

Numerical Analysis of Axisymmetric Flows and Methods for Fluid-Structure Interaction Arising in Blood Flow Simulation.

THÈSE N° 2965 (2004)

PRÉSENTÉE À LA FACULTÉ SCIENCES DE BASE

Institut d'analyse et de calcul scientifique

SECTION DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Simone DEPARIS

mathématicien diplômé EPF
de nationalité suisse et originaire de Lumino (TI)

acceptée sur proposition du jury :

Prof. A. Quarteroni, directeur de thèse
Dr. J.-F. Gerbeau, rapporteur
Prof. J. Rappaz, rapporteur
Prof. A. Robertson, rapporteur

Lausanne, EPFL
2004

Riassunto

Gli obiettivi di questa tesi sono l'analisi numerica di flussi assialsimmetrici e la descrizione di algoritmi adatti alla risoluzione di problemi di interazione fluido-struttura. Questo lavoro è motivato dalla simulazione di flussi emodinamici, ma presenta carattere di generalità.

La prima parte di questo lavoro si concentra su un modello per flussi incomprimibili tridimensionali basato sulla formulazione assialsimmetrica delle equazioni di Stokes o Navier-Stokes; il dominio di calcolo e il campo vettoriale della velocità si riducono a due dimensioni. In particolare si mostrano stime *a priori* ottimali per elementi finiti assialsimmetrici P1isoP2/P1 per le equazioni di Stokes, ipotizzando che il dominio e i dati siano assialsimmetrici e che i dati abbiano componente angolare nulla. Questa analisi utilizza spazi di Sobolev pesati e un operatore di proiezione di tipo Clément.

In seguito si introducono una formulazione assialsimmetrica delle equazioni di Navier-Stokes in domini mobili. Partendo da risultati esistenti in tre dimensioni, si deducono una formulazione *Lagrangiana-Euleriana arbitraria* (ALE) e risultati di stabilità.

Nella seconda parte ci si occupa di algoritmi per la soluzione di problemi di interazione fluido-struttura. Il problema è introdotto in una forma generale con equazioni di Navier-Stokes incomprimibili per il fluido e un modello viscoelastico per la struttura. Si tiene conto di deformazioni della struttura relativamente grandi e si mostra come estendere algoritmi esistenti al fine di ridurre il tempo computazionale.

In primo luogo, si mostra come condizioni al contorno di traspirazione possano essere utilizzate in una strategia di punto fisso. In secondo luogo, come ridurre il tempo di calcolo utilizzando un algoritmo di Newton con approssimazioni della matrice jacobiana basate su modelli fisici semplificati. Inoltre, per accelerare l'algoritmo di Newton vengono proposti un preconditionatore dinamico per la risoluzione della matrice jacobiana e uno schema di accelerazione; entrambi sono stati testati in simulazioni di flussi emodinamici in due e tre dimensioni.

Abstract

In this thesis we propose and analyze the numerical methods for the approximation of axisymmetric flows as well as algorithms suitable for the solution of fluid-structure interaction problems. Our investigation is aimed at, but are not restricted to, the simulation of the blood flow dynamics.

The first part of this work deals with an axisymmetric fluid model based on three-dimensional incompressible Stokes or Navier–Stokes equations which are solved on a two-dimensional half-section of the domain under consideration. In particular we show optimal *a priori* error estimates for P1isoP2/P1 axisymmetric finite elements for the steady Stokes equations under the assumption that the domain and the data are axisymmetric and that the data have no angular component. Our analysis is carried out in the framework of weighted Sobolev spaces and takes advantage of a suitably defined Clément type projection operator.

We then introduce an axisymmetric formulation of the Navier–Stokes equations in moving domains and, starting from existing results in three-dimensions, we set up an *Arbitrary Lagrangian–Eulerian* (ALE) formulation and prove some stability results.

In the second part, we deal with algorithms for the solution of fluid-structure interaction problems. We introduce the problem in a generic form where the fluid is described by means of incompressible Navier–Stokes equations and the structure by a viscoelastic model. We account for large deformations of the structure and we show how existing algorithms may be improved to reduce the computational time.

In particular we show how to use transpiration boundary conditions to approximate the fluid-structure problem in a fixed point strategy. Moreover, in a quasi-Newton strategy we reduce the cost by replacing the Jacobian with inexact Jacobians stemming from reduced physical models for the problem at hand. To speed up the convergence of the Newton algorithm, we also define a dynamic preconditioner and an acceleration scheme which have been successfully tested in haemodynamics simulations in two and three dimensions.

A Cécile, Nicola e Giulia

Acknowledgments¹

First of all, I would like to thank Prof. Alfio Quarteroni who has encouraged me from the beginning, for the dynamism and enthusiasm he passes on to the people he is working with. He gave me the opportunity to work in his team and introduced me to the fascinating modeling and scientific computing. He has given me a lot of his time to guide me in my research and he also stimulated me to work with other people in the same area. This has been very important to me and to the research reported herein. I am particularly grateful to him for the patience and the confidence he always had (and has).

I am especially grateful to Dr. Fabio Nobile who taught me the feelings related to the fluid-structure interaction problem as well as his *cool* in facing coding and maths problems.

Prof. Luca Formaggia and Prof. Alessandro Veneziani deserve my thanks for the time they spent in explanations, suggestions and discussions on mathematical topics and the help in overcoming the numerous difficulties encountered. Luca also for the hints on the Linux operating system.

I am no less grateful to Dr. Xavier Vasseur for his great collaboration in Fortran coding and children growing. Dr. Miguel Fernandez deserves my special thanks for the animated discussions on any mathematical topics and Dr. Jean-Frederic Gerbeau for his precious remarks on the algorithmic aspects of fluid-structure interaction as well as for reading this thesis as member of the jury.

I also thank Dr. Christine Bernardi and Dr. Zakaria Belhachmi for their fruitful discussions and collaborative work on the theoretical aspects of the axisymmetric formulation of the Stokes problem. They were very patient with me and taught me many things on numerical analysis.

Special thanks to Prof. Robert Dalang, not only for being the president of the jury, but also for having supported me in carrying out a PhD since I met him long time ago. I want to thank Prof. Jacques Rappaz for reading this manuscript as member of the jury, as well as for the interesting discussions we had about the organization of the IACS and more generally of the EPFL. Special thanks to Prof. Anne Robertson for her kindness, for carefully reading the manuscript and for the long trip from Pittsburgh to be present as member of the jury.

The whole crew of the CMCS at the EPFL receives also my gratitude, for sharing many experiences, discussions and sandwiches. Special thanks to Nicola Parolini for the discussions on the Navier–Stokes solvers and for the babysitting, to Daniele Lamponi for those on the one-dimensional model for the haemodynamics and the discussions about the Swiss mountains and to Marco Discacciati for the precious remarks on domain decomposition. Too bad that he supports another operating system!

Ringrazio mia madre Annamaria, per avermi sempre spinto e sostenuto nei miei studi. Un grand merci à mes beaux-parents Annie et Pierre pour leur soutien.

Je dédie cette thèse à Cécile, amour de ma vie, que je remercie de tout mon cœur pour son soutien et ses encouragements.

¹The ETH board has supported this work through a stipend for exchange between ETHZ and EPFL.

Contents

Introduction	1
I Numerical Solution of Axisymmetric Flows	7
1 Axisymmetric formulation, analysis and approximation of Stokes equations	9
1.1 Assumptions and definitions	10
1.2 Axisymmetric formulation and analysis	11
1.2.1 Weighted Sobolev spaces	12
1.2.2 Dimension reduction	13
1.2.3 The weak axisymmetric form	14
1.3 Finite element approximation	14
1.4 Finite element analysis	16
1.4.1 Weighted inverse inequalities	16
1.4.2 Inf-Sup condition	19
1.4.3 Existence, uniqueness and a priori error estimates	21
1.5 Technical results	22
1.5.1 Weighted approximation properties	22
1.5.2 Axisymmetric Inf-Sup condition in $(V_1^1(\Omega) \times H_1^1(\Omega)) \times H_1^1(\Omega)$	31
2 Axisymmetric Navier–Stokes equations	35
2.1 The steady case	35
2.2 The unsteady case of a moving domain	37
2.2.1 Conservative weak ALE formulation	37
2.2.2 Construction of the ALE mapping	40
2.2.3 Finite element discretization	41
2.2.4 Time discretization	43
2.3 Algebraic aspects	44
2.3.1 Algebraic formulation	44
2.3.2 Quadrature formula	46
2.3.3 Computational aspects	47
2.3.4 Inexact factorization schemes	49
2.4 Defective boundary conditions	52
2.5 Some numerical results	54

II	Fluid-Structure Interaction	57
3	A fluid-structure interaction problem	59
3.1	Formulation of the fluid-structure problem	59
3.1.1	The governing continuum equation	59
3.1.2	Weak formulation	62
3.2	Time discretization	64
3.3	An abstract formulation	66
3.3.1	Newton based algorithms for the solution of the fixed-point problem .	67
3.3.2	Computation of the Jacobian against a given vector	68
3.3.3	Strategies for the solution of the non-linear coupled system	70
3.4	A domain decomposition formulation approach	71
3.4.1	Stokes problem	71
3.4.2	Fluid-structure interaction	73
4	Efficient solution of BGS iterations	77
4.1	Block Gauss Seidel algorithm	77
4.1.1	Residual of BGS	79
4.1.2	Fluid's residual's norm	80
4.2	The role of Aitken extrapolation	82
4.2.1	Problem setting	82
4.2.2	Scalar Aitken method	82
4.2.3	Extension to the vector case	84
4.2.4	Minimizing on ω^{-1}	85
4.2.5	Approximating the multiplicity	87
4.2.6	Variants	87
4.3	Transpiration interface conditions	90
4.3.1	Confidence interval	94
4.3.2	Description of the algorithm	94
4.4	Numerical experiments	96
4.4.1	Two-dimensional test	96
4.4.2	Three-dimensional test	98
5	Preconditioning of Newton and quasi-Newton algorithms for the solution of the fully implicit problem	101
5.1	Newton	102
5.2	How to compute the Jacobian	103
5.3	How to compute approximate Jacobian	103
5.4	Preconditioned Krylov iterations for the Jacobian system	106
5.4.1	The GMRES iterative method	106
5.4.2	Dynamic initial guess	107
5.4.3	Dynamic preconditioner	108
5.4.4	Invertibility of the preconditioner	109
5.4.5	Application to a sequence of problems	110
5.5	Nonlinear acceleration of the Newton–Krylov algorithm	112
5.5.1	Acceleration strategy	112
5.5.2	Global procedure	114

CONTENTS

5.6	Application to fluid-structure interaction	116
5.6.1	Settings	116
5.6.2	Three-dimensional test case	116
5.6.3	Two-dimensional test case	118
5.7	Conclusion	122
Conclusions		127
A Useful properties needed in the analysis of the axisymmetric Stokes problem		129
A.1	Lemmas needed by proposition 1.5.2	129
A.2	A result on the divergence operator	130

Introduction

The main interest which has driven our work is the mathematical modeling of the arterial vascular system. Since the blood flow is pulsatile and arterial walls comply with the systolic-diastolic heart beat, a mathematical model should account for the blood dynamics, the wall dynamics and their interaction.

Modeling arterial wall mechanics is a very challenging task, due to the complex multi-layered structure of arteries (see, e.g., Fung [Fun55] and Nichols and O'Rourke [NO98]). In the past few years, simplified models have been proposed for reducing the complexity of the numerical simulations (see, e.g., Le Tallec [LT94], Anicic and Léger [AL99], Quarteroni, Tuveri and Veneziani [QTV00], Nobile [Nob01] or Anicic, Le Dret and Raoult [ALDR03]).

Despite the complexity of the physical nature of blood (composite of plasma, red blood cells, etc., see, e.g., Nichols and O'Rourke [NO98] or, from a mathematical view point, Arada and Sequeira [AS03] or Padula and Sequeira [PS98]), it is often useful to model blood as a single phase, incompressible, homogeneous, linearly viscous fluid. However, in large arteries (with radius larger than about 0.3 cm) blood's behavior does not depart significantly from that of a Newtonian fluid (see for example Quarteroni, Tuveri and Veneziani [QTV00]), therefore in this work it will be modeled by three-dimensional unsteady Navier–Stokes equations. However their approximation, even in a single artery, requires substantial computational resources. This pushes forward the use of reduced models, i.e., mathematical models that feature a lower computational complexity. In this respect, one approach consists of using one-dimensional models (the flow being three-dimensional but the computational domain and the velocity flow field are one-dimensional, see for example Sherwin, Formaggia and Peiró [SFP01], Formaggia, Nobile and Quarteroni [FNQ02] or Sherwin et al. [SFPP03, SFPP03]); see also Robertson and Sequeira [RS03] where a director theory derived from Cosserat theory for solid mechanics is successfully employed. A different approach consists of using two-dimensional models which are derived for the mean longitudinal section of the artery (see Formaggia et al. [FGNQ01] or Nobile [Nob01]).

One case where the computational effort is reduced but not the dimension of the model, is when the domain, the initial and the boundary conditions as well as the body forces can be approximated as symmetric with respect to a straight axis. In such a case it is possible to model the blood by three-dimensional axisymmetric Stokes or Navier–Stokes equations which take advantage of the hypothesis of symmetry (cf. Bernardi, Dauge and Maday in [BDM99]). Then the computational domain reduces to a two-dimensional one (see figure 1) and if, in addition, the angular component of the data can be neglected, the velocity of the fluid is characterized by its axial and radial components (the angular one being zero). This is important since the size of the problem is reduced without losing three-dimensional features and without any assumption on the velocity profile.

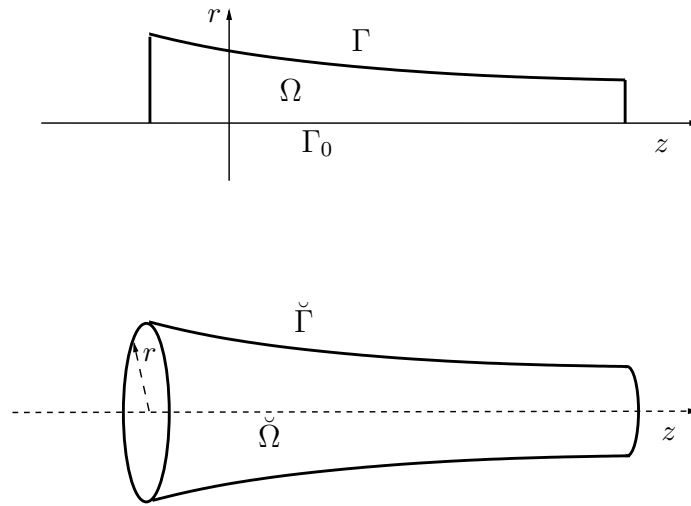


Figure 1: An example of an axisymmetric domain $\tilde{\Omega}$ and its half section Ω .

This reduction has several applications. For example, in view of a modeling of the complete cardiovascular system, the coupling of different models has been considered in Quarteroni, Tuveri and Veneziani [QTV00], Laganà et al. [LDM⁺02], Quarteroni [Qua02], Quarteroni and Formaggia [QF02] and Formaggia and Veneziani [FV03]. The idea is to use a very simple model for the global cardiovascular system (for example lumped models based on the electric analogy) and the full model limitedly to those regions where one wants to accurately simulate the local effects of the circulation. The axisymmetric model may be used as interface between a simple model and a full three-dimensional one. In particular it can be easily coupled with an axisymmetric one-dimensional model (see, e.g., [Lam04]). A coupling with a lumped model is under consideration (see Vergara [Ver]).

Since the axisymmetric model can reproduce three-dimensional effects, it can also be used as “fine model” in the region of interest. For example, when dealing with a stent on a straight tract of artery, this model can replace a full three-dimensional one, since in a first approximation we can consider the geometry and the data axisymmetric.

Part I of this thesis is devoted to the analysis of the axisymmetric model. In particular, we propose a numerical approximation of the axisymmetric Stokes equations by the finite element method. We consider the so-called P1isoP2/P1 elements (see, e.g., Brezzi and Fortin [BF91] or Quarteroni and Valli [QV94]). We prove existence, stability and error estimate for the steady Stokes equations (chapter 1) and stability properties for the time advancing schemes in the case of unsteady Navier–Stokes equations (chapter 2). We build the algebraic system associated to the discrete Navier–Stokes problem and recall some solution techniques based on suitable splitting methods.

Another aspect which is very crucial in blood flow analysis is the dynamical coupling of the blood flow and the arterial wall. Fluid-structure interaction is not an exclusive feature of haemodynamics. Indeed it appears in many other applications in physics and engineering. For instance, aeroelasticity problems or fluttering of wings and structures: the action of the wind on a bridge induces the oscillations of the bridge, while on a sail it generates the lift which moves a sailing boat.

The solution of the mechanical coupling requires algorithms that correctly describe the energy transfer between the blood flow and the vessel wall (see Farhat, Lesoinne and Le Tallec [FLL98], Grandmont [Gra98], Desjardins and Esteban [DE99, DE00], Grandmont and Maday [GM00, GM01], Desjardins et al. [DEGL01]). This aspect is particularly relevant in large arteries, where the vessel wall radius may vary up to 10% between systolic and diastolic pulses. A numerical simulation of fluid-structure interaction helps understanding the effects of geometry or physical changes of the vessel, as in a coronary bypass or a partial replacement of the aortic arc (cf. Migliavacca et al. [MPD⁺01] or Corno et al. [CSB⁺03]).

One numerical challenge when facing fluid-structure problems involving large displacements, is the definition of fast and accurate coupling algorithms, that allow the prediction of long-term time evolution maintaining the stability of the overall system. At each time step, we have to solve a highly coupled non-linear system using efficient methods that preserve, inside inner loops, the fluid-structure subsystem splitting.

Standard strategies to solve this non-linear system are fixed-point based methods such as Block-Jacobi or Block-Gauss-Seidel (BGS) iterations (see, e.g., Codina and Cervera [CC96], Le Tallec and Mouro [LM01], Nobile [Nob01]). A BGS iterations method with constant relaxation parameter may be effective, provided this parameter is chosen judiciously. In fact, if the parameter is too large it may lead to divergence of the algorithm, whereas if it is too small, the convergence is slowed down considerably. As pointed out in the literature (e.g. Nobile [Nob01]), this choice depends, among other factors, on the domain geometry, the characteristics of the fluid and the structure, as well as the boundary data. A better strategy proposed by Irons and Tuck [IT69] and Mok, Wall and Ramm [MWR01], consists of using an Aitken-like method to choose the parameter dynamically, based solely on the computed solutions of the two previous iterations.

Standard BGS iterations are very expensive. Indeed, besides slow convergence one has to account, at each iteration, for the cost of updating the fluid mesh and the corresponding fluid matrices.

More recent approaches make use of Block-Newton methods on the non-linear coupled problem (see, e.g., Matthies and Steindorf [MS00], Gerbeau and Vidrascu [GV03], Heil [Hei03], Fernández and Moubachir [FM03, FM04]). Since the computation of the full Jacobian of the coupled system would be very expensive, the generalized minimal residual (GMRES) iterative method (see [SS86]) is applied so that one only needs to compute the action of the Jacobian on intermediate residuals (see Fernández and Moubachir [FM04]). Still, this can be a difficult operation; for that reason an inexact Jacobian computation has been proposed by Gerbeau and Vidrascu [GV03], which is derived by applying a reduced (much simpler) physical model. The obvious advantage is the fast computation of the Jacobian, however the drawback is that, in some cases, the time step must be smaller than the one allowed for the exact version.

In part II, we show how to modify existing algorithms for fluid-structure interaction problems in order to improve their efficiency. In the two considered cases, BGS and Newton algorithms, the modifications do not change the structure of the algorithms, such that existing codes may be easily adapted. For example, we propose the Aitken extrapolation method (section 4.2) to compute the relaxation parameter, which requires only two additional variables; besides, to apply the zeroth order transpiration condition (section 4.1), the BGS algorithm just needs to be shortened during some iterations. Also, an existing Newton or quasi-Newton algorithm may be completed in order to take into account the proposed preconditioner in the GMRES procedure that inverts the Jacobian (chapter 5). A slightly bigger coding effort is

needed in order to benefit from the acceleration step (section 5.5.1), since we need to store some values of the residual history and to solve a linear minimization problem.

What follows is a short description of the main contributions of this work. We present first some results concerning the numerical modeling of axisymmetric flows and then some improvements of existing algorithms for the solution of fluid-structure interaction problems.

Numerical solution of axisymmetric flows

In part I, we focus on axisymmetric Stokes and Navier–Stokes equations. In [BDM99], Bernardi, Dauge and Maday show the existence of axisymmetric solutions of the steady equations and how to discretize them by spectral methods. We discretize the weak axisymmetric Stokes problem with PlisoP2/P1 finite elements and we show existence and optimal *a priori* error estimates. Let Ω be a two-dimensional half section of the axisymmetric three-dimensional domain $\check{\Omega}$ under consideration (see figure 1) and V and Q be suitable weighted Sobolev spaces. Assume that the data are axisymmetric with zero angular component. The axisymmetric Stokes problem reads:

P0.1 Find (\mathbf{u}, p) , $\mathbf{u} = (u_r, u_z)$, in $V \times Q$ such that

$$\begin{cases} \nu \int_{\Omega} (\nabla \mathbf{u} : \nabla \mathbf{v}) r d\mathbf{x} + \nu \int_{\Omega} u_r v_r \frac{1}{r} d\mathbf{x} - \int_{\Omega} (\operatorname{div} \mathbf{v}) p r d\mathbf{x} - \int_{\Omega} v_r p d\mathbf{x} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} r d\mathbf{x}, \\ - \int_{\Omega} (\operatorname{div} \mathbf{u}) q r d\mathbf{x} - \int_{\Omega} u_r q d\mathbf{x} = 0, \end{cases}$$

for all (\mathbf{v}, q) in $V \times Q$.

To recover the three-dimensional solution $(\check{\mathbf{u}}, \check{p})$ from (\mathbf{u}, p) , the three-dimensional domain $\check{\Omega}$ is described in cylindrical coordinates (r, θ, z) . Then

$$\check{\mathbf{u}}(r, \theta, z) = \begin{pmatrix} \check{u}_x \\ \check{u}_y \\ \check{u}_z \end{pmatrix} = \begin{pmatrix} u_r(r, z) \cos \theta \\ u_r(r, z) \sin \theta \\ u_z(r, z) \end{pmatrix}$$

and

$$\check{p}(r, \theta, z) = p(r, z).$$

We make use of weighted Sobolev spaces and we define a projection operator of Clément type which allows us to prove optimal *a priori* error estimates:

Theorem 0.0.1 *The discretized axisymmetric Stokes problem has a unique axisymmetric solution $(\check{\mathbf{u}}_h, \check{p}_h)$ without angular component. Furthermore, if $\check{\mathbf{u}}$ is in $H^2(\check{\Omega})^2$ and \check{p} in $H^1(\check{\Omega})$, then there exists a constant C such that*

$$\|\check{\mathbf{u}} - \check{\mathbf{u}}_h\|_{H^1(\check{\Omega})^2} + \|\check{p} - \check{p}_h\|_{L^2(\check{\Omega})} \leq Ch \left(\|\check{\mathbf{u}}\|_{H^2(\check{\Omega})^2} + \|\check{p}\|_{H^1(\check{\Omega})} \right),$$

where $h > 0$ is the length of the longest side of the finite element triangulation.

When dealing with the steady Navier–Stokes equations, Bernardi, Dauge and Maday in [BDM99] prove that under the same assumptions on the domain and the data, the Navier–Stokes equations expressed in cylindrical coordinates can be split in two coupled problems,

one for the pressure, the axial and radial components of the velocity and the other for the angular component. We deduce that, under the additional assumptions that the data have zero angular component, the problem can be reduced to the half section Ω with unknown (\mathbf{u}, p) , $\mathbf{u} = (u_r, u_z)$.

We then formulate the unsteady axisymmetric Navier–Stokes equations in moving domains using an *Arbitrary Lagrangian–Eulerian* (ALE) formulation, which derives from the classical three-dimensional equations by a change of variable and integrating over the angular coordinate. We propose a time discretization and a semi-implicit treatment of the convective field, show stability results and write the corresponding linear system in matrix form.

We also perform a numerical test on a Womersley flow on a tube of fixed radius and length. The computational domain is a moving domain inside the flow and we impose exact Dirichlet boundary conditions on the immersed boundary. The rate of convergence of the unsteady Navier–Stokes solution with respect to h obeys the theoretical estimate proved for the steady Stokes case.

Fluid-structure interaction

In part II, we introduce a fluid-structure interaction problem, not restricted to the axisymmetric case previously addressed, and propose two abstract frameworks that allow the development of algorithms for the solution of the coupled problem. The first one concerns Newton algorithms with an approximated Jacobian (see also Fernández and Moubachir [FM04], and Gerbeau and Vidrascu [GV03]). The second one is a domain decomposition interpretation for a class of algorithms for the solution of the fluid-structure interaction problem. This allows to interpret existing sub-domain iterative methods (see for example Quarteroni and Valli [QV94]), such as Dirichlet–Neumann or Neumann–Neumann, in the fluid-structure interaction framework.

We then describe the coupled problem as a fixed-point problem on the interface displacement and we propose a modified fixed-point algorithm which combines the BGS iterations with transpiration boundary conditions on the interface.

During some BGS iterations we freeze the computational domain of the fluid and replace the boundary conditions on the interface by transpiration conditions, which are derived by a Taylor expansion of the fluid velocity on the frozen domain. This allows us to keep the matrices associated with the flow discretization unchanged for some iterations. After convergence on the modified problem, the fluid domain is updated and we follow up with the standard BGS-algorithm. This brings considerable advantage in computational efficiency.

To solve the fully-coupled problem we analyze the Newton algorithm and discuss how to accelerate its convergence. Gerbeau and Vidrascu [GV03] propose to approximate the Jacobian by neglecting the non-linear and viscous terms in the linearized fluid problem and the fluid domain is frozen about its current state. We propose to only freeze the fluid domain and to apply boundary conditions stemming from transpiration techniques. With this approach, the number of Newton iterations is smaller than with the previous one, but numerical experiments show that for a fixed time step, the CPU time is smaller with the model in [GV03]. However, since our approximation is closer to the exact Jacobian, the proposed approach may be more efficient and may allow a larger time step, depending on the situation.

We also propose an original preconditioner for the GMRES iterations to solve the Jacobian system (or an approximation of it). We store the partial QR -decomposition of the Jacobian

J as well as the Krylov space \mathcal{L} computed during the GMRES iterations and we use them at the following Newton iteration. In particular this decomposition yields

$$QR = J|_{\mathcal{L}},$$

where $J|_{\mathcal{L}}$ denotes the restriction of J to \mathcal{L} . Then the preconditioner generated in the first Newton iteration is defined as

$$P_1^{-1} = R_1^{-1} Q_1^T \Pi_{\text{Im}(J_1|_{\mathcal{L}_1})} + \frac{1}{\lambda} \left(Id - \Pi_{\text{Im}(J_1|_{\mathcal{L}_1})} \right),$$

where λ is a scalar to be chosen and $\Pi_{\text{Im}(J_1|_{\mathcal{L}_1})}$ is the orthogonal projection on $\text{Im}(J_1|_{\mathcal{L}_1})$. We prove that this preconditioner is well defined provided that J_1 is non-singular. In the second Newton iteration we apply P_1 as right preconditioner and we show that it is possible to nestle the preconditioners in the next Newton iterations. As a result we have a sequence of nested preconditioners which is cleared at each new time step.

We also propose an acceleration step based on the linearization of the residual near the solution and on the replacement of the Jacobian by the preconditioner built with the previous GMRES iterations.

We successfully apply this approach in the Newton algorithm with two approximations of the Jacobian with a considerable gain in CPU time in two and three-dimensional experiments. We also show in a two-dimensional experiment, that at a fixed time, we can switch from a simplified model for the Jacobian approximation to another. In particular, it is possible to build the first preconditioner with the simplest model (which is cheap) in the first Newton iteration and then use a more sophisticated (which is expensive but more effective) in the following Newton iterations.

Part I

Numerical Solution of Axisymmetric Flows

Chapter 1

Axisymmetric formulation, analysis and approximation of Stokes equations

Introduction

Numerical simulation of three-dimensional incompressible flows by finite elements may feature a very high computational complexity. Reducing the dimension of the problem is sometimes of paramount interest. A simple approach consists of using Stokes or Navier–Stokes equations in two dimensions and solve them with finite elements. This significantly reduces the size of the problem, but several three-dimensional features are not present in the model. If the problem is set in a domain which is symmetric by rotation around an axis, it is proved in [BDM99] that, when using a Fourier expansion with respect to the angular variable, the three-dimensional Stokes problem is equivalent to a system of two-dimensional problems on the meridian domain, each problem being satisfied by a Fourier coefficient of the solution. So it is possible to reduce its size without losing three-dimensional features.

