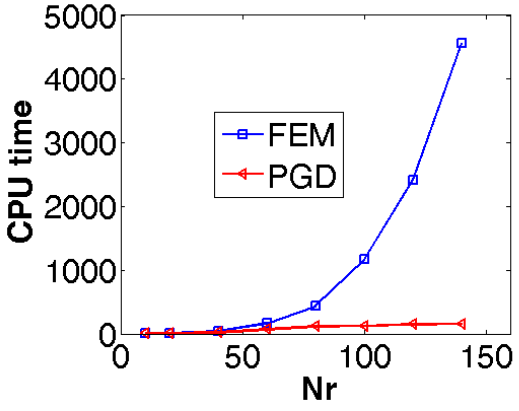


Figure 3.22: Comparison of CPU time between PGD and FEM

3.3.5 Star Domain

We finally show that the proposed change of variable is capable of treating quite complex domains, as illustrated below. We still consider the same PDE:

$$-\Delta u = f \tag{3.59}$$

in a star-like domain which is shown in Fig.(3.23), the domain boundary Γ_1 and Γ_2 is defined by

$$R(\theta) = 0.6 + 0.25sin(5\theta), \theta \in]0, 2\pi[\tag{3.60}$$

and the source term f is defined as:

$$f = \begin{cases} 40 & \text{for } -0.3 \leq x \leq -0.15 \text{ and } -0.2 \leq y \leq 0.2 \\ 2 & \text{otherwise} \end{cases} \tag{3.61}$$

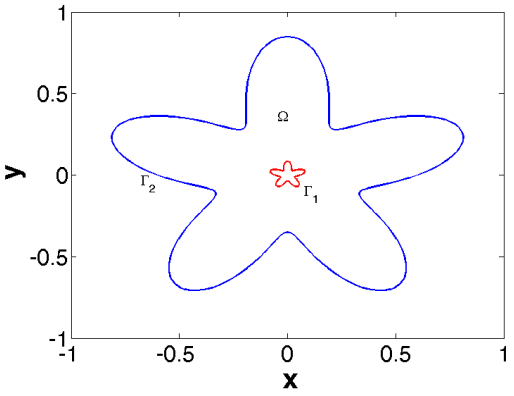


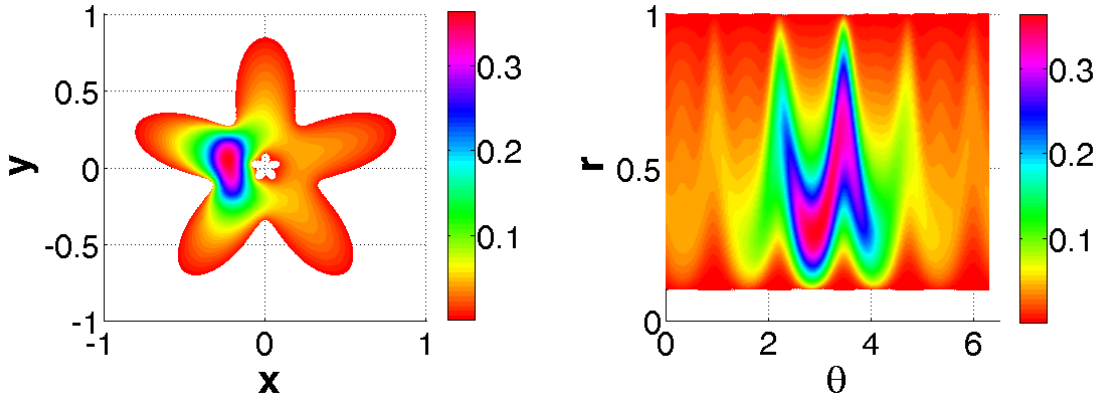
Figure 3.23: Geometry for the Star Domain

The boundary condition is again homogeneous:

$$u = 0 \quad \text{for } \Gamma_1 \text{ and } \Gamma_2 \tag{3.62}$$

The PGD solution (discretization in (θ, r) as (201×201)) in (θ, r) and in cartesian coordinates is given in the Fig.(3.24). The error between PGD solution and FE solution is given in Fig.(3.25). The PGD convergence is shown in Fig.(3.26(a)). We observe that although the solution is quite smooth in cartesian coordinates the change of variables makes it difficult to approximate using the separated representation of the PGD, hence the increased number of modes in the PGD solution and the slower convergence.

In Fig.(3.26(b)), we show again that the PGD converges closer to the FE solution for finer meshes. Although this problem is more of a challenge for the PGD we still observe in Fig.(3.27) that the PGD is much faster than traditional 2D Finite Element. For this complex Geometry problem, more fixed point iterations were also required for each mode to converge during the construction of the PGD solution.



(a) Solution in xy

(b) Solution in (θ, r)

Figure 3.24: Solution for the Star Domain

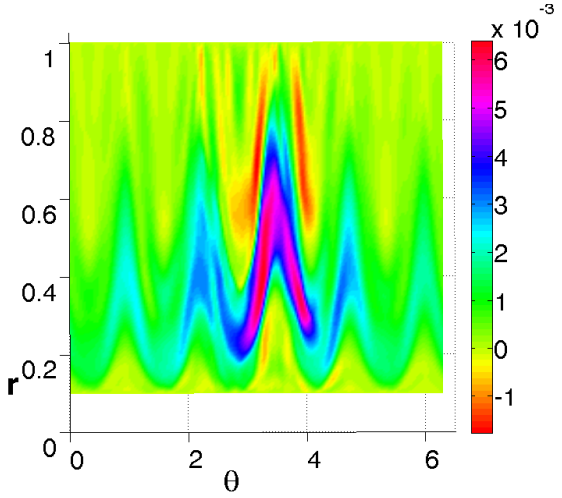
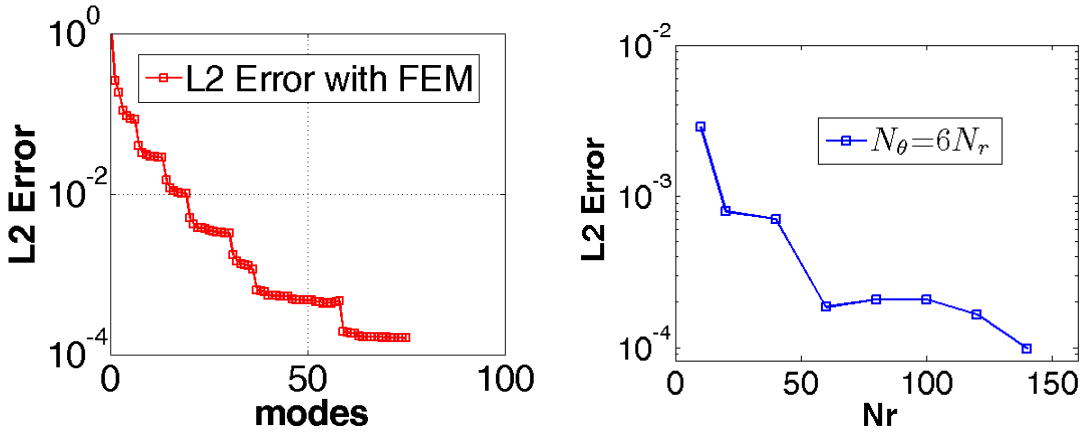


Figure 3.25: PGD error for the Star Domain compare with the FEM solution



(a) PGD convergence

(b) PGD-FEM error for different meshes

Figure 3.26: PGD convergence and PGD-FEM error for different meshes for the star domain

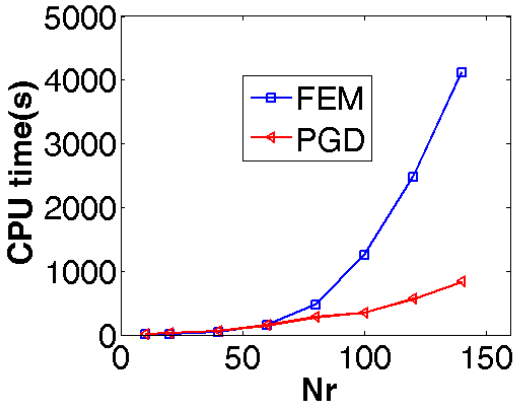


Figure 3.27: FEM-PGD CPU time comparison for the star-like domain.

3.4 Discussion and Conclusion

Through several examples, we have shown that the method proposed in this chapter for applying the PGD on problems defined in complex geometries is efficient and accurate with respect to classical Finite Element method. The only difficulty lies in the proper definition of the change of variable. The separation of variables which lies at the heart of the PGD allows for significant savings in computing time even when the geometry is complex and the solution is difficult to represent in separated form.

Part II

PGD for Resolving the Unsteady Navier-Stokes Equations

Chapter 4

Resolving the Unsteady Navier-Stokes Equations

4.1 Fluid flow and Navier-Stokes Equations 83
4.2 Numerical Method for the Navier-Stokes Equation 85
4.3 Model Reduction Method 88
4.4 Scope and Outline of Part II 89

4.1 Fluid flow and Navier-Stokes Equations

There is a lot of fluid flow problems [White et al., 1987] in the world. In physics, fluid dynamics is a sub-discipline of fluid mechanics that deals with fluid flows. This is the natural science of fluids (liquids and gases) in motion. It has several sub-disciplines itself, including aerodynamics (the study of air and other gases in motion) and hydrodynamics (the study of liquids in motion). Fluid mechanics is the study of fluids either in motion (fluid dynamics) or at rest (fluid statics) and the subsequent effects of the fluid upon the boundaries, which may be either solid surfaces or interfaces with other fluids [White, 2009].

Fluid structure interaction mechanics which is called Fluid structure interaction (FSI) is the coupling of the fluid and structural mechanics. This problem is very complex but has a variety of applications. Fluid structure interaction problems involve the coupling of unsteady fluid flow and structure motion which is of great relevance in the engineering field and the applied science (see examples as: [Whitesides, 2006,

Bazilevs et al., 2006, Squires and Quake, 2005, Coullet et al., 2005]). It is frequently encountered in many areas of civil, mechanical, aerospace and bio-mechanical engineering, such as ship hydrodynamics, off-shore structures, spill-ways in dames, free surface channel flows, liquid containers, stirring reactors, mold filling processes, the analysis of aneurysms in large arteries and artificial heart valves, etc...

Fluid-structure interaction problems involving large motions of the free surface and splashing of waves can also be modelled by the particle finite element method (PFEM) in [Oñate et al., 2006].

Figure 4.1: Flow around submarine

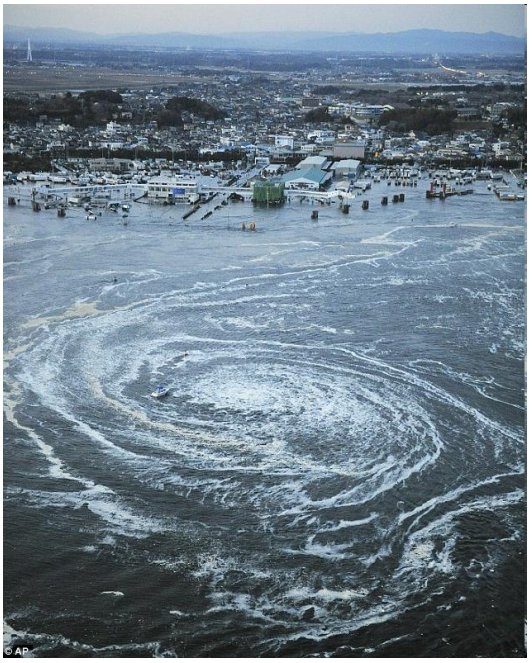


Figure 4.2: Flow generated by an earthquake

Figures (4.1) and (4.2) provide examples of flow around a submarine and a flow created by the Japanese earthquake which happened in the year 2011. Tsunamis are giant waves caused by earthquakes or volcanic eruptions under the ocean. When tsunami waves travel inland, they get larger as the depth of the ocean decreases. If you know the fluid dynamics governing this phenomenon, you can predict how the speed of tsunami waves varies with ocean depth rather than the distance from the earthquake, a knowledge which will reduce the risks of massive destruction.

Another example of our daily life is the flow from a faucet. On the left, there is a turbulent flow from a faucet, and on the right the flow is laminar or slightly transitional (see Figure(4.3)).

All of these flows of fluid can be described by a set of non-linear partial differential

Figure 4.3: The flow from a faucet

equations which is called Navier-Stokes equations.

These problems are characterized by a dynamic and non-linear behavior both in the structure and the fluid mechanics responses. Predicting this behavior is a significant challenge from the numerical simulation [Kirby et al., 2007]. In the past decades, great attention has been paid to the development and application of numerical simulation. Traditional approaches to analysis of such problems, although proving extremely useful tools in engineering practice, necessarily rely on simplified models, which have a narrow range of validity and applications. Because of the development of computational resources, numerical modeling of the problem became possible. However, there is still a lot of work to do in the field of specific numerical strategies for the discretisation of the fluid, solid and time domains and for the modeling of the fluid-structure interface. According to the fluid model, the flow problem can be classified into incompressible, slightly compressible and compressible flow.

4.2 Numerical Method for the Navier-Stokes Equation

The major difficulties in solving the incompressible Navier-Stokes equations lie in the non-linearities and in the determination of the pressure. Traditionnally, pressure is determined through an elliptic pressure equation derived from a combination of the partial differential equations expressing conservation of mass and momentum.

The mathematical models lead to a set of partial differential equations (PDEs). Then in a specific flow, this set of PDEs with appropriate initial conditions and

boundary conditions can be solved numerically. The most common mathematical models are Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes (RANS).

In DNS, the Navier-Stokes equations are directly solved to obtain U(xi, t). In this method all the time and length scales would be solved. That is why this method is computationally very expensive. The computational cost is proportional to Re^3. It will cause limitation of this method to low and moderate Reynolds number flows.

In LES, the filtered equations are solved to obtain filtered velocity field U(xi, t). This filtered velocity field reveals only the larger scale turbulent motions. The effect of smaller scale motions are indirectly included in the set of equations.

In RANS model, the set of Reynolds-Averaged Navier-Stokes equations has to be solved to determine mean velocity field and specific turbulence models relying on algebraic formulation or additional partial differential equations, are used to determine the Reynolds stresses.

There are of course many kinds of methods to solve the Navier-Stokes Equations, such as Finite Difference method, Finite Element method, Boundary Element method, Finite Volume method, Pressure method, SPH, Spectral Method and so on.

Finite difference method was used to solve the fluid and solid motions on a fixed grid, and the numerical accuracy involved in the fluid-structure coupling problems was examined [Sugiyama et al., 2011]. [Gupta and Kalita, 2005] proposed a new paradigm for solving the steady-state two-dimensional (2D) Navier-Stokes (NS) equations using a stream function-velocity ($\Psi - v$) formulation. This idea was extended and a second-order implicit, unconditionally stable ($\Psi - v$) formulation was proposed for the unsteady incompressible NS equations in [Kalita and Gupta, 2009].

Driven cavity flow problem was studied by Pseudo-divergence-free element free Galerkin method for incompressible fluid flow in [Huerta et al., 2004]. Mixed Finite element methods were used to deal with stroke problem, in the method, rectangular elements with bilinear approximations were used for the velocities and constants for pressure[Johnson and Pitkäranta, 1982]. Compressible and incompressible viscous flow were modeled by using the Finite Element methods solution of Navier-Stokes Equations, the various ingredients based on operator splitting and the solutions are discussed with the results of numerical experiments in [Glowinski and Pironneau, 1992], for instance. The development of a numerical method for the solution of viscoelastic flow problem was reviewed, and the numerical solution compared with experiments [P.T. Baaijens, 1998]. In [Holdeman, 2010], the author described Hermite stream function and velocity finite elements and a divergence-free Galerkin finite element

method for the computation of incompressible flow, and this very powerful method was applied to the stationary lid-driven cavity and backward-facing step test problems. Theoretical and available numerical convergence results for the pressure-correction methods, the velocity-correction methods, and the consistent splitting methods are reviewed in [Guermond et al., 2006]. A stabilized finite element method using hierarchical basis functions applied to the incompressible Navier-Stokes equations has been also presented in [Whiting and Jansen, 2001].

A coupled ES-FEM/ BEM method was proposed which is used to analyze acoustic fluid-structure interaction problems, where the ES-FEM is used to model the structure, while the acoustic fluid is represented by boundary element method[He et al., 2011], this method can be implemented in a way without the increase of degree of freedoms.

The finite volume method is a very popular discretization method which is well suited for the numerical simulation of various types (elliptic, parabolic or hyperbolic, for instance) of conservation laws, it has been extensively used in several engineering fields, such as fluid mechanics, heat and mass transfer or petroleum engineering. A collocated finite volume scheme with a projection method and a splitting method for the time discretization was studied in [Faure et al., 2008] for the model problem of the driven cavity flow. Finite volume particle method was used for researching cross flow over a circular cylinder vibrating with prescribed motion and a freely vibrating cylinder [Nestor and Quinlan, 2013]. Three different flow problems (decaying vortices and flows over a cylinder and a sphere) are also simulated with the immersed-boundary method in [Kim et al., 2001].

Finally, there are other methods which were used for solving the Navier-Stokes Equations, such as a Fourier expansion in the azimuthal direction and an expansion in Chebyshev polynomials in the (non periodic) radial and axial directions in [Speetjens and Clercx, 2005], this method being used for the study of laminar flows in a cylindrical container. [Trujillo and Em Karniadakis, 1999] describe a method based on a penalty method and a new vorticity-velocity formulation implemented for the unsteady three-dimensional Navier-Stokes equations, two-dimensional flow past a cylinder and three-dimensional flow past a cylinder with end-plates in the work.

These numerical methods could be mainly divided into two classes. One called segregated approach consists in solving separately momentum and mass conservation equation (transformed into a pressure equation) to compute the pseudo-velocity field firstly and then to get new pressure[Cai and Zhang, 2012]. After several coupling iterations to get the converged solution, the advantage is that we do not need to solve a fully-coupled formulation, but this algorithm will be relatively slow. Another approach is the fully coupled formulation which can reduce the imposed

diagonal dominance through the pseudo-transient term and reduce it to accelerate the convergence; and also the evolution of the CPU effort per point maybe far better than that associated with a segregated approach. But the drawback is that a large memory is needed to store and solve the coupled system. Moreover, the coupled system is ill-conditioned (of saddle-point type), which justifies the use of very powerful linear solvers. In order to avoid body-fitted unstructured meshes and to use fast and efficient spectral, finite differences or finite volumes approximations on Cartesian meshes, a proposed strategy is to add a penalized velocity term in the momentum equation of the incompressible Navier-Stokes equations. Penalization methods are now quite classical to compute the flow of an incompressible fluid around a no-slip boundary. It appears that the penalization has to be extended to the volume of the body to give correct physical solutions at high Reynolds numbers [Saiki and Biringen, 1996]. A rigorous justification of the penalization method was taken into account a solid body immersed in an incompressible viscous fluid in motion and shown the efficiency of the numerically method[Angot et al., 1999]. Based on a physically sound mathematical model for compressible flows through porous media, a Brinkman penalization method has been extended for numerical simulations of compressible flows around solid obstacles of complex geometries [Liu and Vasilyev, 2007]. In order to compute the flow around an obstacle surrounded by a thin layer of porous material, the penalization method was studied[Carbou, 2004]. Based on the idea of modeling solid bodies as porous media whose permeability tends to zero, the volume penalization method was considered. Using time-dependent penalization, moving obstacle problems was solved. It is validated by comparing moving and fixed cylinder simulations, and also by performing a convergence test for Couette flow between rotating cylinders [Kolomenskiy and Schneider, 2009]. [Montagnier et al., 2013] proposed an efficient parallel spectral method for direct numerical simulations of transitional and turbulent flows.

4.3 Model Reduction Method

The in-plate domains an in-plane-out-of-plane decomposition was proposed for solving porous media flow models in laminates [Chinesta et al., 2011], then coupled multi-physics problems [Chinesta et al., 2012] and for modeling squeeze flows in composite laminates [Ghnatios et al., 2013]. In those cases the 3D solution was obtained from the solution of a sequence of 2D problems (the ones involving the in-plane coordinates) and 1D problems (the ones involving the coordinate related to the plate thickness). The PGD decomposition is applied to perform an in-plane out-of-plane separated representation for the steady state heat equation defined in a multi layered plate [Bognet et al., 2012] for solving elasticity problems. It separated the 3D solutions

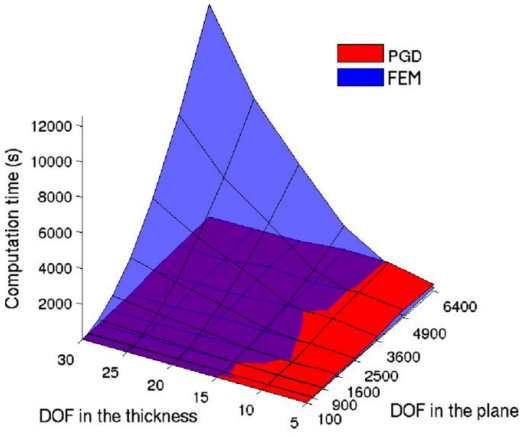


Figure 4.4: Comparison of the PGD and FEM based 3D discretizations

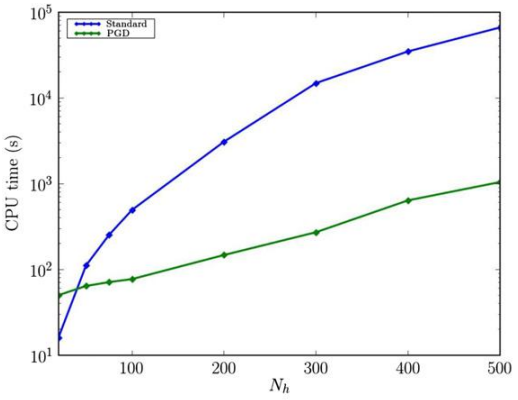


Figure 4.5: CPU time for the PGD and standard solvers

with 2D computational complexity. We can see from figure 4.4 that the PGD solver is faster than the standard solver. [González et al., 2013] attempted to address efficient stabilizations of high dimensional convection-diffusion models encountered in computational physics.

PGD was also considered for solving some problems of fluid mechanics by looking for the solution as sum of tensorial product of functions. PGD was used to solve complex fluid problems [Ammar et al., 2006, Ammar et al., 2007, Ammar et al., 2010c, Chinesta et al., 2011] and Navier-Stokes equations [Dumon et al., 2011, Dumon et al., 2010]. They used a separation of the flow field into functions of x and y, respectively, and demonstrated that PGD is able to solve the Burgers and Stokes equations accurately and with considerable time saving in CPU compared with the standard solver, (see Figure 4.5). [Dumon et al., 2013] also consisted in the association of PGD with a spectral collocation method to solve transfer equations as well as Navier-Stokes equations, however, an incremental scheme is used and PGD method is used only for generating the spatial modes. By using the Proper Generalized Decomposition, [Aghighi et al., 2013] has studied the transient solution of non-linear coupled models related to the Rayleigh-Brd flow model of both Newtonian and non-Newtonian fluids.

4.4 Scope and Outline of Part II

In most of the previous developments related to PGD and Navier-Stokes equations, the modal PGD decomposition was used to separate spatial directions for a simple geometry. In this work, since the physical domains treated by the ISIS-CFD solver are of arbitrary geometrical complexity, Navier-Stokes equations being discretised

with an unstructured finite-volume discretisation, such a spatial decomposition is a priori excluded. Instead, we are going to use the PGD decomposition to separate time and space dimensions, keeping therefore all the generality of the ISIS-CFD software in terms of geometrical complexity. The real objective is to drastically reduce the computational cost of unsteady flows by using a modal representation based on a (hopefully) short decomposition of modes. Another objective is to check if PGD can provide an approximate modal reduction of unsteady flow which could be used in the framework of flow control. Firstly, we will use SVD or POD separation for an ISIS-CFD solution to see whether the unsteady solution can be separated as some finite sum of space and time function and to see how many couples of these functions are needed for an academic unsteady problem. If just several modes are needed, we can easily use the modal decomposition to build a reduced order formulation, otherwise it will be very expensive to use the model reduction method. Then, we will provide a complete description of the PGD formulation of unsteady navier-Stokes equations in the framework of an unstructured Navier-Stokes solver like ISIS-CFD software.

The remainder of this thesis is therefore structured in the following way:

- The chapter 5 will present the basic theory on which the ISIS-CFD solver is based and a numerical example will be presented for the steady Stokes Equations for the lid driven cavity problem.
- The chapter 6 will use SVD for representing the unsteady flow a-priori computed by ISIS-CFD solver. The 2D unsteady flow in lid-driven cavity will be considered as a first academic example. Then, we will give the theories about the PGD formulation of the Unsteady Navier-Stokes Equations by using a same temporal function for the velocity and pressure in the modal decomposition. The discretisatio of these new PGD equations using an ustructured finite volume method will be described in the framework of a fully-coupled pressure-velocity formulation.
- The numerical example for using the proposed PGD method to the flow governed by Stokes and Navier-Stokes equations will be given in the chapter 7.

Chapter 5

The ISIS-CFD flow solver

5.1 Introduction 91
5.2 The ISIS-CFD flow solver 92
5.2.1 Reynolds Averaged Navier-Stokes Equations 92
5.2.2 Discretization of the momentum equation 92
5.2.3 Velocity-pressure coupling algorithm 94
5.2.4 The linear solvers 96
5.3 Numerical results 97
5.3.1 Steady Stokes for Lid driven cavity 97
5.3.2 Analytical solution for Steady Stokes 101
5.4 Discussion and Conclusion 104

5.1 Introduction

The ISIS-CFD flow solver [Deng et al., 2001, Queutey and Visonneau, 2007, Leroyer et al., 2008], created by the ECN and CNRS (DSPM team of the laboratory LHEEA,), solves the incompressible Unsteady Reynolds Averaged Navier Stokes Equations. The software is based on a fully unstructured finite volume method to build the spatial discretization of transport equations. The pseudo-velocity is used to build the pressure equation, by using the Rhie and Chow's reconstruction [Rhie and Chow, 1983]. The ISIS-CFD code uses a linear reconstruction scheme for each control volume and the gradient required for this reconstruction is computed with a least square approach in the present verification exercise. It is worldwide distributed from 2006 by NUMECA Int. under the name FINE^{TM}/Marine and is specialized in the computation of lows in the marin context.

These latter simulations in the context of viscous naval hydrodynamics ([Wackers et al., 2010, Wackers et al., 2011]) show excellent agreement with the experimental results illustrated by comparisons of free-surface elevations as well as velocity field. The unstructured discretization is face-based, which means that cells can be constituted of an arbitrary number of faces. It is reliable as soon as a sufficient density of points is used to capture local gradients like, for instance, the density discontinuity at the free-surface. The use of anisotropic automatic grid refinement controlled by ad-hoc criteria makes possible to improve the local accuracy, even if the finite-volume discretisation is formally only second order accurate.

This chapter is therefore organized as follows.

The general characteristics of ISIS-CFD solver are described in section (5.2) An example of simulation for the steady Stokes Equations is given in section (5.3.1). Finally, some concluding remarks are given.

5.2 The ISIS-CFD flow solver

5.2.1 Reynolds Averaged Navier-Stokes Equations

For an incompressible flow of viscous fluid under isothermal conditions, mass, momentum and volume fraction conservation equations can be written as (using the generalized form of Gauss' theorem):

$$\begin{aligned} &\tfrac{\partial}{\partial t}\int_V \rho dV + \int_S \rho(\vec{U}-\vec{U_d})\cdot\vec{n}\ dS = 0 \\ &\tfrac{\partial}{\partial t}\int_V \rho U_i dV + \int_S \rho U_i(\vec{U}-\vec{U_d})\cdot\vec{n}\ dS = \int_S(\tau ij I_j - pI_i)\cdot\vec{n}\ dS + \int_V \rho g_i dV \end{aligned} \tag{5.1}$$

where V is the domain of interest, or control volume, bounded by the closed surface S moving at the velocity $\vec{U_d}$ with a unit normal vector $\vec{n}$ directed outward. U and p represent respectively the velocity and pressure fields. Density is defined by ρ, τ_{ij} and g_i are the components of viscous stress tensor and the gravity, whereas I_j is a vector whose components vanished, except for the component j which is equal to unity.

5.2.2 Discretization of the momentum equation

A second-order accurate discretized form of the Gauss' theorem leads, for a generic flow variable Q, excluding the pressure, to the following semi-discrete equation available for Eq.(5.1):

$$\frac{\partial}{\partial \tau}(\rho V Q)_c + \frac{\partial}{\partial t}(\rho V Q)_c + \sum_f (CF_f - DF_f) = (S_Q^V) + \sum_f (S_Q^f) \tag{5.2}$$

where CF_f and DF_f are convective and diffusive fluxes through the face f, respectively;

$$\begin{aligned} CF_f &= \dot{m}_f \\ DF_f &= (\Gamma_Q)_f (\vec{\nabla} Q_f \cdot \vec{i_k})(S_k)_f \end{aligned} \tag{5.3}$$

the source terms S_Q^f and S_Q^V are related to the cell volume and cell face, respectively; τ is a local fictitious time variable. Vector $\vec{i_k}$ represents a generic unit vector ($\vec{i_1} = (1, 0, 0)$). The fictitious local time term acts to reinforce the diagonal dominance for the (Picard) linearized equations that are solved successively in a non-coupled (segregated) way.

The mass fluxes through faces are defined by:

$$\dot{m}_f = \rho(\vec{U} - \vec{U}_d)_f \cdot \vec{S}_f \tag{5.4}$$

where $\vec{S}_f$ is the oriented surface vector $\vec{S}_f = S_f \vec{n}_f$.

The time derivatives are evaluated by using three-level Euler second-order accurate approximations, which means:

$$\frac{\partial R}{\partial t} \cong \frac{\delta R}{\delta t} = e^c R^c + e^p R^p + e^q R^q \tag{5.5}$$

the superscript c, p, q refer to the current time t^c, the previous time t^p and the time t^q anterior to p. For example, a constant time step Δt, then $t^p = t^c - \Delta t$ and $t^q = t^c - 2\Delta t$. The coefficients $\{e^c, e^p, e^q\}$ are got from Taylor expansion from t^c, and depend on a possibly variable time step law $\Delta t(t)$. $\frac{\partial R}{\partial \tau}$ is the fictitious local time derivative which is:

$$\frac{\partial R}{\partial \tau} = \frac{R^c - R^{c0}}{\Delta \tau} \tag{5.6}$$

with R^{c0} is a previous estimate of R^c in the non-linear process. Finally, the generic equation can be discretized as:

$$(e^c + \frac{1}{\Delta \tau})(\rho V Q)_C^c + \sum_f (CF_f - DF_f) = (S_Q^V) + \sum_f (S_Q^f) - (e\rho V Q)_C^p - (e\rho V Q)_C^q + \frac{(\rho V Q)_C^{c0}}{\Delta \tau} \tag{5.7}$$

The mass conservation equation can be simplified by using the same kind of second-order accurate approximation,

$$\sum_f \Gamma(\vec{U}) = 0 \tag{5.8}$$

where $\Gamma()$ provides the flux of a vector through the face.

$$\Gamma(\vec{F}) = \vec{F}_f \cdot \vec{S}_f = S_f \ \overrightarrow{F_f} \cdot \overrightarrow{n_f} \tag{5.9}$$

5.2.3 Velocity-pressure coupling algorithm

One of the fundamental feature of the Navier-Stokes Equations for the incompressible fluid is the fact that the pressure acts in the momentum equations to enforce the incompressibility of the flow. In fact, the pressure intervenes only through its gradient in the momentum equation and is not present in the continuity equation. A special treatment is therefore needed to build a pressure equation for the pressure variable.

5.2.3.1 Pressure equation

By using the previous description of the discretization, the momentum equation can be semi-discretized as :

$$\begin{aligned}(e^c + \tfrac{1}{\Delta\tau_C})(\rho\ \overrightarrow{U})^c_C + \tfrac{a_C \overrightarrow{U^c_C} + \sum_{nb} a_{nb} \overrightarrow{U^c_{nb}} + \overrightarrow{S}}{V^c_C} + \overrightarrow{grad}\,(P)|_C \\ - \tfrac{(\rho\ \overrightarrow{U})^{c0}_C}{\Delta\tau_C} + \tfrac{(e\rho V\ \overrightarrow{U})^p_C + (e\rho V\ \overrightarrow{U})^q_C}{V^c_C} = \vec{0}\end{aligned} \tag{5.10}$$

where $\{a_C, a_{nb}\}$ are the matrix coefficients from the discretisation of the implicit part of the diffusive and convective terms; $\vec{S}$ is a source term containing all explicit remaining contributions and external force fields except gravity and pressure. From this equation, the cell-center pseudo-velocity is introduced as:

$$\begin{aligned}\overrightarrow{U^c_C} \ = \ & -C_{PC}(\overrightarrow{\hat{c}_C} + \overrightarrow{grad}\,(P)|_C) \\ & +C_{PC}(\tfrac{(\rho\ \overrightarrow{U})^{c0}_C}{\Delta\tau_C}) \\ & -C_{PC}(\tfrac{(e\rho V\ \overrightarrow{U})^p_C + (e\rho V\ \overrightarrow{U})^q_C}{V^c_C})\end{aligned} \tag{5.11}$$

with

$$C_{PC} = \frac{1}{(e^c + \frac{1}{\Delta\tau_C})\rho^c_C + \frac{a_C}{V^c_C}}$$

where the discretized vector $\overrightarrow{\hat{U}}$, homogeneous to gravity acceleration, includes part of the diffusion, convection and source terms.

$$\overrightarrow{\hat{U}} = \frac{\sum_{nb} a_{nb} \overrightarrow{U^c_{nb}} + \overrightarrow{S}}{V^c_C} \tag{5.12}$$

With the method of Rhie and Chow [Rhie and Chow, 1983], we do not interpolate directly the mass flow at the interface but suppose that the velocity at the face is provided by the following semi-discretized formula, based on the discretisation of the momentum equations:

$$\begin{aligned} \overrightarrow{U^c_f} \;=\; & -C_{Pf}(\overrightarrow{\hat{U}_f} + \overrightarrow{grad}\,(P)|_f) \\ & +C_{Pf}(\tfrac{(\rho \overrightarrow{U})^{c0}_f}{\Delta\tau_f}) \\ & -C_{Pf}(\tfrac{(e\rho V \overrightarrow{U})^p_f + (e\rho V \overrightarrow{U})^q_f}{V^c_f}) \end{aligned} \tag{5.13}$$

with

$$C_{Pf} = \frac{1}{(e^c + \frac{1}{\Delta\tau_f})\rho^c_f + \frac{a_C}{V^c_f}}$$

Using such a formula to rebuild the mass-flux is equivalent to using a local momentum discretisation formula linking together velocity and pressure gradient. From that point of view, we can classify the Rhie and Chow mass reconstruction as a pseudo-physical interpolation approach which, instead of interpolating the quantity make use of an interpolated functional relation provided by the flow physics. Then by using these reconstructed mass flux terms in the continuity equation, we can easily get a pressure equation which is well-posed and does not suffer from checkerboard coupling:

$$\begin{aligned} \textstyle\sum_f C_{pf}\ \overrightarrow{P}\,|_f \cdot \vec{S}_f \;=\; & \textstyle\sum_f (-C_{pf}\ \overrightarrow{\hat{U}_f}) \cdot \vec{S}_f \\ & + \textstyle\sum_f C_{pf}(\tfrac{(\rho \overrightarrow{U})^{c0}_f}{\Delta\tau_f}) \cdot \vec{S}_f \\ & + \textstyle\sum_f -C_{pf}(\tfrac{(e\rho V \overrightarrow{U})^p_f + (e\rho V \overrightarrow{U})^q_f}{V^c_f}) \cdot \vec{S}_f \end{aligned} \tag{5.14}$$

Once the pressure equation is solved, the velocity flux $\mathcal{F}(\vec{U})$ is corrected by:

$$\begin{aligned} \mathcal{F}(\vec{U}^c) = \; & -C_{pf}(\mathcal{F}(\overrightarrow{\hat{U}}) + \mathcal{F}(\vec{\nabla}P)) \\ & +C_{pf}(\tfrac{\mathcal{F}(\rho U^{c0})}{\Delta\tau_f}) \\ & -C_{pf}(\tfrac{(eV)^p \mathcal{F}(\rho \overrightarrow{U^p}) + (eV)^q \mathcal{F}(\rho \overrightarrow{U^q})}{V^c_f}) \end{aligned} \tag{5.15}$$

with

$$\mathcal{F}(\vec{A}) = \vec{A}_f \cdot \vec{S}_f$$

and, therefore, the velocity flux $\mathcal{F}(\vec{U})$ satisfies the continuity equation.