Here we are going to present an axisymmetric model which supposes data with angular component equal to zero. The advantage is that its discretization results in a linear system of the same size as a two-dimensional one. In this case, all the Fourier coefficients of the solution but the one of order zero vanish. So the number of unknowns in its discretization is the same as in the Cartesian two-dimensional one. The only further difficulty is that the variational formulation requires weighted scalar product (and the analysis has to be carried out in weighted Sobolev spaces).

For the discretization of the Stokes problem we have chosen to work with P1isoP2/P1 finite elements: The approximation of the pressure makes use of continuous piecewise affine functions and the approximation of the velocity components relies also on continuous piecewise affine functions but on a finer mesh. We refer to Ying in [Yin86] and Tabata [Tab96] for the numerical analysis of the discretization by other types of finite elements in a similar framework.

As usual, the numerical analysis of the discrete problem relies on an inf-sup condition (Brezzi [Bre74]). For the discretization by Taylor–Hood elements of the two-dimensional Stokes problem in the Cartesian case, Bercovier and Pironneau in [BP79] prove an inf-sup condition and Verfürth in [Ver84] refines the analysis of these elements. Our aim is to extend these results to the axisymmetric case.

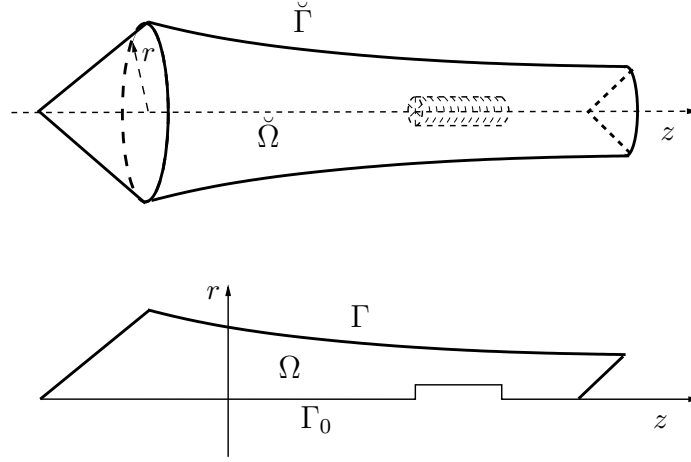


Figure 1.1: An example of an axisymmetric domain $\check{\Omega}$ and its half section Ω . There is an obstacle on the axis, hence Γ_0 is a union of two disjoint segments.

The proof of the inf-sup condition in [Ver84] needs a very accurate approximation property of the discrete spaces, involving both the usual Lagrange interpolation operator and the Clément projection operator (see [Clé75]). One of the main parts of this chapter is devoted to the extension of the properties of these operators to the weighted Sobolev spaces. A first work in this subject is due to Mercier and Raugel (see [MR82]). However the results therein are not sufficient for our needs.

Once these results are established we prove an optimal inf-sup condition for the discrete spaces and optimal a priori error estimates.

An extension of these results to Navier–Stokes equations may be found in the following chapter.

The proofs and the results reported in this chapter have been already submitted in a paper by Belhachmi, Bernardi and Deparis, *Weighted Clément operator and application to the finite element discretization of the axisymmetric Stokes problem* [BBD].

1.1 Assumptions and definitions

We are interested in modeling a flow in a domain $\check{\Omega}$ symmetric with respect to the z axis (see figure 1.1). We use cylindrical coordinates (r, θ, z) and we note Ω the half section $(r, 0, z)$. On the boundary $\check{\Gamma}$ of the physical domain $\check{\Omega}$ we impose a Dirichlet boundary condition. Γ denotes the half section of $\check{\Gamma}$ and Γ_0 the intersection of $\check{\Omega}$ with the axis, such that $\partial\Omega$ is the union of Γ and Γ_0 . All vector-fields on $\check{\Omega}$ are expressed in cylindrical coordinates.

The fluid is modeled by Stokes equations in the domain $\check{\Omega}$ and we suppose that the boundary condition and the external forces are axisymmetric and that their angular component is zero.

Scalar functions \check{p} or vector-fields $\check{\mathbf{u}}$ on $\check{\Omega}$ are *axisymmetric* (with respect to the z -axis), if for any rotation \mathcal{R}_η around the z -axis of an arbitrary angle η in $[-\pi, \pi)$, it holds

$$\begin{aligned}\check{p} \circ \mathcal{R}_\eta &= \check{p}, \\ \mathcal{R}_{-\eta}(\check{\mathbf{u}} \circ \mathcal{R}_\eta) &= \check{\mathbf{u}},\end{aligned}$$

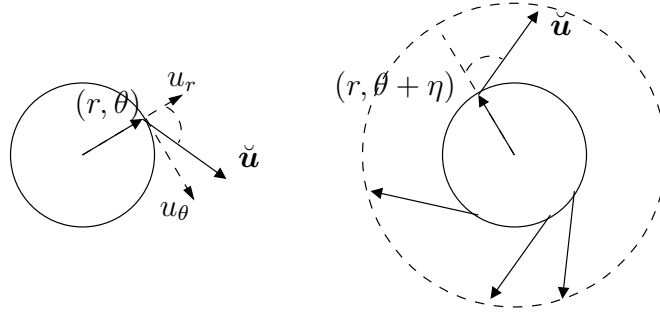


Figure 1.2: Axial section of an axisymmetric vector field.

or explicitly

$$\check{p}(r, \theta + \eta, z) = \check{p}(r, \theta, z),$$

and, denoting respectively by \check{u}_r , \check{u}_θ , \check{u}_z the radial, angular and axial components of a vector-field $\check{\mathbf{u}}$, and recalling that $(\check{u}_r, \check{u}_\theta)$ forms a local coordinate system on an axial section plane (i.e., a plane orthogonal to the axis, see figure 1.2)

$$\begin{aligned}\check{u}_r \circ \mathcal{R}_\eta &= \check{u}_r, \\ \check{u}_\theta \circ \mathcal{R}_\eta &= \check{u}_\theta, \\ \check{u}_z \circ \mathcal{R}_\eta &= \check{u}_z.\end{aligned}$$

In particular each cylindrical component of $\check{\mathbf{u}}$ is also axisymmetric.

An axisymmetric function \check{p} on $\check{\Omega}$ depends only on the radial and axial coordinates, therefore we can associate a function p on Ω such that $p(r, z) = \check{p}(r, 0, z)$. An axisymmetric vector-field $\check{\mathbf{u}}$ depends on (r, z) . If it has zero angular component ($\check{u}_\theta = 0$), we associate a vector-field $\mathbf{u} = (u_r, u_z)$ on Ω such that $u_r = \check{u}_r$ and $u_z = \check{u}_z$.

1.2 Axisymmetric formulation and analysis

In this section we introduce the model, the notation and we recall some results from [BDM99].

Suppose that the axisymmetric domain $\check{\Omega}$ is bounded, has a Lipschitz-continuous boundary, that Γ_0 is a finite union of segments of positive length and that the external forces are axisymmetric with zero angular component.

The stationary homogeneous three-dimensional Stokes problem reads

$$\begin{cases} -\nu \Delta \check{\mathbf{u}} + \nabla \check{p} = \check{\mathbf{f}} & \text{in } \check{\Omega}, \\ \operatorname{div} \check{\mathbf{u}} = 0 & \text{in } \check{\Omega}, \\ \check{\mathbf{u}} = \mathbf{0} & \text{on } \partial \check{\Omega}, \end{cases} \quad (1.1)$$

where $\check{\mathbf{f}}$ is in $H^{-1}(\check{\Omega})^3$. For simplicity we have chosen zero boundary data, however our analysis extends without difficulty to axisymmetric boundary data $\check{\mathbf{g}}$ with zero angular component and zero mean flux through $\partial \check{\Omega}$. The weak form of differential equation (1.1) writes:

P1.1 Find $(\check{\mathbf{u}}, \check{p})$ in $H_0^1(\check{\Omega})^3 \times L_0^2(\check{\Omega})$ such that for all $(\check{\mathbf{v}}, \check{q})$ in $H_0^1(\check{\Omega})^3 \times L_0^2(\check{\Omega})$

$$\begin{cases} \check{a}(\check{\mathbf{u}}, \check{\mathbf{v}}) + \check{b}(\check{\mathbf{v}}, \check{p}) = \int_{\check{\Omega}} \check{\mathbf{f}} \cdot \check{\mathbf{v}} d\check{\mathbf{x}}, \\ \check{b}(\check{\mathbf{u}}, \check{q}) = 0, \end{cases} \quad (1.2)$$

where the bilinear forms \check{a} and \check{b} are defined as

$$\begin{aligned} \check{a}(\check{\mathbf{u}}, \check{\mathbf{v}}) &= \nu \int_{\check{\Omega}} (\nabla \check{\mathbf{u}} : \nabla \check{\mathbf{v}}) d\check{\mathbf{x}}, \\ \check{b}(\check{\mathbf{u}}, \check{q}) &= - \int_{\check{\Omega}} \operatorname{div} \check{\mathbf{u}} \check{q} d\check{\mathbf{x}}, \end{aligned}$$

$H_0^1(\check{\Omega})$ stands for the space of functions in $H^1(\check{\Omega})$ with zero trace and $L_0^2(\check{\Omega})$ for the space of functions in $L^2(\check{\Omega})$ with integral equal to zero.

Bernardi, Dauge and Maday have shown in [BDM99, §IX.1] that, if $\check{\mathbf{f}}$ is axisymmetric, this problem has a unique axisymmetric solution and that it can be split in two separate problems on Ω , one for the angular component \check{u}_θ and the other for $(\check{u}_r, \check{u}_z, \check{p})$. If the data have no rotation as supposed, i.e., the angular component \check{f}_θ is equal to zero, then \check{u}_θ is also zero.

1.2.1 Weighted Sobolev spaces

In this section we introduce some weighted Sobolev spaces (see Kufner, [Kuf80] and [BDM99, §II.1]) that we use for the weak formulation of the axisymmetric problem.

For any real number α and $1 \leq p < \infty$, the space $L_\alpha^p(\Omega)$ is defined as the set of measurable functions w such that

$$\|w\|_{L_\alpha^p(\Omega)} = \left(\int_{\Omega} |w|^p r^\alpha d\mathbf{x} \right)^{\frac{1}{p}} < \infty,$$

where $r = r(\mathbf{x})$ is the radial coordinate of \mathbf{x} , i.e., the distance of a point \mathbf{x} in Ω from the symmetry axis and $d\mathbf{x} = r dr dz$. For $p = \infty$, $L_\alpha^\infty(\Omega)$ is simply equal to $L^\infty(\Omega)$. The subspace $L_{1,0}^2(\Omega)$ of $L_1^2(\Omega)$ denotes the functions q with weighted integral equal to zero:

$$\int_{\Omega} q r d\mathbf{x} = 0.$$

Let ℓ be a positive integer. We define the weighted Sobolev space $W_1^{\ell,p}(\Omega)$ as the space of functions in $L_1^p(\Omega)$ such that their partial derivatives of order less than or equal to ℓ belong to $L_1^p(\Omega)$. The space $W_1^{\ell,p}(\Omega)$ is a Hilbert space endowed with the following semi-norm and norm

$$\begin{aligned} |w|_{W_1^{\ell,p}(\Omega)} &= \left(\sum_{k=0}^{\ell} \|\partial_r^k \partial_z^{\ell-k} w\|_{L_1^p(\Omega)}^p \right)^{\frac{1}{p}}, \\ \|w\|_{W_1^{\ell,p}(\Omega)} &= \left(\sum_{k=0}^{\ell} |w|_{W_1^{k,p}(\Omega)}^p \right)^{\frac{1}{p}}. \end{aligned}$$

When $p = 2$, we note as in the standard case $W_1^{\ell,2}(\Omega)$ by $H_1^\ell(\Omega)$. We also need another weighted space $V_1^1(\Omega)$, defined as

$$V_1^1(\Omega) = H_1^1(\Omega) \cap L_{-1}^2(\Omega)$$

1.2. AXISYMMETRIC FORMULATION AND ANALYSIS

and endowed with the norm

$$\|w\|_{V_1^1(\Omega)} = \left(|w|_{H_1^1(\Omega)}^2 + \|w\|_{L_{-1}^2(\Omega)}^2 \right)^{\frac{1}{2}}.$$

It can be proved that all functions in $V_1^1(\Omega)$ have a null trace on Γ_0 (see Mercier and Raugel [MR82]). The traces on Γ are defined in a nearly standard way, see Bernardi, Dauge and Maday, [BDM, §I], Theorem a.5. Let $H_1^{\frac{1}{2}}(\Gamma)$ be the trace space of $H_1^1(\Omega)$ on Γ ,

$$H_1^{\frac{1}{2}}(\Gamma) = \{w|_{\Gamma}; w \in H_1^1(\Omega)\}.$$

1.2.2 Dimension reduction

In this section we state the correspondence of the standard three-dimensional and weighted two-dimensional Sobolev spaces. See [BDM99, §II.4] for the proofs of the following statements.

The subspace of axisymmetric functions in $H^1(\check{\Omega})$ is isomorphic to $H_1^1(\Omega)$. In the original three-dimensional problem, to take into account the boundary condition, the subspace $H_0^1(\check{\Omega})$ of zero trace functions is introduced. The counterpart for the axial component of the velocity is the weighted subspace

$$H_{1\Diamond}^1(\Omega) = \{w \in H_1^1(\Omega); w = 0 \text{ on } \Gamma\},$$

and the one for the radial component is

$$V_{1\Diamond}^1(\Omega) = \{w \in V_1^1(\Omega); w = 0 \text{ on } \Gamma\}.$$

We describe the axisymmetric domain $\check{\Omega}$ with cylindrical coordinates (r, θ, z) . It is possible to define two isomorphisms, which map axisymmetric functions and vector-fields on $\check{\Omega}$ to functions and vector-fields on Ω . These isomorphisms are called *reduction operators* and are defined in the scalar case as

$$\begin{aligned} \left\{ \check{w} \in H_0^1(\check{\Omega}) \text{ axisymmetric} \right\} &\longrightarrow H_{1\Diamond}^1(\Omega), \\ \check{w} &\longmapsto w : w(r, z) = \check{w}(r, \theta, z) \forall \theta \end{aligned}$$

and in the vector case as

$$\begin{aligned} \left\{ \check{\mathbf{w}} \in H_0^1(\check{\Omega})^3 \text{ axisymmetric and } \check{w}_\theta = 0 \right\} &\longrightarrow V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega), \\ \check{\mathbf{w}} &\longmapsto \mathbf{w} : w_r = \check{w}_r, w_z = \check{w}_z. \end{aligned}$$

Proposition 1.2.1 *The space of axisymmetric vector-fields in $H^1(\check{\Omega})^3$ with zero angular component is isomorphic to $V_1^1(\Omega) \times H_1^1(\Omega)$. The space of axisymmetric vector-fields in $H_0^1(\check{\Omega})^3$ with zero angular component is isomorphic to $V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)$.*

Proof To an axisymmetric vector-field $\check{\mathbf{v}}$ in $H^1(\check{\Omega})^3$ with zero angular component ($\check{v}_\theta = 0$) we associate a vector-field $\mathbf{v} = (v_r, v_z)$ on Ω , such that $v_r = \check{v}_r$, $v_z = \check{v}_z$ and vice-versa. Firstly recall that in cylindrical coordinates

$$\nabla \check{\mathbf{v}} = \begin{pmatrix} \partial_r v_r & 0 & \partial_r v_z \\ 0 & \frac{1}{r} v_r & 0 \\ \partial_z v_r & 0 & \partial_z v_z \end{pmatrix}.$$

Then

$$\begin{aligned}\|\check{\mathbf{v}}\|_{H^1(\check{\Omega})^3}^2 &= \int_{\check{\Omega}} (|\check{\mathbf{v}}|^2 + |\nabla \check{\mathbf{v}}|^2) dx dy dz = 2\pi \int_{\Omega} (|\check{\mathbf{v}}|^2 + |\nabla \check{\mathbf{v}}|^2) r dr dz \\ &= 2\pi \int_{\Omega} (|\mathbf{v}|^2 + |\nabla \mathbf{v}|^2) r d\mathbf{x} + 2\pi \int_{\Omega} v_r^2 \frac{1}{r} d\mathbf{x} = 2\pi \|\mathbf{v}\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2\end{aligned}$$

where $\nabla \mathbf{v}$ is equal to $\begin{pmatrix} \partial_r v_r & \partial_r v_z \\ \partial_z v_r & \partial_z v_z \end{pmatrix}$. Hence \mathbf{v} is in $V_1^1(\Omega) \times H_1^1(\Omega)$ and similarly for the inverse direction. ■

1.2.3 The weak axisymmetric form

The Stokes problem P1.1 on Ω for $(\check{u}_r, \check{u}_z, p)$ is equivalent to the following weak formulation of the Stokes axisymmetric problem.

P1.2 Find (\mathbf{u}, p) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$ such that, for all (\mathbf{v}, q) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$,

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} r d\mathbf{x}, \\ b(\mathbf{u}, q) = 0, \end{cases} \quad (1.3)$$

where the forms a and b are defined by

$$\begin{aligned}a(\mathbf{u}, \mathbf{v}) &= \frac{1}{2\pi} \check{a}(\check{\mathbf{u}}, \check{\mathbf{v}}) = \frac{1}{2\pi} \nu \int_{\check{\Omega}} (\nabla \check{\mathbf{u}} : \nabla \check{\mathbf{v}}) d\check{\mathbf{x}} \\ &= \nu \int_{\Omega} (\nabla \mathbf{u} : \nabla \mathbf{v}) r d\mathbf{x} + \nu \int_{\Omega} u_r v_r \frac{1}{r} d\mathbf{x}, \\ b(\mathbf{u}, q) &= \frac{1}{2\pi} \check{b}(\check{\mathbf{u}}, \check{q}) = -\frac{1}{2\pi} \int_{\check{\Omega}} \operatorname{div} \check{\mathbf{u}} \check{q} d\check{\mathbf{x}} \\ &= -\int_{\Omega} (\operatorname{div} \mathbf{u}) q r d\mathbf{x} - \int_{\Omega} u_r q d\mathbf{x},\end{aligned} \quad (1.4)$$

where $\operatorname{div} \mathbf{u} = \partial_r u_r + \partial_z u_z$ and $d\check{\mathbf{x}} = dx dy dz$. Indeed it can be checked that $a(\mathbf{u}, \mathbf{v}) = \frac{1}{2\pi} \check{a}(\check{\mathbf{u}}, \check{\mathbf{v}})$ and $b(\mathbf{u}, q) = \frac{1}{2\pi} \check{b}(\check{\mathbf{u}}, \check{q})$.

In [BDM99, §IX.1] it is proved that this problem has a unique solution. In particular it is easily derived from its analogue on $\check{\Omega}$ by using the reduction operator that the following inf-sup condition holds: There exists a positive constant β such that for all q in $L_{1,0}^2(\Omega)$,

$$\sup_{\mathbf{v} \in V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega)} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq \beta \|q\|_{L_{1,0}^2(\Omega)}. \quad (1.5)$$

1.3 Finite element approximation

In this section we introduce our finite element approximation to solve the Stokes problem P1.2 based on P1isoP2/P1 two dimensional elements on the half section Ω of the three-dimensional axisymmetric domain. We can interpret a mesh of Ω as a mesh of $\check{\Omega}$ made of toroidal elements

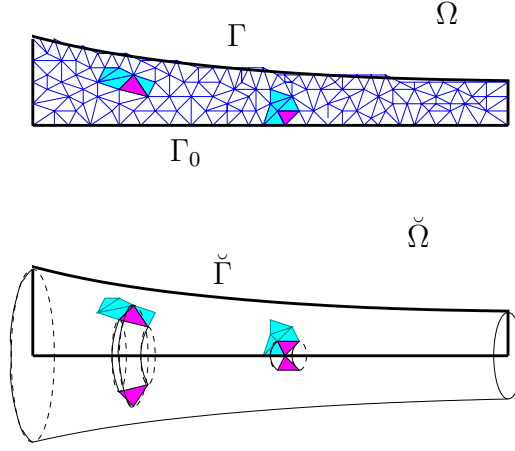


Figure 1.3: The mesh on the half section Ω and its axisymmetric representation in $\tilde{\Omega}$.

with triangular section (see figure 1.3). We then prove the inf-sup condition in a similar way as [BP79] and [Ver84].

The half section Ω represents our computational domain (see figure 1.3). From now on we suppose that Ω is polygonal and we introduce a *regular family of triangulations* $(\mathcal{T}_h)_h$ of Ω with the following properties:

- (i) The domain $\bar{\Omega}$ is the union of the elements of \mathcal{T}_h .
- (ii) If $T_k \neq T_j$ and their intersection is non empty, then $T_k \cap T_j$ is either a side or a node.
- (iii) There exists a constant σ independent of h , such that for all T in \mathcal{T}_h , its diameter h_T is smaller than h and T contains a circle of radius σh_T .

We also suppose that each triangle T in \mathcal{T}_h has at least one vertex inside Ω (not on $\Gamma \cup \Gamma_0$). In all that follows, c, c', \dots , denote generic constants that may depend on σ and vary from one line to the next one but are always independent of h .

Each triangulation \mathcal{T}_h is used for P1 elements for the pressure. Moreover $\mathcal{T}_{h/2}$ denotes the triangulation obtained from \mathcal{T}_h by dividing each triangle into four equal triangles by joining the midpoints of the edges. Indeed $\mathcal{T}_{h/2}$ is used for P1 elements for the velocity and generates the same degrees of freedom as P2 elements defined on \mathcal{T}_h .

Let $P_k(T)$ denote the set of restrictions to T of polynomials of degree less than or equal to k ; then the finite element spaces for the velocity and the pressure are

$$\begin{aligned} V_{h/2} &= \{ \mathbf{v}_h \in \mathcal{C}^0(\bar{\Omega})^2 : \mathbf{v}_h|_{\Gamma} = 0, v_{h,r}|_{\Gamma_0} = 0; \forall T \in \mathcal{T}_{h/2} \mathbf{v}_h|_T \in P_1(T)^2 \}, \\ Q_h &= \left\{ q_h \in \mathcal{C}^0(\bar{\Omega}) : \int_{\Omega} q_h r d\mathbf{x} = 0; \forall T \in \mathcal{T}_h q_h|_T \in P_1(T) \right\}. \end{aligned}$$

Lemma 1.3.1 *The following inclusions hold*

$$\begin{aligned} V_{h/2} &\subset V_{1\phi}^1(\Omega) \times H_{1\phi}^1(\Omega), \\ Q_h &\subset L_{1,0}^2(\Omega). \end{aligned}$$

Proof Q_h is included in $H^1(\Omega)$ and since Ω is bounded, $H^1(\Omega)$ is included in $H_1^1(\Omega)$. Then Q_h is a subset of $L_1^2(\Omega)$ and thanks to the boundary conditions imposed in $V_{h/2}$, this is included in $H_{1\Diamond}^1(\Omega)^2$.

Let \mathbf{v}_h be in $V_{h/2}$. Then $\|v_{h,r}\|_{H_1^1(\Omega)} < \infty$ and thanks to lemmas 1.4.1 and 1.4.3 in section 1.4.1, for any T in $\mathcal{T}_{h/2}$

$$\|\mathbf{v}_h\|_{L_{-1}^2(T)}^2 \leq C(2\sigma h_T)^{-1} \|\mathbf{v}_h\|_{L_1^2(T)}^2 < \infty.$$

Hence $\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2 < \infty$ and $V_{h/2} \subset V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)$. ■

The discrete formulation of Stokes problem P1.2 reads:

P1.3 Find (\mathbf{u}_h, p_h) in $V_{h/2} \times Q_h$ for all (\mathbf{v}_h, q_h) in $V_{h/2} \times Q_h$

$$\begin{cases} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h \, r \, d\mathbf{x}, \\ b(\mathbf{u}_h, q) = 0. \end{cases} \quad (1.6)$$

1.4 Finite element analysis

1.4.1 Weighted inverse inequalities

Preliminary results

In this section we will prove inverse inequalities for vector-fields in $V_{h/2}$. We need the following classification of the triangles. For any T in \mathcal{T}_h , let F_T denote an affine mapping from a reference triangle \hat{T} onto T . Then the vertices of \hat{T} , $\hat{\mathbf{a}}_i$, $i = 1, 2, 3$, are mapped by F_T onto the vertices of T , \mathbf{a}_i , $i = 1, 2, 3$. Let $\hat{\lambda}_i$ be the barycentric coordinate associated with $\hat{\mathbf{a}}_i$. We also define a scalar number r_T which is the minimum of the radial coordinate of the vertices of T not belonging to the axis.

Lemma 1.4.1 For any triangle T of \mathcal{T}_h , there exists a constant c , such that $ch_T \leq r_T$.

Proof Since the family of triangulations is regular, the distance of a vertex (r, z) to the opposite side of T is larger than or equal to $2\sigma h_T$. Since r_T is the distance of a vertex from the axis and no triangle crosses the axis, $r_T \geq 2\sigma h_T$. The only exception is when the vertex is on a triangle which crosses the axis and has a side on Γ . In this case $r_T \geq Ch_T$, where C depends on the angles between Γ and Γ_0 . Since there is a finite number of intersections between Γ and Γ_0 , the constant C is bounded from below by a positive constant. ■

The triangles T of $\mathcal{T}_{h/2}$ can be of three different types:

- Type 1. If $T \cap \Gamma_0$ is empty, the ratio $\frac{\max_{\mathbf{x} \in T} r(\mathbf{x})}{\min_{\mathbf{x} \in T} r(\mathbf{x})}$ is smaller than a constant only depending on the regularity parameter σ of the family of triangulations. Then there exist two positive constants c and c' depending only on σ such that

$$cr_T \leq r(\mathbf{x}) \leq c'r_T \quad \forall \mathbf{x} \in T.$$

We say that r is “equivalent to” r_T .

- Type 2. If $T \cap \Gamma_0$ is an edge with endpoints \mathbf{a}_2 and \mathbf{a}_3 , the function r is equal to $\alpha_1^T \lambda_1$, with the constant α_1^T equal to r_T , so that the ratio α_1^T/h_T is bounded from above and from below by positive constants only depending on σ .
- Type 3. If $T \cap \Gamma_0$ is a vertex, for instance \mathbf{a}_1 , the function r is equal to $\alpha_2^T \lambda_2 + \alpha_3^T \lambda_3$, with the constants α_i^T equal to $r(\mathbf{a}_i)$, so that the ratio α_i^T/h_T is bounded from above and from below by positive constants only depending on σ . So, the function r is “equivalent to” $h_T(\lambda_2 + \lambda_3)$, with equivalence constants only depending on σ .

Here we introduce some technical lemmas which will be used to establish the approximation properties of the Lagrange interpolation operator and the Clément operator.

We fix an integer $k \geq 1$ and, with each T in \mathcal{T}_h , we associate its lattice of order k , i.e., the set of degrees of freedom for polynomial of degree k . Let $\Sigma_h = \{\mathbf{a}_i, 1 \leq i \leq N_h\}$ be the union of these lattices on all T in \mathcal{T}_h .

Lemma 1.4.2 *Let φ_i denote the Lagrange function in $P_k(T)$ associated to the node $\mathbf{a}_i = (r_i, z_i)$ of Σ_h . Then there exists a constant c independent of h_T , such that for all T in \mathcal{T}_h containing \mathbf{a}_i the following inequalities hold*

$$\|\varphi_i\|_{L_1^p(T)} \leq c \left(\max_T r^{\frac{1}{p}} \right) h_T^{\frac{2}{p}}, \quad \|\varphi_i\|_{W_1^{1,p}(T)} \leq c \left(\max_T r^{\frac{1}{p}} \right) h_T^{\frac{2}{p}-1}. \quad (1.7)$$

Proof Since the proof is similar for both inequalities we only give it for the first one. Indeed, it is readily checked by going to the reference element that

$$\|\varphi_i\|_{L_1^p(T)} \leq \left(\max_T r^{\frac{1}{p}} \right) h_T^{\frac{2}{p}} \|\hat{\varphi}_i\|_{L^p(\hat{T})}.$$

■

Note that if T intersect Γ_0 , then r_T and $\max_T r$ are of the same order as h_T .

Inverse Inequalities

Firstly, inequalities are proved for the norm of $L_{-1}^p(T)$, then for the semi-norm of $W_1^{\ell,p}(T)$ and finally the proof is carried out in the norms of $V_1^1(\Omega) \times H_1^1(\Omega)$.

For a triangle T in \mathcal{T}_h , note its area by $|T|$. Let f be a polynomial defined in T , then \hat{f} stands for $f \circ F_T$. In particular $\rho_T = r \circ F_T$ is the affine function representing the radial coordinate.