5.2.3.2 Algorithm

Discretized momentum and pressure equations provide a linear coupled system which is solved by a segregated decoupled approach, which is similar to the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm.

1. Initialized flow field quantities Q^0 at $t = t^0$
2. New time step $t = t + \Delta t$
3. Start iterative procedure with $Q = Q^0$
4. If needed, compute the phase concentration for each fluid phase and update global fluid properties,
5. If needed, compute the turbulent quantities from field of step 3
6. Solve the momentum equations to obtain a new prediction of the velocities
7. Solve the pressure equation (Eq.(5.14)) to obtain a new pressure field
8. Update the velocity face fluxes and correct the velocity components with new pressure field
9. If the nonlinear residuals are not low enough, go to step 3 and update the iteration counter within the time step
10. Go to step 2 and update time, t

5.2.4 The linear solvers

Linear systems resulting from momentum and pressure equations are solved with the help of iterative linear solvers.

The momentum equations are easy to solve, which have the strong diagonal dominance as by using the local fictitious time step. So, simple method such as Gauss-Seidel type methods are used. Dirichlet or Neumann type boundary conditions are used for the velocity.

But, the pressure equation is far more complex to solve. It is singular by nature as the pressure is defined to a constant and its conditioning is very bad when grids are highly stretched close to walls.

When velocity conditions are imposed on all boundaries, to make it solvable, a compatibility criteria should be fulfilled which is equivalent to the global incompressibility constraint applied to the whole fluid domain. Therefore, if pressure is not

imposed on one boundary, the velocity field on the boundaries should verify a global incompressibility relation.

Resolution is made through preconditioned conjugate gradients or GMRES methods with a preconditioning based on an incomplete LU decomposition.

5.3 Numerical results

5.3.1 Steady Stokes for Lid driven cavity

Actually, it was not possible for me to modify directly the original ISIS-CFD code because of the huge complexity of such an industrial CFD solver. It was therefore decided to rebuild a prototype code retaining the same algorithmic principles based on MathLab. This prototype code will be built according to the standard algorithmic principles [Deng et al., 2001, Queutey and Visonneau, 2007, Leroyer et al., 2008] and compared to its PGD variant. Therefore to assess this new prototype code, this Section will show a first illustration of it on the computation of a steady Stokes problem. This new code will be also compared with an other code based on a Finite Element formulation developped previously by Adrien Leygue. This flow is defined by the following coupled partial differential equations:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial p}{\partial x} &= 0 \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} - \frac{\partial p}{\partial y} &= 0 \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned} \tag{5.16}$$

the domain is $(x, y) \in [-1, 1]$, the geometry and boundary conditions for the lid driven cavity problem are shown in Fig.(5.1). The boundary conditions are listed as follows,

$$\begin{aligned} u &= 0 \quad on\ x = -1; y \in (-1, 1) \\ u &= 1 \quad on\ x = 1; y \in (-1, 1) \\ v &= 0 \quad on\ y = -1, 1; x \in (-1, 1) \end{aligned} \tag{5.17}$$

Cartesian grid and a non-uniform grid close to the boundaries are used for these numerical examples. A second-order accurate discretisation is used for the Laplace operator and a first-order discretisation for the pressure gradient.

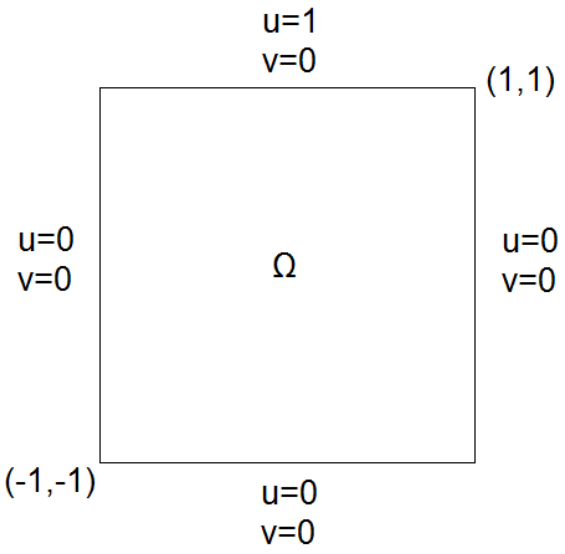


Figure 5.1: Geometry and boundary conditions for the lid driven cavity

For the pressure equation, the under relaxation is used during the iterative process, and new, improved, pressure p^{new} is obtained with:

$$p = (1 - \alpha_p)p^{old} + \alpha_p p^{new} \tag{5.18}$$

where α_p is the pressure under-relaxation factor.

The velocities are also under relaxed. The iteratively improved velocity components u^{new} and v^{new} are obtained from

$$\begin{aligned} u &= (1 - \alpha_u)u^{old} + \alpha_u u^{new} \\ v &= (1 - \alpha_v)v^{old} + \alpha_v v^{new} \end{aligned} \tag{5.19}$$

where α_u and α_v are the $u-$ and $v-$ velocity under-relaxation factors.

For solving the pressure equation, the Successive Over-Relaxation Method(SOR) was used, the relaxation factor being equal to 1.2.

In this case, the under-relaxation factors α_u and α_v for the u and v velocity are equal to 0.5 and the pressure under-relaxation factor α_p is 0.3.

Three meshes (23×23, 43×43 and 63×63) were used to compute the lid-driven cavity problem. The convergence for the three kinds of mesh is shown in Figure(5.2).

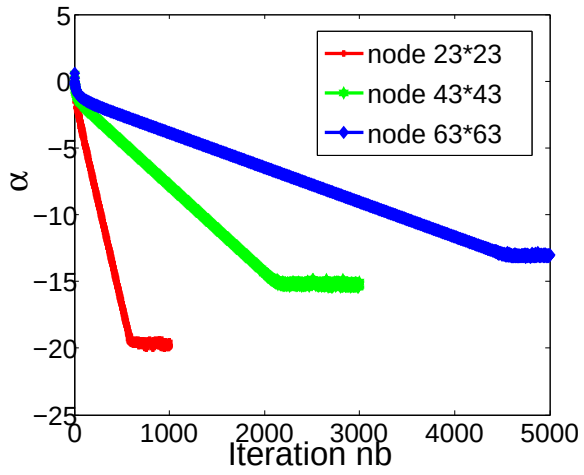
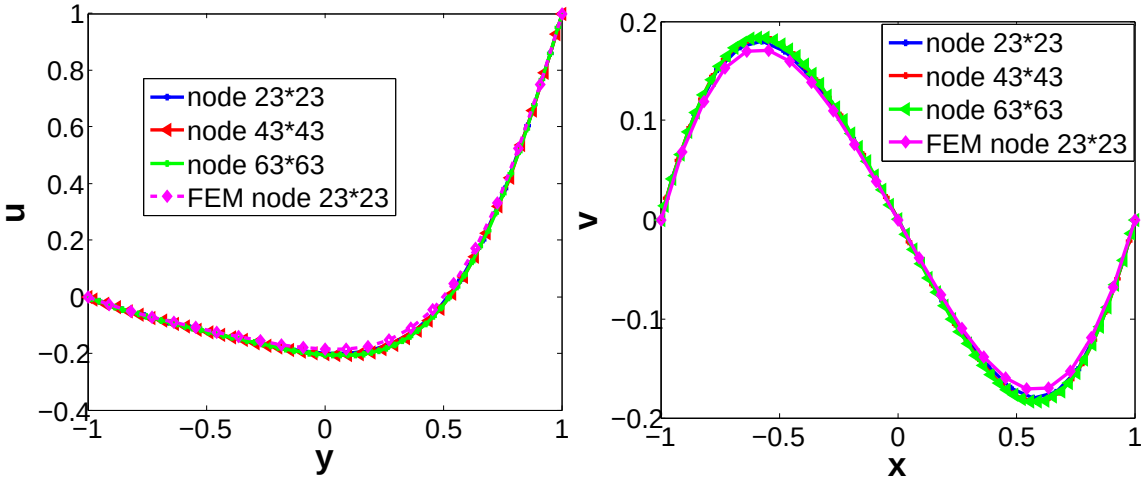


Figure 5.2: Convergence analysis

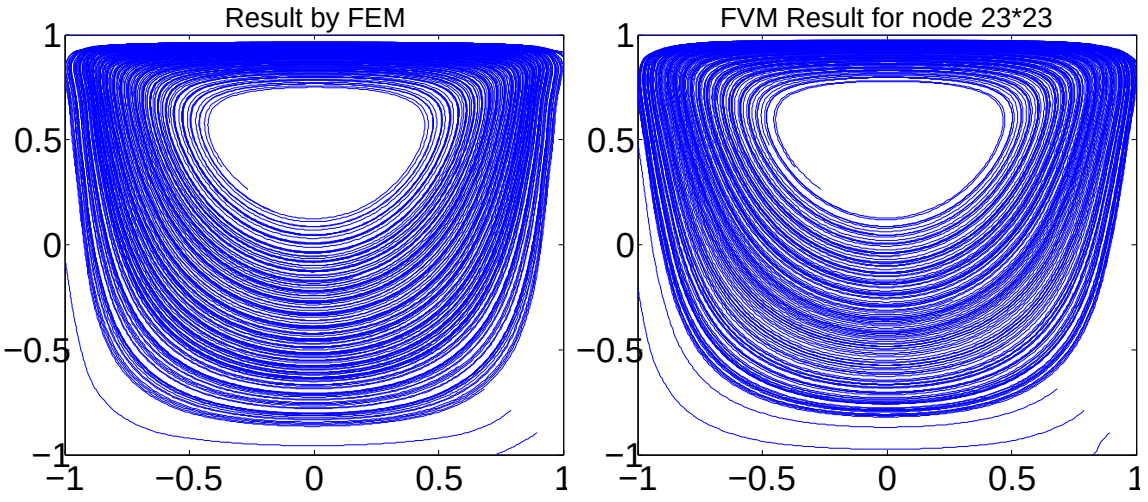


(a) Horizontal velocity component u(0,y)

(b) Vertical velocity component v(x,0)

Figure 5.3: FV and FE solutions for Lid-driven cavity

The approximate horizontal and vertical velocity components along the mid-plane are shown, respectively, in Fig.(5.3(a)) and Fig.(5.3(b)) and the FEM result for the mesh 23×23 is also shown in the same figure. The streamlines for different meshes and FEM results are also shown in Figure(5.4(a)-5.4(d)), respectively.



(a) FEM result(23*23)

(b) FVM result(23*23)

(c) FVM result(43*43)

(d) FVM result(63*63)

Figure 5.4: FEM and FVM solutions for the lid-driven cavity (streamlines)

From the result in Fig.(5.3(a)) and Fig.(5.3(b)), we can observe that the 'ISIS-CFD' results for the lid driven cavity problem are consistent with the FEM solution, althoug the FVM solution seems to be more accurate for the same grid. Figure(5.2) indicates that more and more iterations are needed to reach a converged state when the grid is refined. This is probably due to the use of SOR for solving the pressure equation in the prototype Mathlab code.

The residual for the momentum equations in the chapter is defined as:

$$|Res| = \frac{\sum |Res_i|}{N} \tag{5.20}$$

where Res_i is computed for each node of the grid. For instance, the residual for the x momentum equation is defined at each point by:

$$Res_i = \frac{\partial^2 u_i}{\partial x^2} + \frac{\partial^2 u_i}{\partial y^2} - \frac{\partial p_i}{\partial x} \tag{5.21}$$

The residuals for the momentum equations vs the iteration number for different discretizations are shown in Fig.(5.5). More iterations are needed to reach the same level of residual as expected (see Fig.(5.6)). The CPU time comparison is also shown for these four kinds of discretization in Fig.(5.7). We can observe that the cpu time increases very quickly when the discretisation becomes finer, and of course more iterations are needed to converge for the finer discretizations.

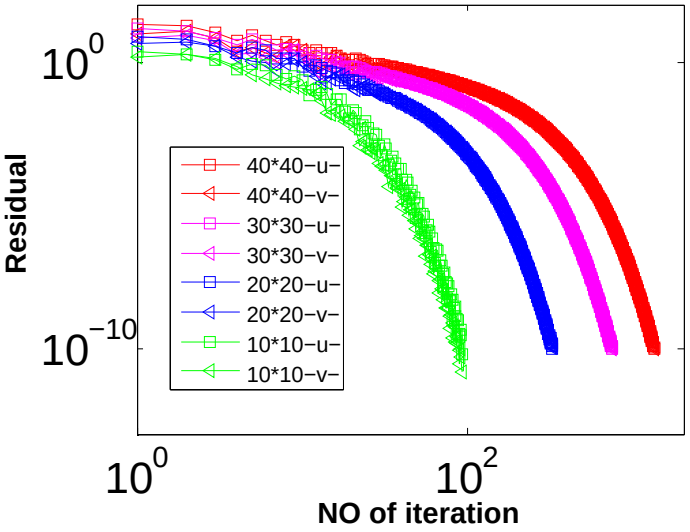


Figure 5.5: Residual vs iteration for 4 cases

5.3.2 Analytical solution for Steady Stokes

Finally, to check the accuracy of this prototype code, we will compute a manufactured solution of the modified Stokes equations.

$$\begin{aligned} u &= cos(2\pi x)sin(\pi y) \\ v &= -sin(2\pi x)cos(\pi y) \\ p &= (1 - 8\pi^2\nu)sin(2\pi x)sin(2\pi y) \end{aligned} \tag{5.22}$$

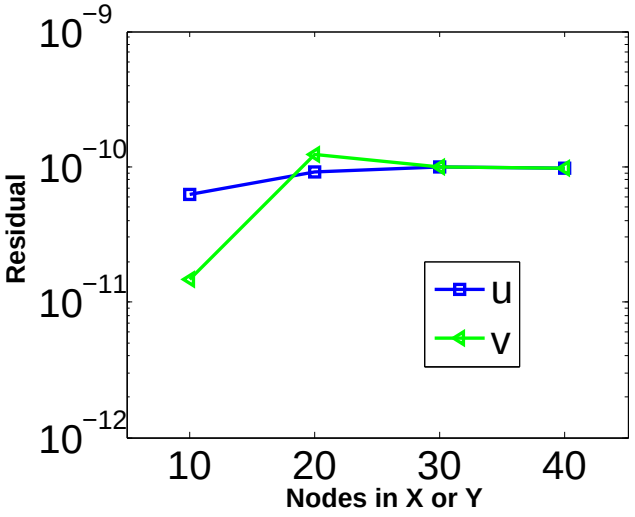


Figure 5.6: Residual vs number of nodes for 4 cases

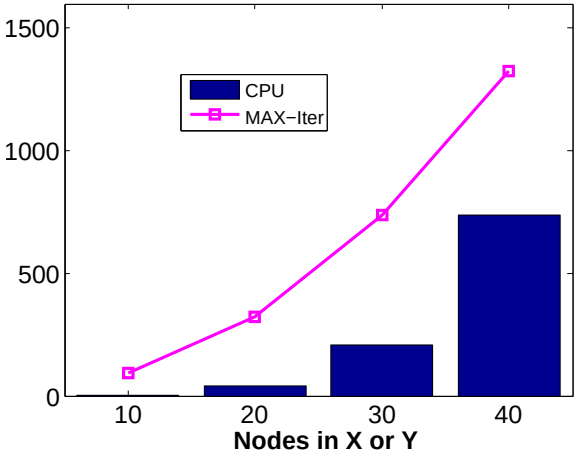


Figure 5.7: Evolution of the CPU time and number of iterations to converge for various grids

is the solution of this following Stokes equation:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial p}{\partial x} &= f_x \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} - \frac{\partial p}{\partial y} &= f_y \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned} \tag{5.23}$$

where the source terms $f(f_x \otimes f_y)$ are derived from the analytical solution,

$$\begin{aligned} f_x &= (8\pi^2\nu + 2\pi(1 - 8\pi^2\nu))cos(2\pi x)sin(2\pi x) \\ f_y &= (2\pi(1 - 8\pi^2\nu) - 8\pi^2\nu)sin(2\pi x)cos(2\pi y) \end{aligned} \tag{5.24}$$

Knowing the analytical solution, we can compute the $L-2$ norm of error ([Cummins et al., 2005]) as:

$$L2Error = \sqrt{\frac{\sum(u^i_{exact} - u^i_{ISIS-CFD})^2}{N}} \tag{5.25}$$

where N is the number of nodes for the discretization, i stands for each nodes, u_{exact} is the exact solution, and $u_{ISIS-CFD}$ is the solution computed with our prototype code.

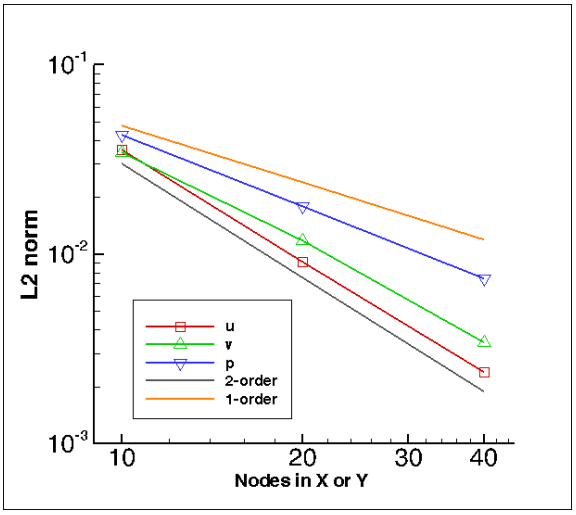


Figure 5.8: L-2 norm of error for the velocities and pressure

The $L-2$ norms of error for the velocities and pressure are shown in the Fig.(5.8) for the above-mentionned manufactured problem. 2-nd order accuracy is obtained for the velocities, and for the pressure, the accuracy is between 1st and 2nd order, due to the fact that the discretisation of the pressure gradient is second-order accurate everywhere except near the boundaries where it is only first-order

accurate. It would have been possible to recover a true second-order accuracy for this special manufactured solution exercise by locating the discretisation point inside the boundary instead of putting it at the boundaries, but this was not done since this *ad-hoc* modification would have been meaningful only for the case of manufactured solutions.

5.4 Discussion and Conclusion

This chapter briefly presented the theory on which ISIS-CFD solver is built, and the steady Stokes flow in a lid driven cavity was considered to assess a prototype code built with MathLab but implemented within the same algorithmic framework as ISIS-CFD. A manufactured solution of the Stokes equation was also built and computed to assess the accuracy of the prototype code.

From the numerical results, we can see that the ISIS-CFD solver is an efficient approach for solving a steady flow problem. The ISIS-CFD solution is consistent with the FEM solution, although a lot of iterations is needed for the convergence of the solution. The second order accuracy is obtained for the velocities, and an accuracy between first and second order for the pressure is observed in agreement with the chosen discretisations.

Chapter 6

PGD for Resolving the Unsteady Navier-Stokes Equations

6.1 Introduction . . . 106

6.2 Singular Value Decomposition (SVD) for ISIS-CFD solution . . . 106

6.2.1 Singular Value Decomposition (SVD) . . . 106

6.2.2 Numerical example for SVD of ISIS-CFD solution . . . 107

6.3 The PGD algorithm illustrated on an unsteady convection-diffusion equation . . . 116

6.3.1 Computing the function $S(x)$ . . . 117

6.3.2 Computing the function $R(t)$ . . . 117

6.4 Comparison between an incremental approach and a separated decomposition . . . 118

6.4.1 In terms of computational effort . . . 118

6.4.2 In terms of memory allocation . . . 119

6.5 PGD formulation for Resolving the Unsteady Navier-Stokes equations . . . 119

6.5.1 Introduction . . . 119

6.5.2 Unsteady Stokes Equations . . . 120

6.5.3 PGD Generalization to the unsteady 2D Stokes equations for incompressible flows . . . 120

6.5.4 A pressure equation formulation to solve the PGD formulation of Navier-Stokes equations for incompressible flows . . . 124

6.6 Treatment of non-linearities for the Navier-Stokes equations . . . 127

6.6.1 Linearization of the non-linear terms in the Navier-Stokes . . . 129

6.6.2 PGD formulation for the linearization . . . 130

6.6.3 Simplifying more the linearization . . . 137

6.7 Discussion and Conclusion 139

6.1 Introduction

In the chapter 5, the ISIS-CFD solver was briefly introduced and some limited illustrations were provided. The present chapter will be devoted to the presentation of a PGD formulation for solving the unsteady Navier-Stokes equations. Since our objective is to use a PGD decomposition to look for separated solutions in time and space, it is interesting to evaluate how many modes are necessary for a simple academic flow in order to check if such a decomposition is a reasonable choice. Consequently, in Section(6.2), we are going to introduce the Singular Value Decomposition (SVD) for representing an academic unsteady flow from a *a priori* solution computed with the ISIS-CFD solver. SVD will be used to decompose the ISIS-CFD solution into spatial and temporal modes to get an *a posteriori* time-space representation of the solution of the problem. Then we will present the Proper Generalized Decomposition Method for building the new equations leading to a separate modal decomposition of the solution. The main objective of this chapter is to provide the theoretical and numerical formulation of an algorithm based on the Proper Generalized Decomposition (PGD) applied to the solution of unsteady viscous flows. By using a separate functional description for the space and time variables, one can formulate an algorithm which may replace the traditional incremental approach, and consequently, may reduce drastically the computational time needed for the simulation of complex unsteady flows if not too many modes are necessary to accurately represent the discrete solution.

6.2 Singular Value Decomposition (SVD) for ISIS-CFD solution

6.2.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a powerful method for matrix factorization introduced by Beltrami and Jordan in 1870s, for complex matrices by Autonne in 1902, extended to the non-square matrices problem by Eckhart and Young in 1939. For a survey on the early history of the SVD, the interested reader is pointed to [Stewart, 1992]. For further details on the SVD and its applications,

see for instance ([Christiansen, 2001, Kleibergen and Paap, 2006, Liang et al., 2007, Luo et al., 2009]). This method is interesting as an extension for eigenvalue decomposition for the rectangular matrix.

We can use the ISIS-CFD solver to compute the incompressible Navier-Stokes equations for the unsteady problem, when we put all time step solution in a matrix, as follows:

$$u(\vec{x},t) = [u_1(\vec{x})\ u_2(\vec{x})\ u_3(\vec{x})...u_t(\vec{x})] \tag{6.1}$$

where the space is discretized with $N = n * n$ points, and M is the number of time steps. The matrix has a rank $N * M$, we can find an approximate compressed representation of $u(\vec{x},t)$ by SVD,

$$u^h(\vec{x},t) \approx \sum_{i=1}^{N} \sigma_i \cdot X_i(\vec{x}) \cdot T_i(t) \tag{6.2}$$

The Singular Value Decomposition (SVD) provides the best L^2 approximation.

$$U = F \cdot S \cdot V^T \tag{6.3}$$

where F are the eigenvectors of $U \cdot U^T$, V are the eigenvectors of $U^T \cdot U$ and S is diagonal and contains the square root of the eigenvalues.

So the solution from the ISIS-CFD solver for the unsteady Navier-Stokes Equations can be represented as:

$$u^h(\vec{x},t) \approx \sum_{i=1}^{n} X_i(\vec{x}) \cdot T_i(t) \tag{6.4}$$

where, the square of eigenvalues are put into the $X_i(\vec{x})$ and $T_i(t)$.

6.2.2 Numerical example for SVD of ISIS-CFD solution

Let us suppose that we have the unsteady solution of lid driven cavity problem (see in the Fig.(6.1)) computed with the ISIS-CFD solver. Here, the top wall is moving with an imposed sinusoidal motion. Then we use SVD for representing the solution of velocity u, v and pressure P to evaluate the modal convergence in terms of spatial and temporal modes.

In this example, the period is $\omega = 2\pi$, and the Reynolds number $Re = 100$, the simulation time being $t = [0, 6]s$.

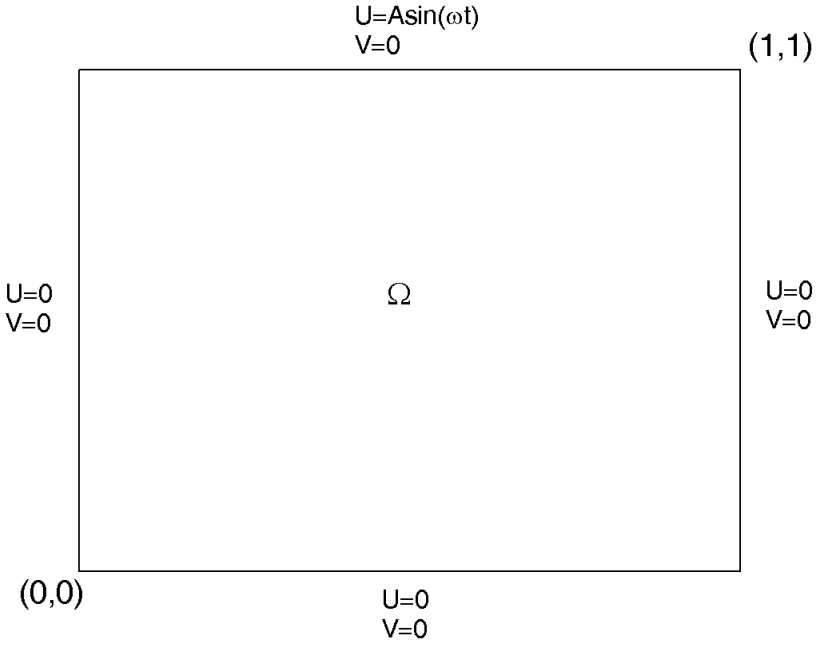


Figure 6.1: Geometry and boundary conditions

6.2.2.1 SVD representation for velocity U

The eigenvalues for the velocity U are shown in the Fig.(6.4). Here, we display several eigenvalues for the velocity U in Table (6.1):

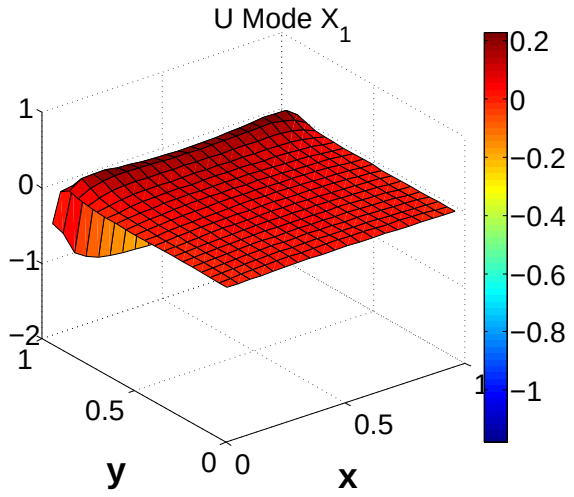
Table 6.1: Several Eigenvalues for the velocity U

$\frac{\alpha_4}{\alpha_1}$	$\frac{\alpha_{12}}{\alpha_1}$	$\frac{\alpha_{112}}{\alpha_1}$
0.0135	$6.4927e^{-4}$	$9.8235e^{-8}$

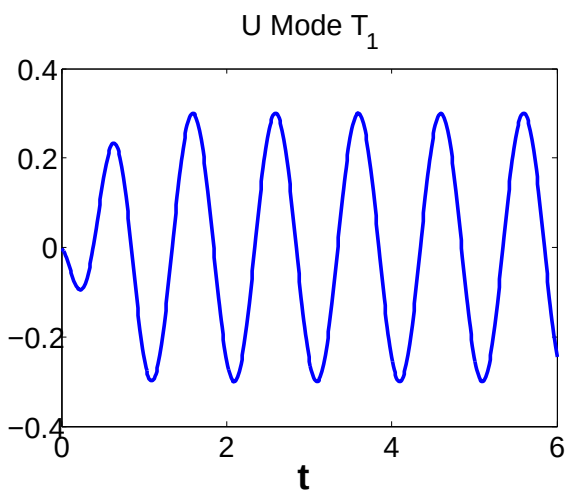
When 12 modes for velocity u were used, the L^2 error norm is reduced to $= 1.3052e^{-5}$, where the L^2 error norm is defined as:

$$L^2 = \sqrt{\frac{\sum(u^i_{SVD} - u^i_{Exact})^2}{N}} \tag{6.5}$$

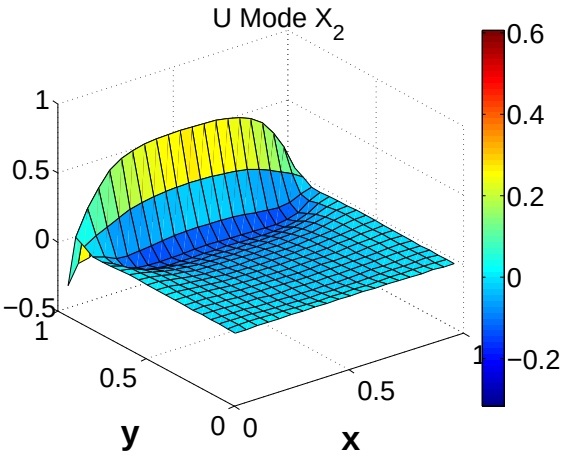
i stands for each nodes, and N is the total number of points for the discretization. u^i_{SVD} and u^i_{SVD} is the SVD solution and the exact solution for node i. The 1-4 modes for the solution of velocity U are given in the Fig.(6.2) and Fig.(6.3), while the evolution of L^2 norm with the modes number is given in Fig.(6.5).