Lemma 1.4.3 *Let $1 \leq p < \infty$ and k be an integer. There exists a constant c , such that for every triangle T in \mathcal{T}_h and any polynomial f in $P_k(T)$, vanishing on the axis if T is of type 2 or 3,*

$$\|f\|_{L_{-1}^p(T)} \leq c r_T^{-2/p} \|f\|_{L_1^p(T)}. \quad (1.8)$$

Proof If T is of type 1, then for any point in T , $r_T \leq r$ and

$$\int_T |f|^p \frac{1}{r} d\mathbf{x} \leq \frac{1}{r_T^2} \int_T |f|^p r d\mathbf{x}.$$

Let T be of type 2. On the reference triangle, the weighted norms

$$\|\hat{f} \hat{\lambda}_1^{-1/p}\|_{L^p(\hat{T})} \text{ and } \|\hat{f} \hat{\lambda}_1^{1/p}\|_{L^p(\hat{T})}. \quad (1.9)$$

on the polynomials \hat{f} of degree k . Then

$$\begin{aligned} \|f\|_{L_{-1}^p(T)}^p &= \frac{|T|}{2} \|\hat{f} (\hat{\lambda}_1 r_T)^{-1/p}\|_{L^p(\hat{T})}^p \quad (\text{owing to the equivalence}) \\ &\leq c \frac{|T|}{2} \frac{1}{r_T^2} \|\hat{f} (\hat{\lambda}_1 r_T)^{1/p}\|_{L^p(\hat{T})}^p = c \frac{1}{r_T^2} \|f\|_{L_1^p(T)}^p. \end{aligned}$$

This proves (1.8).

If T is of type 3, from the equivalence of the norms in (1.9) with $\hat{\lambda}_1$ replaced by $\hat{\lambda}_2 + \hat{\lambda}_3$,

$$\begin{aligned} \|f\|_{L_{-1}^p(T)}^p &\leq c \frac{|T|}{2} \|\hat{f} ((\hat{\lambda}_2 + \hat{\lambda}_3) r_T)^{-1/p}\|_{L^p(\hat{T})}^p \\ &\leq c' \frac{|T|}{2} \frac{1}{r_T^2} \|\hat{f} ((\hat{\lambda}_2 + \hat{\lambda}_3) r_T)^{1/p}\|_{L^p(\hat{T})}^p \leq c' \frac{1}{r_T^2} \|f\|_{L_1^p(T)}^p. \end{aligned}$$

■

Lemma 1.4.4 (Inverse inequality on weighted L^p -spaces) *There exists a constant c , such that for every triangle T in \mathcal{T}_h and any polynomial f in $P_k(T)$,*

$$\|\nabla f\|_{L_1^p(T)} \leq c h_T^{-1} \|f\|_{L_1^p(T)}. \quad (1.10)$$

Proof If T is of type 1, the standard inverse inequality gives

$$\begin{aligned} \|\nabla f\|_{L_1^p(T)} &\leq \max_{\mathbf{x} \in T} r(\mathbf{x})^{\frac{1}{p}} \|\nabla f\|_{L^p(T)} \leq c \left(\max_{\mathbf{x} \in T} r(\mathbf{x}) \right)^{\frac{1}{p}} h_T^{-1} \|f\|_{L^p(T)} \\ &\leq c \left(\frac{\max_{\mathbf{x} \in T} r(\mathbf{x})}{\min_{\mathbf{x} \in T} r(\mathbf{x})} \right)^{\frac{1}{p}} h_T^{-1} \|f\|_{L_1^p(T)}, \end{aligned}$$

and the boundedness of the quantity $\frac{\max_{\mathbf{x} \in T} r(\mathbf{x})}{\min_{\mathbf{x} \in T} r(\mathbf{x})}$ leads to the desired inequality. If T is of type 2, we have by using the transformation F_T

$$\|\nabla f\|_{L_1^p(T)} \leq c h_T^{\frac{2}{p}-1} r_T^{\frac{1}{p}} \|(\nabla \hat{f}) \hat{\lambda}_1^{\frac{1}{p}}\|_{L^p(\hat{T})},$$

and, by using the equivalence of norms on a finite dimensional space, we obtain

$$\|\nabla f\|_{L_1^p(T)} \leq c h_T^{\frac{2}{p}-1} r_T^{\frac{1}{p}} \|\hat{f} \hat{\lambda}_1^{\frac{1}{p}}\|_{L^p(\hat{T})}$$

Thus going back to T yields

$$\|\nabla f\|_{L_1^p(T)} \leq c h_T^{-1} \|f\|_{L_1^p(T)},$$

which is the desired result. When T is of type 3, the inequality follows from the same argument as previously, with $\hat{\lambda}_1$ replaced by $\hat{\lambda}_2 + \hat{\lambda}_3$.

■

Now we are ready to prove a weighted inverse inequality:

Proposition 1.4.1 (Inverse inequality on weighted Sobolev spaces) *There exists a constant c such that for all \mathbf{v}_h in $V_{h/2}$,*

$$\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2 \leq c \sum_{T \in \mathcal{T}_{h/2}} h_T^{-2} \|\mathbf{v}_h\|_{L_1^2(T)^2}^2. \quad (1.11)$$

Proof Let $\mathbf{v}_h = (v_r, v_z)$ be in $V_{h/2}$. From lemmas 1.4.1 and 1.4.3, for any T in $\mathcal{T}_{h/2}$,

$$\|v_r\|_{L_{-1}^2(T)} \leq c h_T^{-1} \|v_r\|_{L_1^2(T)}$$

and from lemma 1.4.4

$$\|\nabla v_r\|_{L_1^2(T)} \leq c h_T^{-1} \|v_r\|_{L_1^2(T)},$$

and this last inequality also holds for v_z . Hence

$$\begin{aligned} \|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2 &= c \sum_{T \in \mathcal{T}_{h/2}} \left(\|\nabla v_r\|_{L_1^2(T)}^2 + \|\nabla v_z\|_{L_1^2(T)}^2 + \|v_r\|_{L_{-1}^2(T)}^2 \right) \\ &\leq c \sum_{T \in \mathcal{T}_{h/2}} h_T^{-2} \left(\|v_z\|_{L_1^2(T)}^2 + 2\|v_r\|_{L_1^2(T)}^2 \right) \\ &\leq c \sum_{T \in \mathcal{T}_{h/2}} h_T^{-2} \|\mathbf{v}\|_{L_1^2(T)^2}^2 \leq \frac{c}{4} \sum_{T' \in \mathcal{T}_h} h_{T'}^{-2} \|\mathbf{v}\|_{L_1^2(T')^2}^2. \end{aligned}$$

■

1.4.2 Inf-Sup condition

Verfürth in [Ver84] has proved the inf-sup condition for the Stokes problem with P1isoP2/P1 elements in the Cartesian case. The proof of Verfürth is based on the following result (due to [BP79]), which we write adapted to the axisymmetric problem and to (not necessarily uniformly) regular families of triangulations.

There exists a positive constant c independent of h such that

$$\forall q_h \in Q_h, \sup_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq c \left(\sum_{T \in \mathcal{T}_h} h_T^2 |q_h|_{H_1^1(T)}^2 \right)^{1/2}. \quad (1.12)$$

We prove this statement at the end of the chapter in section 1.5.2.

The following result is an extension of the Clément operator introduced in [Clé75], see also [BG98] and [BMR04, §IX.3].

There exist positive constants c and c' and an operator $R_h : V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega) \rightarrow V_h$ such that for all functions \mathbf{v} in $V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)$ and for all triangles T in \mathcal{T}_h

$$\|\mathbf{v} - R_h \mathbf{v}\|_{L_1^2(T)} \leq c h_T \|\mathbf{v}\|_{V_1^1(T) \times H_1^1(T)}, \quad (1.13)$$

$$\|\mathbf{v} - R_h \mathbf{v}\|_{V_1^1(T) \times H_1^1(T)} \leq c' \|\mathbf{v}\|_{V_1^1(T) \times H_1^1(T)}, \quad (1.14)$$

where Δ_T is the union of the triangles sharing at least one common vertex with T . Its proof will be given in section 1.5.1. Now we are going to state and prove the inf-sup theorem for axisymmetric P1isoP2/P1 finite elements.

Theorem 1.4.1 *There exists a positive constant c independent of h such that*

$$\forall q_h \in Q_h, \quad \sup_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq c \|q_h\|_{L_1^2(\Omega)}. \quad (1.15)$$

Proof Let q_h be in Q_h with $\|q_h\|_{L_1^2(\Omega)} = 1$, and note

$$\eta = \left(\sum_{T \in \mathcal{T}_h} h_T^2 |q_h|_{H_1^1(T)}^2 \right)^{1/2}.$$

Inequality (1.12) implies

$$\sup_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq c_1 \eta. \quad (1.16)$$

The proof of the inf-sup condition (1.5) at the continuous level is based on the existence of a \mathbf{u} satisfying

$$\operatorname{div} \mathbf{u} + \frac{1}{r} u_r = -q_h \text{ and } \|\mathbf{u}\|_{V_1^1(\Omega) \times H_1^1(\Omega)} \leq 2\beta^{-1} \|q_h\|_{L_1^2(\Omega)} = 2\beta^{-1} \quad (1.17)$$

(see [BDM99, §IX.1] or appendix A.2). This implies $|b(\mathbf{u}, q_h)| = \|q_h\|_{L_1^2(\Omega)}^2 = 1$. On the other hand, taking $\mathbf{u}_h = R_h \mathbf{u}$ and using (1.13) and (1.14) yields

$$\sum_{T \in \mathcal{T}_h} h_T^{-2} \|\mathbf{u}_h - \mathbf{u}\|_{L_1^2(T)}^2 \stackrel{(1.13)}{\leq} c^2 \sum_{T \in \mathcal{T}_h} \|\mathbf{u}\|_{V_1^1(\Delta_T) \times H_1^1(\Delta_T)}^2 \leq c'^2 \|\mathbf{u}\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2 \stackrel{(1.17)}{\leq} 4\beta^{-2} c'^2. \quad (1.18)$$

By integration by parts together with the Schwarz inequality, we derive that

$$\begin{aligned} |b(\mathbf{u}_h - \mathbf{u}, q_h)| &= \left| \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{u}_h - \mathbf{u}) \nabla q_h r dx \right| \leq \sum_{T \in \mathcal{T}_h} \|\mathbf{u} - \mathbf{u}_h\|_{L_1^2(T)} h_T^{-1} h_T \|\nabla q_h\|_{L_1^2(T)} \\ &\leq \left(\sum_{T \in \mathcal{T}_h} h_T^{-2} \|\mathbf{u} - \mathbf{u}_h\|_{L_1^2(T)}^2 \right)^{1/2} \left(\sum_{T \in \mathcal{T}_h} h_T^2 \|\nabla q_h\|_{L_1^2(T)}^2 \right)^{1/2} \stackrel{(1.18)}{\leq} c \eta. \end{aligned} \quad (1.19)$$

Using (1.17), (1.14) and (1.19) yields

$$\begin{aligned} \sup_{\mathbf{v} \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} &\geq \frac{b(\mathbf{u}_h, q_h)}{\|\mathbf{u}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \\ &\geq \frac{b(\mathbf{u}, q_h) - |b(\mathbf{u}_h - \mathbf{u}, q_h)|}{\|\mathbf{u}\|_{V_1^1(\Omega) \times H_1^1(\Omega)} + \|\mathbf{u}_h - \mathbf{u}\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \stackrel{(1.17, 1.14)}{\geq} c(b(\mathbf{u}, q_h) - |b(\mathbf{u}_h - \mathbf{u}, q_h)|) \\ &\stackrel{(1.19)}{\geq} c(1 - c' \eta) = c_2 - c_3 \eta. \end{aligned} \quad (1.20)$$

Inequalities (1.16) and (1.20) imply

$$\sup_{\mathbf{v} \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq \max \{c_1 \eta, c_2 - c_3 \eta\} \geq \min_{t \geq 0} \max \{c_1 t, c_2 - c_3 t\} = \frac{c_1 c_2}{c_1 + c_3}.$$

1.4. FINITE ELEMENT ANALYSIS

This ends the proof in the case $\|q_h\|_{L_1^2(\Omega)} = 1$. Otherwise, if q_h is different from zero, take $\tilde{q}_h = q_h/\|q_h\|_{L_1^2(\Omega)}$, which concludes the proof. ■

It is also possible to replace $V_{h/2}$ by piecewise quadratic functions on \mathcal{T}_h , i.e.,

$$\tilde{V}_h = \{\mathbf{v}_h \in \mathcal{C}^0(\Omega)^2 : \mathbf{v}_h|_\Gamma = 0, v_{h,r}|_{\Gamma_0} = 0; \forall T \in \mathcal{T}_h \mathbf{v}_h|_T \in P_2(T)^2\}.$$

The degrees of freedom of this space are exactly the same as for $V_{h/2}$, so by the same arguments as previously, the inf-sup condition (1.15) still holds with $V_{h/2}$ replaced by \tilde{V}_h . The proof of proposition 1.5.2 on page 31 can also be adapted to \tilde{V}_h . We recall that the couple of finite elements spaces (\tilde{V}_h, Q_h) are called Taylor–Hood elements.

1.4.3 Existence, uniqueness and a priori error estimates

The spaces $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega)$ equipped with $\|\cdot\|_{V_1^1(\Omega) \times H_1^1(\Omega)}$ and $L_{1,0}^2(\Omega)$ with $\|\cdot\|_{L_1^2(\Omega)}$ are Hilbert spaces. In fact (see proposition 1.2.1 and [BDM99, §II.2]) they are isomorphic to subspaces of $H^1(\check{\Omega})^3$ and $L^2(\check{\Omega})$ respectively. The bilinear form $a(\cdot, \cdot)$ is elliptic (property derived from $\check{a}(\cdot, \cdot)$), and the bilinear form $b(\cdot, \cdot)$ satisfies by theorem 1.4.1 the inf-sup condition. Hence the abstract results of Babuška [Bab73], Brezzi [Bre74] (see also Brezzi–Fortin [BF91, §II.2.2] and Girault–Raviart [GR86, §II.1]) yield the well-posedness of the discrete Stokes problem (1.6).

Theorem 1.4.2 *Problem (1.6) has a unique solution (\mathbf{u}_h, p_h) in $V_{h/2} \times Q_h$. Furthermore, if \mathbf{u} is in $H_1^2(\Omega)^2$ and p in $H_1^1(\Omega)$, then there exists a constant C such that*

$$\|\mathbf{u} - \mathbf{u}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)} + \|p - p_h\|_{L_1^2(\Omega)} \leq Ch \left(\|\mathbf{u}\|_{H_1^2(\Omega)^2} + \|p\|_{H_1^1(\Omega)} \right). \quad (1.21)$$

Proof As pointed out in definition (1.4), the bilinear form $a(\cdot, \cdot)$ can be expressed by the three-dimensional bilinear form $\check{a}(\cdot, \cdot)$, which is coercive (see for example [BDM99, §II.2]). In section 1.4.2 the inf-sup condition 1.4.1 is proved, hence from theorem 1.1 in [GR86, §II.1] it follows that

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)} + \|p - p_h\|_{L_1^2(\Omega)} \\ \leq C \left(\inf_{\mathbf{v}_h \in V_{h/2}} \|\mathbf{u} - \mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)} + \inf_{q_h \in Q_h} \|p - q_h\|_{L_1^2(\Omega)} \right). \end{aligned}$$

Mercier and Raugel [MR82] in theorem 4.4 show that the space of functions in $H_1^2(\Omega)$ vanishing on Γ_0 is included in $V_1^1(\Omega)$. Theorems 1.5.1 and 1.5.2 and corollary 1.5.3 in section 1.5.1 lead to

$$\inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)} \leq Ch \|\mathbf{u}\|_{H_1^2(\Omega)^2}$$

and

$$\inf_{q_h \in Q_h} \|p - q_h\|_{L_1^2(\Omega)} \leq Ch \|p\|_{H_1^1(\Omega)},$$

which proves (1.21). ■

In particular, Bernardi, Dauge and Maday in [BDM99, §IX.1] show that if Ω is convex and the angles between Γ and Γ_0 , are not too large (for example less than $\frac{3}{4}\pi$ suffices), and

if \mathbf{f} is in $L_1^2(\Omega)^2$, then \mathbf{u} is in $H_1^2(\Omega)^2$ and p is in $H_1^1(\Omega)$ (in fact they show that \mathbf{u} is even more regular than that). So the error behaves like ch at least when these conditions on the geometry of Ω are satisfied.

1.5 Technical results

1.5.1 Weighted approximation properties

We now prove some properties of the Lagrange interpolation operator and also of some Clément type operators. There are several possible constructions of the Clément operator, we follow here the approach presented in [BMR04, §IX.3] in the Cartesian case. We begin by some technical lemmas which are useful in what follows.

Preliminary results

The next lemma states a polynomial approximation property which is a weighted extension of a more general result due to [DS80] which was however stated for the unweighted case.

For \mathbf{a}_i in Σ_h , $\tilde{\Delta}_i$ denotes the union of two triangles containing \mathbf{a}_i and sharing a common edge, and h_i stands for the diameter of $\tilde{\Delta}_i$.

Lemma 1.5.1 *For all p , $1 \leq p \leq +\infty$, there exists a constant c , independent of h_i , such that for all functions v in $W_1^{1,p}(\tilde{\Delta}_i)$,*

$$\inf_{q \in P_0(\tilde{\Delta}_i)} \left(\|v - q\|_{L_1^p(\tilde{\Delta}_i)} + h_i |v - q|_{W_1^{1,p}(\tilde{\Delta}_i)} \right) \leq c h_i |v|_{W_1^{1,p}(\tilde{\Delta}_i)}. \quad (1.22)$$

Proof Let T and T' denote the two triangles which define $\tilde{\Delta}_i$, and e their common edge. Let h_e denote the diameter of e and \mathbf{m}_e its midpoint. There exists a constant λ depending only on the regularity parameter σ (defined in section 1.3), such that $\tilde{\Delta}_i$ is star shaped with respect to the ball B centered on \mathbf{m}_e and with radius $\frac{\lambda h_e}{2}$. The function : $\mathbf{x} \mapsto \hat{\mathbf{x}} = 2 \frac{\mathbf{x} - \mathbf{m}_e}{\lambda h_e}$, from $\tilde{\Delta}_i$ into a region $\hat{\Delta}$ maps the ball B into the unit ball \hat{B} . Let $\hat{\varphi}$ be in $\mathcal{D}(\hat{B})$, with $\int_{\hat{B}} \hat{\varphi} d\hat{\mathbf{x}} = 1$, then the function φ defined by

$$\varphi(\mathbf{x}) = \left(\frac{\lambda h_e}{2} \right)^{-2} \hat{\varphi} \left(2 \frac{\mathbf{x} - \mathbf{m}_e}{\lambda h_e} \right),$$

belongs to $\mathcal{D}(B)$ and

$$\int_B \varphi d\mathbf{x} = 1.$$

Define q as

$$q = \int_B \varphi(\mathbf{y}) v(\mathbf{y}) d\mathbf{y}.$$

To evaluate the norm of $v - q$ in $L_1^p(\tilde{\Delta}_i)$, we start with the following Taylor formula: For each $\mathbf{x} \in \tilde{\Delta}_i$, and $\mathbf{y} \in B$,

$$v(\mathbf{x}) = v(\mathbf{y}) + \int_0^1 (\mathbf{x} - \mathbf{y}) \cdot \nabla v(\mathbf{x} + s(\mathbf{y} - \mathbf{x})) ds.$$

1.5. TECHNICAL RESULTS

Multiplying by $\varphi(\mathbf{y})$ and integrating over B , we obtain

$$v(\mathbf{x}) - q = \int_B \int_0^1 \varphi(\mathbf{y})(\mathbf{x} - \mathbf{y}) \cdot \nabla v(\mathbf{x} + s(\mathbf{y} - \mathbf{x})) ds d\mathbf{y}.$$

Setting $\mathbf{z} = \mathbf{x} + s(\mathbf{y} - \mathbf{x})$, yields

$$|v(\mathbf{x}) - q| \leq C \int_{\tilde{\Delta}_i} \left| \int_0^1 \varphi(\mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x})) s^{-1}(\mathbf{x} - \mathbf{z}) \cdot \nabla v(\mathbf{z}) s^{-2} ds \right| d\mathbf{z},$$

whence, for any \mathbf{x} in $\tilde{\Delta}_i$,

$$|v(\mathbf{x}) - q| \leq \int_{\tilde{\Delta}_i} |k(\mathbf{x}, \mathbf{z})(\mathbf{x} - \mathbf{z}) \cdot (\nabla v)(\mathbf{z})| d\mathbf{z}, \quad (1.23)$$

where

$$k(\mathbf{x}, \mathbf{z}) = \int_0^1 \varphi(\mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x})) s^{-3} ds.$$

Since $\varphi(\mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x}))$ vanishes when $|\mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x}) - \mathbf{m}_e| \geq \frac{\lambda h_e}{2}$, and particularly for $s \leq (\mu h_e)^{-1}|\mathbf{z} - \mathbf{x}|$, for a constant μ depending only on σ ,

$$|k(\mathbf{x}, \mathbf{z})| \leq \|\varphi\|_{L^\infty(B)} \int_{(\mu h_e)^{-1}|\mathbf{z} - \mathbf{x}|}^1 s^{-3} ds \leq c \|\varphi\|_{L^\infty(B)} (\mu h_e)^2 \left| |\mathbf{x} - \mathbf{z}|^{-2} - (\mu h_e)^{-2} \right|.$$

Using $\|\varphi\|_{L^\infty(B)} = (\frac{\lambda h_e}{2})^{-2} \|\hat{\varphi}\|_{L^\infty(\hat{B})}$, we deduce

$$|k(\mathbf{x}, \mathbf{z})| \leq c \left(|\mathbf{x} - \mathbf{z}|^{-2} + (\mu h_e)^{-2} \right). \quad (1.24)$$

Let \tilde{k} be the function

$$\tilde{k}(\mathbf{z}) = (|\mathbf{z} - \mathbf{m}_e|^{-2} + (\mu h_e)^{-2}) |\mathbf{z} - \mathbf{m}_e|$$

and deduce from (1.23) and (1.24)

$$|(v - q)(\mathbf{x})| \leq c \int_{\tilde{\Delta}_i} \tilde{k}(\mathbf{x} - \mathbf{z}) |\nabla v(\mathbf{z})| d\mathbf{z}.$$

We now check that, for a constant c only depending on the regularity parameter σ ,

$$r(\mathbf{z}) \geq c r(\mathbf{x}). \quad (1.25)$$

Indeed,

- either the intersection of $\tilde{\Delta}_i$ with Γ_0 is empty. Then, we have

$$r(\mathbf{z}) \geq \frac{\min_{\mathbf{t} \in \tilde{\Delta}_i} r(\mathbf{t})}{\max_{\mathbf{t} \in \tilde{\Delta}_i} r(\mathbf{t})} r(\mathbf{x}),$$

and the ratio $\min_{\mathbf{t} \in \tilde{\Delta}_i} r(\mathbf{t}) / \max_{\mathbf{t} \in \tilde{\Delta}_i} r(\mathbf{t})$ is bounded from below by a constant only depending on σ ;

- or it is not empty. We note that

$$r(\mathbf{z}) = (1-s)r(\mathbf{x}) + sr(\mathbf{y}) \geq \min\{r(\mathbf{x}), r(\mathbf{y})\}.$$

Since $r(\mathbf{y}) \geq \mu h_e$, either $r(\mathbf{x}) \leq \mu h_e$, so that $r(\mathbf{z}) \geq r(\mathbf{x})$, or $r(\mathbf{x}) > \mu h_e$, so that $r(\mathbf{z}) \geq \mu h_e$ and, since $r(\mathbf{x}) \leq ch_e$ for a constant c only depending on σ , $r(\mathbf{z}) \geq \frac{\mu}{c} h_e$.

Thus

$$|(v-q)(\mathbf{x})| r(\mathbf{x})^{\frac{1}{p}} \leq c \int_{\tilde{\Delta}_i} \tilde{k}(\mathbf{x}-\mathbf{z}) |\nabla v(\mathbf{z})| r(\mathbf{z})^{\frac{1}{p}} d\mathbf{z}.$$

Applying Young's inequality yields

$$\|v-q\|_{L_1^p(\tilde{\Delta}_i)} \leq \|\tilde{k}\|_{L^1(\tilde{\Delta}_i)} \|\nabla v\|_{L_1^p(\tilde{\Delta}_i)}.$$

Noting that

$$\begin{aligned} \|\tilde{k}\|_{L^1(\tilde{\Delta}_i)} &= \int_{\tilde{\Delta}_i} (|\mathbf{z} - \mathbf{m}_e|^{-1} + |\mathbf{z} - \mathbf{m}_e|(\mu h_e)^{-2}) d\mathbf{z} \\ &\leq c \int_0^{c'h_e} (\varrho^{-1} + (\mu h_e)^{-2} \varrho) \varrho d\varrho = c'' h_e, \end{aligned}$$

we derive the first part of the inequality (1.22).

The second part of (1.22), i.e., the inequality with the second term on the left-hand side, is obvious. ■

The next lemma is an extension of the previous one and its proof is identical to that for the unweighted case, see [BMR04, §IX], lemma 3.4.

Let Δ_i denotes the union of all elements T in \mathcal{T}_h containing \mathbf{a}_i .

Lemma 1.5.2 *For all p , $1 \leq p \leq +\infty$, there exists a constant c , independent of h_i , such that, for all functions v in $W_1^{1,p}(\Delta_i)$,*

$$\inf_{q \in P_0(\Delta_i)} \left(\|v-q\|_{L_1^p(\Delta_i)} + h_i |v-q|_{W_1^{1,p}(\Delta_i)} \right) \leq c h_i \|v\|_{W_1^{1,p}(\Delta_i)}. \quad (1.26)$$

The following lemma is obtained by the same construction as lemma 1.5.1 and lemma 1.5.2. Since the proof is rather long and technical, we only state the result. We refer to Dupont-Scott [DS80] for the analogue in the unweighted case

Lemma 1.5.3 *For all p , $1 \leq p \leq +\infty$, there exists a constant c , independent of h_i , such that, for all functions $v \in W_1^{\ell+1,p}(\Delta_i)$, the following inequality holds*

$$\inf_{q \in P_\ell(\Delta_i)} \left(\|v-q\|_{L_1^p(\Delta_i)} + h_i |v-q|_{W_1^{\ell+1,p}(\Delta_i)} \right) \leq c h_i^{\ell+1} \|v\|_{W_1^{\ell+1,p}(\Delta_i)}. \quad (1.27)$$

Obviously the results of lemma 1.5.1 to lemma 1.5.3 still hold when replacing Δ_i by an element T of \mathcal{T}_h . If we denote by Δ_T the union of all elements of \mathcal{T}_h sharing at least a common vertex with T , then these results still hold also.

Lagrange interpolation operator

We define the Lagrange interpolation operator $\mathcal{I}_h : \mathcal{C}^0(\bar{\Omega}) \rightarrow X_h$, where X_h denotes the space of Lagrange finite elements of order k : $\mathcal{I}_h \varphi$ coincides with φ on all nodes of Σ_h . For any T in \mathcal{T}_h , we introduce a local interpolation operator $i_T : \mathcal{C}^0(T) \rightarrow P_k(T)$, such that for all \mathbf{a}_j in $\Sigma_h \cap T$,

$$(i_T \varphi)(\mathbf{a}_j) = \varphi(\mathbf{a}_j).$$

So, it holds

$$\mathcal{I}_h \varphi|_T = i_T \varphi|_T.$$

Moreover this operator maps the functions that vanish on Γ onto $X_h \cap H_{1\circ}^1(\Omega)$.

The approximation properties of the Lagrange interpolation operator in the framework of weighted Sobolev spaces are proved in [MR82] (lemmas 6.1 and 6.2) in the case $p = 2$ (with some restrictions). However, this is not sufficient for our purpose, and we need the more general results stated in the following proposition.

Proposition 1.5.1 *For all ℓ , $1 \leq \ell \leq k + 1$, and for all p , $1 \leq p \leq +\infty$, such that*

$$\ell > \frac{3}{p} \quad \text{or} \quad p = 1, \ell = 3, \quad (1.28)$$

there exists a constant C , independent of h , such that, for all element T in \mathcal{T}_h , the following inequalities hold for all functions $v \in W_1^{\ell,p}(\Omega)$

$$\|v - \mathcal{I}_h v\|_{L_1^p(T)} \leq C h_T^\ell |v|_{W_1^{\ell,p}(T)}, \quad (1.29)$$

$$|v - \mathcal{I}_h v|_{W_1^{1,p}(T)} \leq C h_T^{\ell-1} |v|_{W_1^{\ell,p}(T)}. \quad (1.30)$$

Proof For any $T \in \mathcal{T}_h$ and for any polynomial p of degree $\ell - 1$,

$$\|u - \mathcal{I}_h u\|_{L_1^p(T)} \leq \|u - p\|_{L_1^p(T)} + \|\mathcal{I}_h(u - p)\|_{L_1^p(T)}.$$

We consider the second term of this inequality. By going to the reference element, we have

$$\|\mathcal{I}_h(u - p)\|_{L_1^p(T)} \leq c \delta h_T^{2/p} \|\hat{\mathcal{I}}(\hat{u} - \hat{p}) \chi^{\frac{1}{p}}\|_{L^p(\hat{T})},$$

where δ and χ depend on the triangle T as follows

$$\begin{cases} \delta = (\max_T r)^{\frac{1}{p}}, \text{ and } \chi = 1 & \text{if } T \text{ is of type 1,} \\ \delta = h_T^{\frac{1}{p}}, \text{ and } \chi = \hat{\lambda}_1 & \text{if } T \text{ is of type 2,} \\ \delta = h_T^{\frac{1}{p}}, \text{ and } \chi = \hat{\lambda}_2 + \hat{\lambda}_3 & \text{if } T \text{ is of type 3.} \end{cases}$$

Let $W_\chi^{\ell,p}(\hat{T})$ be the weighted Sobolev space with weight χ (similarly as for $W_1^{\ell,p}(T)$). The continuous embedding of $W_\chi^{\ell,p}(\hat{T})$ into $\mathcal{C}^0(\bar{\hat{T}})$, which in the first case, derives from the standard Sobolev embedding and in the other two cases from the three-dimensional one, yields

$$\delta \|\hat{\mathcal{I}}(\hat{u} - \hat{p})\|_{L_\chi^p(\hat{T})} \leq \delta \|\hat{u} - \hat{p}\|_{W_\chi^{\ell,p}(\hat{T})},$$

whence

$$\|u - \mathcal{I}_h u\|_{L_1^p(T)} \leq c \delta h_T^{2/p} \|\hat{u} - \hat{p}\|_{W_\chi^{\ell,p}(\hat{T})}.$$

Using the weighted Bramble-Hilbert lemma which follows from the compactness of the embedding $W_\chi^{\ell,p}(\hat{T}) \subset L_\chi^p(\hat{T})$, we obtain (see [GR86, §I], lemma 2.1)

$$\|u - \mathcal{I}_h u\|_{L_1^p(T)} \leq c \delta h_T^{2/p} |\hat{u} - \hat{p}|_{W_\chi^{\ell,p}(\hat{T})}.$$

Returning back to T leads to (1.29). Inequality (1.30) is obtained similarly. ■

Weighted Clément operator

In this section we define a regularization operator Π_h which maps $L_1^2(\Omega)$ into X_h (the space of Lagrange finite elements of order k), and we establish its approximation properties. With each \mathbf{a}_i in Σ_h , we associate an arbitrary triangle T_i of \mathcal{T}_h which contains \mathbf{a}_i . Note that T_i is to be chosen among a finite number of elements (bounded independently of the discretization parameter). Define π_i as the $L_1^2(T_i)$ orthogonal projection operator onto $P_k(T_i)$: For all v in $L_1^1(T_i)$, $\pi_i v$ is in $P_k(T_i)$ and satisfies

$$\forall q \in P_k(T_i), \quad \int_{T_i} (v - \pi_i v)(\mathbf{x}) q(\mathbf{x}) r d\mathbf{x} = 0. \quad (1.31)$$

We define Π_h as

$$\Pi_h v = \sum_{i=1}^{N_h} (\pi_i v)(\mathbf{a}_i) \varphi_i(\mathbf{x}), \quad (1.32)$$

where φ_i is the Lagrange function associated with \mathbf{a}_i , $1 \leq i \leq N_h$. The following lemma states the stability of π_i .

Lemma 1.5.4 *For all p , $1 \leq p \leq +\infty$, there exists a constant c such that, for $1 \leq i \leq N_h$ and for all functions $v \in L_1^p(T_i)$*

$$\|\pi_i v\|_{L_1^p(T_i)} \leq c \|v\|_{L_1^p(T_i)}. \quad (1.33)$$

Proof On the reference element \hat{T} , we define the projection operator $\hat{\pi}$ such that $\hat{\pi} \hat{v} = \hat{\pi}_i \hat{v}$, namely $\hat{\pi}$ satisfies (1.31) with T_i replaced by \hat{T} and the measure $r d\mathbf{x}$ replaced by $\rho_{T_i}(\zeta, \eta) d\zeta d\eta$ with ρ_{T_i} equal to $r \circ F_{T_i}$.

With the notation introduced in the proof of proposition 1.5.1, the function ρ_{T_i} is equivalent to $\delta^p \chi$. So we derive from Hölder's inequality that, for p' such that $\frac{1}{p} + \frac{1}{p'} = 1$,

$$\|\hat{\pi} \hat{v}\|_{L_\chi^2(\hat{T})}^2 \leq c \|\hat{v}\|_{L_\chi^p(\hat{T})} \|\hat{\pi} \hat{v}\|_{L_\chi^{p'}(\hat{T})}.$$

Next, we obtain from the equivalence of weighted norms on $P_k(T_i)$ for the three weights corresponding to the different values of χ that

$$\|\hat{\pi} \hat{v}\|_{L_\chi^p(\hat{T})} \leq c \|\hat{\pi} \hat{v}\|_{L_\chi^2(\hat{T})}, \quad \|\hat{\pi} \hat{v}\|_{L_\chi^{p'}(\hat{T})} \leq c' \|\hat{\pi} \hat{v}\|_{L_\chi^2(\hat{T})}.$$

Combining all this gives

$$\|\hat{\pi} \hat{v}\|_{L_\chi^p(\hat{T})} \leq c \|\hat{v}\|_{L_\chi^p(\hat{T})}.$$

1.5. TECHNICAL RESULTS

By applying now the transformation F_{T_i} leads to

$$\|\pi_i v\|_{L_1^p(T_i)} \leq c \delta \|\hat{\pi} \hat{v}\|_{L_\chi^p(\hat{T})} \leq c' \delta \|\hat{v}\|_{L_\chi^p(\hat{T})} \leq c'' \|\pi_i v\|_{L_1^p(T_i)},$$

which is the desired result. ■

The following theorem states the first approximation properties of Π_h .

Theorem 1.5.1 *For all integers ℓ , $0 \leq \ell \leq k+1$, and for all p , $1 \leq p \leq +\infty$, there exists a constant C , independent of h_T , such that, for all $T \in \mathcal{T}_h$ and all functions $v \in W_1^{\ell,p}(\Delta_T)$, the following inequalities hold*

$$\|v - \Pi_h v\|_{L_1^p(T)} \leq C h_T^\ell |v|_{W_1^{\ell,p}(\Delta_T)} \quad (1.34)$$

and, when $\ell \geq 1$,

$$|v - \Pi_h v|_{W_1^{1,p}(T)} \leq C h_T^{\ell-1} |v|_{W_1^{\ell,p}(\Delta_T)}. \quad (1.35)$$

Proof The proof of (1.34) is divided into three cases: $\ell = 0$, $0 < \ell \leq \frac{3}{p}$ and $\ell > \frac{3}{p}$ (or $\ell = 3$, $p = 1$).

Case $\ell = 0$. It holds

$$\|\Pi_h v\|_{L_1^p(T)} \leq \sum_{i=1}^{N_h} \alpha_i \|\pi_i v(\mathbf{a}_i) \varphi_i\|_{L_1^p(T)}, \quad (1.36)$$

where α_i is equal to 1 if the intersection of the support of φ_i with T is not empty, and zero otherwise.

For any fixed i , we can write

$$\|\pi_i v(\mathbf{a}_i) \varphi_i\|_{L_1^p(T)} \leq \|\pi_i v\|_{L^\infty(T_i)} \|\varphi_i\|_{L_1^p(T)}.$$

If T_i is of type 1, then we obtain from lemma 1.4.2 and a standard inverse inequality

$$\|\pi_i v(\mathbf{a}_i) \varphi_i\|_{L_1^p(T)} \leq \hat{c} \left(\frac{\max_T r}{\min_{T_i} r} \right)^{\frac{1}{p}} h_{T_i}^{-\frac{2}{p}} h_T^{\frac{2}{p}} \|\pi_i v\|_{L_1^p(T_i)}. \quad (1.37)$$

If T_i is of type 2, it follows from lemma 1.4.2 and the fact that $\max_{\mathbf{x} \in T} r(\mathbf{x}) \leq c h_T$ that

$$\|\varphi_i\|_{L_1^p(T)} \leq c h_T^{\frac{3}{p}}. \quad (1.38)$$

On the other hand, we have

$$\|\pi_i v\|_{L^\infty(T_i)} = \|\widehat{\pi_i v}\|_{L^\infty(\hat{T})} \leq c \|\widehat{\pi_i v}(\hat{\lambda}_1)\|_{L^p(\hat{T})}^{\frac{1}{p}},$$

whence

$$\|\pi_i v\|_{L^\infty(T_i)} \leq c h_{T_i}^{-\frac{3}{p}} \|\pi_i v\|_{L_1^p(T_i)}.$$

Combining this with (1.38) gives

$$\|(\pi_i v)(\mathbf{a}_i) \varphi_i\|_{L_1^p(T)} \leq c \|\pi_i v\|_{L_1^p(T_i)}. \quad (1.39)$$

If T_i is of type 3, the same arguments applies and this estimate still holds.

Inserting (1.37), respectively (1.39), into (1.36), we obtain

$$\|\Pi_h v\|_{L_1^p(T)} \leq \hat{c} \sum_{i=1}^{N_h} \alpha_i \|\pi_i v\|_{L_1^p(T_i)},$$

Noting that the number of non zero α_i is bounded only as a function of k , we deduce from lemma 1.5.4 the inequality

$$\|\Pi_h v\|_{L_1^p(T)} \leq c \|v\|_{L_1^p(\Delta_T)}. \quad (1.40)$$

Combining this with a triangle inequality yield (1.34) when $\ell = 0$.

Case $\ell \leq \frac{3}{p}$. We note that for any polynomial $q \in P_k(\Delta_T)$, and for all nodes \mathbf{a}_i in T , $\pi_i q$ is equal to q , therefore the restriction of $\Pi_h q$ to T is also equal to q . Hence

$$\|v - \Pi_h v\|_{L_1^p(T)} = \|v - q + \Pi_h(v - q)\|_{L_1^p(T)} \leq \|v - q\|_{L_1^p(T)} + \|\Pi_h(v - q)\|_{L_1^p(T)}.$$

Using (1.40), we obtain

$$\|v - \Pi_h v\|_{L_1^p(T)} \leq c \|v - q\|_{L_1^p(\Delta_T)}. \quad (1.41)$$

Combining with the result of lemma 1.5.2, respectively lemma 1.5.3 yield (1.34) when $\ell \leq \frac{3}{p}$.

Case $\ell > \frac{3}{p}$ (or $\ell = 3, p = 1$). The functions of $W_1^{\ell,p}(\Delta_T)$ are continuous, therefore we can use the Lagrange interpolation operator \mathcal{I}_h . Noting that for all \mathbf{a}_i , $\pi_i(\mathcal{I}_h v)$ is equal to $(\mathcal{I}_h v)|_{T_i}$, we have $\Pi_h(\mathcal{I}_h v)$ which is equal to $\mathcal{I}_h v$. Whence

$$\|v - \Pi_h v\|_{L_1^p(T)} \leq \|v - \mathcal{I}_h v\|_{L_1^p(T)} + \|\Pi_h(v - \mathcal{I}_h v)\|_{L_1^p(T)},$$

Using once again (1.40) leads to

$$\|v - \Pi_h v\|_{L_1^p(T)} \leq c \|v - \mathcal{I}_h v\|_{L_1^p(\Delta_T)}.$$

The result follows from proposition 1.5.1.

The proof of (1.35) is the same as the previous one with obvious modifications (see lemma 1.4.2). ■

Following the same lines we deduce the following statement.

Corollary 1.5.1 *For all ℓ , $1 \leq \ell \leq k+1$, and for all p , $1 \leq p \leq +\infty$, there exists a constant c , independent of h , such that, for all elements T in \mathcal{T}_h , and all edges e of T which are not contained in Γ_0 , and for all functions $v \in W_1^{\ell,p}(\Delta_T)$, the following inequality holds*

$$\|v - \Pi_h v\|_{L_1^p(e)} \leq c h_T^{\ell - \frac{1}{p}} |v|_{W_1^{\ell,p}(\Delta_T)}. \quad (1.42)$$

Summing over all elements T , we obtain the global result

Corollary 1.5.2 *For all ℓ , $1 \leq \ell \leq k+1$, and for all p , $1 \leq p \leq +\infty$, there exists a constant c , independent of h , such that, for any function $v \in W_1^{\ell,p}(\Omega)$,*

$$\|v - \Pi_h v\|_{L_1^p(\Omega)} \leq c h^\ell |v|_{W_1^{\ell,p}(\Omega)}. \quad (1.43)$$

Other weighted Clément operators

To take into account boundary conditions, we introduce now a modified operator Π_h^0 which preserves the null trace on the boundary Γ of Ω :

$$\Pi_h^0 v = \sum_{i=1, \mathbf{a}_i \notin \Gamma}^{N_h} (\pi_i v)(\mathbf{a}_i) \varphi_i. \quad (1.44)$$

The operator Π_h has the same approximation properties given in theorem 1.5.1 and the proof is similar to the unweighted case, see [BMR04], theorem 3.11.

Corollary 1.5.3 *Estimates (1.34), (1.35) and (1.42) still holds with Π_h replaced by Π_h^0 , for all functions v in $W_1^{\ell,p}(\Delta_T)$ vanishing on $\Gamma \cap \Delta_T$.*

We need also to introduce two other operators $\tilde{\Pi}_h$, which maps $V_1^1(\Omega)$ into $X_h \cap V_1^1(\Omega)$ (the space of Lagrange finite elements of order k vanishing at Γ_0), and $\tilde{\Pi}_h^0$, which maps $V_{1\circ}^1(\Omega)$ into $X_h \cap V_{1\circ}^1(\Omega)$ (the space of Lagrange finite elements of order k vanishing at $\Gamma_0 \cup \Gamma$), defined as follows:

$$\tilde{\Pi}_h v = \sum_{i=1, \mathbf{a}_i \notin \Gamma_0}^{N_h} (\pi_i v)(\mathbf{a}_i) \varphi_i. \quad (1.45)$$

$$\tilde{\Pi}_h^0 v = \sum_{i=1, \mathbf{a}_i \notin (\Gamma_0 \cup \Gamma)}^{N_h} (\pi_i v)(\mathbf{a}_i) \varphi_i. \quad (1.46)$$

Since we do not have any application for the approximation properties of these operators for all the spaces $W_1^{\ell,p}(\Omega)$, we restrict ourselves to the case $p = 2$. We state the main result in the following theorem.

Theorem 1.5.2 *For all ℓ , $1 \leq \ell \leq k + 1$, there exists a constant c , independent of h , such that, for all elements T in \mathcal{T}_h , and for all functions $v \in H_1^\ell(\Delta_T) \cap V_1^1(\Delta_T)$, the following inequality holds*

$$\left(h_T^{-1} \|v - \tilde{\Pi}_h v\|_{L_1^2(T)} + \|v - \tilde{\Pi}_h v\|_{V_1^1(T)} \right) \leq C h_T^{\ell-1} \|v\|_{H_1^\ell(\Delta_T) \cap V_1^1(\Delta_T)}. \quad (1.47)$$

The same estimate holds with $\tilde{\Pi}_h$ replaced by $\tilde{\Pi}_h^0$ and for all v in $H_1^\ell(\Delta_T) \cap V_{1\circ}^1(\Delta_T)$.

Proof We consider two cases.

Case $\ell = 1$. We can write

$$\|\tilde{\Pi}_h v\|_{V_1^1(T)} \leq \sum_{i=1, \mathbf{a}_i \notin \Gamma_0}^{N_h} \|\pi_i v\|_{L^\infty(T_i)} \|\varphi_i\|_{V_1^1(T)}.$$

Using (1.7) and (1.8) for evaluating $\|\varphi_i\|_{V_1^1(T)}$ and the same arguments as previously for bounding $\|\pi_i v\|_{L^\infty(T_i)}$ according to T_i being of type 1, 2 or 3, we derive

$$\|\tilde{\Pi}_h v\|_{V_1^1(T)} \leq c \|v\|_{V_1^1(T)},$$

which yields one part of (1.47). On the other hand, we have

$$\|v - \tilde{\Pi}_h v\|_{L_1^2(T)} \leq \|v - \Pi_h v\|_{L_1^2(T)} + \|\Pi_h v - \tilde{\Pi}_h v\|_{L_1^2(T)},$$

and the first term in the right-hand side satisfies the desired estimate, see (1.34). We also note that the second term vanishes on triangles T of type 1. If T is of type 2 or 3, we derive from (1.7) that

$$\|\Pi_h v - \tilde{\Pi}_h v\|_{L_1^2(T)} = \sum_{a_i \in \Gamma_0 \cap T} h_T^{\frac{3}{2}} \|\pi_i v\|_{L^\infty(T_i)},$$

where T_i is also of type 2 or 3. If T_i is of type 2 for instance, we derive by the same arguments as in the proof of lemma 1.5.4,

$$\|\pi_i v\|_{L^\infty(T_i)} = \|\widehat{\pi_i v}\|_{L^\infty(\hat{T})} \leq c \|\widehat{\pi_i v}(\hat{\lambda}_1)^{\frac{1}{2}}\|_{L^\infty(\hat{T})} \leq c \|\widehat{\pi_i v}(\hat{\lambda}_1)^{\frac{1}{2}}\|_{L^2(\hat{T})} \leq c \|\hat{v}(\hat{\lambda}_1)^{\frac{1}{2}}\|_{L^2(\hat{T})}.$$

By applying the Poincaré–Friedrichs inequality to the function $\widehat{\pi_i v}(\hat{\lambda}_1)^{\frac{1}{2}}$ which vanishes on one edge of \hat{T} , we obtain

$$\|\pi_i v\|_{L^\infty(T_i)} \leq c (\|\nabla \hat{v}(\hat{\lambda}_1)^{\frac{1}{2}}\|_{L^2(\hat{T})^2} + \|\hat{v}(\hat{\lambda}_1)^{-\frac{1}{2}}\|_{L^2(\hat{T})}).$$

Going back to T_i thus gives

$$\|\pi_i v\|_{L^\infty(T_i)} \leq c h_{T_i}^{1-\frac{3}{2}} \|v\|_{V_1^1(T_i)}.$$

The same estimate holds when T_i is of type 3, by using similar arguments and applying the Poincaré–Friedrichs inequality to functions in the space

$$\hat{W} = \left\{ \hat{w}(\hat{\lambda}_2 + \hat{\lambda}_3)^{\frac{1}{2}}; \hat{w} \in H^1(\hat{T}) \right\},$$

see [GR86, chap. I], theorem 2.1. This concludes the proof of (1.47).

Case $\ell \geq 2$. Since $H_1^\ell(T) \subset C^0(\bar{T})$ and $\tilde{\Pi}_h(\mathcal{I}_h v)$ is equal to $\mathcal{I}_h v$ for all functions v in $H_1^\ell(\Delta_T) \cap V_1^1(\Delta_T)$, we derive from inequality (1.47) for $\ell = 1$

$$h_T^{-1} \|v - \tilde{\Pi}_h v\|_{L_1^2(T)} + \|v - \tilde{\Pi}_h v\|_{V_1^1(T)} \leq c \|v - \mathcal{I}_h v\|_{V_1^1(\Delta_T)}.$$

Estimate (1.47) follows by combining proposition 1.5.1 with a further result proved in [MR82], lemma 6.1. ■

To end this section, we give a useful stability property, that we state in the following theorem

Theorem 1.5.3 *There exists a constant c , independent of h , such that, for all elements T in \mathcal{T}_h , and for all functions $v \in L_{-1}^2(T)$, the following inequality holds*

$$\|\tilde{\Pi}_h v\|_{L_{-1}^2(T)} \leq c \|v\|_{L_{-1}^2(\Delta_T)}. \quad (1.48)$$

Proof We have

$$\|\tilde{\Pi}_h v\|_{L_{-1}^2(T)} \leq \sum_{i=1, \mathbf{a}_i \notin \Gamma_0}^{N_h} \|\pi_i v\|_{L^\infty(T_i)} \|\varphi_i\|_{L_{-1}^2(T)}.$$

Combining inequalities (1.7), (1.8) and, when T_i is of type 2 or 3 for instance,

$$\|\pi_i v\|_{L^\infty(T_i)} \leq h_T^{-\frac{1}{2}} \|v\|_{L_{-1}^2(T_i)}$$

yields (1.48). ■

As a consequence of the previous results, we have

Corollary 1.5.4 *Let us define the operator $R_h : V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \rightarrow V_{h/2}$,*

$$R_h(u_r, u_z) = \left(\tilde{\Pi}_h^0 u_r, \Pi_h^0 u_z \right).$$

Then there exist positive constants c and c' such that for all triangles T in \mathcal{T}_h and all functions \mathbf{v} in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega)$

$$\begin{aligned} \|\mathbf{v} - R_h \mathbf{v}\|_{L_1^2(T)^2} &\leq c h_T \|\mathbf{v}\|_{V_1^1(\Delta_T) \times H_1^1(\Delta_T)}, \\ \|\mathbf{v} - R_h \mathbf{v}\|_{V_1^1(T) \times H_1^1(T)} &\leq c' \|\mathbf{v}\|_{V_1^1(\Delta_T) \times H_1^1(\Delta_T)}. \end{aligned}$$

1.5.2 Axisymmetric Inf-Sup condition in $(V_1^1(\Omega) \times H_1^1(\Omega)) \times H_1^1(\Omega)$

The following proposition is the weighted version of a result due to Bercovier and Pironneau in [BP79] about the Cartesian inf-sup condition for P1isoP2/P1 elements. We refer to [BF91, §VI.6] for the idea of the extension to a regular family of triangulations.

Proposition 1.5.2 *There exists a positive constant c independent of h such that*

$$\forall q_h \in Q_h, \sup_{\mathbf{v} \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq c \left(\sum_{T \in \mathcal{T}_h} h_T^2 |q_h|_{H_1^1(T)}^2 \right)^{1/2}.$$

Proof Let q_h be a given element of Q_h . Let T_k and T_j be two triangles of \mathcal{T}_h with a common side, two common vertices denoted by \mathbf{x}^k and \mathbf{x}^j and let \mathbf{x}^{kj} be their midpoint $\frac{1}{2}(\mathbf{x}^k + \mathbf{x}^j)$ (see figure 1.4). Each element T_k of \mathcal{T}_h being divided into four sub-triangles by joining the midpoints of its edges, let D_k be the union of the three sub-triangles of T_k with one vertex being \mathbf{x}^{kj} . Note d_k the weighted measure of D_k :

$$d_k = \int_{D_k} r d\mathbf{x}.$$

Define D_j in the same way. To simplify the notation we neglect $d\mathbf{x}$ in the integrals.

We define an element \mathbf{v}_h of $V_{h/2}$, being equal to zero at each vertex of any T in \mathcal{T}_h and on $\Gamma_0 \cup \Gamma$, and equal to arbitrary real vectors \mathbf{v}_{kj} at all midpoints \mathbf{x}^{kj} . We can also write \mathbf{v}_h as $\sum_{(kj)} \varphi^{kj} \mathbf{v}_{kj}$, where φ^{kj} is a basis function of $V_{h/2}$ which is one at \mathbf{x}^{kj} and zero at the other vertices of triangles of $\mathcal{T}_{h/2}$. The end of the proof is divided into five steps.

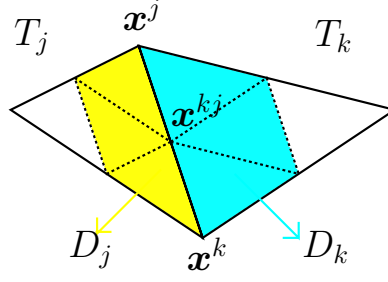


Figure 1.4: The two triangles T_k and T_j .

Step 1. The norm of $\varphi^{kj} \mathbf{v}_{kj}$ is bounded by:

$$\|\varphi^{kj} \mathbf{v}_{kj}\|_{L_1^2(T_k)}^2 \leq \int_{D_k} r |\varphi^{kj} \mathbf{v}_{kj}|^2 \leq |\mathbf{v}_{kj}|^2 d_k.$$

Then

$$\|\mathbf{v}_h\|_{L_1^2(\Omega)}^2 \leq \sum_{(kj)} \|\varphi^{kj} \mathbf{v}_{kj}\|_{L_1^2(\Omega)}^2 \leq \sum_{(kj)} |\mathbf{v}_{kj}|^2 (d_k + d_j).$$

By the inverse inequality (1.11) and noting h_k the diameter h_{T_k} ,

$$\begin{aligned} \|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}^2 &\leq c \sum_{T_k \in \mathcal{T}_h} h_k^{-2} \|\mathbf{v}_h\|_{L_1^2(T_k)}^2 \leq c \sum_{T_k \in \mathcal{T}_h} \sum_{T_j \cap T_k = \text{edge}} h_k^{-2} |\mathbf{v}_{kj}|^2 d_k \\ &\leq c \sum_{(kj)} \left(h_k^{-2} d_k + h_j^{-2} d_j \right) |\mathbf{v}_{kj}|^2 \leq c \sigma^{-2} \sum_{(kj)} h_{kj}^{-2} |\mathbf{v}_{kj}|^2 (d_k + d_j), \end{aligned}$$

where $h_{kj} = \max\{h_k, h_j\}$ and the last inequality holds since T_k and T_j have a common edge. We have shown that

$$\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)} \leq c \sum_{(kj)} h_{kj}^{-2} |\mathbf{v}_{kj}|^2 (d_k + d_j). \quad (1.49)$$

Step 2. The vector field ∇q_h is constant on T_k and it is noted by ∇q_k . Let χ_D be the characteristic function of a set D , then by integration by parts

$$b(\varphi^{kj} \mathbf{v}_{kj}, q_h) = \int_{\Omega} r \varphi^{kj} \mathbf{v}_{kj} \cdot \nabla q_h = \int_{\Omega} r \varphi^{kj} \mathbf{v}_{kj} \cdot (\nabla q_k \chi_{D_k} + \nabla q_j \chi_{D_j}).$$

Now by considering the cases where T_k is of type 1, 2 or 3 and using [BMR04, §VII], proposition 2.3, we check that there exist two positive constants α_1 and α_2 only depending on σ and a scalar ρ_{kj} with $\alpha_1 < \rho_{kj} < \alpha_2$, such that

$$\int_{D_k} \varphi^{kj} r d\mathbf{x} = d_k \rho_{kj}.$$

Explicit values for α_1 and α_2 may be found in lemma A.1.1 in appendix A. This implies that

$$\int_{\Omega} r \varphi^{kj} \mathbf{v}_{kj} \cdot \nabla q_k \chi_{D_k} = \int_{D_k} r (\nabla q_k \cdot \varphi^{kj} \mathbf{v}_{kj}) = (\nabla q_k \cdot \mathbf{v}_{kj}) d_k \rho_{kj},$$

1.5. TECHNICAL RESULTS

hence

$$b(\mathbf{v}_h, q_h) = \sum_{(kj)} (\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j) \cdot \mathbf{v}_{kj}. \quad (1.50)$$

Step 3. Equation (1.49) together with (1.50) gives

$$\max_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{V_1^1(\Omega) \times H_1^1(\Omega)}} \geq c \max_{(\mathbf{v}_{kj})_{(kj)}} \left[\frac{\sum_{(kj)} (\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j) \cdot \mathbf{v}_{kj}}{\sum_{(kj)} h_{kj}^{-2} |\mathbf{v}_{kj}|^2 (d_k + d_j)} \right]. \quad (1.51)$$

We now take $\mathbf{v}_{kj} = h_{kj}^2 (\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j) / (d_k + d_j)$, so we are in the case where the Cauchy–Schwarz inequality becomes an equality:

$$\begin{aligned} \sum_{(kj)} \left[\frac{\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j}{h_{kj}^{-1} (d_k + d_j)^{\frac{1}{2}}} \cdot h_{kj}^{-1} \mathbf{v}_{kj} (d_k + d_j)^{\frac{1}{2}} \right] \\ = \left[\sum_{(kj)} \frac{|\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j|^2}{h_{kj}^{-2} (d_k + d_j)} \right]^{\frac{1}{2}} \left[\sum_{(kj)} h_{kj}^{-2} |\mathbf{v}_{kj}|^2 (d_k + d_j) \right]^{\frac{1}{2}}. \end{aligned} \quad (1.52)$$

Therefore, from (1.51) and (1.52)

$$\max_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_0} \geq c \left[\sum_{(kj)} h_{kj}^2 \frac{|\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j|^2}{d_k + d_j} \right]^{\frac{1}{2}}. \quad (1.53)$$

Step 4. Since in T_k the gradient $\nabla q_h|_{T_k} = \nabla q_k$ is constant, setting $q_k = q_h(\mathbf{x}^k)$ leads to

$$\nabla q_k \cdot \frac{\mathbf{x}^k - \mathbf{x}^j}{|\mathbf{x}^k - \mathbf{x}^j|} = \frac{q_k - q_j}{|\mathbf{x}^k - \mathbf{x}^j|}$$

and the same holds in T_j . Then

$$(\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j) \cdot \frac{\mathbf{x}^k - \mathbf{x}^j}{|\mathbf{x}^k - \mathbf{x}^j|} = \frac{(\rho_{kj} d_k + \rho_{jk} d_j) (q_k - q_j)}{|\mathbf{x}^k - \mathbf{x}^j|}.$$

Note that the ρ_{kj} are larger than α_1 . For a vector \mathbf{a} and a unit vector \mathbf{e} , $|\mathbf{a}|^2 \geq |\mathbf{a} \cdot \mathbf{e}|^2$. Taking $\mathbf{e} = \frac{\mathbf{x}^k - \mathbf{x}^j}{|\mathbf{x}^k - \mathbf{x}^j|}$, the square of the previous inequality gives

$$(\rho_{kj} d_k \nabla q_k + \rho_{jk} d_j \nabla q_j)^2 \geq c \frac{(d_k + d_j)^2 (q_k - q_j)^2}{|\mathbf{x}^k - \mathbf{x}^j|^2}. \quad (1.54)$$

Step 5. We note \mathbf{x}^l the third vertex of T_k . Lemmas A.1.2 and A.1.3 and equality $\nabla q_k \cdot \frac{\mathbf{x}^k - \mathbf{x}^j}{|\mathbf{x}^k - \mathbf{x}^j|} = \frac{q_h(\mathbf{x}^k) - q_h(\mathbf{x}^j)}{|\mathbf{x}^k - \mathbf{x}^j|}$ allow to write for $m = k$ and l

$$|\nabla q_k|^2 \leq c \left[\frac{(q_k - q_l)^2}{|\mathbf{x}^k - \mathbf{x}^l|^2} + \frac{(q_m - q_j)^2}{|\mathbf{x}^m - \mathbf{x}^j|^2} \right]. \quad (1.55)$$

Inequalities (1.53) and (1.54) yield

$$\max_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_0} \geq c \left\{ \sum_{T_k \in \mathcal{T}_h} \sum_{(\mathbf{x}^k, \mathbf{x}^j) \text{ edges of } T_k} h_{kj}^2 \frac{(q_k - q_j)^2}{|\mathbf{x}^k - \mathbf{x}^j|^2} d_{kj} \chi_{\{\mathbf{x}^{kj} \notin \Gamma \cup \Gamma_0\}} \right\}^{\frac{1}{2}},$$

where we have replaced d_k by d_{kj} , which is the weighted measure of the union of the three sub-triangles of T_k which have a vertex on the edge $(\mathbf{x}^k, \mathbf{x}^j)$ and \mathbf{x}^{kj} is the midpoint of the edge. Since for each triangle T in \mathcal{T}_h there is at least one vertex inside Ω , in the previous equation at least two midpoints of the edges are not in $\Gamma \cup \Gamma_0$.