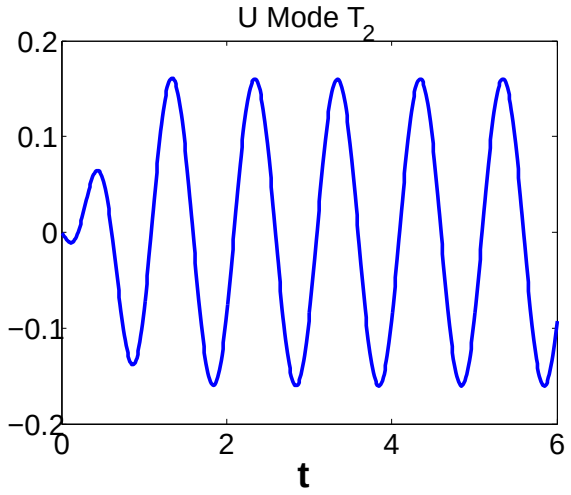
(a) Space mode X1



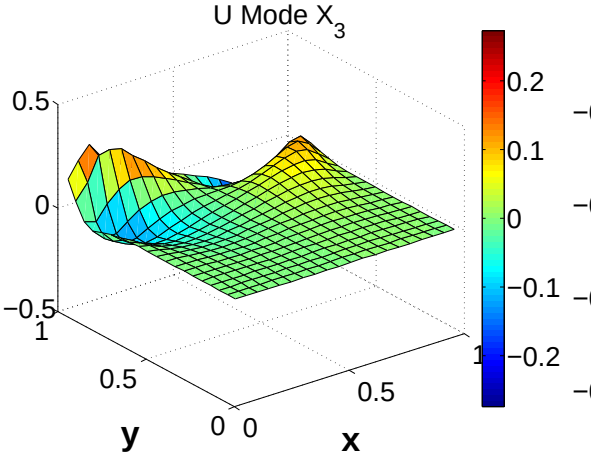
(b) Time mode T1



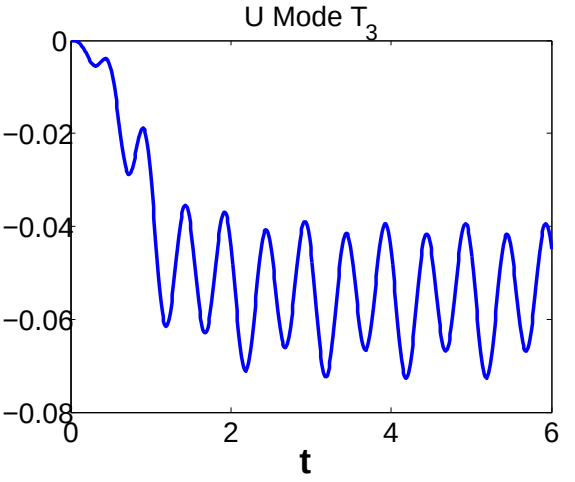
(c) Space mode X2



(d) Time mode T2

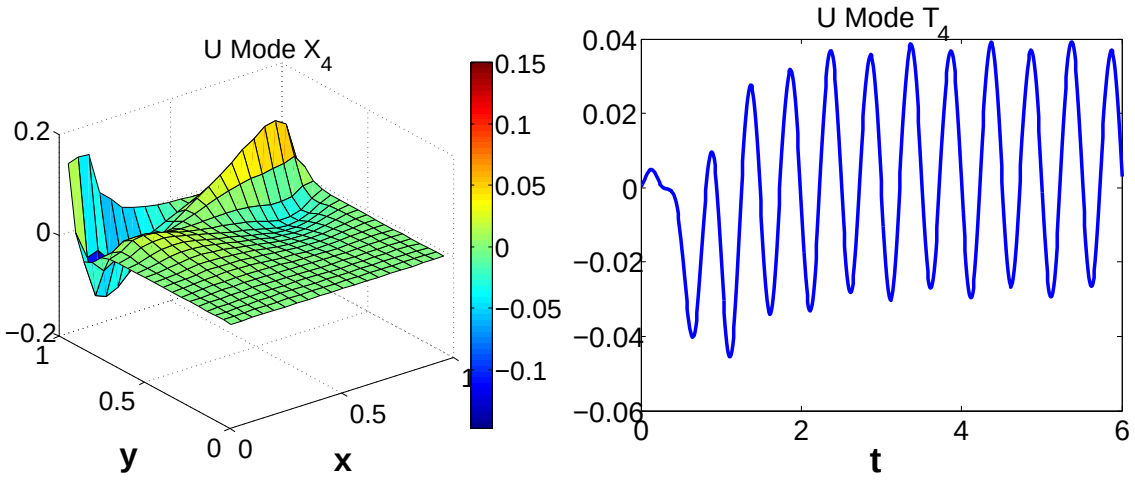


(e) Space mode X3



(f) Time mode T3

Figure 6.2: 3rd mode for u



(a) Space mode X4

(b) Time mode T4

Figure 6.3: 4th mode for u

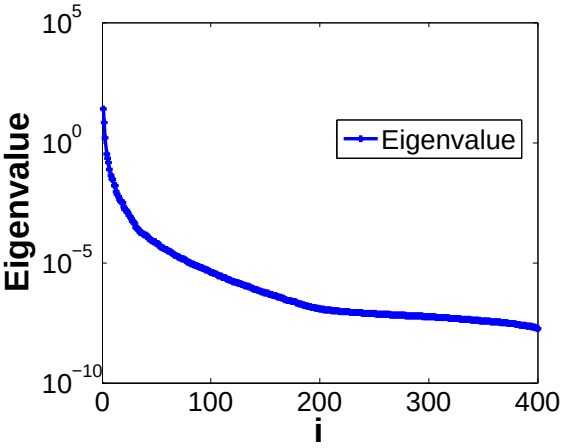


Figure 6.4: Eigenvalue for U

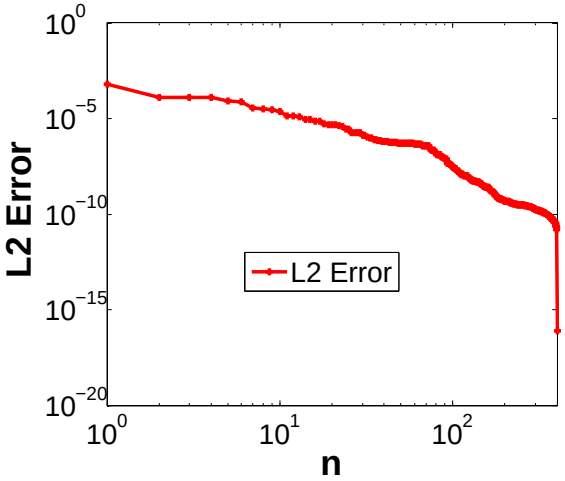


Figure 6.5: Error for U (logplot)

6.2.2.2 SVD representation for velocity V

The eigenvalues for the velocity V are shown in the Fig.(6.8). Several eigenvalues for the velocity V are displayed in Table (6.2):

Table 6.2: Several eigenvalues for the velocity V

$\frac{\alpha_4}{\alpha_1}$	$\frac{\alpha_{12}}{\alpha_1}$	$\frac{\alpha_{126}}{\alpha_1}$
0.0274	$7.0643e^{-4}$	$9.7519e^{-8}$

when 12 modes for velocity v were retained, the L^2 error norm was $7.3265e^{-5}$. The first 4 modes were shown in Fig.(6.6) and Fig.(6.7). The error vs the modes number is shown in Fig.(6.9).

6.2.2.3 SVD representation for pressure P

The eigenvalues for the pressure P are shown in the Fig.(6.12). Here, several eigenvalues for the pressure P are displayed in Table (6.3):

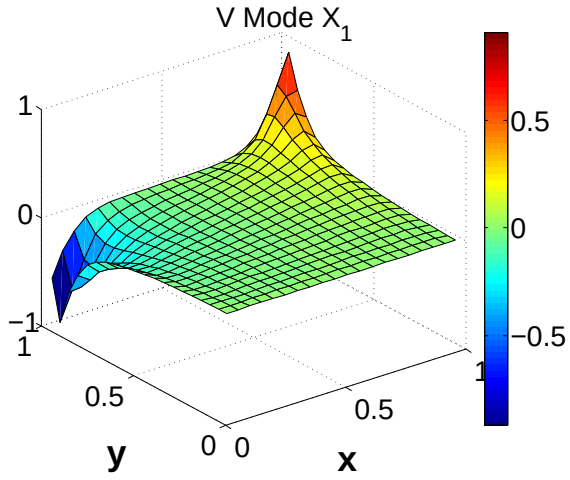
Table 6.3: Several eigenvalues for the pressure P

$\frac{\alpha_4}{\alpha_1}$	$\frac{\alpha_{12}}{\alpha_1}$	$\frac{\alpha_{138}}{\alpha_1}$
0.0095	$5.1713e^{-4}$	$9.9895e^{-8}$

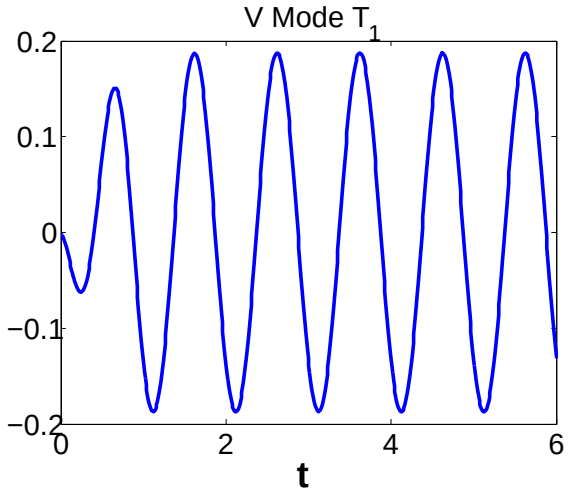
when 12 modes for pressure p for the 6s problem were taken, the L^2 error norm $= 4.4143e^{-4}$, the first 4 modes were shown in Fig.(6.10) and Fig.(6.11), and the evolution of error vs the modes number is shown in the Fig.(6.13).

From the numerical example for the lid driven cavity problem, we can see that L^2 error is under $7.5e^{-5}$ for the velocities representation and $4.5e^{-4}$ for the pressure by using 12 modes instead of the total modes number 400, and the number of modes needed depending on the admissible error of the representation. If more modes are used, an higher accuracy is obtained.

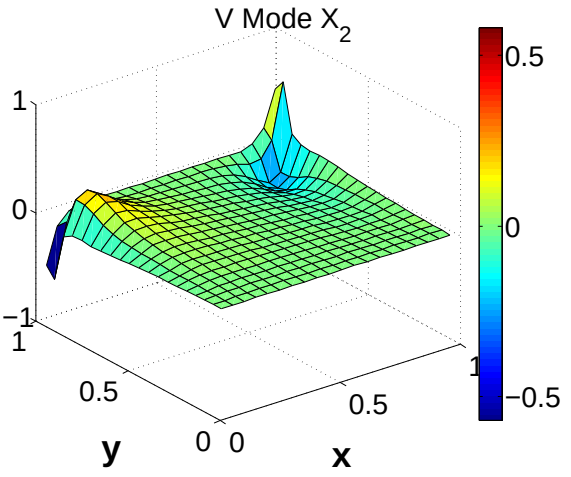
On this elementary example, we have shown that the ISIS-CFD solution of the unsteady problem can be separated into a finite sum of time and space functions by



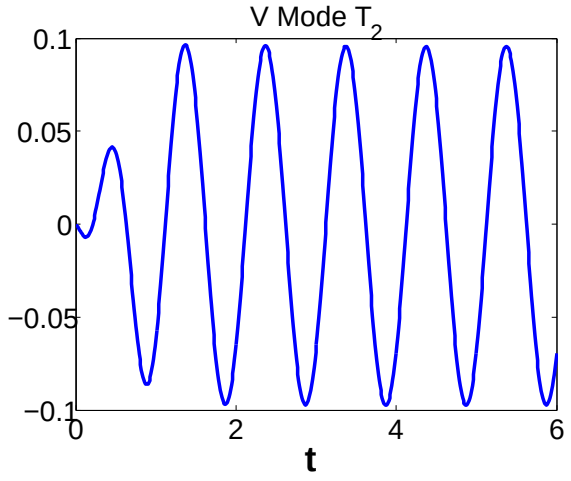
(a) Space mode X1



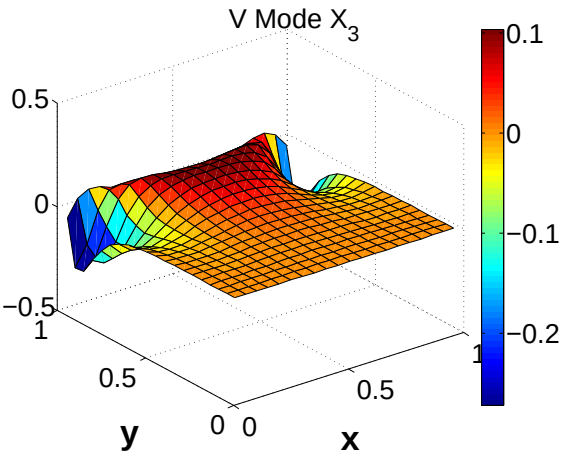
(b) Time mode T1



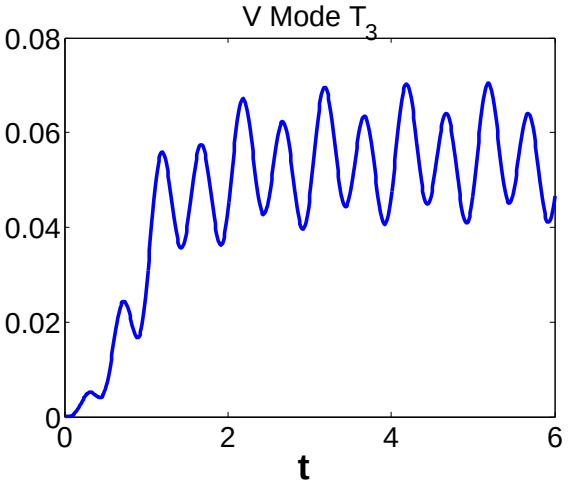
(c) Space mode X2



(d) Time mode T2

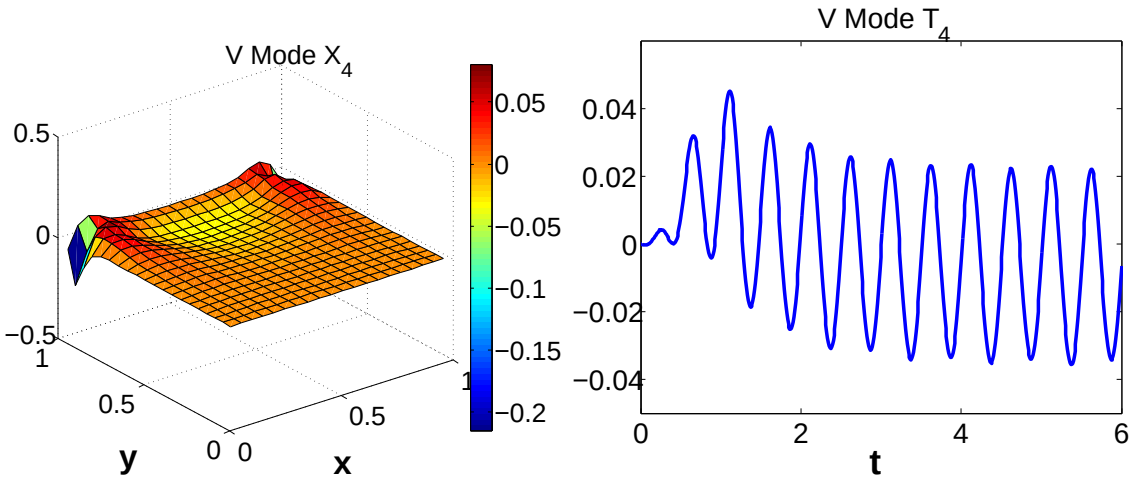


(e) Space mode X3



(f) Time mode T3

Figure 6.6: 1st-3rd modes for v



(a) Space mode X4

(b) Time mode T4

Figure 6.7: 4th mode for v

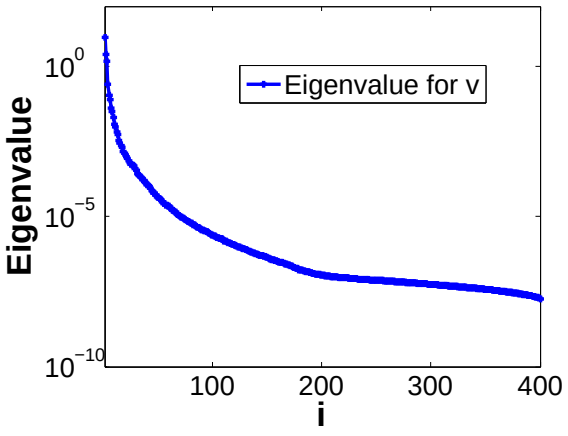


Figure 6.8: Eigenvalue for V

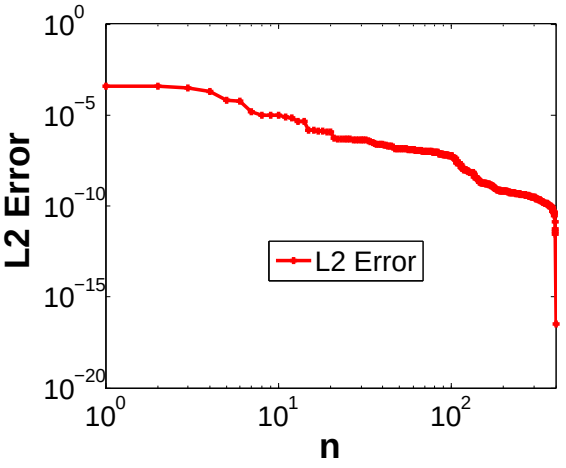
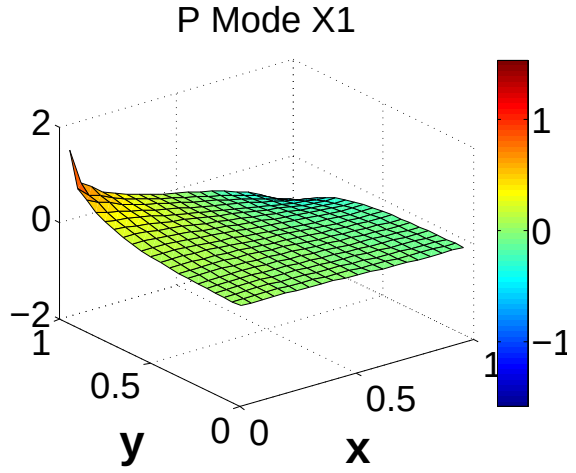
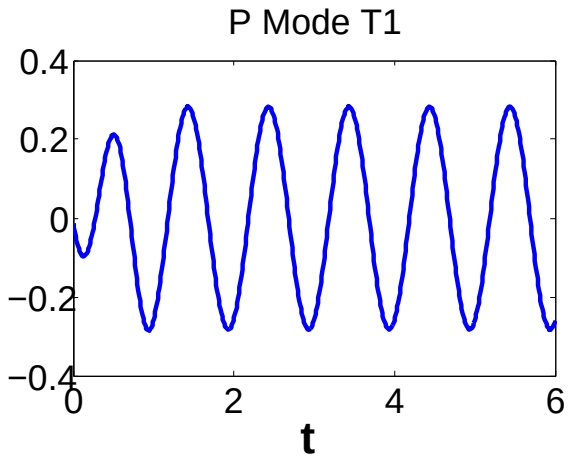


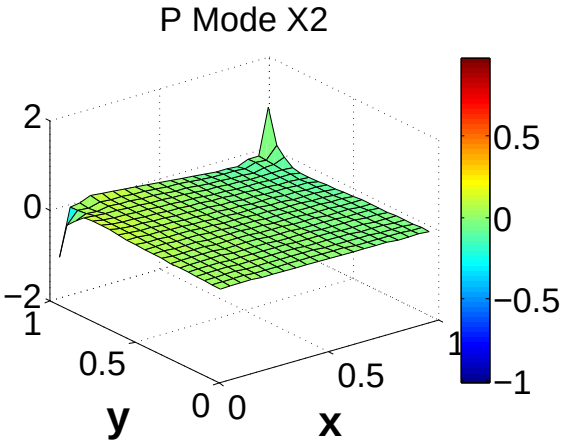
Figure 6.9: Error for V (logplot)



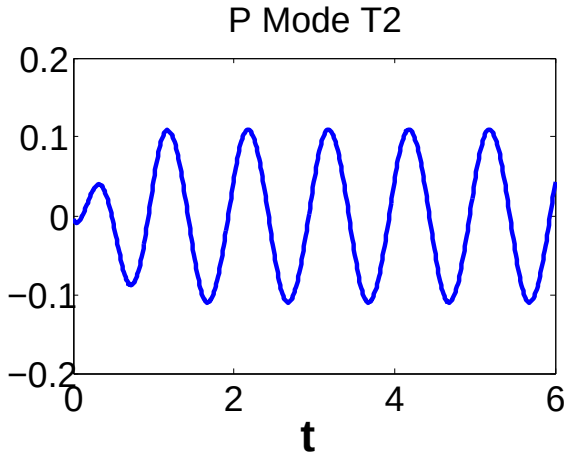
(a) Space mode X1



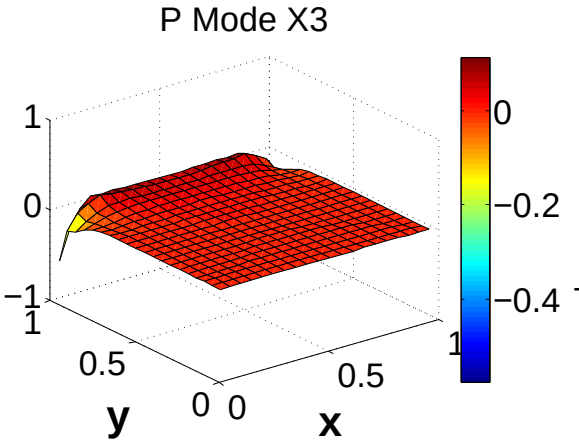
(b) Time mode T1



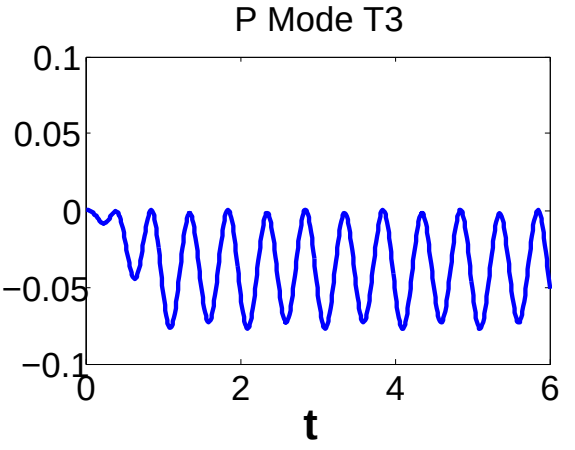
(c) Space mode X2



(d) Time mode T2

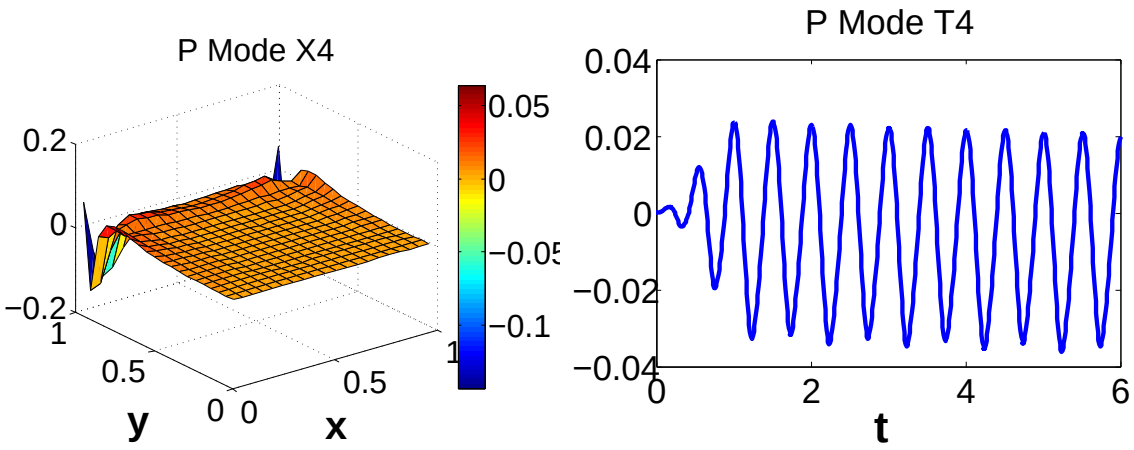


(e) Space mode X3



(f) Time mode T3

Figure 6.10: 1st-3rd modes for P



(a) Space mode X4

(b) Time mode T4

Figure 6.11: 4th mode for Pressure

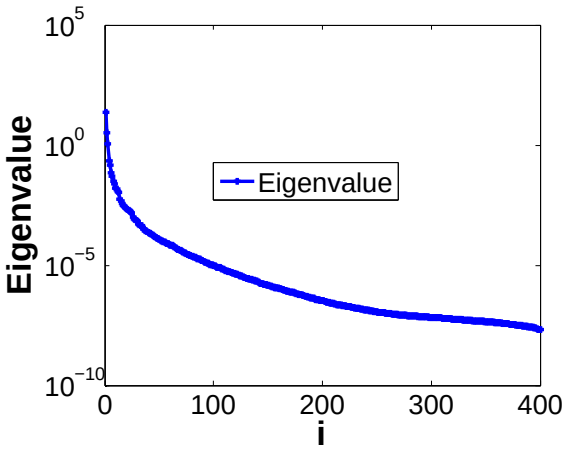


Figure 6.12: Eigenvalue for Pressure

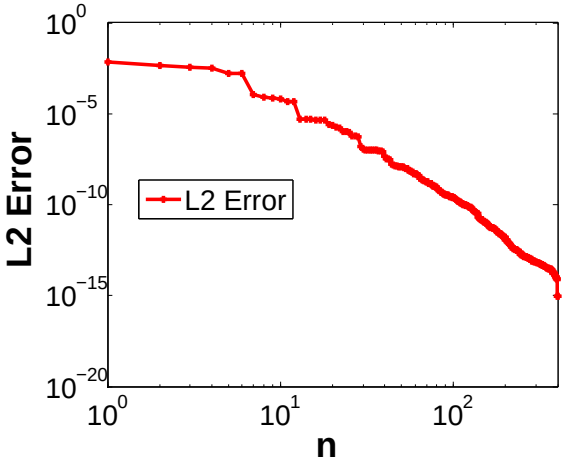


Figure 6.13: Error for Pressure (logplot)

SVD or POD, but both of them are *a posteriori* model reduction for the real problem. However, with a limited number of modes around 12, a very accurate representation is obtained. This is avry encouraging result for the development of a time-space PGD approach, although one must remember that the SVD decomposition is optimal, which is not true for a PGD expansion.

Therefore, in the next section we are going to show how an *a priori* PGD based model reduction can be formulated for the unsteady flow problem. The ISIS-CFD flow solver will be strongly modified to account for the new PGD decomposition and we will see if this new formulation leads to an increased computational efficiency.

6.3 The PGD algorithm illustrated on an unsteady convection-diffusion equation

Let us first illustrate the algorithm on an unsteady convection-diffusion equation. This equation reads:

$$\frac{\partial u}{\partial t} + \vec{v}\vec{\nabla}u - a\Delta u = f(x,t) \tag{6.6}$$

We are looking for a solution which can be expressed by the following functional expansion:

$$u(x,t) \approx \sum_{i=1}^{N} X_i(x)T_i(t) \tag{6.7}$$

Let us suppose that the first terms of this expansion have been already determined, which leaves us with the problem of determining the next group through an iterative process sometimes called “the enrichment step”.

$$u(x,t) \approx u_n(x,t) + R(t)S(x) \tag{6.8}$$

where $u_n(x,t) = \sum_{i=1}^{N} X_i(x)T_i(t)$ is known from previous iterations of enrichment. In order to determine the next group, we build a weak formulation of this problem and introduce the following weighting functions:

$$u^*(x,t) = R^*(t)S(x) + R(t)S^*(x) \tag{6.9}$$

Once injected in the weak formulation, one gets the following equation:

$$\begin{aligned}&\int_0^{T_{max}} \int_\Omega (SR^* + RS^*)[S\tfrac{\partial R}{\partial t} - a\Delta S.R + (\vec{v}\vec{\nabla}S).R]d\overline{x}dt = \\ &\int_0^{T_{max}} \int_\Omega (SR^* + RS^*)[f(x,t) - \sum_{i=1}^N X_i \tfrac{\partial T_i}{\partial t} + a\Delta X_i.T_i - \sum_{i=1}^N (v\nabla\vec{X}_i).T_i]d\overline{x}dt\end{aligned} \tag{6.10}$$

6.3.1 Computing the function $S(x)$

Following the procedure described in [Chinesta et al., 2010a], we suppose first that R is known which implies that $R^* = 0$. Then, the previous equation becomes:

$$\begin{aligned}&\int_\Omega S^*[\alpha_t S - a\beta_t \Delta S + \beta_t v\nabla S]d\overline{x} = \\ &\int_\Omega S^*[\gamma_t(x) - \sum_{i=1}^N \alpha_t^i X_i + a\sum \beta_t^i \Delta X_i - \sum_{i=1}^N \beta_t^i v\nabla X_i]d\overline{x}\end{aligned} \tag{6.11}$$

with:

$$\begin{aligned}\alpha_t &= \int_0^{T_{max}} R(t)\tfrac{\partial R}{\partial t}dt \\ \alpha_t^i &= \int_0^{T_{max}} R(t)\tfrac{\partial T_i}{\partial t}dt \\ \beta_t &= \int_0^{T_{max}} R^2(t)dt \\ \beta_t^i &= \int_0^{T_{max}} R(t)T_i(t)dt \\ \gamma_t &= \int_0^{T_{max}} R(t)f(x,t)dt\end{aligned} \tag{6.12}$$

We can now transform the weak formulation into the following strong formulation:

$$\alpha_t S - a\beta_t \Delta S + \beta_t v\nabla S = \gamma_t(x) - \sum_{i=1}^N \alpha_t^i X_i + a\sum_{i=1}^N \beta_t^i \Delta X_i - \sum_{i=1}^N \beta_t^i v\nabla X_i \tag{6.13}$$

which can be easily solved by using a modified steady version of the available finite volume solver, for instance.

6.3.2 Computing the function $R(t)$

From $S(x)$ just computed, we can search $R(t)$. In this case, we suppose that S is known which means that $S^* = 0$. Then, we get the following simplified weak formulation:

$$\begin{aligned}&\int_0^{T_{max}} \int_\Omega (SR^*)[S\tfrac{\partial R}{\partial t} - a\Delta S.R + (v\nabla S).R]d\overline{x}dt = \\ &\int_0^{T_{max}} \int_\Omega (SR^*)[f(x,t) - \sum X_i \tfrac{\partial T_i}{\partial t} + a\Delta X_i.T_i - \sum_{i=1}^N (v\nabla X_i).T_i]d\overline{x}dt\end{aligned} \tag{6.14}$$

which leads to:

$$\int_0^{T_{max}} R^*[\beta_x \frac{\partial R}{\partial t} + (\lambda_x - a\alpha_x)R - \gamma_x(t) + \sum_{i=1}^{N} \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^{N}(\lambda_x^i - a\alpha_x^i)T_i]dt = 0 \quad (6.15)$$

with:

$$\begin{aligned} \alpha_x &= \int_\Omega S(x)\Delta S(x)d\overline{x} \\ \alpha_x^i &= \int_\Omega S(x)\Delta X_i(x)d\overline{x} \\ \beta_x &= \int_\Omega S^2(x)d\overline{x} \\ \beta_x^i &= \int_\Omega S(x)X_i(x)d\overline{x} \\ \lambda_x &= \int_\Omega S(x)(v.\nabla S(x))d\overline{x} \\ \lambda_x^i &= \int_\Omega S(x)(v.\nabla X_i(x))d\overline{x} \\ \gamma_x &= \int_\Omega S(x)f(x,t)d\overline{x} \end{aligned} \quad (6.16)$$

We can now transform the weak formulation into the following strong formulation:

$$\beta_x \frac{\partial R}{\partial t} = (a\alpha_x - \lambda_x)R + \gamma_x(t) - \sum_{i=1}^{N} \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^{N}(a\alpha_x^i - \lambda_x^i)T_i \quad (6.17)$$

This simple ODE which can be easily solved using an ad-hoc discretisation method.

These two steps will be repeated until convergence is reached, i.e. until the stopping criteria is verified:

$$\left\| R^{(q)}(t).S^{(q)}(x) - R^{(q-1)}(t).S^{(q-1)}(x) \right\| < 10^{-8} \quad (6.18)$$

Finally, we will consider that the enrichment process is achieved when:

$$\left\| \frac{\partial u}{\partial t} + v\nabla u - a\Delta u - f(x,t) \right\| < \varepsilon \quad (6.19)$$

with $\varepsilon = 10^{-8}$.

6.4 Comparison between an incremental approach and a separated decomposition

6.4.1 In terms of computational effort

In terms of computational effort, to compute model of an unsteady flow over n time steps, we should compute n three dimensional fields in an incremental approach. In

the separated representation, to compute n time steps, we need to compute $Q * N$ three dimensional fields, where Q is the number of enrichments of the representation and N is the number of iterations needed to determine each individual functional group. It is clear that a separated will be attractive as soon as $n >> N * Q$. It was observed in previous studies that N is of the order of 10 while Q is more difficult to determine and probably more flow-dependent.

6.4.2 In terms of memory allocation

Let us compare now the respective advantages of a classical incremental approach with respect to the separated representation in terms of memory allocation. With this new formulation, we are able to reconstruct the whole time evolution of the solution $u(x,t)$ as soon as store Q couples $R_i(x), T_i(t)$. The functions $T_i(t)$ are mono dimensional and easy to store even if we have a very large number of time steps. However, $X_i(x)$ is a three-dimensional field which has also to be store Q times. The key question for three-dimensional unsteady computations is the number of groups Q needed to converge on the enrichment process? In an incremental approach with a second-order accurate time discretization, one stores 3 three-dimensional fields, independently of the time step chosen for the discretization but the complete history of the flow field is lost. Only some quantities of interests can be stores for further analysis. In a separated representation, it is mandatory to store several tens of three-dimensional fields to reconstruct the complete time evolution of the solution. Independently of the time step since the cost in terms of storage of $T_i(t)$ is negligible. However, the complete history of the flow field can be easily reconstructed thanks to the separated decomposition.

6.5 PGD formulation for Resolving the Unsteady Navier-Stokes equations

6.5.1 Introduction

Before formulating the time-space PGD decomposition for the non-linear Navier-Stokes equations, we are going to illustrate it on the Stokes equations in order to see how the traditional pressure velocity coupling algorithm will be affected by this new formulation. The treatment of non-linearities will be addressed in a further section.

6.5.2 Unsteady Stokes Equations

The unsteady Stokes Equations in a bidimensional domain in this work are given here as:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} - \frac{\partial}{\partial x}(\nu \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial u}{\partial y}) &= f^u(\overline{x}, t) \\ \frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} - \frac{\partial}{\partial x}(\nu \frac{\partial v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial v}{\partial y}) &= f^v(\overline{x}, t) \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= f^p(\overline{x}, t) \end{aligned} \tag{6.20}$$

where u, v are the velocities in the x, y directions, and p is the pressure, f^i are the source terms for the problems, and ν is the viscous number for the unsteady flow. Most of the time, f^p is null since the flow is strictly incompressible.

6.5.3 PGD Generalization to the unsteady 2D Stokes equations for incompressible flows

The Unsteady Stokes equations are given by:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} - \frac{\partial}{\partial x}(\nu \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial u}{\partial y}) &= f^u(\overline{x}, t) \\ \frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} - \frac{\partial}{\partial x}(\nu \frac{\partial v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial v}{\partial y}) &= f^v(\overline{x}, t) \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned} \tag{6.21}$$

As explained before, if we inject into the 2D Stokes equations the following decomposition for the pressure and velocity fields:

$$\begin{aligned} u(x,t) &\approx \sum_{i=1}^{N} X_i^u(x) T_i^u(t) + X^u T^u \\ v(x,t) &\approx \sum_{i=1}^{N} X_i^v(x) T_i^v(t) + X^v T^v \\ p(x,t) &\approx \sum_{i=1}^{N} X_i^p(x) T_i^p(t) + X^p T^p \end{aligned} \tag{6.22}$$

The fact that the temporal modes are different for each velocity component will result in spatial modes which do not satisfy the classical solenoidal incompressibility condition (i.e. $\frac{\partial X_i^u}{\partial x} + \frac{\partial X_i^v}{\partial y} = 0$ for every mode i). This modification will result into too large modifications of the ISIS-CFD solver, which means that we decided to keep the same temporal modes for each velocity component (i.e. $T_i^u = T_i^v = T_i$). To further simplify, we will also keep the same temporal mode for the pressure (i.e. $T_i^p = T_i$).