Let t_k denote the weighted measure of the triangle T_k . From lemma A.1.4 $d_{kj} \geq \frac{3}{8}t_k$, hence inequality (1.55) leads to

$$\max_{\mathbf{v}_h \in V_{h/2}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_0} \geq c \left(\sum_{T_k \in \mathcal{T}_h} h_k^2 t_k |\nabla q_k|^2 \right)^{\frac{1}{2}} = c \left(\sum_{T \in \mathcal{T}_h} h_T^2 \|\nabla q_h\|_{L_1^2(T)}^2 \right)^{\frac{1}{2}}.$$

The last equality holds since

$$\|\nabla q_h\|_{L_1^2(T_k)}^2 = \int_{T_k} r |\nabla q_h|^2 = (\nabla q_k)^2 \int_{T_k} r = (\nabla q_k)^2 t_k.$$

To prove the proposition it is now enough to show that $(\mathbf{v}_h, \nabla q_h) = -b(\mathbf{v}_h, q_h)$. Since $\check{\mathbf{v}}_h = 0$ on $\check{\Gamma}$, integrating by parts yields

$$(\mathbf{v}_h, \nabla q_h) = \int_{\check{\Omega}} \check{\mathbf{v}}_h \nabla \check{q}_h = - \int_{\check{\Omega}} \operatorname{div} \check{\mathbf{v}}_h q_h = -b(\mathbf{v}_h, q_h).$$

■

Chapter 2

Axisymmetric Navier–Stokes equations

Introduction

In this chapter we formulate the axisymmetric Navier–Stokes problem. The assumptions on the boundary data, on the domain and on the mesh are the same as in chapter 1. In the first section we report some theoretical results on the analytical solution of the weak Navier–Stokes problem in the steady case. They are derived from existing results in [BDM99, §IX.2] and even though we present them in the case of homogeneous boundary data, they still hold in the non-homogeneous one.

The unsteady case with moving domains is the subject of the following section. We present the weak problem in *Arbitrary Lagrangian–Eulerian* (ALE) form. This formulation may be used when dealing with moving domains and consists in recasting the governing differential equation and the related weak formulation in a frame of reference moving with the domain.

We use the definition of the ALE mapping and its discretization presented by Nobile in his PhD thesis [Nob01], and we extend its use to the axisymmetric formulation of the Navier–Stokes equations.

2.1 The steady case

Let Ω be defined as in the previous chapter, i.e., a half section of an axisymmetric three-dimensional domain $\check{\Omega}$. The stationary three-dimensional incompressible homogeneous Navier–Stokes equations reads

$$\begin{cases} -\nu\Delta\check{\mathbf{u}} + (\check{\mathbf{u}} \cdot \nabla)\check{\mathbf{u}} + \nabla\check{p} = \check{\mathbf{f}} & \text{in } \check{\Omega}, \\ \operatorname{div} \check{\mathbf{u}} = 0 & \text{in } \check{\Omega}, \\ \check{\mathbf{u}} = 0 & \text{on } \partial\check{\Omega}, \end{cases} \quad (2.1)$$

Where $\check{\mathbf{u}}$ is a three-dimensional vector field representing the fluid velocity, \check{p} the pressure (divided by the fluid density) and $\check{\mathbf{f}}$ the internal volumic forces. The scalar ν is the kinematic viscosity.

We assume that $\check{\mathbf{f}}$ is in $L^2(\check{\Omega})$ and that it is axisymmetric with angular component equal to zero.

We can write the weak problem as the coupling of the following problems on u_θ , $\mathbf{u} = (u_r, u_z)$ and p respectively. Recall that we describe Ω with Cartesian coordinates $\mathbf{x} = (r, z)$, which represent the axial and radial coordinates. In particular $d\mathbf{x} = drdz$.

P2.1 Find (\mathbf{u}, p) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$, such that for all (\mathbf{v}, q) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$,

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + d(\mathbf{u}, \mathbf{u}, \mathbf{v}) + d_{u_\theta}(u_\theta, v_r) + b(\mathbf{v}, p) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}, \\ b(\mathbf{u}, q) = 0. \end{cases} \quad (2.2)$$

P2.2 Find u_θ in $V_{1\circ}^1(\Omega)$, such that for all v_θ in $V_{1\circ}^1(\Omega)$,

$$a_1(u_\theta, v_\theta) + d_{\mathbf{u}}(u_\theta, v_\theta) - d_{u_\theta}(u_\theta, u_r) = 0. \quad (2.3)$$

The trilinear form $d(\cdot, \cdot, \cdot)$ and the bilinear forms $a_1(\cdot, \cdot)$, $d_{u_\theta}(\cdot, \cdot)$, $d_{\mathbf{u}}(\cdot, \cdot)$ are defined as

$$\begin{aligned} d(\mathbf{w}, \mathbf{u}, \mathbf{v}) &= \int_{\Omega} ((\mathbf{w} \cdot \nabla) \mathbf{u}) \cdot \mathbf{v} \, d\mathbf{x}, \\ a_1(u_\theta, v_\theta) &= \int_{\Omega} \nabla u_\theta \cdot \nabla v_\theta \, d\mathbf{x}, \\ d_{w_\theta}(u_\theta, v_r) &= - \int_{\Omega} w_\theta u_\theta v_r \, d\mathbf{x}, \\ d_{\mathbf{w}}(u_\theta, v_\theta) &= \int_{\Omega} (\mathbf{w} \cdot \nabla u_\theta) v_\theta \, d\mathbf{x}, \end{aligned}$$

while $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are defined in the previous chapter.

Bernardi, Dauge and Maday in [BDM99, §IX.2] show that the coupled problem P2.1-P2.2 has a unique solution if ν is (almost everywhere) greater than a constant ν_0 which depends only on the geometry and on the data. Moreover, for an arbitrary u_θ in $V_1^1(\Omega)$ there is a solution to problem P2.1. The constant function $u_\theta = 0$ is a solution of problem P2.2 for any \mathbf{u} in $V_1^1(\Omega) \times H_1^1(\Omega)$. Hence, the coupled problem P2.1-P2.2 reduces to

P2.3 Find (\mathbf{u}, p) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$, such that for all (\mathbf{v}, q) in $V_{1\circ}^1(\Omega) \times H_{1\circ}^1(\Omega) \times L_{1,0}^2(\Omega)$,

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + d(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}, \\ b(\mathbf{u}, q) = 0. \end{cases} \quad (2.4)$$

In this case a solution (unique if ν is large enough) of the weak three-dimensional Navier–Stokes problem is the axisymmetric solution given by

$$\begin{aligned} \check{\mathbf{u}}(r, \theta, z) &= (u_r(r, z), 0, u_z(r, z)) \quad \text{and} \\ \check{p}(r, \theta, z) &= p(r, z). \end{aligned}$$

2.2 The unsteady case of a moving domain

In this section we deal with the unsteady Navier–Stokes equations in a domain $\check{\Omega}_t \subset \mathbb{R}^3$ moving with a prescribed law. We assume that for all time t in $[0, T]$ the domain remains axisymmetric and bounded. As before, the problem can be reduced to a problem on half sections $\Omega_t \subset \mathbb{R}_+ \times \mathbb{R}$.

The notation is the same used in the previous section but here \mathbf{u} and p are also functions of t and we denote the strain rate tensor as

$$\epsilon(\check{\mathbf{u}}) = \frac{\nabla \check{\mathbf{u}} + \nabla \check{\mathbf{u}}^T}{2}.$$

The equations in the three-dimensional domain read

$$\begin{cases} \partial_t \check{\mathbf{u}} + (\check{\mathbf{u}} \cdot \nabla) \check{\mathbf{u}} - \operatorname{div} (2\nu \epsilon(\check{\mathbf{u}})) + \nabla \check{p} = \check{\mathbf{f}}, \\ \operatorname{div} \check{\mathbf{u}} = 0, \end{cases} \quad \text{in } \check{\Omega}_t, t > 0. \quad (2.5)$$

with axisymmetric initial condition $\check{\mathbf{u}}_0$ with zero angular component and boundary conditions

$$\begin{cases} \check{\mathbf{u}} = \check{\boldsymbol{\phi}} \text{ on } \check{\Gamma}_t^D, \\ -\check{p}\check{\mathbf{n}} + 2\nu \epsilon(\check{\mathbf{u}}) \cdot \check{\mathbf{n}} = \check{\boldsymbol{\sigma}} \text{ on } \check{\Gamma}_t^N. \end{cases} \quad (2.6)$$

Since $\operatorname{div} \check{\mathbf{u}} = 0$, the two expressions for the viscous forms

$$-\operatorname{div} (\nu \nabla \check{\mathbf{u}}) \quad \text{and} \quad -\operatorname{div} (2\nu \epsilon(\check{\mathbf{u}}))$$

do coincide. However, the induced natural conditions on the boundary (surface stresses that we need in our analysis from now on) are more appropriately expressed with the form in (2.5).

The boundary $\partial \check{\Omega}_t$ has been split into two disjoint regions $\check{\Gamma}_t^D$ and $\check{\Gamma}_t^N$. The function $\check{\boldsymbol{\phi}}$ describes the velocity of the fluid on the Dirichlet boundary and is supposed to be axisymmetric with zero angular component. The typical situation is when it is equal to the velocity of the wall in fluid structure interaction (see also chapter 3).

The imposed normal stress $\check{\boldsymbol{\sigma}}$ is also supposed to be axisymmetric with zero angular component, and the unit outward normal vector $\check{\mathbf{n}}$ has the same property because the domain is supposed to be axisymmetric at all times.

2.2.1 Conservative weak ALE formulation

When the computational domain changes, the so called ALE frame is normally adopted in view of the numerical approximation. The ALE frame may be defined similarly to the Lagrangian one, often used in continuum mechanics. See for example [Nob01] for a preliminary analysis and description of its discretization.

We would like to solve the Navier–Stokes equations in a moving domain in a time interval $I = (0, T)$. With this goal in mind, we define an ALE mapping on the half sections, we extend it to the three-dimensional domain, we write the weak form of the problem in the three-dimensional ALE framework and finally we bring it back to two dimensions using weighted integrals.

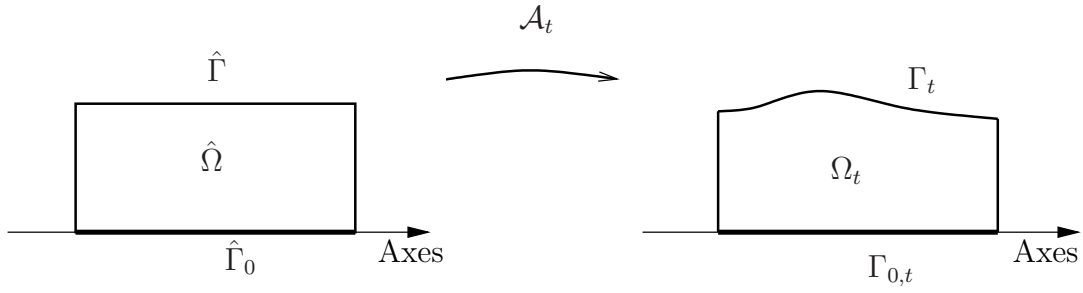


Figure 2.1: ALE mapping between the initial configuration and the configuration at time t .

Let $\hat{\Omega}$ be a reference two-dimensional configuration, for example the initial domain $\Omega_{t=0}$ and let $(\mathcal{A}_t)_t$ be a family of mappings, such that for each t , \mathcal{A}_t maps a point $\hat{\mathbf{x}}$ of $\hat{\Omega}$ to a point \mathbf{x} on Ω_t :

$$\begin{aligned} \mathcal{A} : \hat{\Omega} \times I &\rightarrow \mathbb{R}^2 \\ (\hat{\mathbf{x}}, t) &\mapsto \mathcal{A}(\hat{\mathbf{x}}, t) = \mathcal{A}_t(\hat{\mathbf{x}}). \end{aligned}$$

For simplicity, we note a function of (\cdot, t) with the subscript t . We define $\Omega \times I = \{(\mathbf{x}, t) \in \mathbb{R}^2 \times \mathbb{R}, \mathbf{x} \in \Omega_t\}$, with a little abuse of notation and the domain velocity \mathbf{w} as

$$\mathbf{w}_t(\mathbf{x}) = \frac{d\mathcal{A}_t(\hat{\mathbf{x}})}{dt}, \text{ where } \hat{\mathbf{x}} = \mathcal{A}_t^{-1}(\mathbf{x}).$$

For all t , we assume that \mathcal{A}_t is an homeomorphism from $\hat{\Omega}$ onto Ω_t , i.e., \mathcal{A}_t is continuous from the closure of $\hat{\Omega}$ onto the closure of Ω_t with a continuous inverse. Furthermore, we also assume that the application $t \mapsto \mathcal{A}_t(\hat{\mathbf{x}})$ is differentiable almost everywhere in I for all $\hat{\mathbf{x}}$ in $\hat{\Omega}$. Usually, $\hat{\mathbf{x}}$ is called ALE coordinate while \mathbf{x} is the Eulerian coordinate.

Let $\Gamma_{0,t}$ be the intersection of $\partial\Omega_t$ with the axis and Γ_t be $\partial\Omega_t \setminus \Gamma_{0,t}$. We assume that \mathcal{A}_t restricted to $\hat{\Gamma}_0$ is an homeomorphism onto $\Gamma_{0,t}$. We then can build an ALE mapping $\check{\mathcal{A}}_t$ from $\check{\hat{\Omega}}$ to $\check{\Omega}_t$ with the same properties as \mathcal{A}_t by simply adding the angular coordinate.

To derive the conservative weak ALE formulation of the Navier–Stokes equations on Ω , we start from its three-dimensional equivalent, i.e.,

$$\begin{aligned} \frac{d}{dt} \int_{\check{\Omega}_t} \check{\mathbf{u}} \cdot \check{\mathbf{v}} d\check{\mathbf{x}} + \int_{\check{\Omega}_t} [(\check{\mathbf{u}} - \check{\mathbf{w}}) \cdot \nabla] \check{\mathbf{u}} \cdot \check{\mathbf{v}} d\check{\mathbf{x}} + \int_{\check{\Omega}_t} \text{div}(\check{\mathbf{u}} - \check{\mathbf{w}}) \check{\mathbf{u}} \cdot \check{\mathbf{v}} d\check{\mathbf{x}} \\ + 2\nu \int_{\check{\Omega}_t} \epsilon(\check{\mathbf{u}}) : \epsilon(\check{\mathbf{v}}) d\check{\mathbf{x}} - \int_{\check{\Omega}_t} \text{div} \check{\mathbf{v}} \check{p} d\check{\mathbf{x}} = \int_{\check{\Omega}_t} \check{\mathbf{f}} \cdot \check{\mathbf{v}} d\check{\mathbf{x}} + \int_{\check{\Gamma}_t^N} \check{\boldsymbol{\sigma}} \cdot \check{\mathbf{v}} d\check{s}, \quad (2.7) \end{aligned}$$

$$- \int_{\check{\Omega}_t} \text{div} \check{\mathbf{u}} \check{q} d\check{\mathbf{x}} = 0, \quad (2.8)$$

which has to hold for all test vector fields $\check{\mathbf{v}}$ and all test functions \check{q} . We do not introduce the test spaces explicitly at this stage, since we are only interested in their correspondent on Ω_t , which are defined as follows.

2.2. THE UNSTEADY CASE OF A MOVING DOMAIN

For all t in $[0, T]$ we define

$$\begin{aligned} V(\Omega_t) &= \left\{ \mathbf{v} : \Omega_t \rightarrow \mathbb{R}^2, \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_t^{-1}, \hat{\mathbf{v}} \in V_1^1(\hat{\Omega}) \times H_1^1(\hat{\Omega}) \right\}, \\ V_{\Gamma^D}(\Omega_t) &= \left\{ \mathbf{v} \in V(\Omega_t), \mathbf{v} = 0 \text{ on } \Gamma_t^D \right\}, \\ Q(\Omega_t) &= \left\{ q : \Omega_t \rightarrow \mathbb{R}, q = \hat{q} \circ \mathcal{A}_t^{-1}, \hat{q} \in L_1^2(\hat{\Omega}) \right\}, \end{aligned} \quad (2.9)$$

while

$$\begin{aligned} V(\Omega) &= \left\{ \mathbf{v} : \Omega \times I \rightarrow \mathbb{R}^2, \mathbf{v}_t \in V(\Omega_t) \text{ for a.e. } t \right\}, \\ Q(\Omega) &= \left\{ q : \Omega \times I \rightarrow \mathbb{R}, q_t \in Q(\Omega_t) \text{ for a.e. } t \right\}. \end{aligned} \quad (2.10)$$

We recall that for an axisymmetric vector field on $\check{\Omega}_t$, we have

$$\operatorname{div} \check{\mathbf{v}} = \partial_r v_r + \partial_z v_z + \frac{1}{r} v_r \text{ and } \boldsymbol{\epsilon}(\check{\mathbf{u}}) : \boldsymbol{\epsilon}(\check{\mathbf{v}}) = \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) + \frac{1}{r} u_r v_r,$$

where we have set

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right)}{2}.$$

From now on we use the abridged notation $\operatorname{div} \mathbf{v} = \partial_r v_r + \partial_z v_z$. Remark that div is not the conjugate of $\nabla = (\partial_r, \partial_z)^T$ in the weighted Sobolev spaces.

Finally we can write the conservative formulation of the axisymmetric three-dimensional Navier–Stokes equations in moving domain as

P2.4 Find (\mathbf{u}, p) in $V(\Omega) \times Q(\Omega)$, with $\mathbf{u}(0) = \mathbf{u}_0$ and $\mathbf{u} = \boldsymbol{\phi}$ on $\Gamma \times I$, such that for almost every t in I and for all (\mathbf{v}, q) in $V_{\Gamma^D}(\Omega_t) \times Q(\Omega_t)$

$$\left\{ \begin{aligned} & \frac{d}{dt} \int_{\Omega_t} \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Omega_t} [(\mathbf{u} - \mathbf{w}) \cdot \nabla] \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Omega_t} \operatorname{div}(\mathbf{u} - \mathbf{w}) \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} \\ & + \int_{\Omega_t} (u_r - w_r) \mathbf{u} \cdot \mathbf{v} d\mathbf{x} + 2\nu \int_{\Omega_t} \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) r d\mathbf{x} + 2\nu \int_{\Omega_t} u_r v_r \frac{1}{r} d\mathbf{x} \\ & - \int_{\Omega_t} \operatorname{div} \mathbf{v} p r d\mathbf{x} - \int_{\Omega_t} v_r p d\mathbf{x} = \int_{\Omega_t} \mathbf{f} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Gamma_t^N} \boldsymbol{\sigma} \cdot \mathbf{v} r(s) ds, \\ & - \int_{\Omega_t} \operatorname{div} \mathbf{u} q r d\mathbf{x} - \int_{\Omega_t} u_r q d\mathbf{x} = 0. \end{aligned} \right.$$

Similar arguments may be applied to obtain a non-conservative formulation.

Energy inequality

Here we consider homogeneous Dirichlet boundary conditions and we recall that (see [Cia88]), as a consequence of the Korn inequality, for a given domain $\check{\Omega}_t$, there is a constant κ such that for all t , and all $\check{\mathbf{u}}$ vanishing on a subset of $\check{\Gamma}_t^D$ with positive measure,

$$2\nu \int_{\check{\Omega}_t} \boldsymbol{\epsilon}(\check{\mathbf{u}}) : \boldsymbol{\epsilon}(\check{\mathbf{u}}) d\check{\mathbf{x}} \geq \kappa \|\nabla \check{\mathbf{u}}\|_{L^2(\check{\Omega}_t)}^2.$$

We assume that this constant is independent of t . We also assume that the Poincaré and trace inequality hold uniformly with respect to t , i.e.,

$$\begin{aligned}\|\check{\mathbf{u}}\|_{L^2(\check{\Omega}_t)} &\leq C_P \|\nabla \check{\mathbf{u}}\|_{L^2(\check{\Omega}_t)}, \\ \|\check{\mathbf{u}}\|_{L^2(\check{\Gamma}_t^N)} &\leq \gamma \|\check{\mathbf{u}}\|_{H^1(\check{\Omega}_t)}.\end{aligned}$$

Nobile in [Nob01, §3.2] shows that under these assumptions the solution of equations (2.7) and (2.8) satisfies the following energy inequality,

$$\begin{aligned}\|\check{\mathbf{u}}_t\|_{L^2(\check{\Omega}_t)}^2 + \int_I \int_{\check{\Gamma}^N(\tau)} |\check{\mathbf{u}}|^2 (\check{\mathbf{u}} - \dot{\mathbf{g}}) \cdot \check{\mathbf{n}} d\check{s} d\tau + \kappa \int_I \|\nabla \check{\mathbf{u}}_\tau\|_{L^2(\check{\Omega}_\tau)}^2 \\ \leq \|\check{\mathbf{u}}_0\|_{L^2(\check{\Omega})}^2 + \frac{2(1 + C_P^2)}{\kappa} \int_I \left[\|\check{\mathbf{f}}_\tau\|_{H^{-1}(\Omega_\tau)}^2 + \gamma^2 \|\check{\boldsymbol{\sigma}}_\tau\|_{L^2(\check{\Gamma}_t^N)}^2 \right] d\tau,\end{aligned}$$

where $\dot{\mathbf{g}}$ is a three-dimensional extension of the velocity on the boundary of $\check{\Omega}_t$. On Ω_t , the above inequality yields

$$\begin{aligned}\|\mathbf{u}_t\|_{L_1^2(\Omega_t)}^2 + \int_I \int_{\Gamma^N(\tau)} |\mathbf{u}|^2 (\mathbf{u} - \dot{\mathbf{g}}) \cdot \mathbf{n} r(s) ds d\tau + \kappa \int_I |\mathbf{u}_\tau|_{V_1^1(\Omega_\tau) \times H_1^1(\Omega_\tau)}^2 d\tau \\ \leq \|\mathbf{u}_0\|_{L_1^2(\hat{\Omega})}^2 + \frac{2(1 + C_P^2)}{\kappa} \int_I \left[\|\mathbf{f}_\tau\|_{(V_1^1(\Omega_\tau) \times H_1^1(\Omega_\tau))'}^2 + \gamma^2 \|\boldsymbol{\sigma}_\tau\|_{L_1^2(\Gamma_t^N)}^2 \right] d\tau, \quad (2.11)\end{aligned}$$

where $\|\cdot\|_{(V_1^1(\Omega_t) \times H_1^1(\Omega_t))'}$ denotes the norm of the dual space of $V_1^1(\Omega_t) \times H_1^1(\Omega_t)$.

If Γ^N is empty, the previous inequality provides an a-priori estimate for the solution of the axisymmetric Navier–Stokes equations. In this case the convective term $\int_{\Omega_t} [\mathbf{u} \cdot \nabla] \mathbf{u} \cdot \mathbf{u} r d\mathbf{x}$ does not contribute to the energy inequality since

$$\begin{aligned}\int_{\Omega_t} [\mathbf{u} \cdot \nabla] \mathbf{u} \cdot \mathbf{u} r d\mathbf{x} &= \frac{1}{2\pi} \int_{\check{\Omega}_t} (\check{\mathbf{u}} \cdot \nabla) \check{\mathbf{u}} \cdot \check{\mathbf{u}} d\check{\mathbf{x}} = \frac{1}{4\pi} \int_{\check{\Omega}} \check{\mathbf{u}} \cdot \nabla |\check{\mathbf{u}}|^2 = -\frac{1}{4\pi} \int_{\check{\Omega}} \operatorname{div} \check{\mathbf{u}} |\check{\mathbf{u}}|^2 \\ &= -\frac{1}{2} \int_{\Omega_t} \operatorname{div} \mathbf{u} |\mathbf{u}|^2 r d\mathbf{x} - \frac{1}{2} \int_{\Omega_t} u_r |\mathbf{u}|^2 d\mathbf{x}.\end{aligned} \quad (2.12)$$

The second equation of problem P2.4 implies that the last term is equal to zero. To apply P2.4 we verify that $|\mathbf{u}_t|^2$ is in $Q(\Omega_t)$. Indeed, thanks to the Sobolev embedding in \mathbb{R}^3 ,

$$\|\mathbf{u}_t\|_{L_1^2(\Omega_t)}^2 = \|\mathbf{u}_t\|_{L_1^4(\Omega_t)}^2 = \|\check{\mathbf{u}}_t\|_{L^4(\check{\Omega}_t)}^2 < c \|\check{\mathbf{u}}_t\|_{H^1(\check{\Omega}_t)}^2 = c \|\mathbf{u}_t\|_{V_1^1(\Omega_t) \times H_1^1(\Omega_t)}^2.$$

On the contrary, if Γ^N is not empty, the term $\int_I \int_{\Gamma^N(\tau)} |\mathbf{u}|^2 (\mathbf{u} - \dot{\mathbf{g}}) \cdot \mathbf{n} ds d\tau$ does not have a definite sign and does not allow us to obtain the desired result. However, this is not due to the axisymmetric formulation, since it occurs also in the non-axisymmetric case and on a fixed domain. We remark that if $(\mathbf{u} - \dot{\mathbf{g}}) \cdot \mathbf{n} \geq 0$ on Γ_t^N for all t , the boundary term is positive and a global stability is thus recovered. This occurs when Γ_t^N is an outflow section; indeed, $\mathbf{u} - \dot{\mathbf{g}}$ represents the relative fluid velocity with respect to the moving boundary.