This choice of temporal modes leads to the new PGD decomposition:

$$\begin{aligned} u(x,t) &\approx \sum_{k=1}^{i-1} X_k^u(x)T_k(t) + X_i^u T_i \\ v(x,t) &\approx \sum_{k=1}^{i-1} X_k^v(x)T_k(t) + X_i^v T_i \\ p(x,t) &\approx \sum_{k=1}^{i-1} X_k^p(x)T_k(t) + X_i^p T_i \end{aligned} \tag{6.23}$$

which is now injected into a weak formulation of the Stokes equations in the space and time domain of interest:

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega (X_i^u T_i^* + X_i^{u*} T_i) \Big[X_i^u \tfrac{dT_i}{dt} \\ &+ T_i \tfrac{\partial}{\partial x}(X_i^p) - T_i[\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^u}{\partial y})] \Big] \, d\overline{x}dt = \\ &\int_0^{T_{max}} \int_\Omega (X_i^u T_i^* + X_i^{u*} T_i) \Big[f^u(\overline{x},t) - \sum_{k=1}^{i-1} X_k^u \tfrac{dT_k}{dt} \\ &- \sum_{k=1}^{i-1} T_k \tfrac{\partial}{\partial x}(X_k^p) + \sum_{k=1}^{i-1} T_k[\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^u}{\partial y})] \Big] \, d\overline{x}dt \end{aligned}$$

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega (X_i^v T_i^* + X_i^{v*} T_i) \Big[X_i^v \tfrac{dT_i}{dt} \\ &+ T_i \tfrac{\partial}{\partial y}(X_i^p) - T_i[\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^v}{\partial y})] \Big] \, d\overline{x}dt = \\ &\int_0^{T_{max}} \int_\Omega (X_i^v T_i^* + X_i^{v*} T_i) \Big[f^v(\overline{x},t) - \sum_{k=1}^{i-1} X_k^v \tfrac{dT_k}{dt} \\ &- \sum_{k=1}^{i-1} T_k \tfrac{\partial}{\partial y}(X_k^p) + \sum_{k=1}^{i-1} T_k[\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^v}{\partial y})] \Big] \, d\overline{x}dt \end{aligned}$$

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega (X_i^p T_i^* + X_i^{p*} T_i) \Big[T_i \tfrac{\partial X_i^u}{\partial x} + T_i \tfrac{\partial X_i^v}{\partial y} \Big] \, d\overline{x}dt = \\ &\int_0^{T_{max}} \int_\Omega (X_i^p T_i^* + X_i^{p*} T_i) \Big[f^p(\overline{x},t) - \sum_{k=1}^{i-1} [T_k \tfrac{\partial}{\partial x}(X_k^u) + T_k \tfrac{\partial}{\partial y}(X_k^v)] \Big] \, d\overline{x}dt \end{aligned} \tag{6.24}$$

6.5.3.1 Determining the enrichment in space and time

One implements a fixed-point algorithm to determine iteratively the functions in space and time used to enrich the functional expansion.

Determination of the functions in space X_i^u, X_i^v, X_i^p

We suppose that T_i is known, which implies that $T_i^* = 0$ and leads to the following simplified weak formulation:

$$\begin{array}{l} \int_\Omega X_i^{u*} \Big[(\int_0^{T_{max}} T_i \frac{dT_i}{dt} dt) X_i^u \\ + (\int_0^{T_{max}} T_i T_i dt) \frac{\partial X_i^p}{\partial x} - (\int_0^{T_{max}} (T_i)^2 dt) [\frac{\partial}{\partial x}(\nu \frac{\partial X_i^u}{\partial x}) + \frac{\partial}{\partial y}(\nu \frac{\partial X_i^u}{\partial y})] \Big] \, d\overline{x} \quad = \\ \int_\Omega X_i^{u*} \Big[(\int_0^{T_{max}} T_i f^u(\overline{x}, t) dt) - \sum_{k=1}^{i-1} (\int_0^{T_{max}} T_i \frac{dT_k}{dt} dt) X_k^u \\ + \sum_{k=1}^{i-1} (\int_0^{T_{max}} T_i T_k dt) [\frac{\partial}{\partial x}(\nu \frac{\partial X_k^u}{\partial x}) + \frac{\partial}{\partial y}(\nu \frac{\partial X_k^u}{\partial y})] \\ - \sum_{k=1}^{i-1} (\int_0^{T_{max}} T_i T_k dt) \frac{\partial X_k^p}{\partial x}] \Big] \, d\overline{x} \end{array} \tag{6.25}$$

which means that, after having introduced the following coefficients:

$$\begin{array}{lcl} \alpha_{ii} & = & \int_0^{T_{max}} T_i \frac{dT_i}{dt} dt \\ \beta_{ii} & = & \int_0^{T_{max}} (T_i)^2 dt \\ \delta_i^u(\overline{x}) & = & \int_0^{T_{max}} T_i f^u(\overline{x}, t) dt \\ \alpha_{ik} & = & \int_0^{T_{max}} T_i \frac{dT_k}{dt} dt \\ \beta_{ik} & = & \int_0^{T_{max}} T_i T_k dt \end{array} \tag{6.26}$$

one gets the following strong formulation:

$$\begin{array}{l} \alpha_{ii} X_i^u + \beta_{ii} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_i^u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_i^u}{\partial y}) + \frac{\partial X_i^p}{\partial x}] = \\ \delta_i^u - \sum_{k=1}^{i-1} \{ \alpha_{ik} X_k^u + \beta_{ik} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_k^u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_k^u}{\partial y}) + \frac{\partial X_k^p}{\partial x}] \} \end{array} \tag{6.27}$$

By analogy, it is possible to write the modified momentum equation for the v component of the velocity:

$$\begin{array}{l} \alpha_{ii} X_i^v + \beta_{ii} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_i^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_i^v}{\partial y}) + \frac{\partial X_i^p}{\partial y}] = \\ \delta_i^v - \sum_{k=1}^{i-1} \{ \alpha_{ik} X_k^v + \beta_{ik} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_k^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_k^v}{\partial y}) + \frac{\partial X_k^p}{\partial y}] \} \end{array} \tag{6.28}$$

with the following coefficients:

$$\delta_i^v(\overline{x}) = \int_0^{T_{max}} T_i f^v(\overline{x}, t) dt \tag{6.29}$$

As demonstrated previously, the incompressibility constraint is automatically satisfied by all the space functions for the velocity components. Therefore, the equation :

$$\frac{\partial X_i^u}{\partial x} + \frac{\partial X_i^v}{\partial y} = 0 \tag{6.30}$$

will be satisfied for every group i.

Determination of the temporal modes T_i

In this second iteration, one supposes that the functions in space X_i^u, X_i^v and X_i^p are known, which leads to $X_i^{u*}=0$, $X_i^{v*}=0$ and $X_i^{p*}=0$ and, consequently, to simplified weak formulations. Let us treat first the momentum equation along X.

$$\begin{aligned}
&\int_0^{T_{max}} T_i^* \left[(\int_\Omega (X_i^u)^2 d\overline{x}) \tfrac{dT_i}{dt} + (\int_\Omega X_i^u [-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^u}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^u}{\partial y})] d\overline{x}) T_i \right] dt \\
&+ \int_0^{T_{max}} T_i^* \left[(\int_\Omega X_i^u \tfrac{\partial X_i^p}{\partial x} d\overline{x}) T_i \right] dt = \int_0^{T_{max}} T_i^* \left[(\int_\Omega X_i^u f^u(\overline{x},t) d\overline{x}) \right] dt \\
&- \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X^u [-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^u}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^u}{\partial y})] d\overline{x}) T_k \right] dt \\
&- \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X_i^u X_k^u d\overline{x}) \tfrac{dT_k}{dt} \right] dt - \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X_i^u \tfrac{\partial X_k^p}{\partial x} d\overline{x}) T_k \right] dt
\end{aligned} \tag{6.31}$$

which becomes:

$$a_{ii}^u \frac{dT_i}{dt} + b_{ii}^u T_i = d_i^u - \sum_{k=1}^{i-1} a_{ik}^u \frac{dT_k}{dt} - \sum_{k=1}^{i-1} b_{ik}^u T_k \tag{6.32}$$

once the following coefficients have been introduced:

$$\begin{aligned}
a_{ii}^u &= \int_\Omega (X_i^u)^2 d\overline{x} \\
b_{ii}^u &= \int_\Omega X_i^u \left[-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^u}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^u}{\partial y}) + \tfrac{\partial X_i^p}{\partial x} \right] d\overline{x} \\
d_i^u &= \int_\Omega X_i^u f^u(\overline{x},t) d\overline{x} \\
a_{ik}^u &= \int_\Omega X_i^u X_k^u d\overline{x} \\
b_{ik}^u &= \int_\Omega X_i^u \left[-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^u}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^u}{\partial y}) + \tfrac{\partial X_k^p}{\partial x} \right] d\overline{x}
\end{aligned} \tag{6.33}$$

The momentum equation for V can also be used to determine T_i. Let us see what we are going to get.

$$\begin{aligned}
&\int_0^{T_{max}} T_i^* \left[(\int_\Omega (X_i^v)^2 d\overline{x}) \tfrac{dT_i}{dt} + (\int_\Omega X_i^v [-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^v}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^v}{\partial y})] d\overline{x}) T_i \right] dt \\
&+ \int_0^{T_{max}} T_i^* \left[(\int_\Omega X_i^v \tfrac{\partial X_i^p}{\partial y} d\overline{x}) T_i \right] dt = \int_0^{T_{max}} T_i^* \left[(\int_\Omega X_i^v f^u(\overline{x},t) d\overline{x}) \right] dt \\
&- \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X_i^v [-\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^v}{\partial x}) - \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^v}{\partial y})] d\overline{x}) T_i \right] dt \\
&- \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X_i^v X_k^v d\overline{x}) \tfrac{dT_k}{dt} \right] dt - \int_0^{T_{max}} T_i^* \left[\textstyle\sum_{k=1}^{i-1} (\int_\Omega X_i^v \tfrac{\partial X_k^p}{\partial y} d\overline{x}) T_k \right] dt
\end{aligned} \tag{6.34}$$

which becomes:

$$a_{ii}^v \frac{dT_i}{dt} + b_{ii}^v T_i = d_i^v - \sum_{k=1}^{i-1} a_{ik}^v \frac{dT_k}{dt} - \sum_{k=1}^{i-1} b_{ik}^v T_k \tag{6.35}$$

once the following coefficients have been introduced:

$$\begin{aligned}
a_{ii}^v &= \int_\Omega (X_i^v)^2 d\overline{x} \\
b_{ii}^v &= \int_\Omega X_i^v \left[-\frac{\partial}{\partial x}(\nu \frac{\partial X_i^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_i^v}{\partial y}) + \frac{\partial X_i^p}{\partial y} \right] d\overline{x} \\
d_i^v &= \int_\Omega X_i^v f^v(\overline{x}, t) d\overline{x} \\
a_{ik}^v &= \int_\Omega X_i^v X_k^v d\overline{x} \\
b_{ik}^v &= \int_\Omega X_i^v \left[-\frac{\partial}{\partial x}(\nu \frac{\partial X_k^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_k^v}{\partial y}) + \frac{\partial X_k^p}{\partial y} \right] d\overline{x}
\end{aligned} \tag{6.36}$$

We can also use the sum of these coefficients to determine the equation satisfied by T_i.

As demonstrated previously, the mass conservation is automatically satisfied by the velocity spatial modes and can not be used to determine the temporal functions.

6.5.4 A pressure equation formulation to solve the PGD formulation of Navier-Stokes equations for incompressible flows

6.5.4.1 Solving in space

A new generic form

As in the incremental approach, we can use a segregated formulation to determine the velocity components from the momentum equations and the pressure field from the incompressibility constraint transformed into a pressure equation. Then, this set of equations can be solved in a segregated way using a linear coupling algorithm like SIMPLE, or in a fully coupled way by gathering all the discretised equations into a single linear system. Let us reformulate the previous equations in a more concise and generic formulation.

$$\begin{aligned}
\frac{\alpha_{ii}}{\beta_{ii}} X_i^u + [-\frac{\partial}{\partial x}(\nu \frac{\partial X_i^u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_i^u}{\partial y})] + \frac{\partial X_i^p}{\partial x} &= \frac{\delta_i^{u*}}{\beta_{ii}} \\
\frac{\alpha_{ii}}{\beta_{ii}} X_i^v + [-\frac{\partial}{\partial x}(\nu \frac{\partial X_i^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_i^v}{\partial y})] + \frac{\partial X_i^p}{\partial y} &= \frac{\delta_i^{v*}}{\beta_{ii}} \\
\frac{\partial X_i^u}{\partial x} + \frac{\partial X_i^v}{\partial y} &= 0
\end{aligned} \tag{6.37}$$

with:

$$\begin{aligned}
\delta_i^{u*} &= \delta_i^u - \textstyle\sum_{k=1}^{i-1} \{ \alpha_{ik} X_k^u + \beta_{ik} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_k^u}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_k^u}{\partial y}) + \frac{\partial X_k^p}{\partial x}] \} \\
\delta_i^{v*} &= \delta_i^v - \textstyle\sum_{k=1}^{i-1} \{ \alpha_{ik} X_k^v + \beta_{ik} [-\frac{\partial}{\partial x}(\nu \frac{\partial X_k^v}{\partial x}) - \frac{\partial}{\partial y}(\nu \frac{\partial X_k^v}{\partial y}) + \frac{\partial X_k^p}{\partial y}] \}
\end{aligned} \tag{6.38}$$

A finite volume formulation

Following the notations used in , the momentum equations read, once discretised:

$$\begin{aligned} Vol_c \frac{\alpha_{ii}}{\beta_{ii}} X_c^u + \sum C_{nb} X_{nb}^u + C_d X_c^u + Vol_c Discr\left[\frac{\partial}{\partial x} X^p\right] + SrcU &= 0 \\ Vol_c \frac{\alpha_{ii}}{\beta_{ii}} X_c^v + \sum C_{nb} X_{nb}^v + C_d X_c^v + Vol_c Discr\left[\frac{\partial}{\partial y} X^p\right] + SrcV &= 0 \end{aligned} \tag{6.39}$$

where:

$$\begin{aligned} SrcU &= -Vol_c \frac{\delta_i^{u*}}{\beta_{ii}} \\ SrcV &= -Vol_c \frac{\delta_i^{v*}}{\beta_{ii}} \end{aligned} \tag{6.40}$$

The indices i are no more mentionned for the sake of simplicity. The operator $Discr$ stands for "Discretization of". Vol_c is the volume of the cell of integration, C_{nb} and C_d are the discretisation coefficients of the convective-diffusive common operator present in both momentum equations. In order to build a pressure equation, a new pseudo-velocity field $(\hat{X}_c^u, \hat{X}_c^v)$ is introduced:

$$\begin{aligned} \hat{X}_c^u - \frac{1}{Vol_c} \sum C_{nb} X_{nb}^u &= \frac{SrcU}{Vol_c} = -\frac{\delta_i^{u*}}{\beta_{ii}} \\ \hat{X}_c^v - \frac{1}{Vol_c} \sum C_{nb} X_{nb}^v &= \frac{SrcV}{Vol_c} = -\frac{\delta_i^{v*}}{\beta_{ii}} \end{aligned} \tag{6.41}$$

which leads to:

$$\begin{aligned} C_{Diag-Solv} X_c^u + Vol_c \hat{X}_c^u + Vol_c Discr\left[\frac{\partial}{\partial x} X^p\right] &= 0 \\ C_{Diag-Solv} X_c^v + Vol_c \hat{X}_c^v + Vol_c Discr\left[\frac{\partial}{\partial y} X^p\right] &= 0 \end{aligned} \tag{6.42}$$

with $C_{Diag-Solv} = C_d + Vol_c \frac{\alpha_{ii}}{\beta_{ii}}$, or:

$$\begin{aligned} X_c^u &= -\frac{Vol_c}{C_{Diag-Solv}} \left(\hat{X}_c^u + Vol_c Discr\left[\frac{\partial}{\partial x} X^p\right]\right) \\ X_c^v &= -\frac{Vol_c}{C_{Diag-Solv}} \left(\hat{X}_c^v + Vol_c Discr\left[\frac{\partial}{\partial y} X^p\right]\right) \end{aligned} \tag{6.43}$$

which becomes, after having introduced the usual notations:

$$Cp = \frac{Vol_c}{C_{Diag-Solv}} \tag{6.44}$$

$$\begin{aligned} X_c^u &= -Cp \left(\hat{X}_c^u + Discr\left[\frac{\partial}{\partial x} X^p\right]\right) \\ X_c^v &= -Cp \left(\hat{X}_c^v + Discr\left[\frac{\partial}{\partial y} X^p\right]\right) \end{aligned} \tag{6.45}$$

To build now the pressure equation, the starting point is the mass conservation equation for mono or multi-fluid incompressible flows integrated on a control volume Vol_c.

$$\int_{Vol_c} \left[\frac{\partial}{\partial x} \left(X^u \right) + \frac{\partial}{\partial y} \left(X^v \right) \right] dV = \int_{\partial Vol_c} \left(X^u n_x + X^v n_y \right) dS = 0 \tag{6.46}$$

The mass conservation equation can be rewritten as follows:

$$\int_{\partial Vol_c} \left(X^u n_x + X^v n_y \right) dS = \sum_{faces} \left(X_f^u S x_f + X_f^v S y_f \right) = 0 \tag{6.47}$$

with $Sx_f = n_{xf} S_f$ and $Sy_f = n_{yf} S_f$. We have now to use the decomposition of the velocity introduced before and reconstruct the various terms of this decomposition at the faces of the control volume.

$$\begin{aligned} X_f^u &= -Cp_f \left(\hat{X}_f^u + Discr \left[\tfrac{\partial}{\partial x} X^p \right]_f \right) \\ X_f^v &= -Cp_f \left(\hat{X}_f^v + Discr \left[\tfrac{\partial}{\partial y} X^p \right]_f \right) \end{aligned} \tag{6.48}$$

which leads to the following pressure equation:

$$\begin{aligned} &- \sum_{faces} \left[Cp_f S x_f Discr \left[\frac{\partial X^p}{\partial x} \right]_f + Cp_f S y_f Discr \left[\frac{\partial X^p}{\partial y} \right]_f \right] = \\ &\sum_{faces} \left[Cp_f \hat{X}_f^u S x_f + Cp_f \hat{X}_f^v S y_f \right] \end{aligned} \tag{6.49}$$

Now the pressure derivatives at the face f have to be decomposed into two parts, (i) a part normal to the face which will be treated implicitly, (ii) a part tangent to the face which will be put in the right hand-side term and treated explicitly.

$$\begin{aligned} \tfrac{\partial X^p}{\partial x} &= \tfrac{\partial X^p}{\partial n} n \cdot x + \tfrac{\partial X^p}{\partial \tau} \tau \cdot x \\ \tfrac{\partial X^p}{\partial y} &= \tfrac{\partial X^p}{\partial n} n \cdot y + \tfrac{\partial X^p}{\partial \tau} \tau \cdot y \end{aligned} \tag{6.50}$$

This leads to the following standard pressure equation:

$$\begin{aligned} &- \sum_{faces} \left(Cp_f n_x^2 + Cp_f n_y^2 \right)_f S_f \left(\frac{\partial X^p}{\partial n} \right)_f = \\ &\sum_{faces} \left(Cp_f n_x \tau_x + Cp_f n_y \tau_y \right)_f S_f \left(\frac{\partial X^p}{\partial \tau} \right)_f \\ &+ \sum_{faces} \left(Cp_f \hat{X}_f^u n_x + Cp_f \hat{X}_f^v n_y \right) S_f \end{aligned} \tag{6.51}$$

which can be simplified into:

$$\sum_{faces} Cp_f S_f \left(\frac{\partial X^p}{\partial n} \right)_f + \sum_{faces} \left(Cp_f \hat{X}_f^u n_x + Cp_f \hat{X}_f^v n_y \right) S_f = 0 \tag{6.52}$$

which leads to, once discretised:

$$\sum C_{nb}^{PP} X_{nb}^{p} + \sum C_{nb}^{P\hat{U}} \hat{X}^{u}{}_{nb} + \sum C_{nb}^{P\hat{V}} \hat{X}^{v}{}_{nb} = 0 \tag{6.53}$$

where:

$$\begin{aligned} \textstyle\sum_{faces} Cp_f S_f \left(\frac{\partial X^p}{\partial n}\right)_f &= \sum C_{nb}^{PP} P_{nb} \\ \textstyle\sum_{faces} \left(Cp_f \hat{X}_f^u n_x + Cp_f \hat{X}_f^v n_y\right) S_f &= \sum C_{nb}^{P\hat{U}} \hat{U}_{nb} + \sum C_{nb}^{P\hat{V}} \hat{V}_{nb} \end{aligned} \tag{6.54}$$

The PGD fully coupled formulation

$$\begin{aligned} C_{Diag-Solv} X_c^u + Vol_c \hat{X}^u{}_c + Vol_c \sum C_{nb}^{UP} X_{nb}^p &= 0 \\ C_{Diag-Solv} X_c^v + Vol_c \hat{X}^v{}_c + Vol_c \sum C_{nb}^{VP} X_{nb}^p &= 0 \\ \hat{X}_c^u - \tfrac{1}{Vol_c} \sum C_{nb}^{UU} X_{nb}^u &= -\tfrac{\delta_i^{u*}}{\beta_{ii}} \\ \hat{X}_c^v - \tfrac{1}{Vol_c} \sum C_{nb}^{VV} X_{nb}^v &= -\tfrac{\delta_i^{v*}}{\beta_{ii}} \\ \sum C_{nb}^{PP} X_{nb}^p + \sum C_{nb}^{P\hat{U}} \hat{X}^u{}_{nb} + \sum C_{nb}^{P\hat{V}} \hat{X}^v{}_{nb} &= -S_p \end{aligned} \tag{6.55}$$

with $C_{Diag-Solv} = C_d + Vol_c \frac{\alpha_{ii}}{\beta_{ii}}$.

6.5.4.2 Solving in time

To determine the functions T, it is just a ODE which needs to be solved,

$$(a_{ii}^u + a_{ii}^v)\frac{dT_i}{dt} + (b_{ii}^u + b_{ii}^v)T_i = d_i^u + d_i^v - \sum_{k=1}^{i-1}(a_{ik}^v + a_{ik}^u)\frac{dT_k}{dt} - \sum_{k=1}^{i-1}(b_{ik}^u + b_{ik}^v)T_k \tag{6.56}$$

6.6 Treatment of non-linearities for the Navier-Stokes equations

In the above sections, we have detailed the PGD formulation for the unsteady Stokes equations. We are going to examine now the modifications which have to included to treat the additional non-linear terms included in the Unsteady Navier-Stokes Equations.

These additional terms read for x and y momentum equations, respectively:

$$\begin{aligned} T_{NL}^u &= \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \\ T_{NL}^v &= \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \end{aligned} \tag{6.57}$$

The velocity components are decomposed into :

$$\begin{aligned} u(x,t) &\approx \sum_{k=1}^{i-1} X_k^u(x)T_k(t) + X_i^u(x)T_i(t) \\ v(x,t) &\approx \sum_{k=1}^{i-1} X_k^v(x)T_k(t) + X_i^v(x)T_i(t) \end{aligned} \tag{6.58}$$

Several proposals concerning the treatment of the non-linear terms in the framework of a PGD decomposition, have already been made, see([Pruliere et al., 2010, Ammar et al., 2010d]). We can mainly distinguish three different linearisations for the non-linear term:

$$u(x,t) \approx \sum_{k=1}^{i-1} X_k(x)T_k(t) + X_iT_i \tag{6.59}$$

For instance, to evaluate the non-linear term u^2, the three possibilities are :

1. The non-linear term is evaluated from the solution at the previous iteration u^{i-1}

$$(u(x,t))^2 \approx (u^{i-1})^2 = (\sum_{k=1}^{i-1} X_k(x)T_k(t))^2 \tag{6.60}$$

2. Another possibility lies in partially using the solution just computed within the non-linear solver iteration scheme:

$$(u(x,t))^2 \approx (\sum_{k=1}^{i-1} X_k(x)T_k(t)) \cdot (\sum_{k=1}^{i-1} X_k(x)T_k(t) + X_iT_i) = U \cdot u \tag{6.61}$$

3. A third possibility is a variant of the previous one that considers:

$$(u(x,t))^2 \approx (\sum_{k=1}^{i-1} X_k(x)T_k(t) + X_i^{p-1}T_i^{p-1}) \cdot (\sum_{k=1}^{i-1} X_k(x)T_k(t) + X_i^pT_i^p) \tag{6.62}$$

where p denotes the iteration of the non-linear solver used for computing the current enrichment functions $X_i(\vec{x})$ and $T_i(t)$.

In this work, we will write the 2nd and 3rd kinds of linearization for the treatment of the non-linear term which will presented in a unified way.

6.6.1 Linearization of the non-linear terms in the Navier-Stokes

Let us start with the PGD modal decomposition introduced into the non-linear convective terms u^2, uv present in the x momentum equation:

$$\begin{aligned} u^2 &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k + X_i^u T_i)\dot{(}\sum_{k=1}^{i-1} X_k^u T_k + X_i^u T_i) \\ &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k)(\sum_{l=1}^{i-1} X_l^u T_l) + 2\sum_{k=1}^{i-1} X_k^u T_k X_i^u T_i + (X_i^u T_i)^2 \\ uv &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k + X_i^u T_i)\dot{(}\sum_{k=1}^{i-1} X_k^v T_k + X_i^v T_i) \\ &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k)(\sum_{l=1}^{i-1} X_l^v T_l) + [\sum_{k=1}^{i-1} X_k^u T_k X_i^v T_i] + [\sum_{k=1}^{i-1} X_k^v T_k X_i^u T_i] + X_i^u X_i^v (T_i)^2 \end{aligned} \tag{6.63}$$

Since:

$$(\sum_{k=1}^{i-1} a_k)(\sum_{l=1}^{i-1} b_l) = \sum_{k=1}^{i-1}\sum_{l=1}^{i-1} a_k b_l \tag{6.64}$$

we get:

$$\begin{aligned} u^2 &= \textstyle\sum_{k=1}^{i-1}\sum_{l=1}^{i-1} X_k^u X_l^u T_k T_l + 2\sum_{k=1}^{i-1} X_k^u T_k X_i^u T_i + (X_i^u T_i)^2 \\ uv &= \textstyle\sum_{k=1}^{i-1}\sum_{l=1}^{i-1} X_k^u X_l^v T_k T_l + \sum_{k=1}^{i-1} X_k^u T_k X_i^v T_i + \sum_{k=1}^{i-1} X_k^v T_k X_i^u T_i + X_i^u X_i^v (T_i)^2 \end{aligned} \tag{6.65}$$

and:

$$\begin{aligned} T_{NL}^u &= \textstyle\sum_{k=1}^{i-1}\sum_{l=1}^{i-1}[\frac{\partial}{\partial x}(X_k^u X_l^u) + \frac{\partial}{\partial y}(X_k^u X_l^v)]T_k T_l \\ &+ \textstyle\sum_{k=1}^{i-1}[\frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial y}(X_k^u X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^u)]T_k T_i \\ &+ [\tfrac{\partial}{\partial x}(X_i^u)^2 + \tfrac{\partial}{\partial y}(X_i^u X_i^v)]T_i^2 \end{aligned} \tag{6.66}$$

Similarly, we can compute the non-linear convective term for the y momentum equation:

$$\begin{aligned} T_{NL}^v &= \textstyle\sum_{k=1}^{i-1}\sum_{l=1}^{i-1}[\frac{\partial}{\partial x}(X_k^u X_l^v) + \frac{\partial}{\partial y}(X_k^v X_l^v)]T_k T_l \\ &+ \textstyle\sum_{k=1}^{i-1}[\frac{\partial}{\partial x}(X_k^u X_i^v) + \frac{\partial}{\partial x}(X_k^v X_i^u) + \frac{\partial}{\partial y}(X_k^v X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^v)]T_k T_i \\ &+ [\tfrac{\partial}{\partial x}(X_i^u X_i^v) + \tfrac{\partial}{\partial y}(X_i^v)^2]T_i^2 \end{aligned} \tag{6.67}$$

6.6.2 PGD formulation for the linearization

We have now to include the non-linear terms in the x momentum equation

$$\begin{aligned}
&\int_0^{T_{max}} \int_\Omega (X_i^u T_i^* + X_i^{u*} T_i) \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] d\overline{x} dt = \\
&\int_0^{T_{max}} \int_\Omega (X_i^u T_i^* + X_i^{u*} T_i) \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} (\frac{\partial}{\partial x}(X_k^u X_l^u) + \frac{\partial}{\partial y}(X_k^u X_l^v)) T_k T_l \\
&+ \sum_{k=1}^{i-1} (\frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial y}(X_k^u X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^u)) T_k T_i \\
&+ (\frac{\partial}{\partial x}(X_i^u X_i^u) + \frac{\partial}{\partial y}(X_i^v X_i^u)) T_i^2 \Big] d\overline{x} dt
\end{aligned} \tag{6.68}$$

and in the y momentum equation:

$$\begin{aligned}
&\int_0^{T_{max}} \int_\Omega (X_i^v T_i^* + X_i^{v*} T_i) \left[\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \right] d\overline{x} dt = \\
&\int_0^{T_{max}} \int_\Omega (X_i^v T_i^* + X_i^{v*} T_i) \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_l^v) + \frac{\partial}{\partial y}(X_k^v X_l^v)] T_k T_l \\
&+ \sum_{k=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_i^v) + \frac{\partial}{\partial x}(X_k^v X_i^u) + \frac{\partial}{\partial y}(X_k^v X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^v)] T_k T_i \\
&+ (\frac{\partial}{\partial x}(X_i^u X_i^v) + \frac{\partial}{\partial y}(X_i^v X_i^v) T_i^2 \Big] d\overline{x} dt
\end{aligned} \tag{6.69}$$

6.6.2.1 Determination of the spatial modes X_i^u, X_i^v

We suppose that T_i is known, which implies that $T_i^* = 0$ and leads to the following simplified weak formulation for these non-linear terms:

$$\begin{aligned}
&\int_0^{T_{max}} \int_\Omega X_i^{u*} T_i \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right] d\overline{x} dt = \\
&\int_0^{T_{max}} \int_\Omega X_i^{u*} \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} (\frac{\partial}{\partial x}(X_k^u X_l^u) + \frac{\partial}{\partial y}(X_k^u X_l^v)) T_k T_l T_i \\
&+ \sum_{k=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial y}(X_k^u X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^u)] T_k T_i^2 \\
&+ (\frac{\partial}{\partial x}(X_i^u X_i^u) + \frac{\partial}{\partial y}(X_i^v X_i^u)) T_i^3 \Big] d\overline{x} dt
\end{aligned} \tag{6.70}$$

and for the y momentum equation:

$$\begin{aligned}
&\int_0^{T_{max}} \int_\Omega X_i^{v*} T_i \left[\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \right] d\overline{x} dt = \\
&\int_0^{T_{max}} \int_\Omega X_i^{v*} \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_l^v) + \frac{\partial}{\partial y}(X_k^v X_l^v)] T_k T_l T_i \\
&+ \sum_{k=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_i^v) + \frac{\partial}{\partial x}(X_k^v X_i^u) + \frac{\partial}{\partial y}(X_k^v X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^v)] T_k T_i^2 \\
&+ (\frac{\partial}{\partial x}(X_i^u X_i^v) + \frac{\partial}{\partial y}(X_i^v X_i^v) T_i^3 \Big] d\overline{x} dt
\end{aligned} \tag{6.71}$$

This leads to the strong form of the non-linear terms to be added to the x and y

momentum equations :

$$\begin{array}{l}
\sum_{k=1}^{i-1}\sum_{l=1}^{i-1}\gamma_{ikl}(\frac{\partial}{\partial x}(X_k^uX_l^u)+\frac{\partial}{\partial y}(X_k^uX_l^v))\\
+\sum_{k=1}^{i-1}\gamma_{iik}(\frac{\partial}{\partial x}(X_k^uX_i^u)+\frac{\partial}{\partial x}(X_k^uX_i^u)+\frac{\partial}{\partial y}(X_k^uX_i^v)+\frac{\partial}{\partial y}(X_k^vX_i^u))\\
+\gamma_{iii}(\frac{\partial}{\partial x}(X_i^uX_i^u)+\frac{\partial}{\partial y}(X_i^vX_i^u))\\
\sum_{k=1}^{i-1}\sum_{l=1}^{i-1}\gamma_{ikl}(\frac{\partial}{\partial x}(X_k^uX_l^v)+\frac{\partial}{\partial y}(X_k^vX_l^v))\\
+\sum_{k=1}^{i-1}\gamma_{iik}(\frac{\partial}{\partial x}(X_k^uX_i^v)+\frac{\partial}{\partial x}(X_k^vX_i^u)+\frac{\partial}{\partial y}(X_k^vX_i^v)+\frac{\partial}{\partial y}(X_k^vX_i^v))\\
+\gamma_{iii}(\frac{\partial}{\partial x}(X_i^uX_i^v)+\frac{\partial}{\partial y}(X_i^vX_i^v))
\end{array} \tag{6.72}$$

with:

$$\gamma_{ikl}=\int_0^{T_{max}}T_iT_kT_ldt \tag{6.73}$$

We get finally the PGD formulation for the spatial momentum equation along x:

$$\begin{array}{l}
\alpha_{ii}X_i^u-\beta_{ii}[\frac{\partial}{\partial x}(\nu\frac{\partial X_i^u}{\partial x})+\frac{\partial}{\partial y}(\nu\frac{\partial X_i^u}{\partial y})-\frac{\partial X_i^p}{\partial x}]+\gamma_{iii}[\frac{\partial}{\partial x}(X_i^uX_i^u)+\frac{\partial}{\partial y}(X_i^vX_i^u)]\\
+\sum_{k=1}^{i-1}\gamma_{iik}[\frac{\partial}{\partial x}(X_k^uX_i^u)+\frac{\partial}{\partial x}(X_k^uX_i^u)+\frac{\partial}{\partial y}(X_k^vX_i^u)+\frac{\partial}{\partial y}(X_k^uX_i^v)]=\delta_i^u\\
-\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^u+\sum_{l=1}^{i-1}\gamma_{ikl}[\frac{\partial}{\partial x}(X_k^uX_l^u)+\frac{\partial}{\partial y}(X_k^vX_l^u)]-\beta_{ik}[\frac{\partial}{\partial x}(\nu\frac{\partial X_k^u}{\partial x})+\frac{\partial}{\partial y}(\nu\frac{\partial X_k^u}{\partial y})-\frac{\partial X_k^p}{\partial x}]\}
\end{array} \tag{6.74}$$