2.2.2 Construction of the ALE mapping

In the literature several techniques have been proposed (see for example [FLL98] and [Nob01]) to construct an ALE mapping. The fundamental problem is

2.2. THE UNSTEADY CASE OF A MOVING DOMAIN

P2.5 *Given the evolution of the moving boundary*

$$\mathbf{g} : \partial\hat{\Omega} \times I \rightarrow \partial\Omega_t,$$

find an ALE mapping \mathcal{A}_t , such that

$$\mathcal{A}_t(\hat{\mathbf{x}}) = \mathbf{g}(\hat{\mathbf{x}}, t) \quad \forall t \in I, \forall \hat{\mathbf{x}} \in \partial\hat{\Omega}.$$

Nobile in [Nob01, §1.4] proposes to solve this problem by a harmonic extension, i.e.,

P2.6 *For almost all t , find $\mathcal{A}_t : \hat{\Omega} \rightarrow \Omega_t$, such that*

$$\begin{cases} \Delta \mathcal{A}_t = 0 & \hat{\mathbf{x}} \in \hat{\Omega}, \\ \mathcal{A}_t(\hat{\mathbf{x}}) = \mathbf{g}_t(\hat{\mathbf{x}}) & \hat{\mathbf{x}} \in \partial\hat{\Omega}. \end{cases}$$

This approach is feasible as long as we can guarantee the invertibility of the mapping. Another possible choice is to solve problem P2.6 on $\hat{\Omega}$. In fact, the formulations differ only in the kind of integrals which are used: In the former, standard unweighted integrals are used, while the latter uses weighted ones.

The weighted version can be useful when one would like each element in the triangulation to always have nearly the same proportion of weighted measure, i.e., $\int_{T_t} r / \int_{\Omega_t} r = \int_{T_0} r / \int_{\Omega_0} r$ for almost every t . The weak formulation of P2.6 with weighted integral is

P2.7 *Find $\mathcal{A} : \hat{\Omega} \times I \rightarrow \Omega_t$, such that for almost all t in I , \mathcal{A}_t in $V_1^1(\hat{\Omega}) \times H_1^1(\hat{\Omega})$, $\mathcal{A}_t = \mathbf{g}_t$ on $\partial\hat{\Omega}$ and for all \mathbf{y} in $V_{1\circ}^1(\hat{\Omega}) \times H_{1\circ}^1(\hat{\Omega})$ with $\mathbf{y}|_{\partial\hat{\Omega}} = 0$,*

$$\int_{\hat{\Omega}} \nabla \mathcal{A}_t : \nabla \mathbf{y} r d\mathbf{x} + \int_{\hat{\Omega}} \mathcal{A}_{t,r} y_r \frac{1}{r} d\mathbf{x} = 0.$$

Since in the discretization of the Navier–Stokes problem we will start with a uniformly regular mesh, we will use the unweighted formulation to find the ALE mapping. The weak formulation of P2.6 with unweighted integral is

P2.8 *Find $\mathcal{A} : \hat{\Omega} \times I \rightarrow \Omega_t$, such that for almost all t in I , \mathcal{A}_t in $H^1(\hat{\Omega})^2$, $\mathcal{A}_t = \mathbf{g}_t$ on $\partial\hat{\Omega}$ and for all \mathbf{y} in $H^1(\hat{\Omega})^2$ with $\mathbf{y}|_{\partial\hat{\Omega}} = 0$,*

$$\int_{\hat{\Omega}} \nabla \mathcal{A}_t : \nabla \mathbf{y} d\mathbf{x} = 0.$$

The weighted form would be with $d\mathbf{x}$ replaced by $r d\mathbf{x}$, an extra term $x_{t,r} y_r \frac{1}{r} d\mathbf{x}$ and with test vector fields in $V_{1\circ}^1(\hat{\Omega}) \times H_{1\circ}^1(\hat{\Omega})$.

2.2.3 Finite element discretization

A finite element approximation of problem P2.4 involves both Navier–Stokes and ALE's discretization. For the Navier–Stokes problem, we choose, as in the first chapter, P1isoP2/P1 elements for which we have proved the discrete inf-sup compatibility condition. In particular, the finite element spaces depend on the discretization of the ALE mapping, which we choose to discretize with linear finite elements:

$$Y_h(\hat{\Omega}) = \left\{ \mathbf{y}_h \in \mathcal{C}^0(\hat{\Omega}, \mathbb{R}^2) : \mathbf{y}|_T \in P_1(T) \forall T \in \mathcal{T}_h \right\},$$

where $\{\mathcal{T}_h\}_h$ is a regular family of triangulations of $\hat{\Omega}$. For simplicity we suppose that the initial domain $\hat{\Omega}$ is polygonal and we note by \mathbf{g}_h a projection of the boundary movement to the traces of vector fields in $Y_h(\hat{\Omega})$. From now on, we abusively note by \mathcal{A}_t the discrete ALE mapping and by Ω_t the computational domain at time t . Then for a fixed t in I , the discrete ALE mapping, is the solution of

P2.9 Find \mathcal{A}_t in $Y_h(\hat{\Omega})$, with $\mathcal{A}_t(\hat{\mathbf{x}}) = \mathbf{g}_h(\hat{\mathbf{x}}, t)$ for all $\hat{\mathbf{x}}$ in $\partial\hat{\Omega}$ and such that for all \mathbf{y}_h in $Y_h(\hat{\Omega})$ with $\mathbf{y}_h|_{\partial\hat{\Omega}} = 0$,

$$\int_{\hat{\Omega}} \nabla \mathcal{A}_t : \nabla \mathbf{y}_h d\mathbf{x} = 0.$$

The finite element formulation of the Navier–Stokes equations depend on the two spaces

$$\begin{aligned} V_{h/2}(\hat{\Omega}) &= \left\{ \mathbf{v}_h : \hat{\Omega} \rightarrow \mathbb{R}^2, \mathbf{v}_h \in \mathcal{C}^0(\hat{\Omega}), v_r|_{\hat{\Gamma}_0} = 0, \mathbf{v}_h|_T \in P_1(T)^2, \forall T \in \mathcal{T}_{h/2} \right\}, \\ Q_h(\hat{\Omega}) &= \left\{ q_h : \hat{\Omega} \rightarrow \mathbb{R}, q_h \in \mathcal{C}^0(\hat{\Omega}), q_h|_T \in P_1(T), \forall T \in \mathcal{T}_h \right\}. \end{aligned}$$

The finite element spaces on Ω_t are defined as

$$\begin{aligned} V_{h/2}(\Omega_t) &= \left\{ \mathbf{v}_h : \Omega_t \rightarrow \mathbb{R}^2, \mathbf{v}_h = \hat{\mathbf{v}}_h \circ \mathcal{A}_t^{-1}, \hat{\mathbf{v}}_h \in V_{h/2}(\hat{\Omega}) \right\}, \\ Q_h(\Omega_t) &= \left\{ q_h : \Omega_t \rightarrow \mathbb{R}, q_h = \hat{q}_h \circ \mathcal{A}_t^{-1}, \hat{q}_h \in Q_h(\hat{\Omega}) \right\}. \end{aligned}$$

Then $V_{\Gamma^D, h/2}(\Omega_t)$, $V_{h/2}(\Omega)$ and $Q_h(\Omega)$ are defined in a similar way to the continuous case in (2.9) and (2.10). Note that the choice of finite elements for the discretization of the ALE mapping implies that vector fields in $V_{h/2}(\Omega_t)$ and functions in $Q_h(\Omega_t)$ are piecewise linear, such that they in fact define P1isoP2/P1 finite elements spaces on Ω_t .

Let $\mathbf{u}_{0,h}$ and \mathbf{f}_h be projections on the space $V_{h/2}(\Omega)$ of the initial condition \mathbf{u}_0 and the internal forces \mathbf{f} and ϕ_h that of the boundary condition on the trace of $V_{h/2}(\Omega)$. The semi-discrete formulation of the axisymmetric Navier–Stokes equation reads

P2.10 Find (\mathbf{u}_h, p_h) in $V_{h/2}(\Omega) \times Q_h(\Omega)$, with $\mathbf{u}_h(0) = \mathbf{u}_{0,h}$ and $\mathbf{u}_h = \phi_h$ on $\Gamma \times I$, such that for almost every t in I and for all (\mathbf{v}_h, q_h) in $V_{\Gamma^D, h/2}(\Omega_t) \times Q_h(\Omega_t)$

$$\left\{ \begin{aligned} &\frac{d}{dt} \int_{\Omega_t} \mathbf{u}_h \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Omega_t} [(\mathbf{u}_h - \mathbf{w}_h) \cdot \nabla] \mathbf{u}_h \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Omega_t} \operatorname{div} \left(\frac{1}{2} \mathbf{u}_h - \mathbf{w}_h \right) \mathbf{u}_h \cdot \mathbf{v}_h r d\mathbf{x} \\ &\quad + \int_{\Omega_t} \left(\frac{1}{2} u_{hr} - w_{hr} \right) \mathbf{u}_h \cdot \mathbf{v}_h d\mathbf{x} + 2\nu \int_{\Omega_t} \boldsymbol{\epsilon}(\mathbf{u}_h) : \boldsymbol{\epsilon}(\mathbf{v}_h) r d\mathbf{x} + 2\nu \int_{\Omega_t} u_{hr} v_{hr} \frac{1}{r} d\mathbf{x} \\ &\quad - \int_{\Omega_t} \operatorname{div} \mathbf{v}_h p_h r d\mathbf{x} - \int_{\Omega_t} v_{hr} p_h d\mathbf{x} = \int_{\Omega_t} \mathbf{f}_h \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Gamma_t^N} \boldsymbol{\sigma}_h \cdot \mathbf{v}_h r(s) ds, \\ & - \int_{\Omega_t} \operatorname{div} \mathbf{u}_h q_h r d\mathbf{x} - \int_{\Omega_t} u_{hr} q_h d\mathbf{x} = 0, \end{aligned} \right.$$

where we have added a consistent stabilizing term

$$-\frac{1}{2} \int_{\Omega_t} \operatorname{div} \mathbf{u}_h \mathbf{u}_h \cdot \mathbf{v}_h r d\mathbf{x} - \frac{1}{2} \int_{\Omega_t} (u_{hr}) \mathbf{u}_h \cdot \mathbf{v}_h d\mathbf{x},$$

to recover the same energy inequality as in the differential case. In fact equality (2.12) is not true in the discrete formulation, since $|\mathbf{u}_h|^2$ may not belong to $Q_h(\Omega_t)$.

2.2.4 Time discretization

Here we present an implicit Euler scheme applied to the conservative formulation given in problem P2.10. Other possibilities are presented for the Cartesian case in [Nob01] (including non-conservative formulation and second order scheme). We consider a scheme satisfying the Geometric Conservation Laws (GCL) introduced by Farhat et al. in [FGG01, LF95] and discussed in the case of Navier–Stokes equations in [NV99].

Geometric Conservation Laws have been originally investigated in the context of finite difference and finite volume schemes for fluid dynamic problems. It stems from the basic idea that the solution should be minimally affected by the domain movement law. Indeed, at the continuous level, the ALE formulation is formally equivalent to the original problem; yet this is not generally true when the fully discrete system is considered. It has been proposed that some ‘simple’ solution of the differential problem should be also solutions of the discrete system. In particular, the attention has been concentrated on the capability of the discrete system of representing a constant solution, which is clearly a solution of the differential equation (in the absence of the source term and with the appropriate boundary and initial conditions). Following this approach we can state that *a numerical scheme satisfies the Geometric Conservation Laws if it is able to reproduce a constant solution*. It is therefore, similar to the “patch test” often used by finite element practitioners. As we will see, the GCL constraint involves only mesh geometrical quantities and the domain velocity field. The significance of this condition is still not completely clear. Results are available for special type of finite-volume schemes in [FGG01, LF95] where the GCL have been linked to convergence properties of the proposed scheme.

Let Δt be the time step, let \mathbf{u}_h^0 be equal to the projection of the initial condition on the discrete velocities space and let \mathbf{u}_h^n be the approximation of the solution at time $t^n = n\Delta t$. The implicit Euler scheme reads:

P2.11 For any $n = 0, \dots, \frac{T}{\Delta t} - 1$, find $(\mathbf{u}_h^{n+1}, p^{n+1})$ in $V_{h/2} \times Q_h$ such that $\mathbf{u}_h^{n+1} = \phi_h$ on $\Gamma_{t^n}^D$ and for all (\mathbf{v}_h, q_h) in $V_{\Gamma^D, h/2} \times Q_h$,

$$\left\{ \begin{aligned} & \frac{1}{\Delta t} \int_{\Omega_{t^{n+1}}} \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h r d\mathbf{x} - \frac{1}{\Delta t} \int_{\Omega_{t^n}} \mathbf{u}_h^n \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Omega_{t^{n+1/2}}} [(\mathbf{u}_h^* - \mathbf{w}_h^{n+1/2}) \cdot \nabla] \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h r d\mathbf{x} \\ & + \int_{\Omega_{t^{n+1/2}}} \operatorname{div} \left(\frac{1}{2} \mathbf{u}_h^* - \mathbf{w}_h^{n+1/2} \right) \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Omega_{t^{n+1/2}}} \left(\frac{1}{2} u_{hr}^* - w_{hr}^{n+1/2} \right) \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h d\mathbf{x} \\ & + 2\nu \int_{\Omega_{t^{n+1/2}}} \epsilon(\mathbf{u}_h^{n+1}) : \epsilon(\mathbf{v}_h) r d\mathbf{x} + 2\nu \int_{\Omega_{t^{n+1/2}}} u_{hr}^{n+1} v_{hr} \frac{1}{r} d\mathbf{x} \\ & - \int_{\Omega_{t^{n+1/2}}} \operatorname{div} \mathbf{v}_h p_h^{n+1} r d\mathbf{x} - \int_{\Omega_{t^{n+1/2}}} v_{hr} p_h^{n+1} d\mathbf{x} \\ & = \int_{\Omega_{t^{n+1/2}}} \mathbf{f}_h \cdot \mathbf{v}_h r d\mathbf{x} + \int_{\Gamma_{t^{n+1/2}}^N} \boldsymbol{\sigma}_h \cdot \mathbf{v}_h r(s) ds, \\ & - \int_{\Omega_{t^{n+1/2}}} \operatorname{div} \mathbf{u}_h^{n+1} q_h r d\mathbf{x} - \int_{\Omega_{t^{n+1/2}}} u_{hr}^{n+1} q_h d\mathbf{x} = 0, \end{aligned} \right. \quad (2.13)$$

where $\Omega_{t^{n+1/2}}$ is the middle configuration between times t^n and t^{n+1} and \mathbf{u}_h^* may be chosen as \mathbf{u}_h^{n+1} for a fully implicit scheme or as \mathbf{u}_h^n for an explicit linearization. The latter choice

leads to a truncation error of the same order as the implicit Euler scheme, yet the former would produce a non-linear system of equations. Both choices leads to the following stability results. This is a direct consequence of the stability of the same scheme in the Cartesian three-dimensional case (see [Nob01], lemma 3.5.1).

Lemma 2.2.1 *Scheme (2.13) applied to a fully homogeneous Dirichlet problem is unconditionally stable and the discrete solution satisfies*

$$\begin{aligned} & \| \mathbf{u}_h^{n+1} \|_{L_1^2(\Omega_{t_{n+1}})}^2 + \Delta t \kappa \sum_{i=0}^n | \mathbf{u}_h^{n+1} |_{V_1^1(\Omega_{t_{n+1}}) \times H_1^1(\Omega_{t_{n+1}})}^2 \\ & \leq \| \mathbf{u}_h^0 \|_{L_1^2(\Omega_0)}^2 + \Delta t \frac{2(1 + C_P^2)}{\kappa} \sum_{i=0}^n \| \mathbf{f}_{h,t^{i+1/2}} \|_{(V_1^1(\Omega_{t^{i+1/2}}) \times H_1^1(\Omega_{t^{i+1/2}}))'}^2. \end{aligned}$$

2.3 Algebraic aspects

In this section we build the algebraic system associated to problem P2.11 and propose some solution techniques based on suitable splitting methods.

2.3.1 Algebraic formulation

In the following, we will denote as N the generic dimension of a finite element space (in our application, N will take the value of $N_{\mathcal{A}}$ when dealing with the ALE problem P2.8 and N_v and N_p when dealing with the velocity and the pressure in problem P2.4). We order the finite elements basis such that the last N^d are the functions relative to the Dirichlet nodes. Then $N^f = N - N^d$ denotes the number of degrees of freedom. We will also split a vector or a matrix into their blocks corresponding to nodes on a Dirichlet boundary (subscript d) or not (subscript f). Hence for a matrix A and a vector \mathbf{x} we may write

$$A\mathbf{x} = \begin{pmatrix} A^{ff} & A^{fd} \\ A^{df} & A^{dd} \end{pmatrix} \begin{pmatrix} \mathbf{x}_f \\ \mathbf{x}_d \end{pmatrix} = (A^{ff}\mathbf{x}_f + A^{fd}\mathbf{x}_d).$$

ALE mapping

Let $\{\hat{\varphi}_i\}_{i=1}^{N_{\mathcal{A}}}$ be the Lagrange basis associated to the space $Y_h(\hat{\Omega})$ (two basis-vectors for each node). To find the discrete ALE mapping which solves problem P2.9, we need to compute once and for all a matrix $K_{\mathcal{A}}$ with components

$$(K_{\mathcal{A}})_{ij} = \int_{\hat{\Omega}} \nabla \varphi_i : \nabla \varphi_j d\mathbf{x}, \quad 1 \leq i, j \leq N_{\mathcal{A}}$$

Let \mathbf{g}^n be a vector in $\mathbb{R}^{N_{\mathcal{A}}^d}$, such that

$$\mathbf{g}_h(\cdot, t^n) = \sum_{j=1}^{N_{\mathcal{A}}^d} g_j^n \varphi_{j+N_{\mathcal{A}}^f} \Big|_{\partial \hat{\Omega}}$$

and \mathbf{x}^n in $\mathbb{R}^{N_{\mathcal{A}}}$ such that

$$\mathbf{x}_d^n = \mathbf{g}_h^n \text{ and } A_{\mathcal{A}}^{ff} \mathbf{x}_f^n = -A_{\mathcal{A}}^{fd} \mathbf{g}_h^n.$$

2.3. ALGEBRAIC ASPECTS

Then the discrete ALE mapping at time t^n is given by

$$\mathcal{A}_{t^n} = \sum_{j=1}^{N_A} x_j^n \varphi_j.$$

Note that this system can be split into two independent systems of the same size, each expressing one component of the ALE mapping.

Navier–Stokes equations

Let $\{\psi_i\}_{i=1}^{N_p}$ be the Lagrange basis associated to the space $Q_h(\Omega_t)$ and $\{\varphi_i\}_{i=1}^{N_v}$ the one associated to $V_{h/2}(\Omega_t)$, such that $\{\varphi_i\}_{i=1}^{N_v}$ is a basis of $V_{\Gamma^D, h/2}(\Omega_t)$. Recall that the radial component of any vector-field in $V_{h/2}(\Omega_t)$ vanishes on the axis. For simplicity we omit the dependence on t and we recall that, since we discretize the ALE mapping by piecewise affine vector-fields with triangulation \mathcal{T}_h on $\hat{\Omega}$, the elements of both basis are piecewise affine on $\mathcal{A}_t(\mathcal{T}_h)$ and $\mathcal{A}_t(\mathcal{T}_{h/2})$ respectively. In fact, if for example ψ_i^0 is a basis element of $Q_h(\hat{\Omega})$, we could write $\psi_i^0 \circ \mathcal{A}_t^{-1}$ instead of ψ_i .

We introduce some matrices and vectors with components, for $1 \leq i, j \leq N_v$ and $1 \leq \ell \leq N_p$, (here \mathbf{u}_h^* is supposed to be known from previous iterations)

$$\begin{aligned} M_{ij}(t) &= \int_{\Omega_t} \varphi_i \cdot \varphi_j r d\mathbf{x}, \\ B_{ij}(t; \mathbf{w}_h, \mathbf{u}_h^*) &= \int_{\Omega_t} [(\mathbf{u}_h^* - \mathbf{w}_h) \cdot \nabla] \varphi_i \cdot \varphi_j r d\mathbf{x} \\ &\quad + \int_{\Omega_t} \operatorname{div} \left(\frac{1}{2} \mathbf{u}_h^* - \mathbf{w}_h \right) \varphi_i \cdot \varphi_j r d\mathbf{x} + \int_{\Omega_t} \left(\frac{1}{2} u_{hr}^* - w_{hr} \right) \varphi_i \cdot \varphi_j d\mathbf{x}, \\ K_{ij}(t) &= 2\nu \int_{\Omega_t} \epsilon(\varphi_i) : \epsilon(\varphi_j) r d\mathbf{x} + 2\nu \int_{\Omega_t} \varphi_{ir} \varphi_{jr} \frac{1}{r} d\mathbf{x}, \\ D_{\ell j}(t) &= - \int_{\Omega_{t^{n+1/2}}} \operatorname{div} \varphi_j \psi_\ell r d\mathbf{x} - \int_{\Omega_{t^{n+1/2}}} \varphi_{jr} \psi_\ell d\mathbf{x}, \\ \mathbf{F}_i(t) &= \int_{\Omega_t} \mathbf{f}_h \cdot \varphi_i r d\mathbf{x} + \int_{\Gamma_t^N} \boldsymbol{\sigma}_h \cdot \varphi_i r(s) ds, \end{aligned}$$

Let $\boldsymbol{\phi}^n$ be a vector in $\mathbb{R}^{N_v^d}$, such that

$$\boldsymbol{\phi}_{h, t^n} = \sum_{j=1}^{N_v^d} \phi_j^n \varphi_{j+N_v^f} \Big|_{\Gamma_{t^n}^D}.$$

Then problem P2.11 is equivalent to

P2.12 For any $n = 0, \dots, \frac{T}{\Delta t} - 1$, find $(\mathbf{U}^{n+1}, \mathbf{P}^{n+1})$ in $\mathbb{R}^{N_v} \times \mathbb{R}^{N_p}$ such that $\mathbf{U}_d^{n+1} = \boldsymbol{\phi}^{n+1}$ and

$$\begin{cases} \frac{1}{\Delta t} M^{\mathbf{f}\mathbf{f}}(t^{n+1}) \mathbf{U}_f^{n+1} + B^{\mathbf{f}\mathbf{f}}(t^{n+1}; \mathbf{w}_h^{n+1}; \mathbf{u}_h^*) \mathbf{U}_f^{n+1} + K^{\mathbf{f}\mathbf{f}}(t^{n+1}) \mathbf{U}_f^{n+1} \\ \quad + (D^{\mathbf{f}}(t^{n+1}))^T \mathbf{P}^{n+1} = \mathbf{b}_1(t^{n+1}), \\ D^{\mathbf{f}}(t^{n+1}) \mathbf{U}_f^{n+1} = \mathbf{b}_2(t^{n+1}), \end{cases}$$

where \mathbf{b}_1 and \mathbf{b}_2 account for the volumic forces and the non-homogeneous Dirichlet boundary condition:

$$\begin{aligned} \mathbf{b}_1(t^{n+1}) &= \frac{1}{\Delta t} M^{\mathbf{f}\mathbf{f}}(t^n) \mathbf{U}_{\mathbf{f}}^n + \mathbf{F}(t^{n+1}) - \frac{1}{\Delta t} \left(M^{\mathbf{f}\mathbf{d}}(t^{n+1}) \mathbf{U}_{\mathbf{d}}^{n+1} - M^{\mathbf{f}\mathbf{d}}(t^n) \mathbf{U}_{\mathbf{d}}^n \right) \\ &\quad - B^{\mathbf{f}\mathbf{d}}(t^{n+1}; \mathbf{w}_h^{n+1}; \mathbf{u}_h^*) \mathbf{U}_{\mathbf{d}}^{n+1} - K^{\mathbf{f}\mathbf{d}}(t^{n+1}) \mathbf{U}_{\mathbf{d}}^{n+1}, \\ \mathbf{b}_2(t^{n+1}) &= -D^{\mathbf{d}}(t^{n+1}) \mathbf{U}_{\mathbf{d}}^{n+1}. \end{aligned}$$

At each time step, the pressure and the velocity are given by

$$\mathbf{u}_h^{n+1} = \sum_{j=1}^{N_v} \mathbf{U}_j^{n+1} \varphi_j \quad \text{and} \quad p_h^{n+1} = \sum_{\ell=1}^{N_p} P_{\ell}^{n+1} \psi_{\ell}.$$

2.3.2 Quadrature formula

When the weight in the integral is either 1 or r , the integrands are polynomials and the matrix computation can be done with standard quadrature formulas of appropriate degree.

In contrast, the second integrand in K is a rational function and must be treated conveniently. First of all, since there are edges on the axis, the quadrature points must be internal, otherwise the quadrature is not defined for triangles with a vertex or a side on the axis.

An exact formula may be found but it is not numerically stable: In fact one could check that if, for example, T is a triangle (and $|T|$ its area) with vertices' radial coordinates r_1 , r_2 and r_3 all strictly positive, and λ the barycentric coordinate of the first vertex, then

$$\int_T \lambda^2 \frac{1}{r} d\mathbf{x} = \frac{|T|}{r_3} \int_0^1 \int_0^{1-\xi} \xi^2 \left(\frac{\xi r_1 + \eta r_2}{r_3} + 1 - \eta - \xi \right)^{-1} d\eta d\xi.$$

If $r_1 \neq r_2$, then this integral can be evaluated analytically, giving

$$\begin{aligned} &\frac{|T|}{6r_3(r_2-1)(r_1-r_2)^3(r_1-1)^3} \left(-r_1^5 + 4r_1^4 + 6\ln(r_1)r_1^2r_2 - 6\ln(r_1)r_1r_2^2 - 6\ln(r_2)r_2^3r_1 \right. \\ &\quad + 6\ln(r_2)r_2^3r_1^2 - 2\ln(r_2)r_2^3r_1^3 + 6\ln(r_1)r_1r_2^3 - 6\ln(r_1)r_1^2r_2^3 + 2\ln(r_1)r_1^3r_2^3 - 3r_1^3 - 2\ln(r_1)r_1^3 \\ &\quad \left. + 8r_1^2r_2 - 5r_1r_2^2 + 2\ln(r_2)r_2^3 + 3r_2^3r_1^3 + r_1^5r_2 - 4r_1^4r_2^2 + 5r_1r_2^3 - 8r_2^3r_1^2 + 9r_1^3r_2^2 - 9r_2r_1^3 \right), \end{aligned} \quad (2.14)$$

however if $r_1 = r_2$, then the integral turns out to be

$$-\frac{|T|}{18} \frac{9r_1^2 - 2r_1^3 - 18r_1 + 6\ln(r_1) + 11}{(r_1 - 1)^4}. \quad (2.15)$$

In fact the expression (2.14) converges to (2.15) when $r_2 \rightarrow r_1$, but may lead to divergence in floating point arithmetic, depending on the precision used in the computations (see figures 2.2 and 2.3). This means that if a triangle has an almost horizontal edge, formula (2.14) introduces non-negligible error.

In fact we have tested this approach with a structured mesh. While moving the mesh, a horizontal edge became not horizontal anymore because of rounding errors. Then the solution blows up just on the two nodes on that edge.

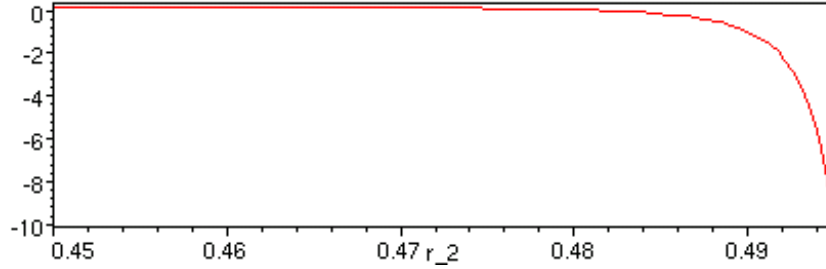


Figure 2.2: Function (2.14) evaluated with 10 precision's digits and with $r_1 = 0.5$, $r_3 = 0.55$ and $r_2 \in [0.45, 0.495]$. The function should converge to approximately 0.1633.