By analogy, it is possible to write the modified spatial momentum equation along y:

$$\begin{array}{l}
\alpha_{ii}X_i^v-\beta_{ii}[\frac{\partial}{\partial x}(\nu\frac{\partial X_i^v}{\partial x})+\frac{\partial}{\partial y}(\nu\frac{\partial X_i^v}{\partial y})-\frac{\partial X_i^p}{\partial y}]+\gamma_{iii}[\frac{\partial}{\partial x}(X_i^uX_i^v)+\frac{\partial}{\partial y}(X_i^vX_i^v)]\\
+\sum_{k=1}^{i-1}\gamma_{iik}[\frac{\partial}{\partial x}(X_k^uX_i^v)+\frac{\partial}{\partial x}(X_k^vX_i^u)+\frac{\partial}{\partial y}(X_k^vX_i^v)+\frac{\partial}{\partial y}(X_k^vX_i^v)]=\delta_i^v\\
-\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^v+\sum_{l=1}^{i-1}\gamma_{ikl}[\frac{\partial}{\partial x}(X_k^uX_l^v)+\frac{\partial}{\partial y}(X_k^vX_l^v)]-\beta_{ik}[\frac{\partial}{\partial x}(\nu\frac{\partial X_k^v}{\partial x})+\frac{\partial}{\partial y}(\nu\frac{\partial X_k^v}{\partial y})-\frac{\partial X_k^p}{\partial y}]\}
\end{array} \tag{6.75}$$

with:

$$\begin{array}{rcl}
\delta_i^u(\overline{x}) & = & \int_0^{T_{max}}T_if^u(\overline{x},t)dt\\
\delta_i^v(\overline{x}) & = & \int_0^{T_{max}}T_if^v(\overline{x},t)dt
\end{array} \tag{6.76}$$

If we introduce the new PGD convective velocities U_{ik}^{cv} whose components are defined by:

$$\begin{array}{rcl}
U_{ik}^{cv} & = & \sum_{l=1}^{i}\gamma_{ikl}X_l^u=\sum_{l=1}^{i}\gamma_{ilk}X_l^u\\
V_{ik}^{cv} & = & \sum_{l=1}^{i}\gamma_{ikl}X_l^v=\sum_{l=1}^{i}\gamma_{ilk}X_l^v
\end{array} \tag{6.77}$$

We can rewrite the PGD spatial momentum equations. Along the direction x, we have to solve:

$$\begin{aligned}
&\alpha_{ii}X_i^u - \beta_{ii}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_i^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_i^u}{\partial y}) - \tfrac{\partial X_i^p}{\partial x}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv}X_i^u) + \tfrac{\partial}{\partial y}(V_{ii}^{cv}X_i^u) = \delta_i^u \\
&-\textstyle\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^u + \sum_{l=1}^{i-1}\gamma_{ikl}[\tfrac{\partial}{\partial x}(X_k^uX_l^u) + \tfrac{\partial}{\partial y}(X_k^uX_l^v)] + \gamma_{iik}[\tfrac{\partial}{\partial x}(X_k^uX_i^u) + \tfrac{\partial}{\partial y}(X_k^uX_i^v)] \\
&-\beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^u}{\partial y}) - \tfrac{\partial X_k^p}{\partial x}]\}
\end{aligned} \tag{6.78}$$

and:

$$\begin{aligned}
&\alpha_{ii}X_i^u - \beta_{ii}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_i^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_i^u}{\partial y}) - \tfrac{\partial X_i^p}{\partial x}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv}X_i^u) + \tfrac{\partial}{\partial y}(V_{ii}^{cv}X_i^u) = \delta_i^u \\
&-\textstyle\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^u + \tfrac{\partial}{\partial x}(X_k^u\sum_{l=1}^{i-1}\gamma_{ikl}X_l^u) + \tfrac{\partial}{\partial y}(X_k^u\sum_{l=1}^{i-1}\gamma_{ikl}X_l^v) + \tfrac{\partial}{\partial x}(\gamma_{iki}X_k^uX_i^u) + \tfrac{\partial}{\partial y}(\gamma_{iki}X_k^uX_i^v) \\
&-\beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^u}{\partial y}) - \tfrac{\partial X_k^p}{\partial x}]\}
\end{aligned} \tag{6.79}$$

then:

$$\begin{aligned}
&\alpha_{ii}X_i^u - \beta_{ii}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_i^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_i^u}{\partial y}) - \tfrac{\partial X_i^p}{\partial x}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv}X_i^u) + \tfrac{\partial}{\partial y}(V_{ii}^{cv}X_i^u) = \delta_i^u \\
&-\textstyle\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^u + \tfrac{\partial}{\partial x}(U_{ik}^{cv}X_k^u) + \tfrac{\partial}{\partial y}(V_{ik}^{cv}X_k^u) - \beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^u}{\partial y}) - \tfrac{\partial X_k^p}{\partial x}]\}
\end{aligned} \tag{6.80}$$

and the spatial momentum equation along y:

$$\begin{aligned}
&\alpha_{ii}X_i^v - \beta_{ii}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_i^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_i^v}{\partial y}) - \tfrac{\partial X_i^p}{\partial y}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv}X_i^v) + \tfrac{\partial}{\partial y}(V_{ii}^{cv}X_i^v) = \delta_i^v \\
&-\textstyle\sum_{k=1}^{i-1}\{\alpha_{ik}X_k^v + \tfrac{\partial}{\partial x}(U_{ik}^{cv}X_k^v) + \tfrac{\partial}{\partial y}(V_{ik}^{cv}X_k^v) - \beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^v}{\partial y}) - \tfrac{\partial X_k^p}{\partial y}]\}
\end{aligned} \tag{6.81}$$

This leads to a more condensed form of the two coupled spatial equations:

$$\begin{aligned}
&\textstyle\sum_{k=1}^{i}\{\alpha_{ik}X_k^u + \tfrac{\partial}{\partial x}(U_{ik}^{cv}X_k^u) + \tfrac{\partial}{\partial y}(V_{ik}^{cv}X_k^u) - \beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^u}{\partial y}) - \tfrac{\partial X_k^p}{\partial x}]\} = \delta_i^u \\
&\textstyle\sum_{k=1}^{i}\{\alpha_{ik}X_k^v + \tfrac{\partial}{\partial x}(U_{ik}^{cv}X_k^v) + \tfrac{\partial}{\partial y}(V_{ik}^{cv}X_k^v) - \beta_{ik}[\tfrac{\partial}{\partial x}(\nu\tfrac{\partial X_k^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu\tfrac{\partial X_k^v}{\partial y}) - \tfrac{\partial X_k^p}{\partial y}]\} = \delta_i^v
\end{aligned} \tag{6.82}$$

Strategies of linearization of the PGD momentum equations

Since the determination of the spatial modes X_i^u and X_i^v is performed inside a fixed point iterative algorithm, we can linearize the momentum equations in such a way that the linearized partial differential operator is identical for the x momentum equation (used to determine X_i^u) and for the y momentum equation (used to determine X_i^v).

This means that the PGD convective velocities will be linearized using a Picard linearization:

$$\begin{aligned} U_{ik}^{cv(p-1)} &= \textstyle\sum_{l=1}^{i-1} \gamma_{ikl} X_l^u + \gamma_{iki} (X_i^u)^{(p-1)} \\ V_{ik}^{cv(p-1)} &= \textstyle\sum_{l=1}^{i-1} \gamma_{ikl} X_l^v + \gamma_{iki} (X_i^v)^{(p-1)} \end{aligned} \tag{6.83}$$

which leads to the final linearized PGD momentum equations.

Along x:

$$\begin{aligned} &\alpha_{ii} X_i^u - \beta_{ii} [\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^u}{\partial y}) - \tfrac{\partial X_i^p}{\partial x}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv(p-1)} X_i^u) + \tfrac{\partial}{\partial y}(V_{ii}^{cv(p-1)} X_i^u) = \delta_i^u \\ &- \textstyle\sum_{k=1}^{i-1} \{\alpha_{ik} X_k^u + \tfrac{\partial}{\partial x}(U_{ik}^{cv(p-1)} X_k^u) + \tfrac{\partial}{\partial y}(V_{ik}^{cv(p-1)} X_k^u) - \beta_{ik} [\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^u}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^u}{\partial y}) - \tfrac{\partial X_k^p}{\partial x}]\} \end{aligned} \tag{6.84}$$

And along y:

$$\begin{aligned} &\alpha_{ii} X_i^v - \beta_{ii} [\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_i^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_i^v}{\partial y}) - \tfrac{\partial X_i^p}{\partial y}] + \tfrac{\partial}{\partial x}(U_{ii}^{cv(p-1)} X_i^v) + \tfrac{\partial}{\partial y}(V_{ii}^{cv(p-1)} X_i^v) = \delta_i^v \\ &- \textstyle\sum_{k=1}^{i-1} \{\alpha_{ik} X_k^v + \tfrac{\partial}{\partial x}(U_{ik}^{cv(p-1)} X_k^v) + \tfrac{\partial}{\partial y}(V_{ik}^{cv(p-1)} X_k^v) - \beta_{ik} [\tfrac{\partial}{\partial x}(\nu \tfrac{\partial X_k^v}{\partial x}) + \tfrac{\partial}{\partial y}(\nu \tfrac{\partial X_k^v}{\partial y}) - \tfrac{\partial X_k^p}{\partial y}]\} \end{aligned} \tag{6.85}$$

6.6.2.2 Determination of the temporal mode T_i

The temporal ordinary differential equation which has to be satisfied by T_i is of course modified by the introduction of non-linear terms. We suppose that X_i^u and X_i^v are known, which implies that $X_i^{u*} = 0$ and $X_i^{v*} = 0$ leads to the following simplified weak formulations for these non-linear terms for the x momentum equation:

$$\begin{aligned} &\textstyle\int_\Omega X_i^u T_i^* \left[\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y}\right] d\overline{x} = \\ &\textstyle\int_\Omega T_i^* \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} (\frac{\partial}{\partial x}(X_k^u X_l^u) + \frac{\partial}{\partial y}(X_k^v X_l^u)) X_i^u T_k T_l \\ &+ \textstyle\sum_{k=1}^{i-1} X_i^u (\frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial x}(X_k^u X_i^u) + \frac{\partial}{\partial y}(X_k^u X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^u)) T_k T_i \\ &+ X_i^u (\tfrac{\partial}{\partial x}(X_i^u X_i^u) + \tfrac{\partial}{\partial y}(X_i^v X_i^u)) T_i^2 \Big] d\overline{x} \end{aligned} \tag{6.86}$$

and to:

$$\begin{aligned} &\textstyle\int_\Omega X_i^v T_i^* \left[\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y}\right] d\overline{x} = \\ &\textstyle\int_\Omega T_i^* \Big[\sum_{k=1}^{i-1} \sum_{l=1}^{i-1} [\frac{\partial}{\partial x}(X_k^u X_l^v) + \frac{\partial}{\partial y}(X_k^v X_l^v)] X_i^v T_k T_l \\ &+ \textstyle\sum_{k=1}^{i-1} X_i^v (\frac{\partial}{\partial x}(X_k^u X_i^v) + \frac{\partial}{\partial x}(X_k^v X_i^u) + \frac{\partial}{\partial y}(X_k^v X_i^v) + \frac{\partial}{\partial y}(X_k^v X_i^v)) T_k T_i \\ &+ X_i^v (\tfrac{\partial}{\partial x}(X_i^u X_i^v) + \tfrac{\partial}{\partial y}(X_i^v X_i^v) T_i^2 \Big] d\overline{x} \end{aligned} \tag{6.87}$$

for the y momentum equation.

We get finally the additional terms for the x momentum equations:

$$\sum_{k=1}^{i-1}\sum_{l=1}^{i-1} c^u_{ikl} T_k T_l + \sum_{k=1}^{i-1} e^u_{ik} T_k T_i + c^u_{iii} T_i^2 \tag{6.88}$$

with:

$$\begin{aligned} c^u_{ikl} &= \int_\Omega X^u_i \left(\frac{\partial}{\partial x}(X^u_k X^u_l) + \frac{\partial}{\partial y}(X^v_k X^u_l)\right) d\overline{x} \\ e^u_{ik} &= \int_\Omega X^u_i \left(\frac{\partial}{\partial x}(X^u_k X^u_i) + \frac{\partial}{\partial x}(X^u_k X^u_i) + \frac{\partial}{\partial y}(X^u_k X^v_i) + \frac{\partial}{\partial y}(X^v_k X^u_i)\right) d\overline{x} \\ &= c^u_{iki} + c^u_{iik} \end{aligned} \tag{6.89}$$

and for the y momentum equation:

$$\sum_{k=1}^{i-1}\sum_{l=1}^{i-1} c^v_{ikl} T_k T_l + \sum_{k=1}^{i-1} e^v_{ik} T_k T_i + c^v_{iii} T_i^2 \tag{6.90}$$

with:

$$\begin{aligned} c^v_{ikl} &= \int_\Omega X^v_i \left(\frac{\partial}{\partial x}(X^u_k X^v_l) + \frac{\partial}{\partial y}(X^v_k X^v_l)\right) d\overline{x} \\ e^v_{ik} &= \int_\Omega X^v_i \left(\frac{\partial}{\partial x}(X^u_k X^v_i) + \frac{\partial}{\partial x}(X^v_k X^u_i) + \frac{\partial}{\partial y}(X^v_k X^v_i) + \frac{\partial}{\partial y}(X^v_k X^v_i)\right) d\overline{x} \\ &= c^v_{iki} + c^v_{iik} \end{aligned} \tag{6.91}$$

This leads to the new temporal equations:

$$\begin{aligned} a^u_{ii}\frac{dT_i}{dt} + (b^u_{ii} + \textstyle\sum_{k=1}^{i-1}(c^u_{iki} + c^u_{iik})T_k)T_i + c^u_{iii}T_i^2 &= d^{u*}_i \\ a^v_{ii}\frac{dT_i}{dt} + (b^v_{ii} + \textstyle\sum_{k=1}^{i-1}(c^v_{iki} + c^v_{iik})T_k)T_i + c^v_{iii}T_i^2 &= d^{v*}_i \end{aligned} \tag{6.92}$$

with:

$$\begin{aligned} d^{u*}_i &= d^u_i - \textstyle\sum_{k=1}^{i-1} a^u_{ik}\frac{dT_k}{dt} - \sum_{k=1}^{i-1}[b^u_{ik} + \sum_{l=1}^{i-1} c^u_{ikl}T_l]T_k \\ d^{v*}_i &= d^v_i - \textstyle\sum_{k=1}^{i-1} a^v_{ik}\frac{dT_k}{dt} - \sum_{k=1}^{i-1}[b^v_{ik} + \sum_{l=1}^{i-1} c^v_{ikl}T_l]T_k \end{aligned} \tag{6.93}$$

which, once linearized with a Picard linearization, become:

$$\begin{aligned} a^u_{ii}\frac{dT_i^{(p)}}{dt} + [b^u_{ii} + \textstyle\sum_{k=1}^{i-1}(c^u_{iki} + c^u_{iik})T_k + c^u_{iii}T_i^{(p-1)}]T_i^{(p)} &= d^{u*}_i \\ a^v_{ii}\frac{dT_i^{(p)}}{dt} + [b^v_{ii} + \textstyle\sum_{k=1}^{i-1}(c^v_{iki} + c^v_{iik})T_k + c^v_{iii}T_i^{(p-1)}]T_i^{(p)} &= d^{v*}_i \end{aligned} \tag{6.94}$$

It is interesting to notice that one can reformulate the temporal equations as the

spatial equations by noticing that :

$$\begin{aligned} a^u_{ii}\frac{dT_i^{(p)}}{dt} + b^u_{ii}T_i + \sum_{k=1}^{i-1} c^u_{iik}T_kT_i + c^u_{iii}T_iT_i &= d^u_i \\ &- \sum_{k=1}^{i-1} a^u_{ik}\frac{dT_k}{dt} - \sum_{k=1}^{i-1} b^u_{ik}T_k \\ &- \sum_{k=1}^{i-1}\sum_{l=1}^{i-1} c^u_{ikl}T_lT_k - \sum_{k=1}^{i-1} c^u_{iki}T_kT_i \end{aligned} \tag{6.95}$$

which leads to :

$$\begin{aligned} a^u_{ii}\frac{dT_i^{(p)}}{dt} + b^u_{ii}T_i + \sum_{k=1}^{i} c^u_{iik}T_kT_i &= d^u_i \\ &- \sum_{k=1}^{i-1} a^u_{ik}\frac{dT_k}{dt} - \sum_{k=1}^{i-1} b^u_{ik}T_k \\ &- \sum_{k=1}^{i-1}\sum_{l=1}^{i} c^u_{ikl}T_lT_k \end{aligned} \tag{6.96}$$

This leads to the more condensed form for the two temporal equations similar to the expressions we got for the two spatial equations:

$$\begin{aligned} \sum_{k=1}^{i} a^u_{ik}\frac{dT_k}{dt} + \sum_{k=1}^{i} b^u_{ik}T_k + \sum_{k=1}^{i}\sum_{l=1}^{i} c^u_{ikl}T_lT_k &= d^u_i \\ \sum_{k=1}^{i} a^v_{ik}\frac{dT_k}{dt} + \sum_{k=1}^{i} b^v_{ik}T_k + \sum_{k=1}^{i}\sum_{l=1}^{i} c^v_{ikl}T_lT_k &= d^v_i \end{aligned} \tag{6.97}$$

Finally, we can give a synthesis of the PGD formulation for solving the unsteady Navier-Stokes equations. It consists in solving the following non-linear spatial and temporal equations coupled together by the definition of their coefficients:

$$
\begin{aligned}
&\textstyle\sum_{k=1}^{i}\{\alpha_{ik}X_k^u + \frac{\partial}{\partial x}(U_{ik}^{cv}X_k^u) + \frac{\partial}{\partial y}(V_{ik}^{cv}X_k^u) - \beta_{ik}[\frac{\partial}{\partial x}(\nu\frac{\partial X_k^u}{\partial x}) + \frac{\partial}{\partial y}(\nu\frac{\partial X_k^u}{\partial y}) - \frac{\partial X_k^p}{\partial x}]\} = \delta_i^u \\
&\textstyle\sum_{k=1}^{i}\{\alpha_{ik}X_k^v + \frac{\partial}{\partial x}(U_{ik}^{cv}X_k^v) + \frac{\partial}{\partial y}(V_{ik}^{cv}X_k^v) - \beta_{ik}[\frac{\partial}{\partial x}(\nu\frac{\partial X_k^v}{\partial x}) + \frac{\partial}{\partial y}(\nu\frac{\partial X_k^v}{\partial y}) - \frac{\partial X_k^p}{\partial y}]\} = \delta_i^v \\
&\textstyle\sum_{k=1}^{i} a_{ik}^u\frac{dT_k}{dt} + \sum_{k=1}^{i} b_{ik}^u T_k + \sum_{k=1}^{i}\sum_{l=1}^{i} c_{ikl}^u T_l T_k = d_i^u \\
&\textstyle\sum_{k=1}^{i} a_{ik}^v\frac{dT_k}{dt} + \sum_{k=1}^{i} b_{ik}^v T_k + \sum_{k=1}^{i}\sum_{l=1}^{i} c_{ikl}^v T_l T_k = d_i^v \\
&U_{ik}^{cv(p-1)} = \textstyle\sum_{l=1}^{i-1}\gamma_{ikl}X_l^u + \gamma_{iki}(X_i^u)^{(p-1)} \\
&V_{ik}^{cv(p-1)} = \textstyle\sum_{l=1}^{i-1}\gamma_{ikl}X_l^v + \gamma_{iki}(X_i^v)^{(p-1)} \\
&\alpha_{ik} = \textstyle\int_0^{T_{max}} T_i\frac{dT_k}{dt}dt \\
&\beta_{ik} = \textstyle\int_0^{T_{max}} T_i T_k dt \\
&\gamma_{ikl} = \textstyle\int_0^{T_{max}} T_i T_k T_l dt \\
&\delta_i^u(\overline{x}) = \textstyle\int_0^{T_{max}} T_i f^u(\overline{x},t)dt \\
&\delta_i^v(\overline{x}) = \textstyle\int_0^{T_{max}} T_i f^v(\overline{x},t)dt \\
&a_{ik}^u = \textstyle\int_\Omega X_i^u X_k^u d\overline{x} \\
&b_{ik}^u = \textstyle\int_\Omega X_i^u\left[\frac{\partial}{\partial x}(U_c X_k^u) + \frac{\partial}{\partial y}(V_c X_k^u) - \frac{\partial}{\partial x}(\nu\frac{\partial X_k^u}{\partial x}) - \frac{\partial}{\partial y}(\nu\frac{\partial X_k^u}{\partial y}) + \frac{\partial X_k^p}{\partial x}\right]d\overline{x} \\
&c_{ikl}^u = \textstyle\int_\Omega X_i^u(\frac{\partial}{\partial x}(X_k^u X_l^u) + \frac{\partial}{\partial y}(X_k^v X_l^u))d\overline{x} \\
&d_i^u = \textstyle\int_\Omega X_i^u f^u(\overline{x},t)d\overline{x} \\
&a_{ik}^v = \textstyle\int_\Omega X_i^v X_k^v d\overline{x} \\
&b_{ik}^v = \textstyle\int_\Omega X_i^v\left[\frac{\partial}{\partial x}(U_c X_k^v) + \frac{\partial}{\partial y}(V_c X_k^v) - \frac{\partial}{\partial x}(\nu\frac{\partial X_k^v}{\partial x}) - \frac{\partial}{\partial y}(\nu\frac{\partial X_k^v}{\partial y}) + \frac{\partial X_k^p}{\partial y}\right]d\overline{x} \\
&c_{ikl}^v = \textstyle\int_\Omega X_i^v(\frac{\partial}{\partial x}(X_k^u X_l^v) + \frac{\partial}{\partial y}(X_k^v X_l^v))d\overline{x} \\
&d_i^v = \textstyle\int_\Omega X_i^v f^v(\overline{x},t)d\overline{x}
\end{aligned}
\tag{6.98}
$$

Linearization strategies for determining the temporal mode T_i

If one uses a Picard linearization for the non-linear terms, the previous equations become:

$$
\begin{aligned}
a_{ii}^u\frac{dT_i^{(p)}}{dt} + [b_{ii}^u + \textstyle\sum_{k=1}^{i-1}(c_{iki}^u + c_{iik}^u)T_k + c_{iii}^u T_i^{(p-1)}]T_i^{(p)} &= d_i^{u*} \\
a_{ii}^v\frac{dT_i^{(p)}}{dt} + [b_{ii}^v + \textstyle\sum_{k=1}^{i-1}(c_{iki}^v + c_{iik}^v)T_k + c_{iii}^v T_i^{(p-1)}]T_i^{(p)} &= d_i^{v*}
\end{aligned}
\tag{6.99}
$$

As observed for the Stokes problem, we will use the sum of these two equations as the unique ODE to determine the temporal function T_i:

$$(a^u_{ii}+a^v_{ii})\frac{dT_i^{(p)}}{dt}+[b^u_{ii}+b^v_{ii}+\sum_{k=1}^{i-1}(c^u_{iki}+c^v_{iki}+c^u_{iik}+c^v_{iik})T_k+(c^u_{iii}+c^v_{iii})T_i^{(p-1)}]T_i^{(p)} = d_i^{u*}+d_i^{v*} \tag{6.100}$$

However, in the case of the Navier-Stokes equations, it is no more possible to ensure the unconditional positivity of the coefficient multiplying $T_i^{(p)}$.
We can also use a Newton-Raphson linearisation based on :

$$T_i^2 \approx 2T_i^{(p)}T_i^{(p-1)} - (T_i^{(p-1)})^2 \tag{6.101}$$

which leads to a more balanced formulation:

$$\begin{array}{rcl} a^u_{ii}\frac{dT_i^{(p)}}{dt} + [b^u_{ii} + \sum_{k=1}^{i-1}(c^u_{iki} + c^u_{iik})T_k + 2c^u_{iii}T_i^{(p-1)}]T_i^{(p)} & = & d_i^{u*} + c^u_{iii}(T_i^{(p-1)})^2 \\ a^v_{ii}\frac{dT_i^{(p)}}{dt} + [b^v_{ii} + \sum_{k=1}^{i-1}(c^v_{iki} + c^v_{iik})T_k + 2c^v_{iii}T_i^{(p-1)}]T_i^{(p)} & = & d_i^{v*} + c^v_{iii}(T_i^{(p-1)})^2 \end{array} \tag{6.102}$$

As previously, we will use again the sum of these two equations to determine the temporal mode T_i:

$$\begin{array}{rcl} (a^u_{ii} + a^v_{ii})\frac{dT_i^{(p)}}{dt} & + & [b^u_{ii} + b^v_{ii} + \sum_{k=1}^{i-1}(c^u_{iki} + c^v_{iki} + c^u_{iik} + c^v_{iik})T_k + 2(c^u_{iii} + c^v_{iii})T_i^{(p-1)}]T_i^{(p)} \\ & = & d_i^{u*} + d_i^{v*} + (c^u_{iii} + c^v_{iii})(T_i^{(p-1)})^2 \end{array} \tag{6.103}$$

6.6.3 Simplifying more the linearization

The previous section described a linearization for which each current spatial and temporal obey the true non-linear equations, which should result in a reduced number of modes for a higher computational cost for determining each mode, because of the non-linear iterative loop needed to determine these modes. We can also design a less implicit and less expensive linearization by using the 2nd kind of linearisation described above, which means shifting the constraint for each individual mode to satisfy the non-linear equations, to the global solution built from successive enrichments. In that case, the determination of each mode will be less expensive but we will need more modes to satisfy the non-linear equations.

In that case, the linearization for the convective terms will be ass follows:

$$\begin{aligned} u^2 &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k)\dot{(}\sum_{k=1}^{i-1} X_k^u T_k + X_i^u T_i) \\ &= (\textstyle\sum_{k=1}^{i-1} X_k^u T_k)(\sum_{l=1}^{i-1} X_l^u T_l) + \sum_{k=1}^{i-1} X_k^u T_k X_i^u T_i \\ uv &= (\textstyle\sum_{k=1}^{i-1} X_k^v T_k)\dot{(}\sum_{k=1}^{i-1} X_k^u T_k + X_i^u T_i) \\ &= (\textstyle\sum_{k=1}^{i-1} X_k^v T_k)(\sum_{l=1}^{i-1} X_l^u T_l) + (\sum_{k=1}^{i-1} X_k^v T_k X_i^u T_i) \end{aligned} \tag{6.104}$$

6.6.3.1 Determination of the spatial modes X_i^u, X_i^v in the framework of a simplified linearization

Based on the previous developments conducted for the linearization, we can deduce quickly the strong forms of the PGD Navier-Stokes equations by just modifying the spatial PGD convective velocities as:

$$\begin{aligned} U_{ik}^{cv} &= \textstyle\sum_{l=1}^{i-1} \gamma_{ikl} X_l^u \\ V_{ik}^{cv} &= \textstyle\sum_{l=1}^{i-1} \gamma_{ikl} X_l^v \end{aligned} \tag{6.105}$$

6.6.3.2 Determination of the temporal mode T_i in the framework of a simplified linearization

In the framework of this second linearisation, the coefficients $c_{iik} = 0$ and $c_{iii} = 0$ in the Eq.(6.88) and Eq.(6.89)),

Then we can build two new ordinary temporal equations to be solved which will be linear:

$$\begin{aligned} a_{ii}^u \tfrac{dT_i}{dt} + (b_{ii}^u + \textstyle\sum_{k=1}^{i-1} c_{iki}^u T_k) T_i &= d_i^{u*} \\ a_{ii}^v \tfrac{dT_i}{dt} + (b_{ii}^v + \textstyle\sum_{k=1}^{i-1} c_{iki}^v T_k) T_i &= d_i^{v*} \end{aligned} \tag{6.106}$$

with:

$$\begin{aligned} d_i^{u*} &= d_i^u - \textstyle\sum_{k=1}^{i-1} a_{ik}^u \tfrac{dT_k}{dt} - \sum_{k=1}^{i-1} [b_{ik}^u - \sum_{k=1}^{i-1} \sum_{l=1}^{i-1} c_{ikl}^u T_l] T_k \\ d_i^{v*} &= d_i^v - \textstyle\sum_{k=1}^{i-1} a_{ik}^v \tfrac{dT_k}{dt} - \sum_{k=1}^{i-1} [b_{ik}^v - \sum_{k=1}^{i-1} \sum_{l=1}^{i-1} c_{ikl}^v T_l] T_k \end{aligned} \tag{6.107}$$

We will use the sum of these two equations as the unique ODE to determine the temporal function T_i:

$$(a_{ii}^u + a_{ii}^v)\frac{dT_i}{dt} + (b_{ii}^u + b_{ii}^v + \sum_{k=1}^{i-1}(c_{iki}^u + c_{iki}^v) T_k) T_i = d_i^{u*} + d_i^{v*} \tag{6.108}$$

However, in the case of the Navier-Stokes equations, it is no more possible to ensure the unconditional positivity of $b_{ii}^u + b_{ii}^v + \sum_{k=1}^{i-1}(c_{iki}^u + c_{iki}^v)T_k$.

6.7 Discussion and Conclusion

In this chapter, a SVD representation for the solution of the unsteady lid driven cavity problem computed with the ISIS-CFD solver was performed. Then, a comprehensive description of the PGD decomposition implemented into an unstructured incompressible finite volume solver (here ISIS-CFD) was described. This choice was based on a common temporal function for the velocities and pressure. The semi-discrete finite volume formulation for the PGD method for solving the Unsteady Stokes Equations was also given in this chapter and several linearizations of the non-linear terms for dealing with the Navier-Stokes equations were provided, leading to a complete formulation for the Navier-Stokes equations.

Chapter 7

Application of PGD for solving Unsteady Navier-Stokes equations

7.1 Introduction 141

7.2 Analytical flow problems 142

7.2.1 Unsteady Diffusion problem 142

7.2.2 Analytical Stokes problem 145

7.2.3 Burgers problem 149

7.3 Real flow problem 153

7.3.1 Stokes flow in a lid-driven cavity 153

7.3.2 Navier-Stokes flow in a lid-driven cavity 157

7.3.3 2D Couette flow 161

7.4 Discussion and Conclusion 163

7.1 Introduction

In this Chapter, we will use the PGD formulation which is described in the Chapter(6) for solving several numerical problems and assess the main characteristics of this new approach.

This chapter is thus organized as follows. First, we will use this method for solving some analytical flow problems, such as the unsteady diffusion problem, an analytical Stokes flow and the Burgers problem.

Then we will compute real viscous flows such as a 2D lid-driven cavity problem and 2D Couette flow between two concentric cylinders.

7.2 Analytical flow problems

This section will present results for analytical flow problems.

7.2.1 Unsteady Diffusion problem

The unsteady diffusion problem is defined by the Stokes equations without gradient of pressure. Consequently, the following equations will be used:

$$\begin{aligned} &\tfrac{\partial u}{\partial t} - \nu \Delta u = f, \quad on\ \partial\Omega \times [0,T] \\ &u(t=0) = u_0 \\ &u = g_D \quad on\ \partial\Omega \end{aligned} \tag{7.1}$$

These equations are defined over a 2D square domain $\Omega = (0,1) \times (0,1)$, the viscosity is $\nu = 0.01$, and the simulation time is $T = 1s$.

The boundary conditions g_D and the initial condition u_0 are specified from the analytical solution which is as follows:

$$\begin{aligned} u_1 &= cos(2\pi x) sin(2\pi y) e^{-t} \\ u_2 &= -sin(2\pi x) cos(2\pi y) e^{-t} \end{aligned} \tag{7.2}$$

From the PGD algorithm described in the Chapter 6, we get the PGD solution which are shown in the Fig.(7.1)-Fig.(7.3), respectively, for the different times $t = 0.01s$, $0.2s$ and $0.9s$. In each figure, the analytical solutions are shown in the top line for u_1 and u_2 and the error compared with the analytical solution in the bottom line of the figures. For each time step solution, we can see that the difference between the analytical solution and the PGD solution were not beyond the order 10^{-3}.

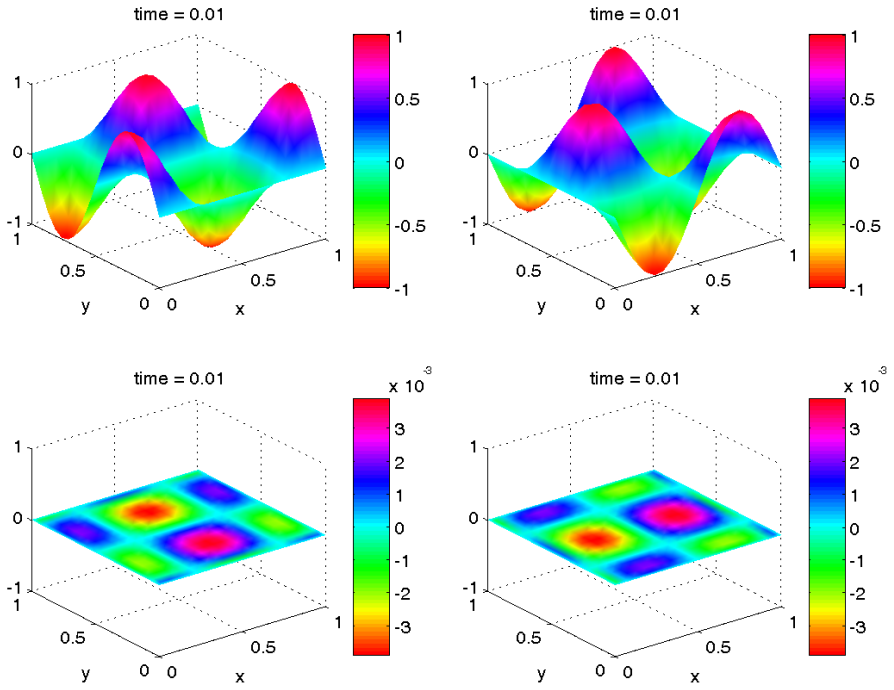


Figure 7.1: PGD solution (top) & error with analytical solution (bottom) at 0.01s

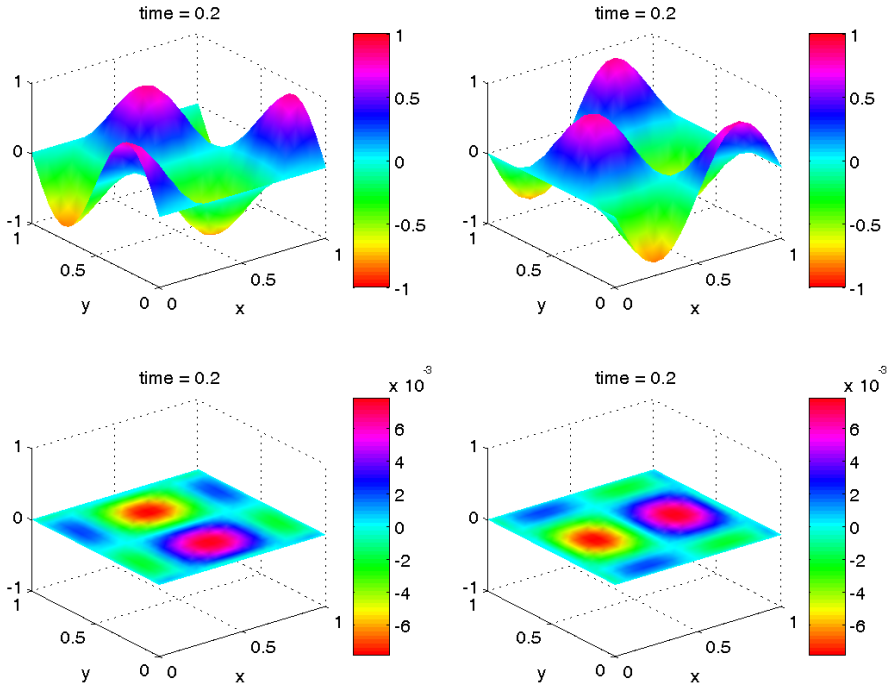


Figure 7.2: PGD solution (top) & error with analytical solution (bottom) at 0.2s

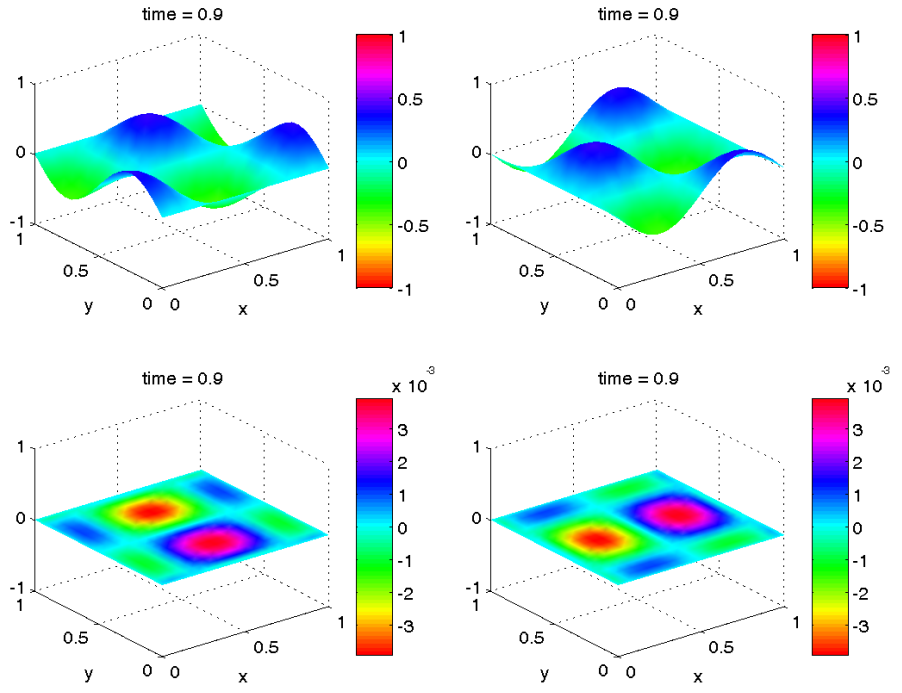
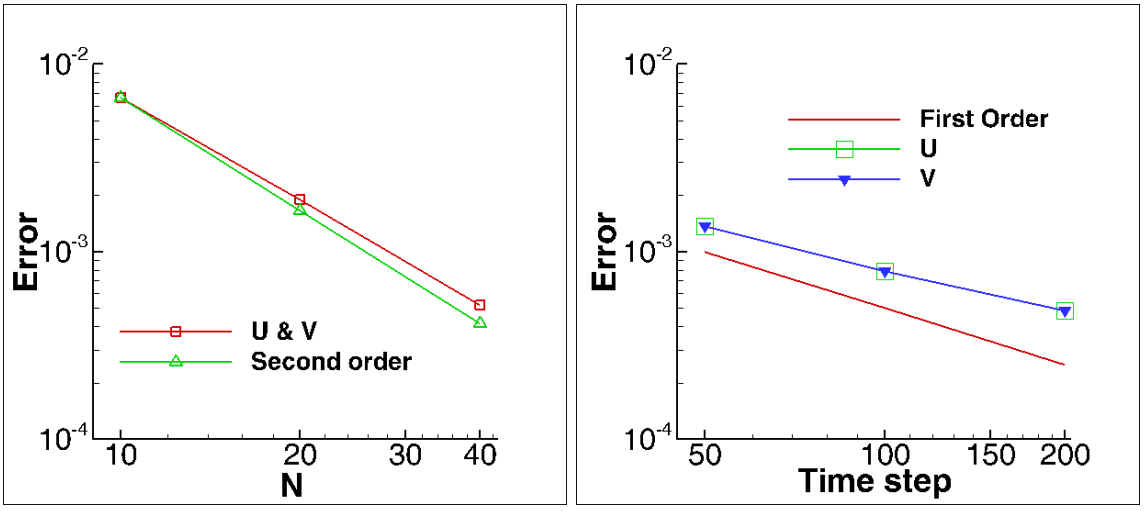


Figure 7.3: PGD solution (top) & error with analytical solution (bottom) at 0.9s



(a) Space

(b) Time

Figure 7.4: Space and time discretization effects for the diffusion problem

The error is defined in the Fig.(7.4) as used in [Cummins et al., 2005]:

$$L_2\ Error\ norm = \sqrt{\frac{\sum(u^i_{PGD} - u^i_{analy})^2}{N}} \tag{7.3}$$

where i stands for each nodes, and N is the total number of nodes for the discretization.

Influence on the error of the space-step size and time-step size for the PGD are shown in Fig.(7.4(a)) and Fig.(7.4(b)). For the space-step size, the second-order convergence was achieved for this problem, and the first-order convergence is achieved for the temporal discretisation. Finally, Fig.(7.5) shows the convergence of the Proper Generalized Decomposition for the unsteady diffusion problem.

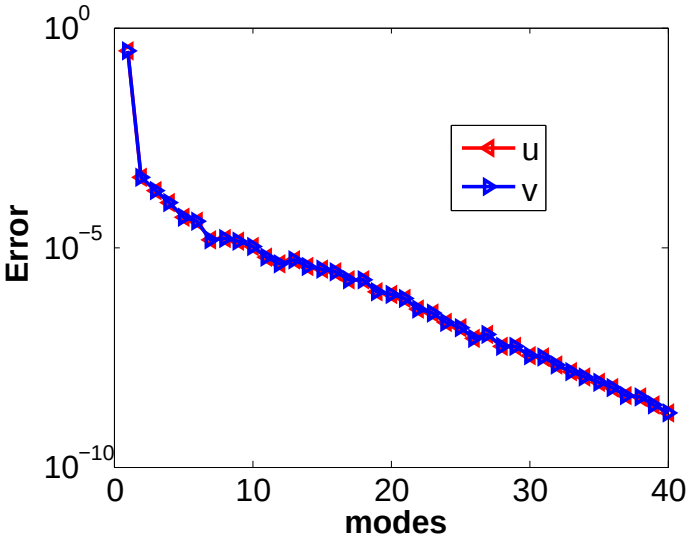


Figure 7.5: Convergence for the diffusion problem

7.2.2 Analytical Stokes problem

Now, let us consider the Stokes problem. The incompressible Stokes problem can be expressed as follows:

$$\begin{aligned} &\tfrac{\partial u}{\partial t} - \nu \Delta u = -\nabla p + f, \quad on\ \partial\Omega \times [0, T] \\ &\nabla u = 0 \\ &u(t = 0) = u_0 \\ &u = g_D \quad on\ \partial\Omega \end{aligned} \tag{7.4}$$

These equations are defined over a 2D square domain $\Omega = (0.25, 1.25) \times (0.5, 1.5)$, we impose the following source terms $f(f_x \otimes f_y)$, and the viscosity $\nu = 0.01$, evaluation time $T = 1s$,

$$\begin{aligned} f_x &= (1-2\pi)(8\pi^2\nu - 1)cos(2\pi x)sin(2\pi y)e^{-t} \\ f_y &= (1+2\pi)(1-8\pi^2\nu)sin(2\pi x)cos(2\pi y)e^{-t} \end{aligned} \tag{7.5}$$

In this case the problem has the following analytical solution:

$$\begin{aligned} u_1 &= cos(2\pi x)sin(2\pi y)e^{-t} \\ u_2 &= -sin(2\pi x)cos(2\pi y)e^{-t} \\ p &= (1-8\pi^2\nu)sin(2\pi x)sin(2\pi y)e^{-t} \end{aligned} \tag{7.6}$$

The boundary conditions were chosen to verify the analytical solution at the boundaries. Fig.(7.6) shows the domain geometry and the boundary conditions which are used for this problem.

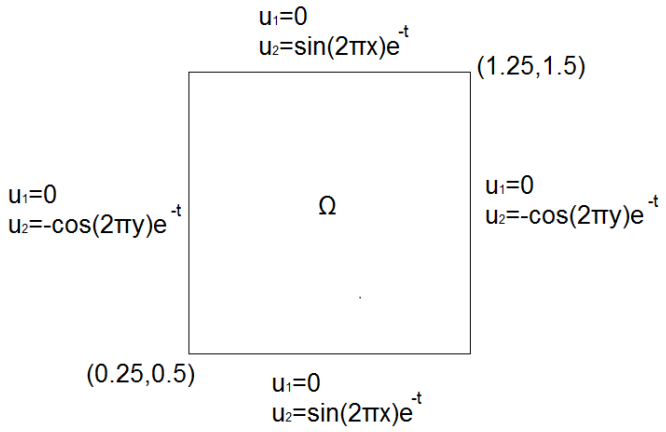


Figure 7.6: Geometry and boundary conditions for the Stokes analytical flow

From the PGD method algorithm described in the Chapter 6, we get the solution which is shown in Fig.(7.7)-Fig.(7.9), respectively. In each figure, the PGD solution is given in the top line and the analytical solution in the bottom line.

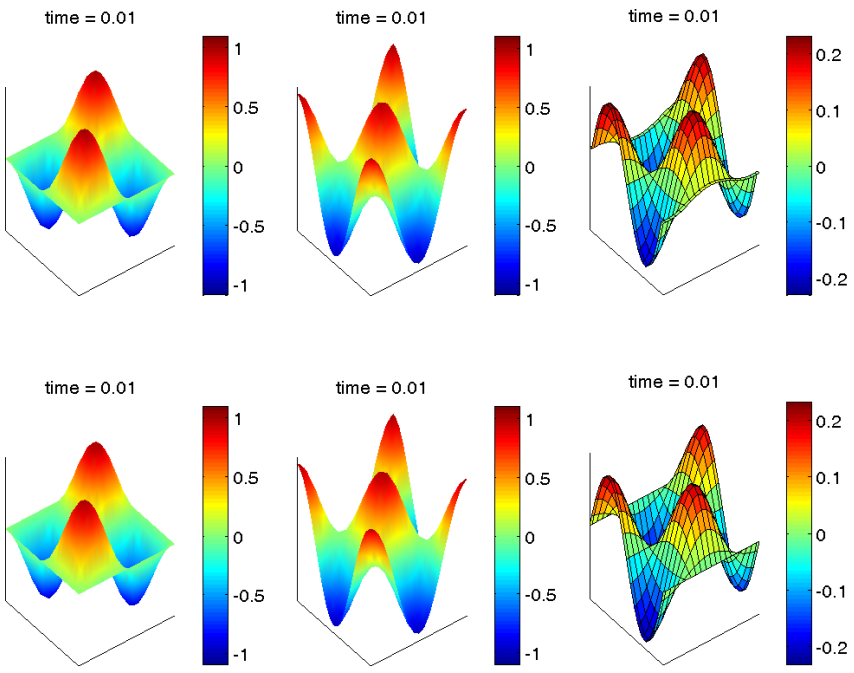


Figure 7.7: PGD solution (top) & exact solution (bottom) at 0.01s

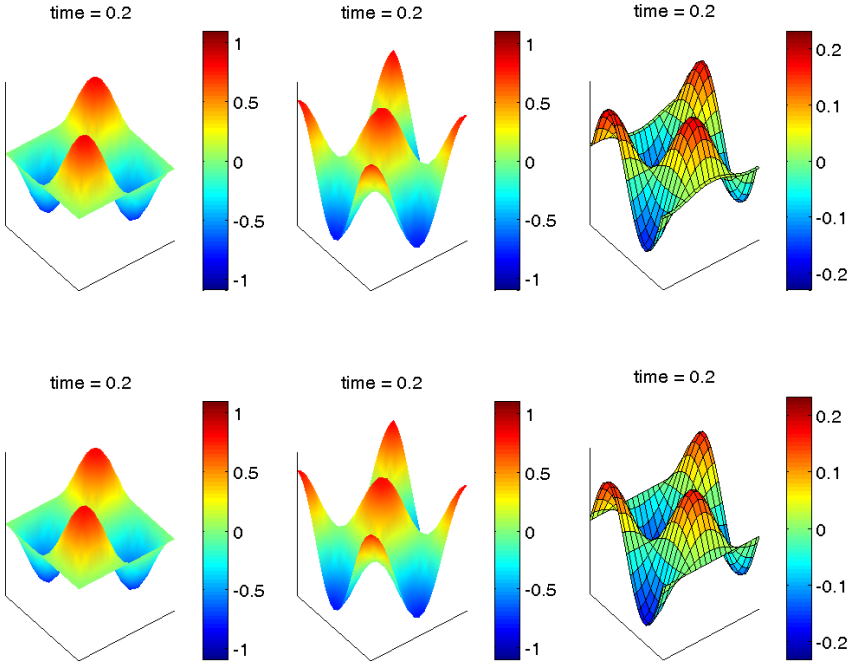


Figure 7.8: PGD solution (top) & exact solution (bottom) at 0.2s

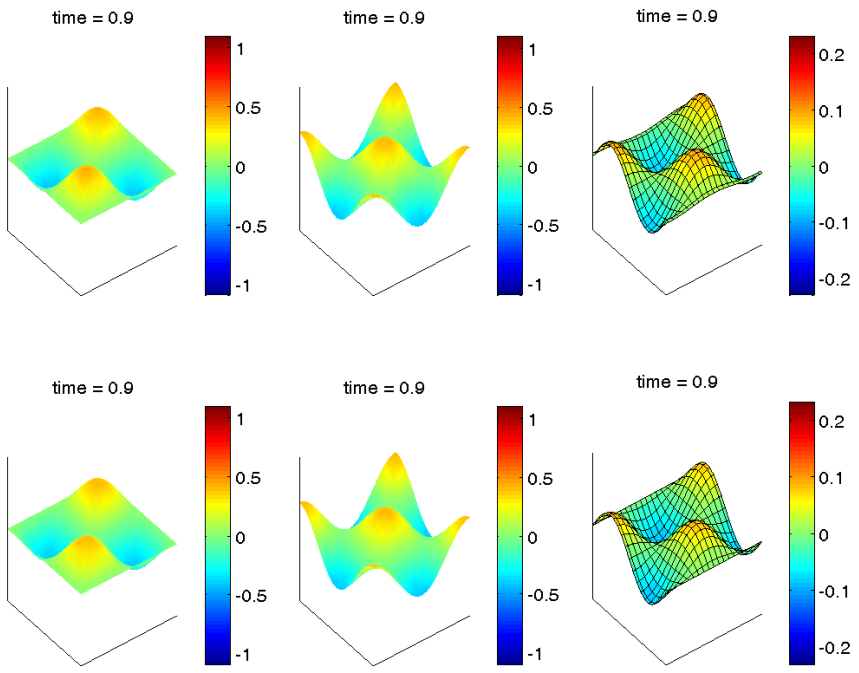
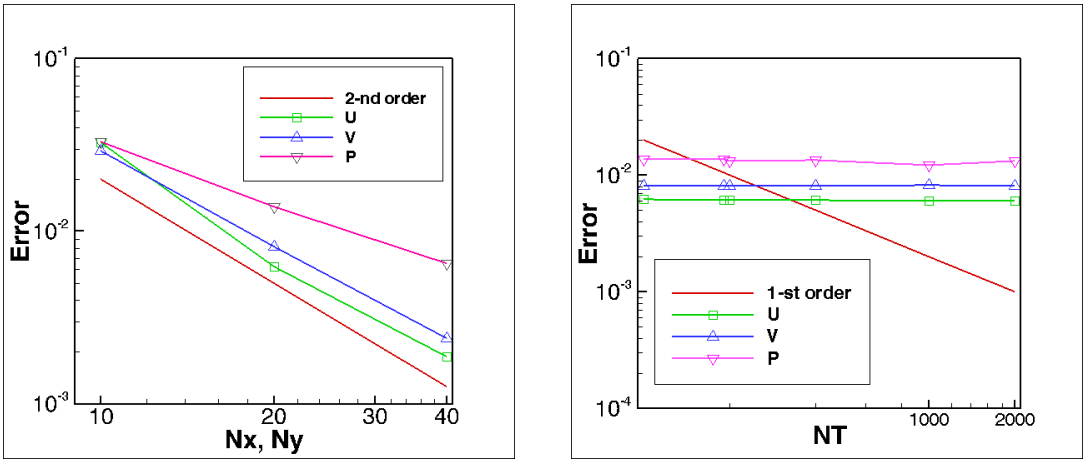


Figure 7.9: PGD solution (top) & exact solution (bottom) at 0.9s



(a) Space

(b) Time

Figure 7.10: Space and time discretization effect for the analytical Stokes problem

Effect of the space-step size and time-step size for the PGD are shown in the Fig.(7.10(a)) and Fig.(7.10(b)). For the space-step size, the second order convergence is achieved for this problem. For the time step size, we can see that 0-order convergence was observed for unclear reasons.

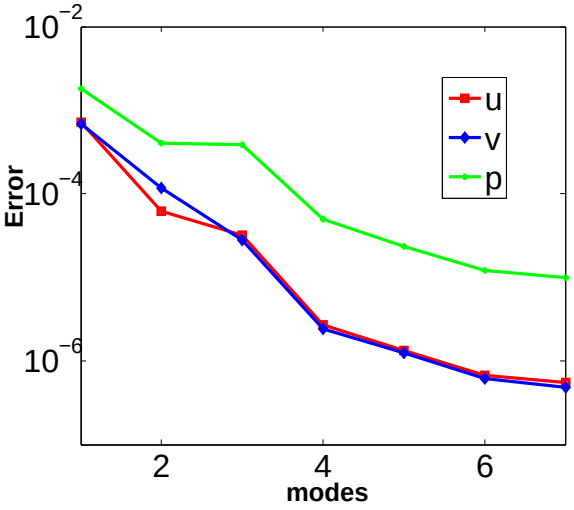


Figure 7.11: Convergence for the analytical Stokes problem

The convergence of the PGD solution for the analytical Stokes problem was shown in Fig.(7.11). We can see that the solution converges very fast. Moreover, the pressure convergence has a higher error compared with the velocities.

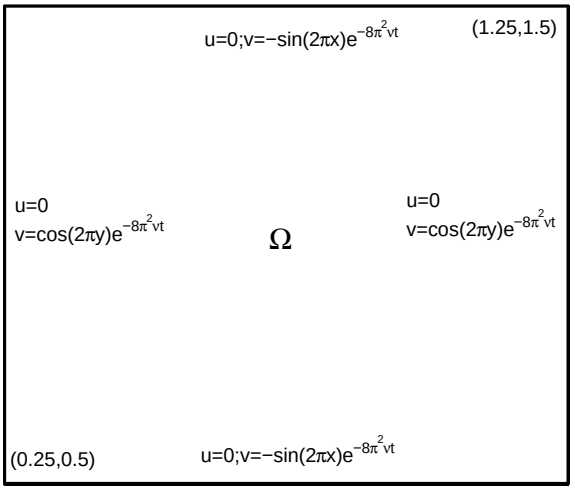


Figure 7.12: Geometry and boundaries for Burgers equation

7.2.3 Burgers problem

The Burgers equation is a convection-diffusion equation, and the convection term is described by a non-linear term. It is a kind of Navier Stokes Equation but without the term of pressure gradient. Here, we use the 2D Burgers equations as follows:

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \nu\Delta u + u\nabla u = f, \quad on\ \partial\Omega \times [0,T] \\
&\nabla u = 0 \\
&u(t=0) = u_0 \\
&u = g_D \quad on\ \partial\Omega
\end{aligned} \tag{7.7}$$

These equations are defined over a 2D square domain $\Omega = (0.25, 1.25) \times (0.5, 1.5)$, we imposed the following source terms $f(f_x \otimes f_y)$:

$$\begin{aligned}
f_x &= -\pi sin(4\pi x)e^{-16\pi^2\nu t} \\
f_y &= -\pi sin(4\pi y)e^{-16\pi^2\nu t}
\end{aligned} \tag{7.8}$$

The geometry of the domain and the boundary conditions for the Burgers equation are shown in the Fig.(7.12). With the given source terms, this problem has the following analytical solution:

$$\begin{aligned}
u_1 &= -cos(2\pi x)sin(2\pi y)e^{-8\pi^2\nu t} \\
u_2 &= sin(2\pi x)cos(2\pi y)e^{-8\pi^2\nu t}
\end{aligned} \tag{7.9}$$

From the PGD approach described in Chapter 6 based on the 2^{nd} linearisation for the non-linear terms, we got the solution which is shown in the Fig.(7.13)-Fig.(7.15) for the time at $t = 0.01s$, $t = 0.2s$ and $t = 0.9s$, respectively. In each figure, we give the PGD solutions in the top line and the errors with the analytical solutions in the bottom line. We can see that the error is of the order of 10^{-3} for each time step with a viscosity $\mu = 0.05$

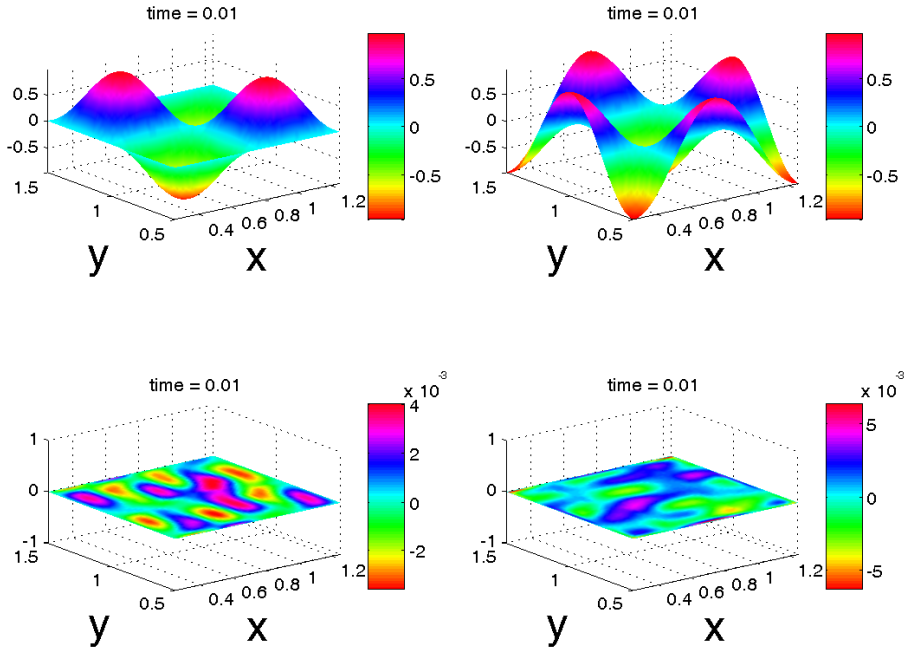


Figure 7.13: PGD solution for Burgers equation at 0.01s

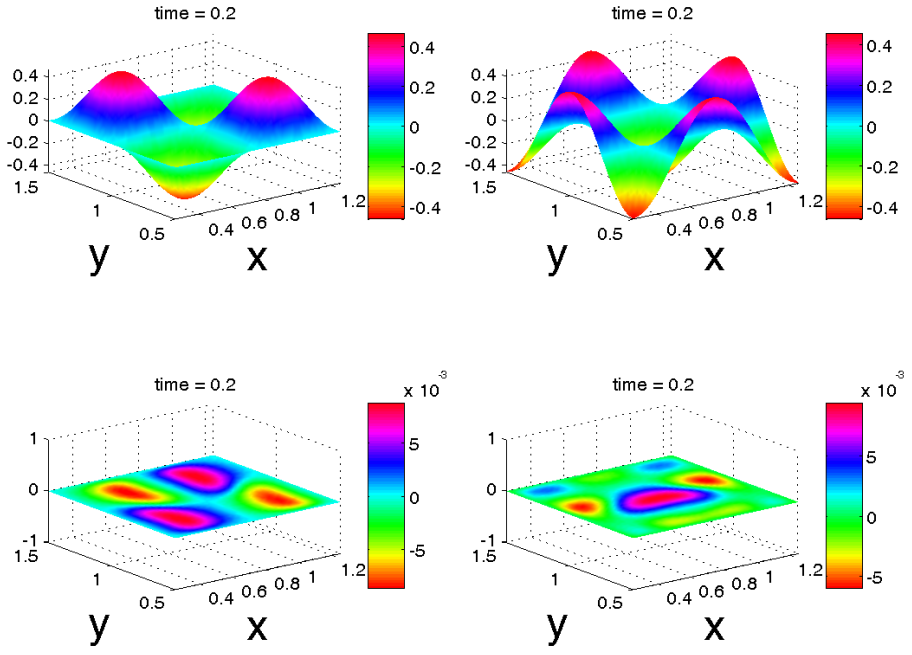


Figure 7.14: PGD solution for Burgers equation at 0.2s

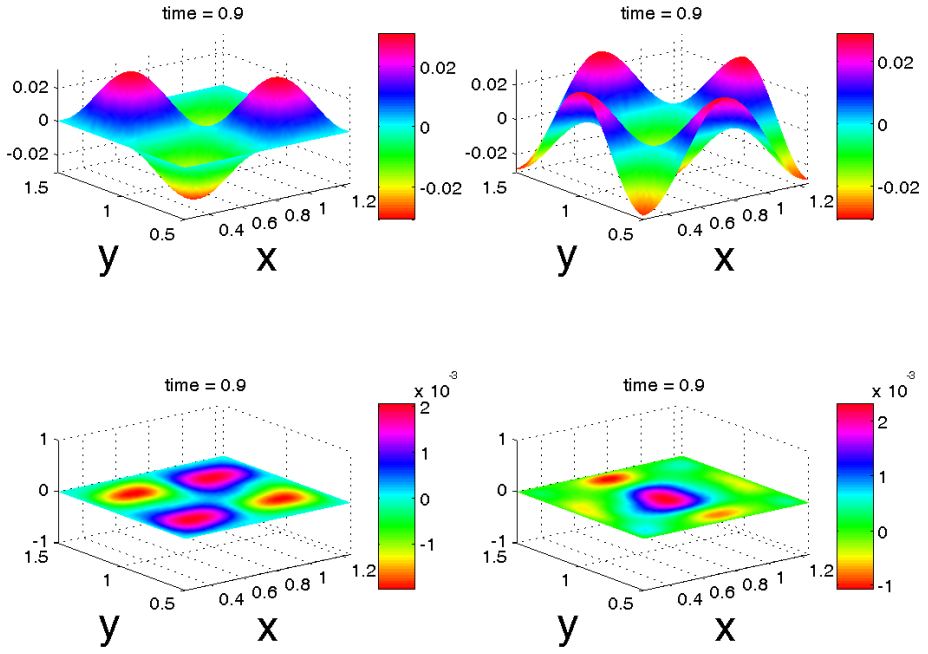


Figure 7.15: PGD solution for Burgers equation at 0.9s

The convergence for the Burgers equation is shown in Fig.(7.16).

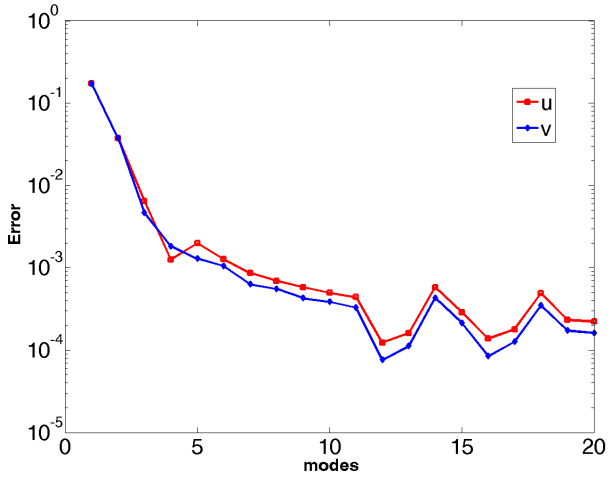
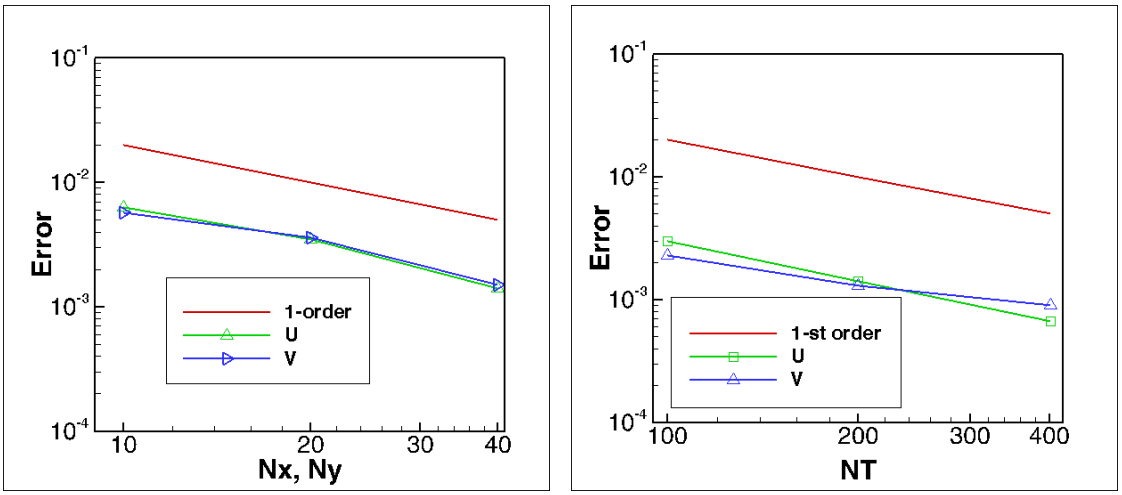


Figure 7.16: Error vs PGD modes for Burgers equation

Effect of the space-step size and time-step size for the PGD are shown in Fig.(7.17(a)) and Fig.(7.17(b)). For the space-step size, the second-order convergence is achieved for this problem, as the second-order scheme was used for the diffusion

terms and the first-order scheme for the convection term. For the time step size, we can see that a 1-order convergence was achieved.



(a) Space (b) Time

Figure 7.17: Space and time discretization effects for Burgers equation

In this section, three numerical examples of analytical flow problems have been examined to assess the PGD algorithm. These numerical examples showed that the PGD method is an efficient approach for resolving these problems by introducing a non-incremental method. If the number of modes can be kept low as it is observed in these analytical examples, the CPU savings can be really impressive.

7.3 Real flow problem

All of the numerical examples in the Section(7.2) concerned analytical flow problems. In this section, we will provide some numerical examples for a real unsteady viscous flow problem by using the PGD approach already introduced in the chapter 6.

7.3.1 Stokes flow in a lid-driven cavity

To start the assessment of the method described in the previous chapter, we will first compute an unsteady Stokes flow in a lid-driven cavity. In that case, we will not add any source terms in the momentum equations, since we do not look for an analytical solution.