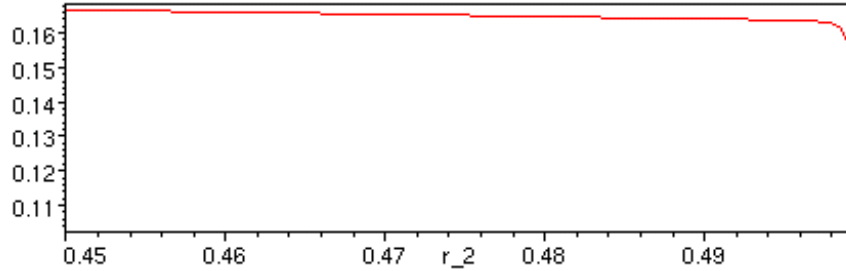


Figure 2.3: Function (2.14) evaluated with 15 precision's digits and with $r_1 = 0.5$, $r_3 = 0.55$ and $r_2 \in [0.45, 0.4995]$. The function should converge to approximately 0.1633.

Hence it is preferable to use a quadrature formula with internal nodes. For example one can use for all the integrals a direct Hammer quadrature formula of order three (cf. [ZT89]), which on a reference triangle $\{(\xi, \eta), 0 \leq \xi \leq 1, 0 \leq \eta \leq 1 - \xi\}$ reads

$$\int_0^1 \int_0^{1-\xi} f(\xi, \eta) d\eta d\xi \simeq \sum_{i=1}^4 w_i f(\xi_i, \eta_i),$$

where the weights are $w_1 = -27/96$ and $w_i = 25/96$ for $i = 2, 3, 4$ and the quadrature points are respectively $(1/3, 1/3)$, $(1/5, 1/5)$, $(3/5, 1/5)$, $(1/5, 3/5)$. Then the integral with weight $\frac{1}{r}$ of the product of two affine functions, is approximated by

$$\int_K f g \frac{1}{r} d\mathbf{x} = \frac{2|T|}{96} \left(-9 \frac{f_\sigma g_\sigma}{h_\sigma} + 5 \sum_{j=1}^3 \frac{(2f_j + f_\sigma)(2g_j + g_\sigma)}{2h_j + h_\sigma} \right),$$

where f_j , $j = 1, 2, 3$ is the value of the affine function f on the vertices of T and $f_\sigma = \sum_{j=1}^3 f_j$ (and similarly for g and r).

2.3.3 Computational aspects

Residual's norms

From classic *a posteriori* error estimates (see [ALT99] or [ALT01]), we expect that the L^2 -norm of the residual of the Stokes equations in the case of P1isoP2/P1 finite elements should have order h . Work in this direction in the axisymmetric case is under consideration.

A posteriori error estimates motivate a precision of the order h in the resolution of the linear system in P2.12. If this is carried out with an iterative method, then we have to choose a norm and a tolerance to decide when to stop the iterations.

The choice of the norm (as the L^2 -norm or L_1^2) is imposed by the a-posteriori analysis. However, for computational reasons, it is sometimes necessary to use discrete norms, such as the maximum norm or the vector-Euclidean norm $\|\mathbf{U}\|_2 = \sqrt{\mathbf{U}^T \mathbf{U}}$. This last choice is useful in the Cartesian case of uniformly regular family of triangulations, since in that case we have the uniform equivalence

$$h\|\mathbf{U}\|_2 \sim \left\| \sum_{j=1}^{N_v} \mathbf{U}_j \varphi_j \right\|_{L^2(\Omega)}. \quad (2.16)$$

In this case, the tolerance to choose in the iterative method is independent from h . Note that in the case of a left preconditioned BiCGStab, the norm is implicitly changed and the tolerance must be adapted to the dependence on h of the norm (see[VdV03, §13.1]).

Another possibility in the Cartesian case is to use the unweighted mass matrix, such that the equivalence in (2.16) is dependent neither on h nor on the uniformity of the triangulation. In fact, if $\|\mathbf{U}\|_M = \sqrt{\mathbf{U}^T \mathbf{M} \mathbf{U}}$, then

$$\|\mathbf{U}\|_M \sim \left\| \sum \mathbf{U}_j \varphi_j \right\|_{L^2(\Omega)}$$

and in this case to recover the convergence rate with respect to h , the tolerance in the iterative method must be proportional to h .

Sometimes, for programming purpose, it is preferable to use the maximum discrete norm $\|\cdot\|_\infty$. Then for all \mathbf{U} in \mathbb{R}^{N_v} ,

$$\|\mathbf{U}\|_2 < c \frac{1}{h} \|\mathbf{U}\|_\infty \text{ and } \|\mathbf{U}\|_M < c \|\mathbf{U}\|_\infty,$$

where c is constant independent of h and Δt , and in particular

$$\left\| \sum \mathbf{U}_j \varphi_j \right\|_{L^2(\Omega)} < c \|\mathbf{U}\|_\infty.$$

Again, the tolerance must be proportional to h .

In the axisymmetric case the equivalence constants of the discrete Euclidean and the L_1^2 norms depends also on the radial coordinates, it is therefore inappropriate to use the former in the iterative method. In contrast the norm $\|\cdot\|_M$, with M the weighted mass matrix, is uniformly equivalent to the L_1^2 norm,

$$\|\mathbf{U}\|_M \sim \left\| \sum \mathbf{U}_j \varphi_j \right\|_{L_1^2(\Omega)},$$

which implies that the tolerance must be proportional to h to be consistent with the convergence rate.

Similarly to the Cartesian case, it is possible to use the maximum norm with the remark that also in the axisymmetric case

$$\left\| \sum \mathbf{U}_j \varphi_j \right\|_{L_1^2(\Omega)} < c \|\mathbf{U}\|_\infty$$

and again we choose the tolerance proportionally to h .

In conclusion, it is in general convenient to use one of these two norms to evaluate the residual. An exception is when the mass matrix is used as left preconditioner of an iterative method such as the conjugate gradient. In this case we suggest to use the Euclidean norm and a tolerance proportional to h , since the Euclidean norm $\|\cdot\|_2$ of the preconditioned problem is equal to M -norm of the actual residual.

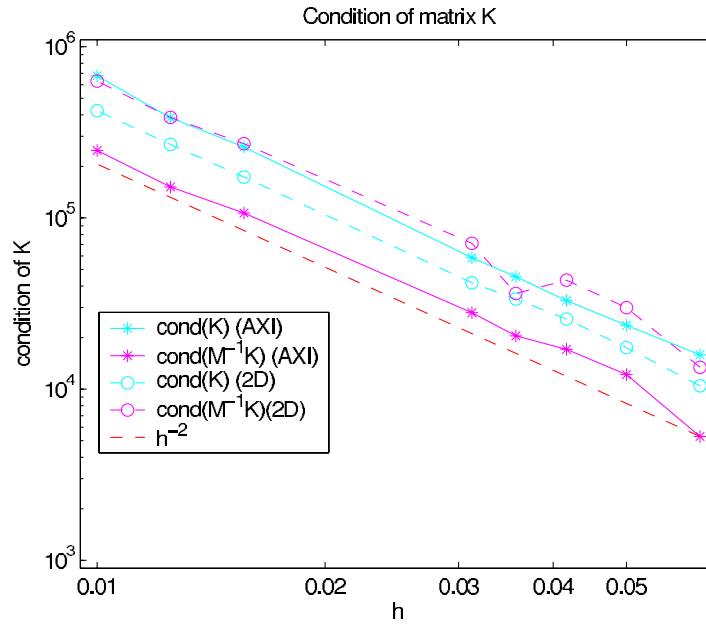


Figure 2.4: Condition number of the stiffness matrix in the case of a 1cm square. Cartesian and axisymmetric formulations compared with the preconditioned cases.

Mass lumping

The lumping of the mass, i.e., the reduction of the mass matrix to a diagonal matrix, is particularly interesting in the computation of the residuals M -norm and in the two factorization schemes of the following section.

The lumping can be performed by summing up each line and replacing the diagonal element by this sum. In the Cartesian case with P1isoP2/P1 elements, this is equivalent to computing the mass matrix with a trapezoidal quadrature formula (see [QV94, §11.4] and [Han94]). In the axisymmetric case this equivalence is no longer true. In fact the trapezoidal quadrature formula would lead to a singular mass matrix. Anyway, in the literature the sum of the lines is proposed as an effective choice (see [Zie67, §11]).

Spectral properties of the stiffness matrix

Here we would like to present a numerical approximation of the condition number of the stiffness matrix and to see the effect of the lumped mass as preconditioner.

In figure 2.4 we report an example of the condition number of K in a square in the Cartesian and in the axisymmetric case and also when the respective lumped mass matrices are used as preconditioners. The condition number is roughly equivalent in the four cases.

2.3.4 Inexact factorization schemes

Here we consider two factorization schemes for the solution of problem P2.12 which are widely used in the solution of the incompressible Navier–Stokes equations (see for example [Qua93], [Ven98], [QSV00] and [QSV99]).

Let \mathbf{U}^* and \mathbf{P}^* be extrapolations of the velocity and pressure vectors at time t^{n+1} . For

example they can be taken equal to \mathbf{U}^n and \mathbf{P}^n or else to $2\mathbf{U}^n - \mathbf{U}^{n-1}$ and $2\mathbf{P}^n - \mathbf{P}^{n-1}$. In this section we neglect the appendices \mathbf{f} and $\mathbf{f}\mathbf{f}$ and the time dependence for sake of simplicity. For a vector $\mathbf{v} = \sum V_j \boldsymbol{\varphi}_j$, we write $B(\mathbf{V})$ instead of $B(\mathbf{w}_h^{n+1}; \mathbf{v})$. Let

$$\begin{aligned} A &= \frac{1}{\Delta t} M + B(\mathbf{U}) + K, \\ A^* &= \frac{1}{\Delta t} M + B(\mathbf{U}^*) + K. \end{aligned}$$

Finally, let H be an approximation of the inverse of A^* , for example equal to $\Delta t M^{-1}$.

Incremental Yosida

This scheme was firstly presented in [QSV99] and its incremental variant is

- 1) $A^* \tilde{\mathbf{U}} = \mathbf{b}_1 - D^T \mathbf{P}^*$,
- 2) $D H D^T \delta \mathbf{P} = \mathbf{b}_2 + D \tilde{\mathbf{U}}$,
- 3) $\mathbf{P}^{n+1} = \mathbf{P}^* + \delta \mathbf{P}$,
- 4) $A^* \mathbf{U}^{n+1} = \mathbf{b}_1 - D^T \mathbf{P}^{n+1}$.

In general, steps 1, 2 and 4 are solved by iterative methods, which are stopped whenever the residuals are smaller than given tolerances. Let $\boldsymbol{\epsilon}_i$ be the residual vector obtained at convergence for the step i , $i = 1, 2, 4$ and let $\epsilon_i = \|\boldsymbol{\epsilon}_i\|$, where now the norm is $\|\mathbf{U}\| = \|\sum \mathbf{U}_j \boldsymbol{\varphi}_j\|_{L_1^2(\Omega)}$.

The following remarks hold also for the Cartesian case.

The residual of problem P2.12 associated to the Yosida scheme can be computed as follows. The first equation has residual

$$\mathbf{R}_1 = A \mathbf{U}^{n+1} + D^T \mathbf{P}^{n+1} - \mathbf{b}_1 = (B(\mathbf{U}^{n+1}) - B(\mathbf{U}^*)) \mathbf{U}^{n+1} + \boldsymbol{\epsilon}_4,$$

then the residual's norm depends on the tolerance used to solve step 4 and on the linearization of the problem.

For a given \mathbf{V} , the mapping $\mathbf{U} \mapsto B(\mathbf{U})\mathbf{V}$ is continuous and linear, hence $\|(B(\mathbf{U}^{n+1}) - B(\mathbf{U}^*))\mathbf{V}\|$ is smaller than $\beta_1 \|\mathbf{U}^{n+1} - \mathbf{U}^*\|$ for a finite positive β_1 dependent on the geometry, on the velocity of the mesh and of the fluid, and possibly on the discretization parameter h . However, we suppose that β_1 is uniformly bounded independently from \mathbf{u} , Ω , h and n and we replace β_1 by its bound. In practice, β_1 is typically small in comparison with the norm of $A\mathbf{U}$. To summarize, setting $\delta \mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^*$, the residual's norm of the first equation is

$$\|\mathbf{R}_1\| \leq \beta_1 \|\delta \mathbf{U}\| + \epsilon_4. \quad (2.17)$$

The second equation of problem P2.12 has residual

$$\begin{aligned} \mathbf{R}_2 &= D \mathbf{U}^{n+1} - \mathbf{b}_2 = -\mathbf{b}_2 + D [A^{*-1}(\mathbf{b}_1 - D^T \mathbf{P}^*)] - D A^{*-1} D^T \delta \mathbf{P} + D A^{*-1} \boldsymbol{\epsilon}_4 \\ &= -\mathbf{b}_2 + D \tilde{\mathbf{U}} - D A^{*-1} D^T \delta \mathbf{P} + D A^{*-1} [\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_4] \\ &= \boldsymbol{\epsilon}_2 + D [H - A^{*-1}] D^T \delta \mathbf{P} + D A^{*-1} [\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_4]. \end{aligned} \quad (2.18)$$

2.3. ALGEBRAIC ASPECTS

This depends on the tolerance used in solving each step, but above all on the norm of A^{*-1} , on the accurate approximation of A^{*-1} and on the extrapolation of the pressure \mathbf{P}^* . We denote by β_2 the matrix norm of $D[H - A^{-1}]D^T$ and by β_3 the one of DA^{-1} . Then

$$\|\mathbf{R}_2\| \leq \epsilon_2 + \beta_2 \|\delta \mathbf{P}\| + \beta_3 (\epsilon_1 + \epsilon_4). \quad (2.19)$$

In order to save iterations in the solution of steps 1, 2 and 4, it is possible to use expression (2.18) to optimize the tolerances to be used. The following strategy may be adopted.

- (i) Extrapolate the magnitude of $\|\delta \mathbf{P}\|$ such that

$$m_{\delta \mathbf{P}} < \|\delta \mathbf{P}\| \quad (2.20)$$

and stop the iterative method which solves the first Yosida's step as soon as

$$\epsilon_1 = \left\| \mathbf{b}_1 - A^* \tilde{\mathbf{U}} - D^T \mathbf{P}^* \right\| < \frac{1}{10} \frac{\beta_2}{\beta_3} m_{\delta \mathbf{P}};$$

- (ii) Stop the iterative method which solves the second Yosida's step as soon as

$$\epsilon_2 = \left\| \mathbf{b}_2 - DHD^T \delta \mathbf{P} + D\tilde{\mathbf{U}} \right\| < \frac{1}{10} \beta_2 \|\delta \mathbf{P}\|;$$

- (iii) Define $\mathbf{P}^{n+1} = \mathbf{P}^* + \delta \mathbf{P}$;

- (iv) Stop the iterative method which solves the forth Yosida's step as soon as

$$\epsilon_4 = \left\| \mathbf{b}_1 - A^* \mathbf{U}^{n+1} - D^T \mathbf{P}^{n+1} \right\| < \min \left\{ \frac{\beta_2}{\beta_3} \|\delta \mathbf{P}\|, \frac{1}{10} \beta_1 \left\| \mathbf{U}^{n+1} - \mathbf{U}^* \right\| \right\}$$

As a consequence of estimates (2.17) and (2.19) and under the assumption that inequality (2.20) holds,

$$\begin{aligned} \|\mathbf{R}_1\| &\leq 1.1 \beta_1 \|\delta \mathbf{U}\|, \\ \|\mathbf{R}_2\| &\leq 1.3 \beta_2 \|\delta \mathbf{P}\|. \end{aligned}$$

If $D[H - A^{-1}]D^T$ is invertible, let β'_2 be the norm of its inverse. Then

$$\|\mathbf{R}_2\| \geq \frac{1}{\beta'_2} \|\delta \mathbf{P}\| - \epsilon_2 - \beta_3 (\epsilon_1 + \epsilon_4). \quad (2.21)$$

Hence, even if the linear problems in steps 1, 2 and 4 are solved with a very small tolerance, the residual of the Yosida scheme is dominated by the error in the extrapolation of the pressure.

Indeed the same remark holds for the extrapolation of the velocity. As a result, the extrapolation of the physical unknowns affects not only the convergence on Δt , but also the one on h . This means that in some situations, even if using a first order scheme for the time discretization, it may be necessary to do a higher order extrapolation. Another choice can be to bound the time and the space discretization parameters.

For example, with a first order extrapolation of the physical unknowns ($\mathbf{U}^* = \mathbf{U}^n$ and $\mathbf{P}^* = \mathbf{P}^n$), Δt must be smaller than ch . Then tolerances in the steps 1, 2 and 4 have to be proportional to Δt . The resulting scheme is of order Δt in time and h in space.

In contrast, with a second order extrapolation ($\mathbf{U}^* = 2\mathbf{U}^n - \mathbf{U}^{n-1}$ and $\mathbf{P}^* = 2\mathbf{P}^n - \mathbf{P}^{n-1}$), Δt must be smaller than $c\sqrt{h}$ and the tolerances in the Yosida scheme proportional to Δt^2 . Still, the resulting scheme is only of order Δt in time and h in space, since a (semi-)implicit Euler scheme is of order Δt . Hence this choice leads to a more accurate resolution and is useful to relax the dependence between h and ΔT .

In the case of a second order time scheme such as Crank-Nicolson it is necessary to use a second order extrapolation, since the Yosida scheme with a first order extrapolation would downgrade the accuracy of the scheme.

Incremental Chorin-Temam

The Yosida scheme allows to solve the momentum equation more accurately. Here we present a scheme which instead solves the continuity equation more accurately.

- 1) $A^*\tilde{\mathbf{U}} = \mathbf{b}_1 - D^T \mathbf{P}^*$,
- 2) $DHD^T \delta \mathbf{P} = \mathbf{b}_2 + D\tilde{\mathbf{U}}$,
- 3) $\mathbf{P}^{n+1} = \mathbf{P}^* + \delta \mathbf{P}$,
- 4) $\mathbf{U}^{n+1} = \tilde{\mathbf{U}} - HD^T \delta \mathbf{P}$.

As for the Yosida scheme we analyze the residual vectors ϵ_i , $i = 1, 2$ related with this scheme. The second equation of problem P2.12 has residual

$$D\mathbf{U}^{n+1} - \mathbf{b}_2 = D\tilde{\mathbf{U}} - DHD^T \delta \mathbf{P} = \epsilon_2, \quad (2.22)$$

i.e., the residual has the same order as the tolerance used to solve step 2. The first equation has residual

$$\begin{aligned} A\mathbf{U}^{n+1} + D^T \mathbf{P}^{n+1} - \mathbf{b}_1 &= A\tilde{\mathbf{U}} - AHD^T \delta \mathbf{P} + D^T \mathbf{P}^{n+1} - \mathbf{b}_1 \\ &= (A - A^*)\tilde{\mathbf{U}} + \epsilon_1 - D^T \mathbf{P}^* + D^T \mathbf{P}^{n+1} - D^T \delta \mathbf{P} + (Id - AH)D^T \delta \mathbf{P} \\ &= (B(\mathbf{U}^{n+1}) - B(\mathbf{U}^*))\tilde{\mathbf{U}} + \epsilon_1 + (Id - AH)D^T \delta \mathbf{P} \end{aligned} \quad (2.23)$$

and its norm is

$$\|A\mathbf{U}^{n+1} + D^T \mathbf{P}^{n+1} - \mathbf{b}_1\| \leq \beta_1 \|\delta \mathbf{U}\| + \epsilon_1 + \beta_4 \|\delta \mathbf{P}\|,$$

where β_4 is the matrix norm of $(Id - AH)D^T$. Again, this depends on the tolerance used in solving each step, but above all on the accurate approximation of A^{-1} and on the extrapolation of the velocity and of the pressure. This means that when using the Chorin-Temam scheme, it is reasonable to choose a tolerance for steps 1 such that ϵ_1 is smaller than an extrapolation of $\beta_1 \|\delta \mathbf{U}\| + \beta_4 \|\delta \mathbf{P}\|$.

2.4 Defective boundary conditions

In this section we present a technique to impose a mean flux on several disjoint sections $\check{S}_0, \check{S}_1, \dots, \check{S}_n$, $n \geq 1$ of the domain $\check{\Omega}_t$, derived from its Cartesian counterpart presented in [FGNQ00] and [Nob01, §5.2]. In axisymmetric blood flow simulations we are in general

2.4. DEFECTIVE BOUNDARY CONDITIONS

interested in one inflow or one outflow section. We neglect all the dependency on time to simplify the notations.

We formalize the problem of fluid equations with defective boundary conditions in the following way: We are interested in solving the axisymmetric Navier–Stokes equations (2.5) in the axisymmetric domain $\tilde{\Omega}$ whose boundary may be rigid or deformable. As usual we suppose zero angular component of the data.

The boundary of the half-section Ω is decomposed into Γ^D and $\Gamma^N = \bigcup S_i$. The Navier–Stokes equations are supplemented by Dirichlet boundary conditions

$$\mathbf{u} = \boldsymbol{\phi} \text{ on } \Gamma^D$$

and prescribed mean flux conditions on sections S_i , $i = 0, \dots, n$,

$$\int_{S_i} \mathbf{u} \cdot \mathbf{n} r(s) ds = Q_i, \quad i = 1, \dots, n, \quad \text{and} \quad \left(p\mathbf{n} - \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \Big|_{S_0} = 0, \quad (2.24)$$

where the Q_i 's are assigned functions of time and $r(s)$ is the value of the radial coordinate at the point with tangential coordinate s .

We could also have imposed the flux on every section S_i , $i = 0, \dots, n$. In this case, due to the incompressibility of the fluid, a compatibility relation must exist among the fluxes Q_i

$$\int_{\Gamma^D} \boldsymbol{\phi} \cdot \mathbf{n} r(s) ds + \sum_{i=0}^n Q_i = 0.$$

and the pressure is defined up to a constant. Note that imposed three-dimensional mean flows are equal to $2\pi Q_i$, $i = 1, \dots, n$.

The fulfillment of the flux conditions can be obtained through the use of Lagrange multipliers.

P2.13 Find (\mathbf{u}, p) in $V(\Omega) \times Q(\Omega)$, with $\mathbf{u}(0) = \mathbf{u}_0$ and $\mathbf{u} = \boldsymbol{\phi}$ on $\Gamma \times I$, and $\lambda_1, \dots, \lambda_n : t \mapsto \mathbb{R}^n$, such that for almost every t in I and for all (\mathbf{v}, q) in $V_{\Gamma^D}(\Omega_t) \times Q(\Omega_t)$,

$$\left\{ \begin{array}{l} \frac{d}{dt} \int_{\Omega_t} \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Omega_t} [(\mathbf{u} - \mathbf{w}) \cdot \nabla] \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Omega_t} \operatorname{div}(\mathbf{u} - \mathbf{w}) \mathbf{u} \cdot \mathbf{v} r d\mathbf{x} \\ \quad + \int_{\Omega_t} (u_r - w_r) \mathbf{u} \cdot \mathbf{v} d\mathbf{x} + 2\nu \int_{\Omega_t} \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) r d\mathbf{x} \\ \quad + 2\nu \int_{\Omega_t} u_r v_r \frac{1}{r} d\mathbf{x} - \int_{\Omega_t} \operatorname{div} \mathbf{v} p r d\mathbf{x} - \int_{\Omega_t} v_r p d\mathbf{x} \\ \quad + \sum_{i=1}^n \lambda_i \int_{S_i} \mathbf{v} \cdot \mathbf{n} r(s) ds = \int_{\Omega_t} \mathbf{f} \cdot \mathbf{v} r d\mathbf{x} + \int_{\Gamma_t^N} \boldsymbol{\sigma} \cdot \mathbf{v} r(s) ds, \\ - \int_{\Omega_t} \operatorname{div} \mathbf{u} q r d\mathbf{x} - \int_{\Omega_t} u_r q d\mathbf{x} = 0, \\ \int_{S_i} \mathbf{u} \cdot \mathbf{n} r(s) ds = Q_i. \end{array} \right. \quad (2.25)$$

Formaggia et al. show in [FGNQ00] that any smooth solution of problem P2.13 satisfies the additional boundary conditions

$$\left(p - \nu \frac{\partial u_n}{\partial \mathbf{n}} \right) \Big|_{S_i} = \lambda_i, \quad \text{and} \quad \frac{\partial \mathbf{u}_\tau}{\partial \mathbf{n}} \Big|_{S_i} = 0, \quad i = 1, \dots, n, \quad (2.26)$$

where $u_n = \mathbf{u} \cdot \mathbf{n}$ and $\mathbf{u}_\tau = \mathbf{u} - u_n \mathbf{n}$. In particular, this yields that both $\frac{\partial \mathbf{u}_\tau}{\partial \mathbf{n}}$ and $p - \nu \frac{\partial u_n}{\partial \mathbf{n}}$ are indeed constant over S_i for $i = 1, \dots, n$. Moreover, the Lagrange multipliers represent the normal component of the normal stress on each section S_i and have then the dimension of a pressure.

In the cited paper, it is shown that, for a stationary Stokes problem, problem P2.13) is well posed. Moreover different strategies are proposed to efficiently solve problem (2.25) discretized with finite elements. Here we recall the ones based on the Yosida algebraic factorization scheme with P1isoP2/P1 finite elements.

At a given time step, we note by Λ the set of the Lagrange multipliers $(\lambda_{1,h}, \dots, \lambda_{n,h})^T$, by Φ the $n \times N_v$ matrix with coefficients

$$\Phi_{i,j} = \int_{S_i} \varphi_j \cdot \mathbf{n} r(s) ds$$

and by $\mathbf{Q} = (Q_1, \dots, Q_n)^T$ the vector of imposed fluxes.

The discrete problem to solve at each time step is

$$\begin{bmatrix} A & D^T & \Phi^T \\ D & 0 & 0 \\ \Phi & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{Q} \end{bmatrix}$$

We can rewrite the block matrices as

$$\tilde{D} = \begin{bmatrix} D \\ \Phi \end{bmatrix}, \quad \tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P} \\ \Lambda \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{b}}_2 = \begin{bmatrix} \mathbf{b}_2 \\ \mathbf{Q} \end{bmatrix}.$$

Then the system of equations can be written in a similar form as in problem P2.12,

$$\begin{bmatrix} A & \tilde{D}^T \\ \tilde{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \tilde{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \tilde{\mathbf{b}}_2 \end{bmatrix}$$

and the Yosida or Chorin-Temam schemes applies in the same way as already described. The algorithm can be easily implemented starting from an existing Navier-Stokes solver which uses factorization methods. Indeed, it suffices to add to the matrix D the few lines of matrix Φ , and apply the chosen factorization method.

On the contrary, the constraints on the fluxes are not satisfied exactly. In fact the error $|\Phi \mathbf{U}^k - \mathbf{Q}^k|$ behaves as $O(\Delta t^2)$. This result has been confirmed numerically in the cited paper.

2.5 Some numerical results

We have tested the axisymmetric formulation of the Navier-Stokes equations in a moving domain with a Womersley flow (for Womersley flow see for example [Ven98]). The flow is defined on a cylinder of radius 1.2cm and we impose a Womersley number equal to 15. The fluid characteristics are $\mu = 0.035$ poise, $\rho = 1\text{g/cm}^3$. The pressure at the inlet is given by

$$P_{\text{in}} = 150 \cdot L \cdot \cos\left(\frac{2\pi}{0.8s}t\right),$$

where $L = 1$ is equal to the length of the tube, and we impose the pressure equal to zero at the outlet.

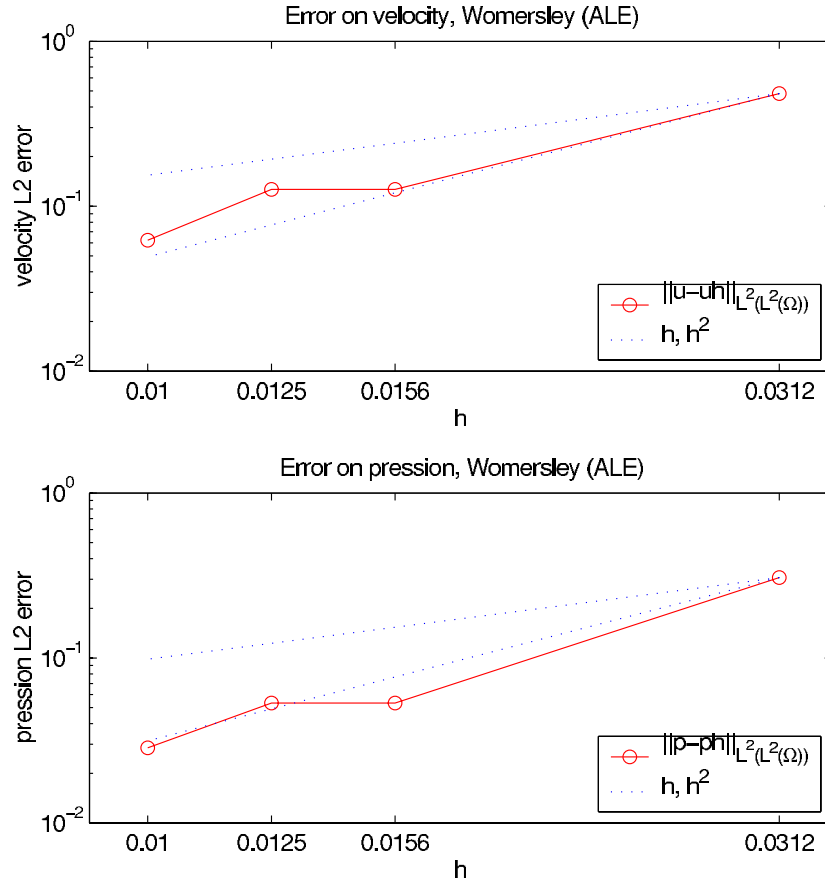


Figure 2.5: Convergence rate w.r.t. h in $L^2((0, T), L_1^2(\Omega))$ -norm for the velocity and pressure for a moving domain immersed in a Womersley flow.