The incompressible unsteady Stokes equations are expressed as follows:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \nu \Delta u + \nabla p = 0, \quad on\ \partial\Omega \times [0,T] \\ &\nabla u = 0 \end{aligned} \tag{7.10}$$

the geometry and the boundary conditions are listed as follows, which are shown in Fig.(7.18):

$$\begin{aligned} &u = 0 \qquad on\ x = -1; y \in (-1,1) \\ &u = cos(t) \quad on\ x = 1; y \in (-1,1) \\ &v = 0 \qquad on\ y = -1,1; x \in (-1,1) \end{aligned} \tag{7.11}$$

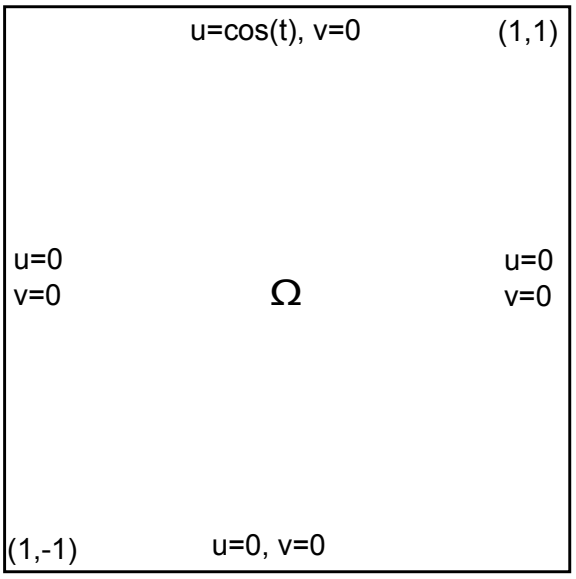


Figure 7.18: Geometry and boundary conditions for the lid-driven cavity flow

Firstly, we use the ISIS-CFD solver for solving the steady Stokes problem as follows,

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial p}{\partial x} &= 0 \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} - \frac{\partial p}{\partial y} &= 0 \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned} \tag{7.12}$$

with the boundary conditions as follows:

$$
\begin{aligned}
u &= 0 \quad on\ x = -1; y \in (-1, 1) \\
u &= 1 \quad on\ x = 1; y \in (-1, 1) \\
v &= 0 \quad on\ y = -1, 1; x \in (-1, 1)
\end{aligned}
\tag{7.13}
$$

In order to formulate a problem with homogeneous boundary conditions for the PGD formulation, we need to choose first spatial and temporal modes such that the unsteady boundary conditions are satisfied, which means that the next modes determined by the PGD approach will satisfy simple homogeneous boundary conditions. Therefore, we take the steady solution as the first spatial mode and $T_1 = cos(t)$ will be retained as the first temporal mode. With this choice, the next modes determined with the PGD approach will satisfy homogeneous boundary conditions (7.11).

From the PGD algorithm described in the last chapter, we can obtain the solution which is shown in the Fig.(7.19)-Fig.(7.21), respectively. In each figure, we give the solution for the velocities u, v, the pressure p, and the streamlines.

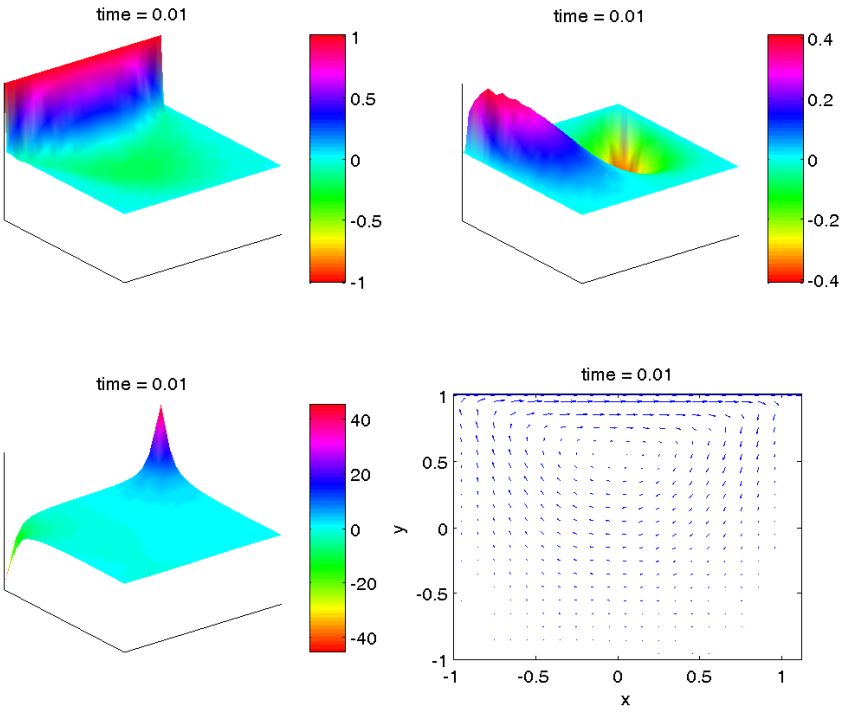


Figure 7.19: PGD solution at 0.01s for the lid-driven cavity problem

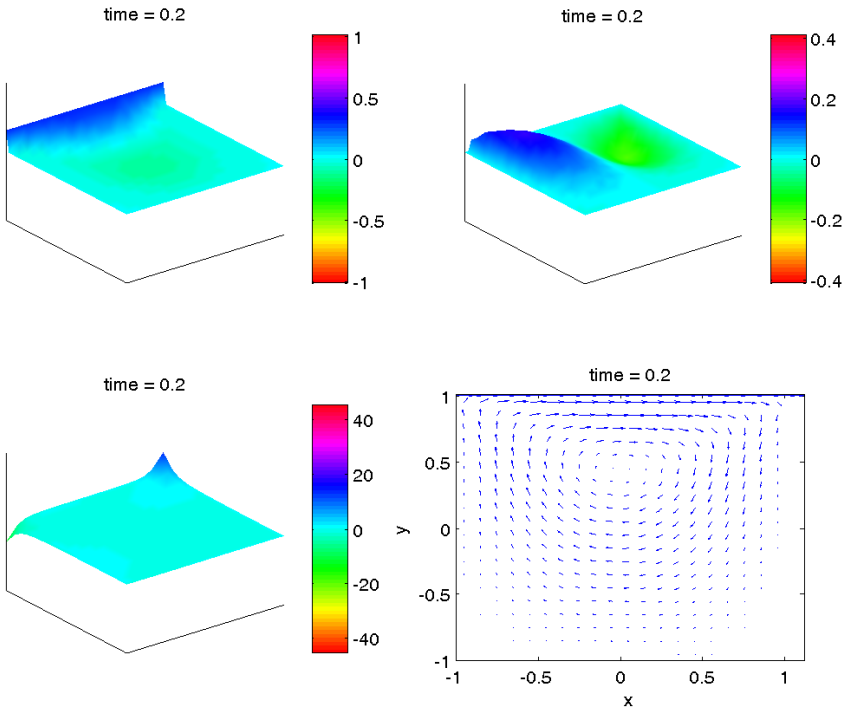


Figure 7.20: PGD solution at 0.2s for the lid-driven cavity problem

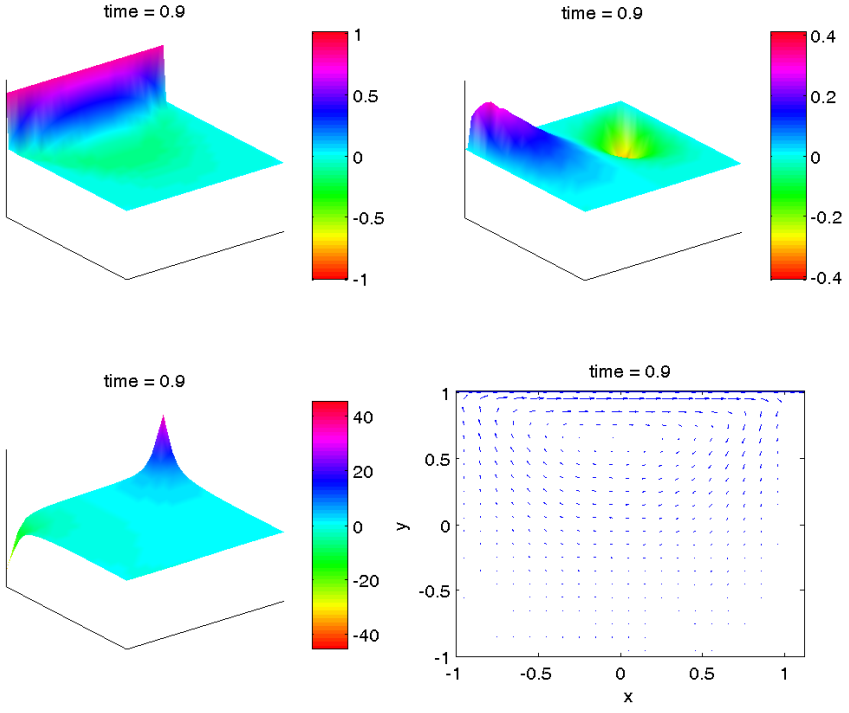


Figure 7.21: PGD solution at 0.9s for the lid-driven cavity problem

The convergence of the PGD solution for the lid-driven cavity problem is shown in Fig.(7.22) while the CPU time comparison between the PGD method and the ISIS-CFD solver is provided in Fig.(7.23). We can observe that the PGD method is definitely faster than the simulation based on the standard unsteady ISIS-CFD solver.

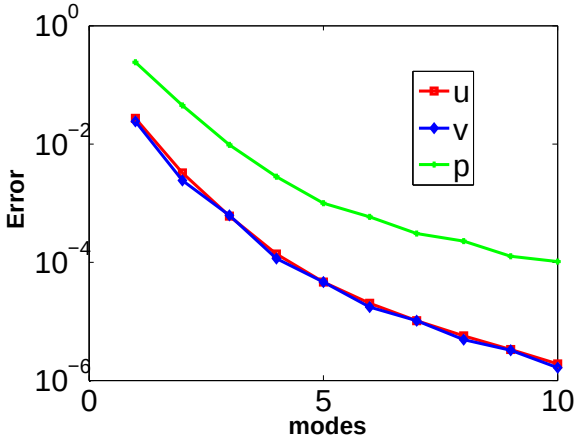


Figure 7.22: Convergence for the lid-driven cavity problem

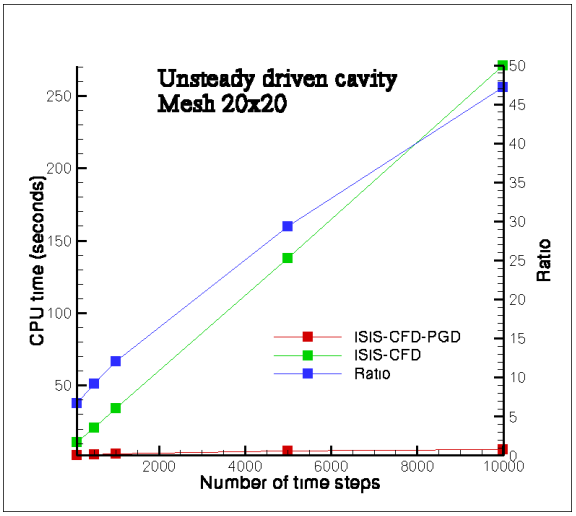
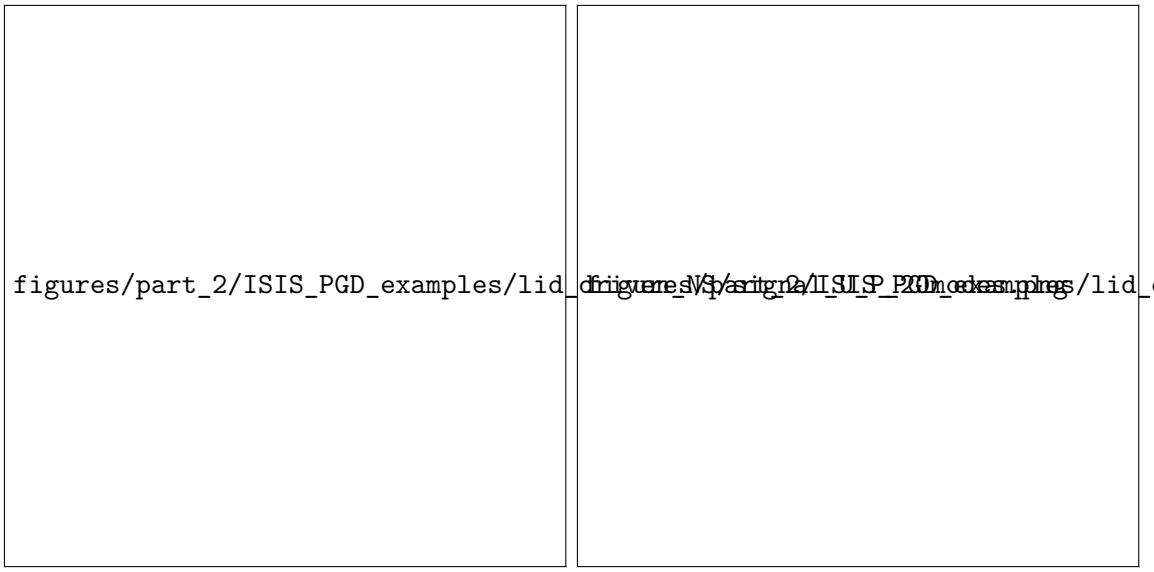


Figure 7.23: CPU time comparison for the lid-driven cavity problem

7.3.2 Navier-Stokes flow in a lid-driven cavity

In this section, we are going to solve the full unsteady Navier-Stokes equations to compute the unsteady flow in a lid-driven cavity, the unsteadiness being provided by the periodic motion of the top wall. The Reynolds number is ($Re = 100$) and the simulation time is $8s$. The geometry is the same as the one described in the previous section and the governing equations are:

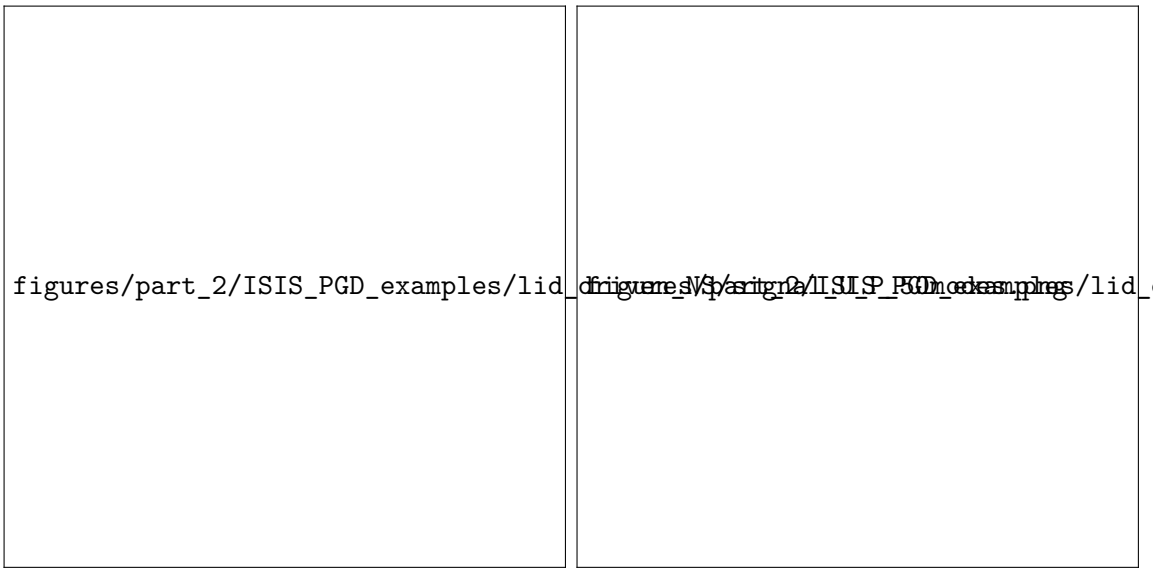
$$\begin{aligned} &\frac{\partial u}{\partial t} - \nu \Delta u + u \nabla u = \nabla P, \quad on\ \partial\Omega \times [0,T] \\ &\nabla u = 0 \end{aligned} \tag{7.14}$$



(a) Velocity U and Pressure P

(b) Pressure P

Figure 7.24: PGD solution with 20 modes and the ISIS-CFD solution for the lid-driven cavity problem by Navier-Stokes (continuous line: ISIS-CFD solution, dashed line: ISIS-CFD-PGD solution)



(a) Velocity U and Pressure P

(b) Pressure P

Figure 7.25: PGD solution with 50 modes and the ISIS-CFD solution for the lid-driven cavity problem by Navier-Stokes ((continuous line: ISIS-CFD solution, dashed line: ISIS-CFD-PGD solution)

The evolution in time at a specific point of the axial velocity U and pressure P are shown in Figure(7.24) and Figure(7.25) for the 20 modes and 50 modes PGD solution. We can see that for the velocity, the PGD solution is very close to the ISIS-CFD solution, but for the pressure, there is still some visible difference even when 50 modes are used. And the evolution of the modes amplitude $\|X_i\| \times \|T_i\|$ is also provided for checking the PGD solution convergence, we can observe a first monotonous convergence, but after eight modes, a kind of saturation is also met. Let us recall that the spatial discretisation is coarse (20*20 points in the square domain) and there is no reason why the PGD version of ISIS should converge towards ISIS-CFD since the numerical errors committed in the evaluation of the PGD coefficients will have a non-negligible influence on the modal convergence. Of course, both methods should converge towards the same solution as both time and space discretisations are refined. From that point of view, a systematic grid convergence study should be performed but the lack of time did not permit to include it in this study.

Figure 7.26: PGD convergence vs modes number

Finally, Figure(7.27) shows a comparison of the computational time between PGD and standard ISIS for several numbers of time steps. We can see that the PGD method is very fast. For the different numbers of time steps, the CPU time does not increase significantly for ISIS-CFD-PGD, while, for the standard ISIS-CFD method, the CPU time increases linearly with the number of time steps, as expected.

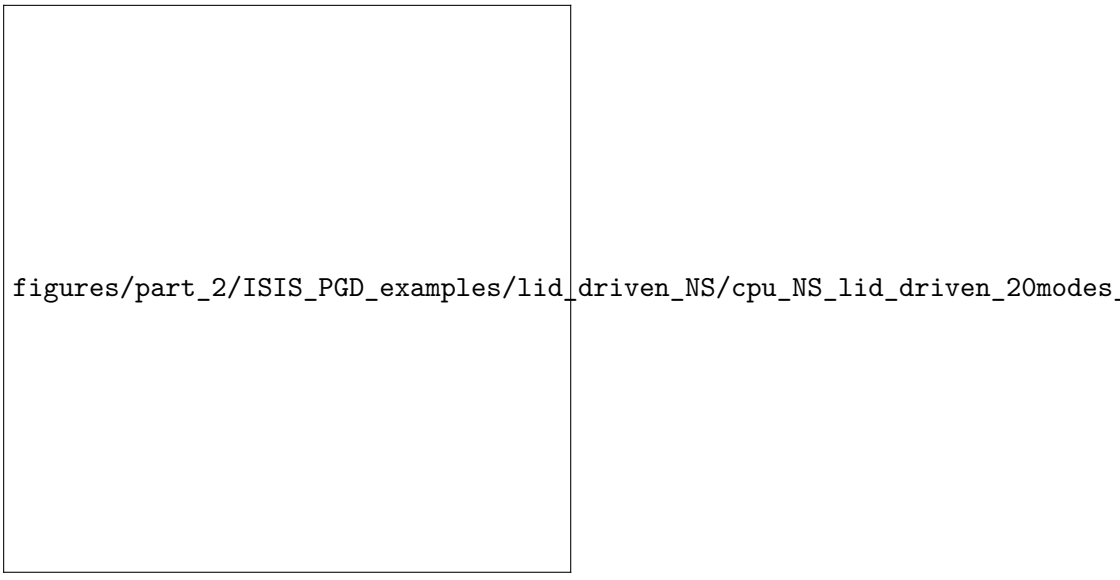


Figure 7.27: CPU comparison for the lid-driven cavity

7.3.3 2D Couette flow

As a last example, the Couette flow between two cylinders is considered. The geometry for this problem is shown in the Figure(7.28), the flow taking place in the gap between two cylinders. Unlike the Taylor-Couette flow, just the inner cylinder is rotating with a periodic motion. The governing equations for Navier-Stokes problem are given by:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \nu\Delta u + u\nabla u = \nabla P, \quad on\ \partial\Omega \times [0,T] \\ &\nabla u = 0 \\ &u(t=0) = u_0 \\ &u = g_D \quad on\ \partial\Omega \end{aligned} \tag{7.15}$$

The boundary condition Γ_{out} for the outer cylinder is a no-slip boundary condition, while a rotation $\Omega sin(\omega t)$ is imposed at the boundary Γ_{in} of the inner cylinder.

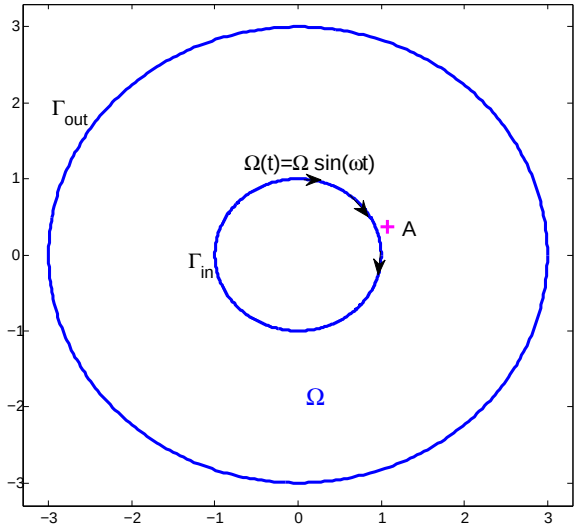
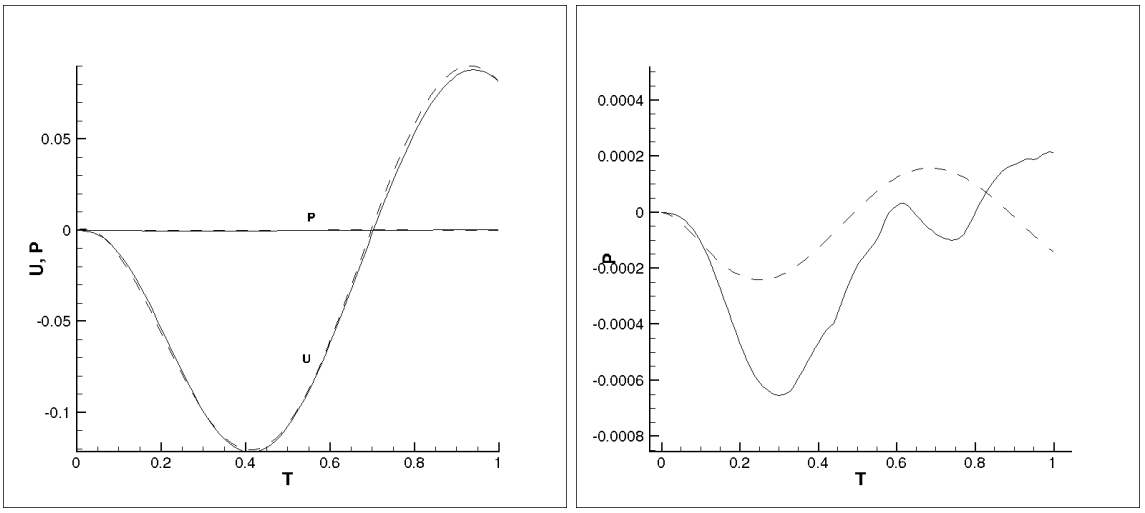


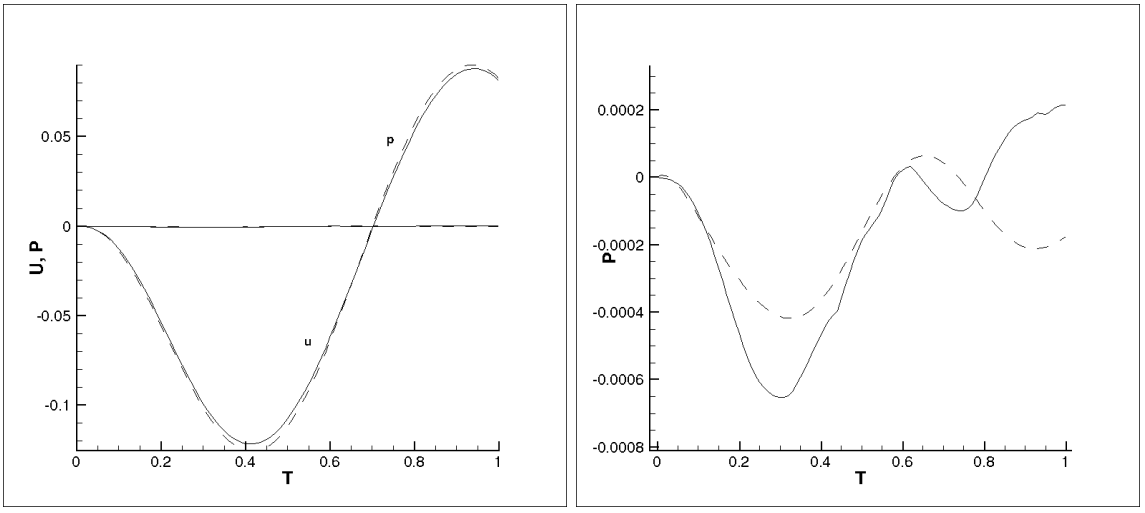
Figure 7.28: Geometry for the Couette flow problem

By using the PGD method, the temporal evolution of velocity U and pressure P at a specified point A (see Figure(7.28) for its location) is shown in Fig.(7.29) and Fig.(7.30) for 20 modes and 32 modes, respectively. The continuous line standing for ISIS-CFD solution and dashed line corresponding to ISIS-CFD-PGD solution, we can notice that when more modes are used, the PGD solution for the velocity is closer to the ISIS-CFD solution, while, for the pressure, there is still a remaining small difference in the temporal evolution.



(a) Velocity U and Pressure P (b) Pressure P

Figure 7.29: PGD solution with 20 modes and the ISIS-CFD solution (continuous line is ISIS-CFD solution and dashed line is ISIS-CFD-PGD solution)



(a) Velocity U and Pressure P (b) Pressure P

Figure 7.30: PGD solution with 32 modes and the ISIS-CFD solution(continuous line is ISIS-CFD solution and dashed line is ISIS-CFD-PGD solution)

The $L-2$ norm (over the whole temporal and spatial domain) of the differences between the ISIS-CFD-PGD and the standard ISIS-CFD solutions with respect to the modes number for the velocity and pressure is shown in Fig.(7.31). As expected, the difference on the velocity is reduced when more modes are used; and for pressure also, we also can notice that this difference is reduced, although in a somewhat less extent.

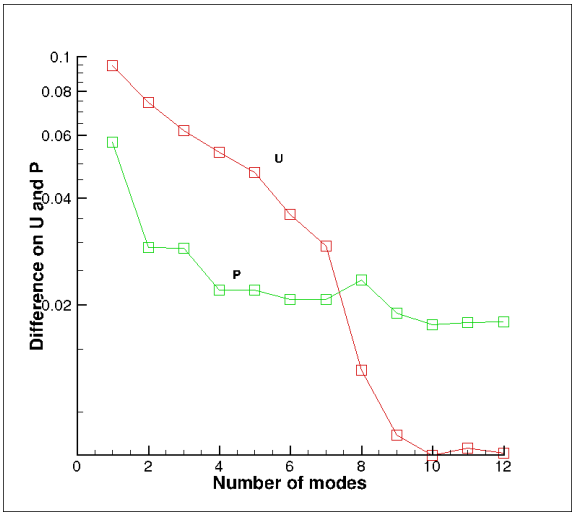


Figure 7.31: $L-2$ norm of the difference for U, p vs modes number

Finally, the PGD convergence is shown in the Fig.(7.32), in the Figure, $\|X_i\| \times \|T_i\|$ is concerning as the Modes amplitude, we can see that this value became smaller as the more modes are used, from the 14th modes, the amplitude is smaller enough.

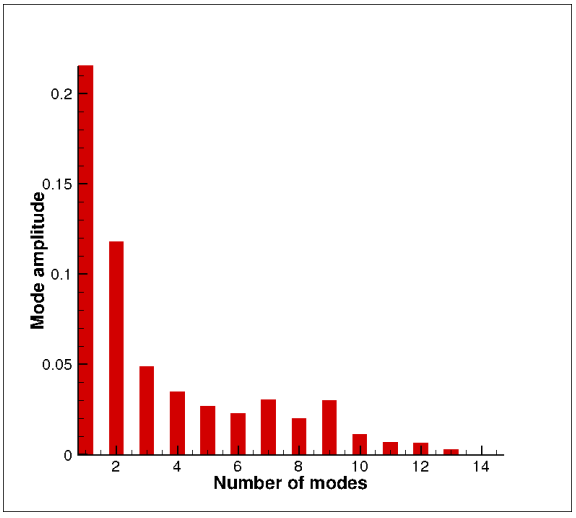


Figure 7.32: PGD convergence vs modes number

7.4 Discussion and Conclusion

In this chapter, several numerical examples were chosen to illustrate the proposed PGD method coupled with the ISIS-CFD solver which was described in the chapter 6. This new method was implemented from scratch in a prototype code in MathLab, based on the same methodological framework as ISIS-CFD. It is also being

implemented in the original ISIS-CFD commercial software. Despite the formidable complexity of this task in terms of implementation and code validation, the first results appear quite encouraging. The considered test cases were either analytical or real with or without non-linear terms but all were confined to bidimensional laminar flows whose unsteadiness is imposed by the motion of one of the boundaries. Incremental and modal representations of unsteadiness were compared with a good agreement for a limited number of modes (less than 20, most of the time). A very good agreement for the temporal evolution was observed on the velocity components while the agreement on the pressure evolution was less good, for a given spatial discretisation. Reasons for this observation have to be found but one can underline right now the accuracy of evaluation of PGD coefficients which depend on domain integral of differential operators which are difficult to evaluate accurately, especially of one uses a finite-volume formulation. Although not proved here, one can expect that the modal convergence of a PGD expansion will strongly depend on the accuracy of the PGD coefficients.

The PGD method proposed in this work for the solution of the unsteady Navier-stokes equations is based on the same temporal function for the velocities and the pressure. It could be interesting to choose a different temporal mode for the pressure to reduce the number of terms of the modal expansion.

In terms of performance, the Proper Generalized Decomposition is very fast as soon as we do not need too many modes and need many time steps. A ratio of more than ten between an incremental approach and a PGD formulation has been observed on most of these first bidimensional testcases, which should even more dramatic for three-dimensional configurations.

The boundary condition problem related to the fact that PGD works only for homogeneous boundary conditions has been solved by determining a priori first modes satisfying the unsteady boundary conditions. Generalizing this procedure to any kind of boundary conditions (like for instance a propeller rotating in a current) may be more problematic.

Finally, this first implementation in the framework of the ISIS-CFD platform is encouraging but a lot of work remains to be done to determine the best numerical strategy to be used (under-relaxation parameters, residual reduction inside the fixed point loop, sensitivity to the accuracy of the PGD coefficients, best linearization, etc...) to get a robust and fast numerical tool replacing the incremental methodology.

Conclusions and Perspectives

Conclusions

PGD based Model Order Reduction techniques are numerical tools which allows to solve complex problem with a little computational effort. This ROM technique showed excellent performance on reduction of complex problem without any a priori knowledge of the solution. This PhD work has been focused on the research of a numerical tool which concerned PGD for solving the problem for the high order PDEs problem, concern with the complex geometry problem, and resolving the Unsteady Navier-Stokes equations by using the PGD algorithm in the ISIS-CFD solver which can be improve the computed efficiency, and some numerical results are also shown in the work.

In short, a series of conclusions and advantages are received in this thesis as follows:

- Most of the PGD works were focus on the lower order PDEs problem, by using the pusdo-Chebyshev method which is an efficiency numerical method, we proposed the PGD method for solving the high order PDEs, to avoiding the high compute time when we use fine enough discretization, there is a clear need for improving computational efficiency. The PGD method was proposed for solving the high order PDEs problem, especially for the 2D lid driven cavity flow problem in the Vorticity-velocity form. The PGD method was coupling with the spectral discretization method for the fourth order PDEs. Firstly, the method was used to analysis the Laplace problem, based on the high accuracy of the chebyshev method, each loop for converge the Fix point iteration can have a very small error, in the framework of PGD method, we can have a higher accuracy solution than the Finite Difference method. Numerical example for the 4th order PDE problem with the homogeneous boundary condition problem was studied, then by using the strategy for change the non-homogeneous problem into the homogeneous problem, we gave the results for the 2D lid driven cavity

flow problem in the Vorticity-velocity form.

- In the space decomposition filed, a lot of work concerning the PGD have been done for the domain can be separated, when the geometry becomes complex, the current PGD framework can not be done for these problems. In the part of work, the curvilinear coordinate was used for changing the complex geometry into the regular geometry.The FD method were used for solving the complex domain problem in the new regular compute domain. Then we proposed the PGD method to resolve the problem in the new compute domain. The PGD method coupling the Finite element method were presented for the complex geometry, the numerical example were done for the Circle domain and the Ellipse domain and the star domain with a whole in the center under the un-uniform source term.

- ISIS-CFD solver is a fully unstructured Unsteady Navier-Stokes solver, which is efficient for the complex geometries, but there is need of fine grid discretization across the interface and the time step is very small, in this case, a huge cpu time for the calculation are needed. In this work, the ISIS-CFD solver was shortly introduced with a numerical example as steady stokes equation; after the SVD was done for the ISIS-CFD solution for unsteady lid driven cavity problem to see whether the ISIS-CFD can be separated or not; then the PGD formulation for Resolving the Unsteady Navier-Stokes equations are given. And two kinds of non-linear treatment for the Navier-stokes were presented. The Numerical examples were given for the Stokes problem, Burgers equation and the Navier-Stokes equation, which is based on a personal Matlab code concerning the Finite Difference discretization. The numerical examples shows that the PGD method is a very effective method as a model reduction method for solving the unsteady Navier-stokes equation, but also a lot of improvement need to be done.

Perspectives

Because of the time limited, the work in this thesis is just concentrated in 2D domain problem, but there is still a lot of work need to do. Several perspective of this work are given as follow:

- For the PGD method concerning with the high order PDEs problem, we just focused on a several sample problems, and the method described in this work can be extend to the 2D rectangular method for studying the plate problem

by using the Jacobi matrix, and we can do the parameter control by using the PGD method.

- The PGD for the complex geometry problem, there are more numerical example need to be done for the complex geometry by using the Curvilinear Coordinates System; and maybe this techniques can be extended to the 3D cases...
- For the part of PGD for Resolving Unsteady Navier-Stokes Equations, in this work, we just derived the formulation for the 2D domain problem, and the numerical example is only for some simple geometry, more complex geometry problem has been include into the ISIS-CFD solver in the DSMP group by the Fortran solver. And more example work are being done. For the 3D problem, we separated the solution $u(x, y, z, t)$ as the space function $X^{\vec{u}}(x, y, z)$ and the time function $T^u(t)$ by the separated representation method.
- Also there is possibility for using the ISIS-CFD coupled with PGD method for the fluid-Structure interaction problem for improving the computational efficiency. If the boundary condition and the initial condition for the flow problem are treated in the proper way, many problem can be used by PGD for improving the computational efficiency.

Bibliography

[Aghighi et al., 2013] Aghighi, M., Ammar, A., Metivier, C., Normandin, M., and Chinesta, F. (2013). Non-incremental transient solution of the Rayleigh–Bénard convection model by using the PGD. Journal of Non-Newtonian Fluid Mechanics, 200:65–78. (Cited on page 89.)

[Ali and Raslan, 2007] Ali, A. and Raslan, K. (2007). Variational iteration method for solving biharmonic equations. Physics Letters A, 370(5-6):441–448. (Cited on page 8.)

[Allery et al., 2005] Allery, C., Béghein, C., and 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. (Cited on page 11.)

[Allery et al., 2011] Allery, C., Hamdouni, A., Ryckelynck, D., and Verdon, N. (2011). A priori reduction method for solving the two-dimensional Burgers ' equations. Applied Mathematics and Computation, 217(15):6671–6679. (Cited on page 11.)

[Ammar and Chinesta, 2008] Ammar, A. and Chinesta, F. (2008). Circumventing Curse of Dimensionality in the Solution of Highly Multidimensional Models Encountered in Quantum Mechanics Using Meshfree Finite Sums Decomposition. Lecture Notes on Computational Science and Engineering, 65:1–17. (Cited on page 12.)

[Ammar et al., 2010a] Ammar, A., Chinesta, F., Diez, P., and 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. (Cited on page 12.)

[Ammar et al., 2010b] Ammar, A., Chinesta, F., González, D., and Cueto, E. (2010b). Recent advances on the use of separated representations. International Journal for Numerical Methods in Engineering, 81(5):637–659. (Cited on page 12.)

[Ammar et al., 2008] Ammar, A., Chinesta, F., and 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. (Cited on page 12.)

[Ammar et al., 2014] Ammar, A., Huerta, A., Chinesta, F., Cueto, E., and Leygue, A. (2014). Parametric solutions involving geometry: A step towards efficient shape optimization. Computer Methods in Applied Mechanics and Engineering, 268:178–193. (Cited on pages 12 and 56.)

[Ammar et al., 2006] Ammar, A., Mokdad, B., Chinesta, F., and Keunings, R. (2006). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Journal of Non-Newtonian Fluid Mechanics, 139(3):153–176. (Cited on pages 11 and 89.)

[Ammar et al., 2007] Ammar, A., Mokdad, B., Chinesta, F., and Keunings, R. (2007). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids:: Part II: Transient simulation using space-time separated representations. Journal of non-newtonian fluid mechanics, 144(2-3):98–121. (Cited on pages 11 and 89.)

[Ammar et al., 2010c] Ammar, A., Normandin, M., and Chinesta, F. (2010c). Solving parametric complex fluids models in rheometric flows. Journal of Non-Newtonian Fluid Mechanics, 165(23-24):1588–1601. (Cited on pages 12 and 89.)

[Ammar et al., 2010d] Ammar, A., Normandin, M., Daim, F., Gonz, D., and Daim, M. N. F. (2010d). Non Incremental Strategies Based on Separated Representations : Applications in Computational Rheology. Communications in Mathematical Sciences, 8(3):671–695. (Cited on pages 12 and 128.)

[Angot et al., 1999] Angot, P., Bruneau, C.-h., and Fabrie, P. (1999). Numerische Mathematik A penalization method to take into account obstacles in incompressible viscous flows. Numer. Math., 81:497–520. (Cited on page 88.)

[Avudainayagam and Vani, 2000] Avudainayagam, A. and Vani, C. (2000). A Domain Decomposition Method for Biharmonic Equation. Computers & Mathematics with Applications, 40:865–876. (Cited on page 8.)

[Baur et al., 2009] Baur, C., Bontoux, P., Kornhaas, M., Minguez, M., Pasquetti, R., Schäfer, M., Serre, E., and Séverac, E. (2009). High-order methods for large-eddy simulation in complex geometries. Numerical Simulation of Turbulent Flows and Noise Generation, 104:309–334. (Cited on page 10.)

[Bazilevs et al., 2006] Bazilevs, Y., Calo, V. M., Zhang, Y., and Hughes, T. J. R. (2006). Isogeometric Fluid–structure Interaction Analysis with Applications to Arterial Blood Flow. Computational Mechanics, 38(4-5):310–322. (Cited on page 84.)

[Biros et al., 2004] Biros, G., Ying, L., and Zorin, D. (2004). A fast solver for the Stokes equations with distributed forces in complex geometries. Journal of Computational Physics, 193(1):317–348. (Cited on page 10.)

[Bognet et al., 2012] Bognet, B., Bordeu, F., Chinesta, F., Leygue, A., and Poitou, A. (2012). Advanced simulation of models defined in plate geometries : 3D solutions with 2D computational complexity. Computer Methods in Applied Mechanics and Engineering, 201-204:1–12. (Cited on page 88.)

[Bonithon et al., 2011] Bonithon, G., Joyot, P., Chinesta, F., and 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. (Cited on page 12.)

[Bruneau and Saad, 2006] Bruneau, C. and Saad, M. (2006). The 2D lid-driven cavity problem revisited. Computers & Fluids, 35(3):326–348. (Cited on page 48.)

[Bruno-alfonso et al., 2012] Bruno-alfonso, A., Cabezas-Gomez, L., Omez, and Navarro, H. A. (2012). Alternate treatments of jacobian singularities in polar coordinates within finite-difference schemes. World Journal of Modelling and Simulation, 8(3):163–171. (Cited on page 59.)

[Buffat et al., 2011] Buffat, M., Penven, L. L., and Cadiou, A. (2011). An efficient spectral method based on an orthogonal decomposition of the velocity for transition analysis in wall bounded flow. Computers & Fluids, 42(1):62–72. (Cited on page 11.)

[Bui, 2000] Bui, T. T. (2000). A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. Computers & fluids, 29:877–915. (Cited on page 10.)

[Burkardt et al., 2006] Burkardt, J., Gunzburger, M., and Lee, H.-C. (2006). POD and CVT-based reduced-order modeling of Navier-Stokes flows. Computer Methods in Applied Mechanics and Engineering, 196(1-3):337–355. (Cited on page 11.)

[Cai and Zhang, 2012] Cai, Z. and Zhang, S. (2012). Mixed methods for stationary Navier-Stokes equations based on pseudostress-pressure-velocity formulation. Mathematics of Computation, 81(280):1903–1927. (Cited on page 87.)

[Cant, 2002] Cant, S. (2002). High-performance computing in computational fluid dynamics: progress and challenges. Philosophical transactions. Series A,

Mathematical, physical, and engineering sciences, 360(1795):1211–25. (Cited on page 10.)

[Carbou, 2004] Carbou, G. (2004). Penalization method for viscous incompressible flow around a porous thin layer. Nonlinear Analysis: Real World Applications, 5(5):815–855. (Cited on page 88.)

[Chady Ghnatios, 2012] Chady Ghnatios (2012). Simulation avancée des problèmes thermiques rencontrés lors de la mis en forme des composites. PhD thesis-Ecole Centrale Nantes. (Cited on pages 12 and 56.)

[Chantasiriwan, 2006] Chantasiriwan, S. (2006). solutions to harmonic and biharmonic problems with discontinuous boundary conditions by collocation methods using multiquadrics as basis functions. International Communications in Heat and Mass Transfer, 34(3):313–320. (Cited on page 9.)

[Chen et al., 2008] Chen, G., Li, Z., and Lin, P. (2008). A fast finite difference method for biharmonic equations on irregular domains and its application to an incompressible Stokes flow. adv comput math, 29:113–133. (Cited on page 9.)

[Chikhaoui, 2007] Chikhaoui, O. E. C. N. (2007). Simulations des grandes échelles et modélisations hybrides RANSE/LES pour le calcul d"ecoulements turbulents de grande complexité. PhD thesis-Ecole Centrale Nantes. (Not cited.)

[Chinesta et al., 2010a] Chinesta, F., Ammar, A., and Cueto, E. (2010a). Proper generalized decomposition of multiscale models. International Journal for Numerical Methods in Engineering, 83(8-9):1114–1132. (Cited on page 117.)

[Chinesta et al., 2010b] Chinesta, F., Ammar, A., and Cueto, E. (2010b). 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. (Cited on page 11.)

[Chinesta et al., 2011] Chinesta, F., Ammar, A., Leygue, A., and Keunings, R. (2011). An overview of the proper generalized decomposition with applications in computational rheology. Journal of Non-Newtonian Fluid Mechanics, 166(11):578–592. (Cited on pages 1, 12, 88 and 89.)

[Chinesta et al., 2012] Chinesta, F., Leygue, a., Bognet, B., Ghnatios, C., Poulhaon, F., Bordeu, F., Barasinski, a., Poitou, a., Chatel, S., and Maison-Le-Poec, S. (2012). First steps towards an advanced simulation of composites manufacturing by automated tape placement. International Journal of Material Forming. (Cited on page 88.)

[Chinesta et al., 2013] Chinesta, F., Leygue, A., Bordeu, F., Aguado, J. V., Cueto, E., González, D., Alfaro, I., Ammar, A., and Huerta, A. (2013). Pgd-based computational vademecum for efficient design, optimization and control. Archives of Computational Methods in Engineering, 20(1):31–59. (Cited on page 11.)

[Christiansen, 2001] Christiansen, S. (2001). Detecting non-uniqueness of solutions to biharmonic integral equations through SVD. Journal of Computational and Applied Mathematics, 134:23–35. (Cited on page 107.)

[Coullet et al., 2005] Coullet, P., Mahadevan, L., and Riera, C. S. (2005). Hydrodynamical models for the chaotic dripping faucet. Journal of Fluid Mechanics, 526:1–17. (Cited on page 84.)

[Cremonesi et al., 2013] Cremonesi, M., Néron, D., Guidault, P.-a., and Ladevèze, P. (2013). A PGD-based homogenization technique for the resolution of nonlinear multiscale problems. Computer Methods in Applied Mechanics and Engineering, 267:275–292. (Cited on pages 1 and 11.)

[Cummins et al., 2005] Cummins, S. J., Francois, M. M., and Kothe, D. B. (2005). Estimating curvature from volume fractions. Computers & Structures, 83(6-7):425–434. (Cited on pages 103 and 144.)

[Dauby, 2007] Dauby, D. (2007). Simulation d'écoulements cavitants par résolution numérique des équations de Navier-Stokes en moyenne de reynolds : application à la cavitation de tourbillon d'extrémité. PhD thesis-Ecole Centrale Nantes. (Not cited.)

[Day et al., 1998] Day, M. S., Colella, P., Lijewski, M. J., Rendleman, C., and Marcus, D. (1998). Embedded boundary algorithms for solving the poisson equation on complex domains. Technical report, Lawrence Berkeley National Laboratory (LBNL-41811). (Cited on page 10.)

[Deng et al., 2001] Deng, G. B., Piquet, J., Vasseur, X., and Visonneau, M. (2001). A new fully coupled method for computing turbulent flows. Computers & Fluids, 30:445–472. (Cited on pages 2, 91 and 97.)

[Deng et al., 2006] Deng, G. B., Queutey, P., and Visonneau, M. (2006). A Code Verification Exercise for the Unstructured Finite Volume CFD Solver ISIS-CFD. In European Conference on Computational Fluid Dynamics,TU Delft, The Netherlands, 2006, pages 1–18. (Not cited.)

[Dubois et al., 2003] Dubois, F., Salaün, M., and Salmon, S. (2003). Vorticity-velocity-pressure and stream function-vorticity formulations for the Stokes problem. Journal de Mathématiques Pures et Appliquées, 82(11):1395–1451. (Cited on page 48.)

[Duggleby et al., 2011] Duggleby, A., Camp, J. L., and Fischer, P. F. (2011). MASSIVELY PARALLEL COMPUTATIONAL FLUID DYNAMICS WITH LARGE EDDY. ASME Conference Proceedings- IMECE2011-62489, page 817. (Cited on page 10.)

[Dumon et al., 2010] Dumon, A., Allery, C., and Ammar, A. (2010). Proper generalized decomposition method for incompressible flows in stream-vorticity formulation. European Journal of Computational Mechanics, 19(5-7):591–617. (Cited on page 89.)

[Dumon et al., 2011] Dumon, A., Allery, C., and Ammar, A. (2011). Proper general decomposition (PGD) for the resolution of Navier-Stokes equations. Journal of Computational Physics, 230(4):1387–1407. (Cited on pages 48 and 89.)

[Dumon et al., 2013] Dumon, A., Allery, C., and Ammar, A. (2013). Proper Generalized Decomposition method for incompressible Navier-Stokes equations with a spectral discretization. Applied Mathematics and Computation, 219(15):8145–8162. (Cited on pages 12 and 89.)

[Durand, 2012] Durand, M. (2012). Interaction fluide-structure souple et légère, applications aux voiliers. PhD thesis-Ecole Centrale Nantes. (Not cited.)

[Erturk and Dursun, 2007] Erturk, E. and Dursun, B. (2007). Numerical solutions of 2-D steady incompressible flow in a driven skewed cavity . Journal of Applied Mathematics and Mechanics, 87:377–392. (Cited on pages 8 and 17.)

[Faure et al., 2008] Faure, S., Laminie, J., and Temam, R. (2008). Colocated finite volume schemes for fluid flows. Communications in computational physics, 4(1):1–25. (Cited on page 87.)

[Geuzaine and Remacle, 2009] Geuzaine, C. and Remacle, J. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79:1309–1331. (Cited on page 10.)

[Ghnatios et al., 2012] Ghnatios, C., Ammar, A., Cimetiere, A., Hamdouni, A., Leygue, A., and Chinesta, F. (2012). FIRST STEPS IN THE SPACE SEPARATED REPRESENTATION OF MODELS DEFINED IN COMPLEX DOMAINS. 11th Biennial Conference on Engineering Systems Design and Analysis, ASME, Nantes, pages ESDA2012–82489. (Cited on pages 12 and 56.)

[Ghnatios et al., 2013] Ghnatios, C., Chinesta, F., and Binetruy, C. (2013). 3D Modeling of squeeze flows occurring in composite laminates. International Journal of Material Forming. (Cited on page 88.)

[Glowinski and Pironneau, 1992] Glowinski, R. and Pironneau, O. (1992). Finite element methods for Navier-Stokes equations. Annu. Rev. Fluid Mech., 24:167–204. (Cited on page 86.)

[González et al., 2010] González, D., Ammar, A., Chinesta, F., and Cueto, E. (2010). Recent advances on the use of separated representations. International journal for numerical methods in Engineering, 81:637–659. (Cited on pages 12 and 55.)

[González et al., 2013] González, D., Cueto, E., Chinesta, F., Díez, P., and Huerta, A. (2013). Streamline upwind / Petrov-Galerkin-based stabilization of proper generalized decompositions for high-dimensional advection-diffusion equations. International Journal for Numerical Methods in Engineering, 94:1216–1232. (Cited on page 89.)

[Greer et al., 2006] Greer, J. B., Bertozzi, A. L., and Sapiro, G. (2006). Fourth order partial differential equations on general geometries. Journal of Computational Physics, 216(1):216–246. (Cited on pages 7 and 10.)

[Gudi et al., 2008] Gudi, T., Nataraj, N., and Pani, A. K. (2008). Mixed Discontinuous Galerkin Finite Element Method for the Biharmonic Equation. Journal of Scientific Computing, 37(2):139–161. (Cited on page 8.)

[Guermond et al., 2006] Guermond, J., Minev, P., and Shen, J. (2006). An overview of projection methods for incompressible flows. Computer Methods in Applied Mechanics and Engineering, 195(44-47):6011–6045. (Cited on page 87.)

[Gumerov and Duraiswami, 2006] Gumerov, N. and Duraiswami, R. (2006). Fast multipole method for the biharmonic equation in three dimensions. Journal of Computational Physics, 215(1):363–383. (Cited on page 8.)

[Gupta and Kalita, 2005] Gupta, M. M. and Kalita, J. C. (2005). A new paradigm for solving Navier-Stokes equations: streamfunction-velocity formulation. Journal of Computational Physics, 207(1):52–68. (Cited on pages 48 and 86.)

[Hay, 2004] Hay, A. (2004). Étude des stratégies d'estimation d'erreur numérique et d'adaptation locale de maillages non-structurés pour les équations de Navier-Stokes en moyenne de Reynolds. PhD thesis-Ecole Centrale Nantes. (Not cited.)

[He et al., 2011] He, Z., Liu, G., Zhong, Z., Zhang, G., and a.G. Cheng (2011). A coupled ES-FEM/BEM method for fluid-structure interaction problems. Engineering Analysis with Boundary Elements, 35(1):140–147. (Cited on page 87.)

[Holdeman, 2010] Holdeman, J. T. (2010). A Hermite finite element method for incompressible fluid flow. International Journal for Numerical Methods in Fluids, 64(4):376–408. (Cited on page 86.)

[Huerta et al., 2004] Huerta, A., Vidal, Y., and Villon, P. (2004). Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. Computer methods in applied mechanics and engineering, 193(September 2003):1119–1136. (Cited on page 86.)

[Johnson and Pitkäranta, 1982] Johnson, B. C. and Pitkäranta, J. (1982). Analysis of Some Mixed Finite Element Methods Related to Reduced Integration. Mathematics of Computation, 38(158):375–400. (Cited on page 86.)

[Kalita and Gupta, 2009] Kalita, J. and Gupta, M. (2009). A streamfunction-velocity approach for 2D transient incompressible viscous flows. International journal for numerical methods in fluids, (February 2009):237–266. (Cited on pages 48 and 86.)

[Kerschen et al., 2005] Kerschen, G., Golinval, J.-c., Vakakis, A. F., and Bergman, L. a. (2005). The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview. Nonlinear Dynamics, 41(1-3):147–169. (Cited on pages 1 and 11.)

[Kim et al., 2001] Kim, J., Kim, D., and Choi, H. (2001). An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. Journal of Computational Physics, 171(1):132–150. (Cited on page 87.)

[Kim et al., 2007] Kim, Y., Kim, D. W., Jun, S., and 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. (Cited on page 48.)

[Kirby and Yosibash, 2003] Kirby, R. M. and Yosibash, Z. (2003). On the Chebyshev-Collocation Method for Bi-Harmonic Problems with Homogeneous Dirichlet Boundary Conditions. Applied Numerical Mathematics, 24:1–11. (Cited on page 31.)

[Kirby et al., 2007] Kirby, R. M., Yosibash, Z., and Karniadakis, G. E. (2007). Towards stable coupling methods for high-order discretization of fluid-structure interaction: Algorithms and observations. Journal of Computational Physics, 223(2):489–518. (Cited on page 85.)

[Kleibergen and Paap, 2006] Kleibergen, F. and Paap, R. (2006). Generalized reduced rank tests using the singular value decomposition. Journal of Econometrics, 133(1):97–126. (Cited on page 107.)

[Kolomenskiy and Schneider, 2009] Kolomenskiy, D. and Schneider, K. (2009). A Fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles. Journal of Computational Physics, 228(16):5687–5709. (Cited on page 88.)

[Ladevèze, 1999] Ladevèze, P. (1999). Nonlinear Computational Structural Mechanics: New Approaches and Non-Incremental Methods of Calculation. Springer. (Cited on pages 1 and 11.)

[Ladevèze and Nouy, 2003] Ladevèze, P. and Nouy, A. (2003). On a multiscale computational strategy with time and space homogenization for structural mechanics. Computer Methods in Applied Mechanics and Engineering, 28-30:3061–3087. (Cited on pages 1 and 11.)

[Ladevèze et al., 2010] Ladevèze, P., Passieux, J.-C., and Néron, D. (2010). The LATIN multiscale computational method and the Proper Generalized Decomposition. Computer Methods in Applied Mechanics and Engineering, 21-22:1287–1296. (Cited on pages 1 and 11.)

[Lai, 2001] Lai, M.-C. (2001). A note on finite difference discretizations for Poisson equation on a disk. Numerical Methods for Partial Differential Equations, 17(3):199–203. (Cited on page 9.)

[Leroyer et al., 2008] Leroyer, A., Barré, S., Kobus, J.-M., and Visonneau, M. (2008). Experimental and numerical investigations of the flow around an oar blade. Journal of Marine Science and Technology, 13(1):1–15. (Cited on pages 91 and 97.)

[Li et al., 2008] Li, B., Sun, Y., and Yu, Y. (2008). Iterative and direct Chebyshev collocation spectral methods for one-dimensional radiative heat transfer. International Journal of Heat and Mass Transfer, 51(25-26):5887–5894. (Cited on pages 9 and 17.)

[Li et al., 2009] Li, X., Lowengrub, J., Rätz, A., and Voigt, A. (2009). Solving PDEs in complex geometries: a diffuse domain approach. Communications in Mathematical Sciences, 7(1):81–107. (Cited on page 10.)

[Li et al., 2011] Li, Z.-C., Lee, M.-G., Chiang, J. Y., and Liu, Y. P. (2011). The Trefftz method using fundamental solutions for biharmonic equations. Journal of Computational and Applied Mathematics, 235(15):4350–4367. (Cited on pages 8 and 17.)

[Liang et al., 2007] Liang, Q., Taylor, P. H., and Borthwick, A. G. (2007). Particle mixing and reactive front motion in unsteady open shallow flow modelled using singular value decomposition. Computers & Fluids, 36(2):248–258. (Cited on page 107.)

[Liberge, 2010] Liberge, E. (2010). Reduced-order modelling by POD-multiphase approach for fluid-structure interaction. European Journal of ..., (November 2012):37–41. (Cited on pages 1 and 11.)

[Liu and Vasilyev, 2007] Liu, Q. and Vasilyev, O. (2007). A Brinkman penalization method for compressible flows in complex geometries. Journal of Computational Physics, 227(2):946–966. (Cited on page 88.)

[Lohner, 2007] Lohner, R. (2007). Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods. Wiley & Sons, page 2nd ed. (Cited on page 10.)

[Luo et al., 2009] Luo, Z., Yang, X., and Zhou, Y. (2009). A reduced finite difference scheme based on singular value decomposition and proper orthogonal decomposition for Burgers equation. Journal of Computational and Applied Mathematics, 229(1):97–107. (Cited on page 107.)

[Mahesh et al., 2004] Mahesh, K., Constantinescu, G., and Moin, P. (2004). A numerical method for large-eddy simulation in complex geometries. Journal of Computational Physics, 197:215–240. (Cited on page 10.)

[Mai-Duy and Tran-Cong, 2009] Mai-Duy, N.and See, H. and Tran-Cong, T. (2009). A spectral collocation technique based on integrated Chebyshev polynomials for biharmonic problems in irregular domains. Applied Mathematical Modelling, 33(1):284–299. (Cited on pages 9 and 20.)

[Mai-Duy and Tanner, 2005] Mai-Duy, N. and Tanner, R. (2005). Solving high-order partial differential equations with indirect radial basis function networks. International Journal for Numerical Methods in Engineering, 63(11):1636–1654. (Cited on pages 7 and 8.)

[Mai-Duy and Tanner, 2007] Mai-Duy, N. and Tanner, R. (2007). A spectral collocation method based on integrated Chebyshev polynomials for two-dimensional biharmonic boundary-value problems. Journal of Computational and Applied Mathematics, 201(1):30–47. (Cited on pages 8, 20, 21 and 40.)

[Mai-Duy et al., 2006] Mai-Duy, N., Trancong, T., and Tanner, R. (2006). A domain-type boundary-integral-equation method for two-dimensional biharmonic Dirichlet problem. Engineering Analysis with Boundary Elements, 30(10):809–817. (Cited on page 8.)

[Marin and Lesnic, 2005] Marin, L. and Lesnic, D. (2005). The method of fundamental solutions for inverse boundary value problems associated with the two-dimensional biharmonic equation. Mathematical and Computer Modelling, 42(3-4):261–278. (Cited on page 8.)

[Martinez and Esperança, 2007] Martinez, J. D. J. and Esperança, P. D. T. T. (2007). A Chebyshev Collocation Spectral Method for Numerical Simulation of Incom-

pressible Flow Problems. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 29(3):317–328. (Cited on pages 9, 17 and 20.)

[Mayo, 1984] Mayo, A. (1984). The fast solution of Poisson and the biharmonic equations on irregular regions. SIAM Journal on Numerical Analysis, 21:285–299. (Cited on page 10.)

[McCorquodale et al., 2004] McCorquodale, P., Colella, P., Grote, D. P., and Vay, J.-L. (2004). A node-centered local refinement algorithm for Poisson's equation in complex geometries. Journal of Computational Physics, 201:34–60. (Cited on page 9.)

[Mokdad et al., 2010] Mokdad, B., Ammar, a., Normandin, M., Chinesta, F., and Clermont, J. (2010). A fully deterministic micro-macro simulation of complex flows involving reversible network fluid models. Mathematics and Computers in Simulation, 80(9):1936–1961. (Cited on page 12.)

[Montagnier et al., 2013] Montagnier, J., Cadiou, A., Buffat, M., and Penven, L. L. (2013). Towards petascale spectral simulations for transition analysis in wall bounded flow. International Journal for Numerical Methods in Fluids, 72:709–723. (Cited on pages 2 and 88.)

[Montlaur et al., 2008] Montlaur, A., Fernandez-Mendez, S., and Huerta, A. (2008). Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. International Journal for Numerical Methods in fluids, 57:1071–1092. (Cited on pages 8, 17 and 48.)

[Montlaur et al., 2010] Montlaur, A., Fernandez-Mendez, S., Peraire, J., and Huerta, A. (2010). Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations. International Journal for Numerical Methods in fluids, 64:549–564. (Cited on page 48.)

[Münster et al., 2012] Münster, R., Mierka, O., and Turek, S. (2012). Finite element fictitious boundary methods (FEM FBM) for 3D particulate flow. Internatinal Journal for Numerical Methods in Fluids, 69(May 2011):294–313. (Cited on page 10.)

[Nayer, 2008] Nayer, G. D. (2008). Interaction fluide-structure pour les corps élancés. PhD thesis-Ecole Centrale Nantes. (Not cited.)

[Nestor and Quinlan, 2013] Nestor, R. and Quinlan, N. (2013). Application of the meshless finite volume particle method to flow-induced motion of a rigid body. Computers & Fluids, 88:386–399. (Cited on page 87.)

[Oñate et al., 2006] Oñate, E., Idelsohn, S. R., Celigueta, M. A., and Rossi, R. (2006). Advances in the particle finite element method for fluid-structure interaction problems. In II European Conference on Computational Mechanics, number 2006. (Cited on page 84.)

[Perssont and Strangt, 2004] Perssont, P.-o. and Strangt, G. (2004). A Simple Mesh Generator in MATLAB *. SIAM Review, 46(2):329–345. (Cited on page 10.)

[Pruliere et al., 2008] Pruliere, E., Ammar, A., ElKissi, N., and Chinesta, F. (2008). Recirculating Flows Involving Short Fiber Suspensions: Numerical Difficulties and Efficient Advanced Micro-Macro Solvers. Archives of Computational Methods in Engineering, 16(1):1–30. (Cited on page 12.)

[Pruliere et al., 2010] Pruliere, E., Chinesta, F., and 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. (Cited on pages 12 and 128.)

[Prulière et al., 2010] Prulière, E., Férec, J., Chinesta, F., and Ammar, A. (2010). An efficient reduced simulation of residual stresses in composite forming processes. International Journal of Material Forming, 3(S2):1339–1350. (Cited on page 12.)

[P.T. Baaijens, 1998] P.T. Baaijens, F. (1998). Mixed finite element methods for viscoelastic flow analysis: a review. Journal of Non-Newtonian Fluid Mechanics, 79(2-3):361–385. (Cited on page 86.)

[Queutey and Visonneau, 2007] Queutey, P. and Visonneau, M. (2007). An interface capturing method for free-surface hydrodynamic flows. Computers & Fluids, 36(9):1481–1510. (Cited on pages 2, 91 and 97.)

[Rhie and Chow, 1983] Rhie, C. M. and Chow, W. L. (1983). Numerical study of the turbulent flow past an airfoil with trailing edge separation. American Institute of Aeronautics and Astronautics, 21(11):1525–1532. (Cited on pages 91 and 95.)

[Rozza et al., 2008] Rozza, G., Huynh, D. B. P., and Patera, a. T. (2008). Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. Archives of Computational Methods in Engineering, 15:229–275. (Cited on pages 1 and 11.)

[Sadat et al., 2012] Sadat, H., Wang, C.-A., and Le Dez, V. (2012). Meshless method for solving coupled radiative and conductive heat transfer in complex multi-dimensional geometries. Applied Mathematics and Computation, 218(20):10211–10225. (Cited on page 10.)

[Saiki and Biringen, 1996] Saiki, E. and Biringen, S. (1996). Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. Journal of Computational Physics, 465(123):450–465. (Cited on page 88.)

[Sakami et al., 1996] Sakami, M., Charette, a., and Le Dez, V. (1996). Application of the discrete ordinates method to combined conductive and radiative heat transfer in a two-dimensional complex geometry. Journal of Quantitative Spectroscopy and Radiative Transfer, 56(4):517–533. (Cited on page 10.)

[Shang and He, 2009] Shang, Y.-q. and He, Y.-n. (2009). Fourier analysis of Schwarz domain decomposition methods for the biharmonic equation. Applied Mathematics and Mechanics, 30(9):1177–1182. (Cited on page 8.)

[Shao et al., 2012] Shao, W., Wu, X., and Chen, S. (2012). Chebyshev tau meshless method based on the integration–differentiation for Biharmonic-type equations on irregular domain. Engineering Analysis with Boundary Elements, 36(12):1787–1798. (Cited on page 9.)

[Speetjens and Clercx, 2005] Speetjens, M. and Clercx, H. (2005). A spectral solver for the Navier-Stokes equations in the velocity-vorticity formulation. International Journal of Computational Fluid Dynamics, 19(3):191–209. (Cited on page 87.)

[Squires and Quake, 2005] Squires, T. and Quake, S. (2005). Microfluidics: Fluid physics at the nanoliter scale. Reviews of modern physics, 77:977–1026. (Cited on page 84.)

[Stewart, 1992] Stewart, G. (1992). On the Early History of the Singular Value Decomposition. SIAM Review, 35(4):551–566. (Cited on page 106.)

[Stykel, 2006] Stykel, T. (2006). Balanced truncation model reduction for semidiscretized Stokes equation. Linear Algebra and its Applications, 415(2-3):262–289. (Cited on page 11.)

[Sugiyama et al., 2011] Sugiyama, K., Ii, S., Takeuchi, S., Takagi, S., and Matsumoto, Y. (2011). A full Eulerian finite difference approach for solving fluid-structure coupling problems. Journal of Computational Physics, 230(3):596–627. (Cited on page 86.)

[Trujillo and Em Karniadakis, 1999] Trujillo, J. and Em Karniadakis, G. (1999). A Penalty Method for the Vorticity-Velocity Formulation. Journal of Computational Physics, 149(1):32–58. (Cited on page 87.)

[Vierendeels et al., 2007] Vierendeels, J., Lanoye, L., Degroote, J., and Verdonck, P. (2007). Implicit coupling of partitioned fluid-structure interaction problems with

reduced order models. Computers & Structures, 85(11-14):970–976. (Cited on page 11.)

[Wackers et al., 2011] Wackers, J., Koren, B., Raven, H. C., Ploeg, a., Starke, a. R., Deng, G. B., Queutey, P., Visonneau, M., Hino, T., and Ohashi, K. (2011). Free-Surface Viscous Flow Solution Methods for Ship Hydrodynamics. Archives of Computational Methods in Engineering, 18(1):1–41. (Cited on page 92.)

[Wackers et al., 2010] Wackers, J., Said, K. A., Deng, G., Queutey, P., and Visonneau, M. (2010). Adaptive grid refinement applied to RANS ship flow computation. 28th Symposium on Naval Hydrodynamics Pasadena, California, 12-17 September 2010, (September):1–18. (Cited on page 92.)

[Wang et al., 2010] Wang, C.-A., Sadat, H., Ledez, V., and Lemonnier, D. (2010). Meshless method for solving radiative transfer problems in complex two-dimensional and three-dimensional geometries. International Journal of Thermal Sciences, 49(12):2282–2288. (Cited on page 10.)

[Wang, 2012] Wang, D. (2012). Sensitivity Analysis and Shape Optimization of a Hole in a Vibrating Rectangular Plate for Eigenfrequency Maximization. Journal of Engineering Mechanics, 138:662–674. (Cited on page 9.)

[Wang, 2004] Wang, T. (2004). A mixed finite volume element method based on rectangular mesh for biharmonic equations. Journal of Computational and Applied Mathematics, 172(1):117–130. (Cited on page 8.)

[Weibin and Xionghua, 2009] Weibin, K. and Xionghua, W. (2009). Chebyshev tau matrix method for Poisson-type equations in irregular domain. Journal of computational and applied mathematics, 228(1):158–167. (Cited on page 9.)

[White, 2009] White, F. M. (2009). FLUID MECHANICS. (Cited on page 83.)

[White et al., 1987] White, S. A., Gotsis, A. D., and Baird, D. G. (1987). Review of the entry flow problem:Experimental and numerical. Journal of Non-Newtonian Fluid Mechanics, 24:121–160. (Cited on page 83.)

[Whitesides, 2006] Whitesides, G. M. (2006). The origins and the future of microfluidics. Nature, 442(7101):368–73. (Cited on page 84.)

[Whiting and Jansen, 2001] Whiting, C. and Jansen, K. (2001). A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis. Internatinal Journal for Numerical Methods in Fluids, 35:93–116. (Cited on page 87.)

[Witelski and Bowen, 2003] Witelski, T. and Bowen, M. (2003). ADI schemes for higher-order nonlinear diffusion equations. Applied Numerical Mathematics, 45(2-3):331–351. (Cited on page 7.)

[Yoon and Sigmund, 2008] Yoon, G. H. and Sigmund, O. (2008). A monolithic approach for topology optimization of electrostatically actuated devices. Computer Methods in Applied Mechanics and Engineering, 197(45-48):4062–4075. (Cited on page 9.)