The computational domain is a cylinder which radius is a prescribed function of time:

$$r(t) = 1 + 0.16 \sin\left(\frac{2\pi t}{10s}\right)$$

and the time horizon T is equal to 1.2s.

Indeed, we do not impose the pressure but the $\sigma \mathbf{n}$. For a time step equal to 10^{-4} s we have the convergence rate w.r.t. h shown in figure 2.5 and w.r.t. Δt for a fixed $h = 0.01$ in figure 2.6. Note that for simplicity reasons we computed the velocity error in the L_1^2 -norm instead of the H_1^1 -norm. This explains the super-convergence shown in figure 2.5 of the velocity field. The super-convergence of the pressure is due to the fact that the exact pressure is linear and it belongs to the finite element space.

We also tested Dirichlet boundary conditions on the inlet, imposed mean fluxes on the inlet as well as on both the inlet and outlet with the same results on the convergence rate. In last test, we have rescaled the pressure at each time step by the mean value of the pressure at the outlet.

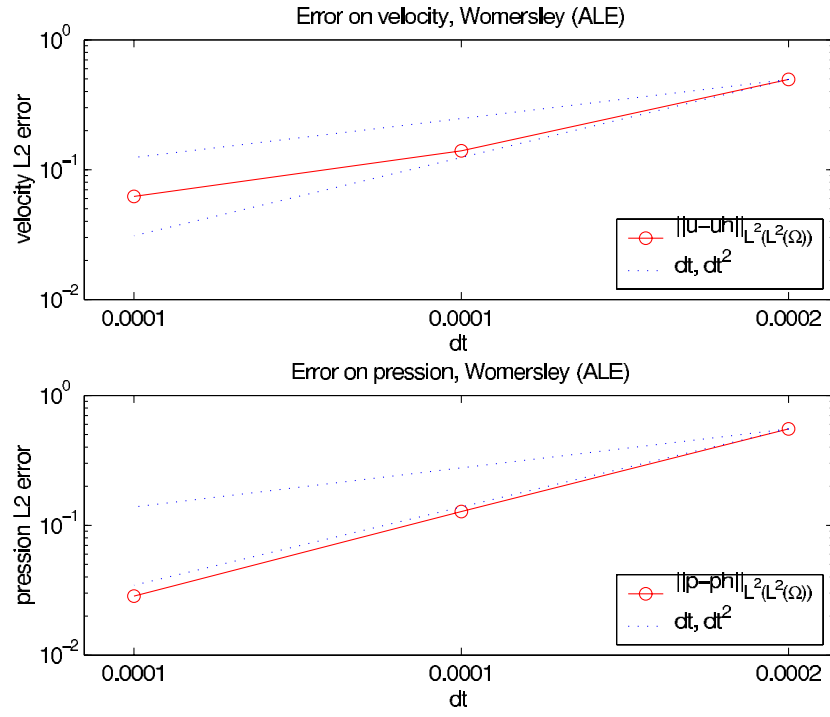


Figure 2.6: Convergence rate w.r.t. Δt in $L^2((0, T), L_1^2(\Omega))$ norm for the velocity and pressure for a moving domain immersed in Womerlsey flow.

Part II

Fluid-Structure Interaction

Chapter 3

A fluid-structure interaction problem

Introduction

Large displacement low speed problems (in which a flexible elastic structure interacts with the flow of an external or internal fluid) occur in many engineering fields, from civil engineering (aeroelasticity) to bio-mechanics (biomedical flows). One challenge arising in the numerical approximation of these fluid-structure problems is the definition of fast and accurate coupling algorithms that allow to predict the long-term time evolution maintaining the stability of the overall system. This issue is particularly difficult to face when the fluid and the solid densities are of the same order, for example in haemodynamics, since only implicit schemes can ensure stability of the resulting method (see [LM01, Nob01, GV03]). Thus, at each time step, the rule is to solve a highly coupled non-linear system (the fluid domain depends on the structural motion) using efficient methods that preserve, inside inner loops, the fluid-structure subsystem splitting. Standard strategies to solve this non-linear system are fixed-point based methods as Block-Jacobi or Block-Gauss-Seidel (BGS) iterations, see [CC96, LM01, Nob01, MWR01]. More recent approaches make use of Block Newton methods [MS00, FM03, FM04, GV03, Hei03] on the non-linear coupled problem.

In this chapter we introduce a fluid-structure interaction problem in its coupled form and we set the basis for the development of algorithms to accelerate the convergence of BGS or Newton algorithms. The solution algorithms presented here and in the following chapters do not depend on the dimension of the original problem. In fact, they can be applied arbitrarily to two-dimensional, three-dimensional or three-dimensional axisymmetric Navier–Stokes equations for the fluid and to shell or string models for the structure. For the sake of clarity, we will introduce the problem in a general framework, so that the reader can easily adapt the methods to specific situations.

3.1 Formulation of the fluid-structure problem

3.1.1 The governing continuum equation

In order to address each problem in its natural setting, we choose to consider the fluid in an ALE formulation, already presented in the previous chapter, and the structure in a pure

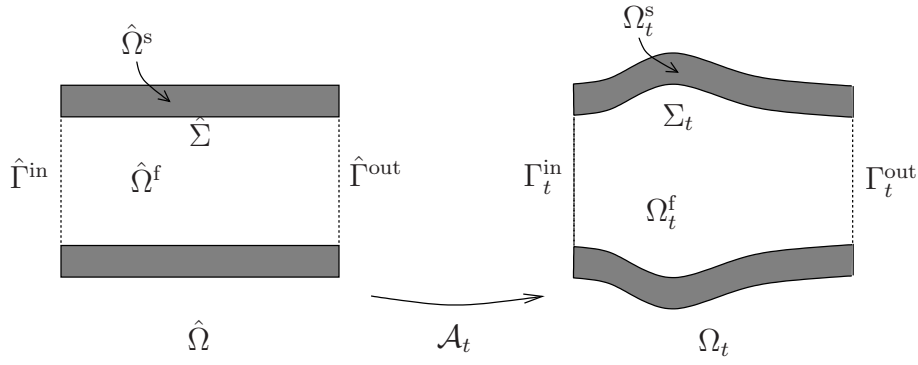


Figure 3.1: ALE mapping between the initial configuration and the configuration at time t .

Lagrangian framework.

The system under study occupies a moving domain Ω_t in its actual configuration. It is made of a deformable structure Ω_t^s (vessel wall, pipe-line, ...) surrounding a fluid under motion (blood, oil, ...) in the complement Ω_t^f of Ω_t^s in Ω_t (see figure 3.1). The problem consists in finding the time evolution of the configuration Ω_t^f , as well as the velocity and Cauchy stress tensor for both the fluid and the structure.

We assume the fluid to be Newtonian, viscous, homogeneous and incompressible. Its behavior is described by its velocity and pressure. The elastic solid under large displacements is described by its velocity and its stress tensor. The classical conservation laws of the continuum mechanics drive the evolution of these unknowns.

We denote by Γ_t^{in} and Γ_t^{out} the inflow and outflow sections of the fluid domain, by \mathbf{n} the fluid domain's outward normal and by \mathbf{n}_s the one of the structure. In particular on the fluid-structure interface Σ_t , $\mathbf{n} = -\mathbf{n}_s$. The boundary conditions to impose on the fluid inlet and outlet can be of Dirichlet type, Neumann or defective as described in section 2.4, while on the interface we impose that the fluid and structure velocities match as well as normal stresses. For simplicity, we assume zero body forces on both the structure and the fluid and that the conditions on the remaining structure boundary are homogeneous of Dirichlet or of Neumann type.

We denote by \mathbf{u} and p the velocity and pressure of the fluid and by \mathbf{d} the displacement of the structure and define a mapping

$$\forall t \in I, \mathcal{A}_t : \hat{\Omega} \rightarrow \Omega_t,$$

such that its restriction to the fluid reference domain is the ALE mapping defined in chapter 2 and its restriction to the structure reference domain is the Lagrangian mapping related to the structure displacement. Indeed, we will only need the restriction to the closure of the fluid reference domain and we will denote it also by \mathcal{A}_t .

We recall that $\hat{\mathbf{x}}$ denotes the coordinates on the reference configuration $\hat{\Omega}$ and $\mathbf{w} = \frac{d\mathcal{A}_t}{dt}$ the domain velocity.

For example, if at the inlet and outlet we impose the stress tensor, the fluid problem reads

3.1. FORMULATION OF THE FLUID-STRUCTURE PROBLEM

P3.1 (Fluid problem)

$$\begin{cases} \rho_f \left(\frac{\partial \mathbf{u}}{\partial t} \Big|_{\hat{\mathbf{x}}} + (\mathbf{u} - \mathbf{w}) \nabla \mathbf{u} \right) = \operatorname{div}(2\mu \boldsymbol{\epsilon}(\mathbf{u})) - \nabla p & \text{on } \Omega_t^f, \\ \operatorname{div} \mathbf{u} = 0 & \text{on } \Omega_t^f, \\ \mathbf{u}(\mathbf{x}, t) = \frac{\partial \mathbf{d}}{\partial t} (\mathcal{A}_t^{-1}(\mathbf{x}), t) & \text{on } \Sigma_t, \\ \boldsymbol{\sigma}_f \cdot \mathbf{n} = \mathbf{g} & \text{on } \Gamma_t^{\text{in}} \cup \Gamma_t^{\text{out}}, \end{cases}$$

where ρ_f is the fluid density, μ its viscosity, $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)}{2}$ is the strain rate tensor and $\boldsymbol{\sigma}_f = -pId + 2\mu \boldsymbol{\epsilon}(\mathbf{u})$ the Cauchy stress tensor.

The equation of the structure can be written as

P3.2 (Structure Problem)

$$\begin{cases} \rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} = \hat{\operatorname{div}}(\hat{\mathbf{T}}) & \text{on } \hat{\Omega}^s, \\ \hat{\mathbf{T}} \cdot \hat{\mathbf{n}}_s = \det \left(\frac{\partial \mathcal{A}_t}{\partial \hat{\mathbf{x}}} \right) \boldsymbol{\sigma}_f \cdot \frac{\partial \mathcal{A}_t}{\partial \hat{\mathbf{x}}} \hat{\mathbf{n}}_s & \text{on } \hat{\Sigma}, \\ \mathbf{d} = 0 \text{ or } \hat{\mathbf{T}} \cdot \hat{\mathbf{n}}_s = 0 & \text{on } \partial \hat{\Omega}^s \setminus \hat{\Sigma}, \end{cases}$$

where $\hat{\mathbf{T}}$ is the first Piola–Kirchhoff stress tensor. It is possible to choose appropriate models for the structure depending on the simulation. The reader may refer to [LT94, QTV00, LL75, LMP91, CB03, ALDR03, Ani02, AL99] for other models.

A generalized string model

When we are dealing with an axisymmetric model for the fluid, we often use a generalized string model presented in [QTV00], which is a simplified wall model. This model is very simple and is based on the hypotheses, among others, of relative small displacements, which is not the case in haemodynamics. Even then, we will use it in testing the interaction algorithms, since it is easily implemented and provides the relevant properties in the mechanical coupling. It should be clear to the reader that, for a satisfactory blood-flow simulation, one have to employ more sophisticated structure models.

The generalized string model derives from a membrane model, which is in general applicable for the arterial wall when bending is of minor importance. Starting from the general equations for large deformations of a cylindrical nonlinear membranes (see, e.g., [MLL92]), the following assumptions have been introduced:

- We allow only axisymmetric radial displacement of the wall;
- The radial deformation of the wall is small ($|d_r/R_o| \ll 1$);
- The long wave assumption holds ($|\partial d_r / \partial z| \ll 1$);
- No-preloading is applied prior to the small deformations.

The equation of the generalized string model on d_r are

$$\rho_s h \frac{\partial^2 d_r}{\partial t^2} - k G h \frac{\partial^2 d_r}{\partial z^2} + \frac{E h}{1 - \nu^2} \frac{d_r}{R_0^2} - \gamma \frac{\partial^3 d_r}{\partial z^2 \partial t} = \sigma_\Sigma,$$

where h is the wall thickness, k is the so-called Timoschenko shear correction factor, G the shear modulus, E the Young modulus, ν the Poisson ratio, ρ_s the wall density and γ a viscoelastic parameter. R_0 is a reference radius independent from the axial coordinate z and σ_Σ is the radial component of the stress vector of the fluid acting on the structure. The term $kGh\frac{\partial^2 d_r}{\partial z^2}$ accounts for the shear deformation but can also be considered as an axial preloading. The term $\gamma\frac{\partial^3 d_r}{\partial z^2 \partial t}$ introduces a viscoelastic behavior and its presence is motivated largely by numerical convergence issues.

The small deformation assumption enters the formula in two ways. One is in the linearization of the governing equations for the radial displacement of the membrane, independent of the particular constitutive equation used. The second is the use of a linear elastic model for the wall material. Once the small deformation approximation is used for the radial deformation, it is consistent to use the linearized elasticity approximation for the wall behavior.

We allow only axisymmetric radial displacement such that the assumption of axisymmetric domain for the fluid is not broken. Note that without the long wave assumption, the wall curvature would enter the equations in the form of $\sqrt{1 + (\partial d_r / \partial z)^2}$.

A discussion on the energy balance of this model coupled with a three-dimensional fluid may be found in [Nob01, §4] and the results therein are applicable to the axisymmetric case.

3.1.2 Weak formulation

A global weak formulation of the coupled problem has been proposed by Le Tallec and Mouro in [LM01]. As far as we know only a few partial results on the existence and uniqueness of a solution are available. In [Nob01] or [Gra98] a sub-problem decomposition is presented. In the latter reference, existence and uniqueness of weak solutions are proved for the stationary problem consisting of the coupling between Stokes equations and a fourth order structural equation under the hypothesis of small data. Other available results concern the motion in a fluid of rigid bodies or deformable ones, whose deformations are described by a linear combination of a finite number of elastic eigenmodes. Existence of weak solutions has been proved in [DE99], [DE00] and [DEGL01] for all $T > 0$ provided there isn't any collision between the bodies.

A proof on the existence of a strong solution of the coupling of a two-dimensional Navier–Stokes model for the fluid and a string model for structure with periodic boundary condition is given in [BdV04]. Existence of strong solutions of the motion of a rigid body in a viscous incompressible fluid, at least for a short time interval, has instead been proved in [GM00] under the hypothesis that the mass and the inertia of the body are sufficiently large.

See also [GM01] for a review of results concerning fluid-structure interaction problems or [LM01], [GV03] or [GVF03] for a more comprehensive presentation.

Since our aim in the following chapters is not to further analyze the coupled problem, but to improve algorithmic aspects, we only present the sub-problem decomposition. The fluid test function spaces are presented in chapter 2 for the axisymmetric case. In a general framework, in the case of natural inflow and outflow condition, the fluid function spaces are

3.1. FORMULATION OF THE FLUID-STRUCTURE PROBLEM

defined through the ALE mapping as

$$V(\Omega_t^f) = \left\{ \mathbf{v} : \Omega_t^f \rightarrow \mathbb{R}^3, \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_t^{-1}, \hat{\mathbf{v}} \in H^1(\hat{\Omega}^f)^3 \right\}, \quad (3.1)$$

$$V_\Sigma(\Omega_t^f) = \left\{ \mathbf{v} \in V(\Omega_t^f), \mathbf{v} \circ \mathcal{A}_t = 0 \text{ on } \Sigma_t \right\}, \quad (3.2)$$

$$Q(\Omega_t^f) = \left\{ q : \Omega_t^f \rightarrow \mathbb{R}, q = \hat{q} \circ \mathcal{A}_t^{-1}, \hat{q} \in L^2(\hat{\Omega}^f) \right\}, \quad (3.3)$$

$$V(\Omega^f) = \left\{ \mathbf{v} : \Omega^f \times I \rightarrow \mathbb{R}^3, \mathbf{v}_t \in V(\Omega_t^f) \right\},$$

$$Q(\Omega^f) = \left\{ q : \Omega^f \times I \rightarrow \mathbb{R}, q_t \in Q(\Omega_t^f) \right\}.$$

We introduce the structure function space as

$$X(\hat{\Omega}^s) = \left\{ \mathbf{d} \in H^1(\hat{\Omega}^s)^3, \mathbf{d}|_{\partial\hat{\Omega} \setminus \hat{\Sigma}} = 0 \right\},$$

with the obvious modifications in case of Dirichlet inlet or outlet conditions for the fluid or Neumann boundary conditions for the structure. Under the assumptions described in section 2.2.2 on the ALE mapping, the fluid-structure interaction problem in weak form is the coupling of the following three problems,

P3.3 (ALE mapping) *For almost every t in I , find $\mathcal{A}_t : \hat{\Omega}^f \rightarrow \Omega_t^f$ such that \mathcal{A}_t in $H^1(\hat{\Omega}^f)^3$ and for all \mathbf{y} in $H^1(\hat{\Omega}^f)^3$ with $\mathbf{y}|_{\partial\hat{\Omega}^f} = 0$,*

$$\begin{aligned} \int_{\hat{\Omega}^f} \nabla \mathcal{A}_t : \nabla \mathbf{y} d\mathbf{x} &= 0, \\ \mathcal{A}_t &= \mathbf{d}_t \quad \text{on } \hat{\Sigma}, \\ \mathcal{A}_t &= 0 \quad \text{on } \hat{\Omega}^f \setminus \hat{\Sigma}. \end{aligned} \quad (3.4)$$

Define $\mathbf{w}_t = \frac{d\mathcal{A}_t}{dt}$ on $\hat{\Omega}^f$ and $\Omega_t^f = \mathcal{A}_t(\hat{\Omega}^f)$.

P3.4 (Fluid) *Find (\mathbf{u}, p) in $V(\Omega^f) \times Q(\Omega^f)$ such that for almost every t in I and for all (\mathbf{v}, q) in $V_\Sigma(\Omega_t^f) \times Q(\Omega_t^f)$,*

$$\begin{cases} \frac{d}{dt} \int_{\Omega_t^f} \rho_f \mathbf{u}_t \cdot \mathbf{v} + \int_{\Omega_t^f} \rho_f (\mathbf{u}_t - \mathbf{w}) \cdot \nabla \mathbf{u}_t \cdot \mathbf{v} - \int_{\Omega_t^f} \rho_f \mathbf{u}_t \cdot \mathbf{v} \operatorname{div} \mathbf{w} \\ \quad - \int_{\Omega_t^f} p_t \operatorname{div} \mathbf{v} + \int_{\Omega_t^f} 2\mu \epsilon(\mathbf{u}_t) \cdot \epsilon(\mathbf{v}) = \int_{\Gamma_t^{\text{in}} \cup \Gamma_t^{\text{out}}} \mathbf{g}_t \cdot \mathbf{v} ds, \\ \int_{\Omega_t^f} q \operatorname{div} \mathbf{u}_t = 0, \end{cases} \quad (3.5)$$

$$\mathbf{u}_t = \mathbf{w}_t \circ \mathcal{A}_t^{-1} \quad \text{on } \Sigma_t.$$

P3.5 (Structure) *For almost every t in I find $\mathbf{d}(\cdot, t)$ in $X(\hat{\Omega}^s)$ such that for all $\hat{\varphi}$ in $X(\hat{\Omega}^s)$,*

$$\int_{\hat{\Omega}^s} \rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} \cdot \hat{\varphi} d\hat{\mathbf{x}} + a_s(\mathbf{d}, \hat{\varphi}) = \int_{\Sigma_t} (p\mathbf{n} - 2\mu \epsilon(\mathbf{u}) \cdot \mathbf{n}) \cdot \hat{\varphi} \circ \mathcal{A}_t^{-1} ds. \quad (3.6)$$

The right hand side of (3.6) imposes the equality in fluid and structure normal stresses and can be computed as the residual of the first equation in (3.5) with test functions \mathbf{v} equal to the lifts of $\varphi \circ \mathcal{A}_t^{-1}|_{\hat{\Sigma}}$. This can be easily verified by integrating by parts (3.5).

The operator a_s is in general non-linear and can be written for example in terms of the free energy function (see for example [LT94]). In case of a generalized string model, the operator a_s is linear and as described in [Nob01, §4], $\hat{\Omega}^s$ reduces to $\Sigma_t \times [0, h]$ and

$$a_s(d_r, \hat{\varphi}) = \frac{Eh}{1 - \nu^2} \int_{\Sigma_t} d_r \hat{\varphi} dz + h \int_{\Sigma_t} \left(kG \frac{\partial d_r}{\partial z} + \frac{\gamma}{h} \frac{\partial^2 d_r}{\partial z \partial t} \right) \frac{\partial \hat{\varphi}}{\partial z} dz.$$

Here the pressure is scaled with respect to the external pressure, i.e., to the real pressure we subtract the external pressure.

The system arising from P3.3-P3.4-P3.5 is coupled and non-linear. Concerning its time discretization several schemes can be considered. A first strategy leads to a *loosely coupled algorithm* [FL00, PFL95], which consists in using an explicit scheme for the fluid (respectively for the structure) and an implicit scheme for the structure (respectively for the fluid). Thus, at each time step, the fluid solution is completely determined starting from the solution of the previous time step and, once the fluid load at the interface has been computed, the structure can be advanced on time updating the position of the interface. In short, the geometry and the interface coupling are treated explicitly. This strategy is computationally cheap and performs well in many practical situations, for example, in aeroelasticity applications [FL00, PFL95]. However, numerical experiments and analysis on simplified models (see [LM01, Nob01, GV03]) indicate that these staggered algorithms are unstable when the structure is light, more precisely when the fluid and structure densities are comparable, as it happens in haemodynamics applications. In these situations, fluid-structure equilibrium must be ensured accurately at each time step. In other words, the geometry and the interface coupling have to be treated implicitly, and then implicit coupling schemes must be considered. For these reasons we will focus on fully coupled implicit schemes.

3.2 Time discretization

In the sequel, we consider a time discretization of the coupled system based on the implicit Euler method for the fluid equations and a mid-point rule for the structure. For simplicity we write a scheme for the fluid equation which does not satisfy the GCL condition. The resulting time discretized problem reads: For $n = 0, 1, \dots, \frac{T}{\Delta t} - 1$,

P3.6 (Discrete ALE mapping) Find $\mathcal{A}_{t^{n+1}}$ in $H^1(\hat{\Omega}^f)^3$ such that for all \mathbf{y} in $H_0^1(\hat{\Omega}^f)^3$

$$\begin{aligned} \int_{\hat{\Omega}^f} \nabla \mathcal{A}_{t^{n+1}} : \nabla \mathbf{y} dx &= 0, \\ \mathcal{A}_{t^{n+1}} &= \mathbf{d}^{n+1} \quad \text{on } \hat{\Sigma}, \\ \mathcal{A}_{t^{n+1}} &= 0 \quad \text{on } \partial \hat{\Omega}^f \setminus \hat{\Sigma}. \end{aligned} \tag{3.7}$$

Define $\mathbf{w}^{n+1} = \frac{\mathcal{A}_{t^{n+1}} - \mathcal{A}_{t^n}}{\Delta t}$ and $\Omega_t^f = \mathcal{A}_t(\hat{\Omega}^f)$.

3.2. TIME DISCRETIZATION

P3.7 (Semi-discretizaion of the fluid equations) Set $\Omega_{t^{n+1}}^f = \mathcal{A}_{t^{n+1}}(\hat{\Omega}^f)$. Find $(\mathbf{u}^{n+1}, p^{n+1})$ in $V(\Omega_{t^{n+1}}^f) \times Q(\Omega_{t^{n+1}}^f)$ such that for all (\mathbf{v}, q) in $V_\Sigma(\Omega_{t^{n+1}}^f) \times Q(\Omega_{t^{n+1}}^f)$ and for $\mathbf{u}^* = \mathbf{u}^{n+1}$,

$$\begin{cases} \frac{1}{\Delta t} \int_{\Omega_{t^{n+1}}^f} \rho_f \mathbf{u}^{n+1} \cdot \mathbf{v} + \int_{\Omega_{t^{n+1}}^f} \rho_f (\mathbf{u}^* - \mathbf{w}^{n+1}) \cdot \nabla \mathbf{u}^{n+1} \cdot \mathbf{v} - \int_{\Omega_{t^{n+1}}^f} \rho_f \mathbf{u}^{n+1} \cdot \mathbf{v} \operatorname{div} \mathbf{w}^{n+1} \\ - \int_{\Omega_{t^{n+1}}^f} p^{n+1} \operatorname{div} \mathbf{v} + \int_{\Omega_{t^{n+1}}^f} 2\mu \epsilon(\mathbf{u}^{n+1}) \cdot \epsilon(\mathbf{v}) = \frac{1}{\Delta t} \int_{\Omega_{t^n}^f} \rho_f \mathbf{u}^n \cdot \mathbf{v} + \int_{\Gamma_{t^{n+1}}^{\text{in}} \cup \Gamma_{t^{n+1}}^{\text{out}}} \mathbf{g} \cdot \mathbf{v} ds, \\ \int_{\Omega_{t^{n+1}}^f} q \operatorname{div} \mathbf{u}^{n+1} = 0, \end{cases} \quad (3.8)$$

$$\mathbf{u}^{n+1} = \mathbf{w}^{n+1} \circ \mathcal{A}_{t^{n+1}}^{-1} \quad \text{on } \Sigma_{t^{n+1}}.$$

P3.8 (Semi-discretization of the structure equation) Find $(\mathbf{d}^{n+1}, \dot{\mathbf{d}}^{n+1})$ in $X(\hat{\Omega}^s) \times L^2(\hat{\Omega}^s)$ such that for all $\hat{\varphi}$ in $X(\hat{\Omega}^s)$,

$$\begin{aligned} \frac{1}{\Delta t} \int_{\hat{\Omega}^s} \rho_s (\dot{\mathbf{d}}^{n+1} - \dot{\mathbf{d}}^n) \cdot \hat{\varphi} d\hat{\mathbf{x}} + \frac{1}{2} \left(a_s(\mathbf{d}^{n+1}, \hat{\varphi}) + a_s(\mathbf{d}^n, \hat{\varphi}) \right) \\ = \int_{\Sigma_t} (p^{n+1} \mathbf{n} - 2\mu \epsilon(\mathbf{u}^{n+1}) \cdot \mathbf{n}) \cdot \hat{\varphi} \circ \mathcal{A}_t^{-1} ds, \quad (3.9) \\ \frac{\mathbf{d}^{n+1} - \mathbf{d}^n}{\Delta t} = \frac{\dot{\mathbf{d}}^{n+1} + \dot{\mathbf{d}}^n}{2}. \end{aligned}$$

We consider sub-problem P3.7 solved when $\mathbf{u}^* = \mathbf{u}^{n+1}$, but often an extrapolation of the velocity is used to linearize the problem. In sub-problem P3.8, $\dot{\mathbf{d}}$ is an approximation of the structure velocity. When using conforming finite elements, the forcing term on the right hand side of (3.9) can be computed as the residual of the fluid equations.

We assume that each problem has been appropriately discretized in space, for instance by a finite element formulation. We formally write the operators associated to each subproblem as follows:

1. For a given displacement of the structure at the interface $\mathbf{d}_\Sigma = \mathbf{d}|_\Sigma$, the ALE map and its velocity are computed on the whole fluid domain:

$$(\mathcal{A}_{t^{n+1}}, \mathbf{w}^{n+1}) = \mathcal{D}(\mathbf{d}_\Sigma^{n+1}); \quad (\text{P3.6})$$

2. The Navier–Stokes equations in ALE formulation are solved on the new domain to obtain the velocity and the pressure of the fluid:

$$(\mathbf{u}^{n+1}, p^{n+1}) = \mathcal{F}(\mathcal{A}_{t^{n+1}}, \mathbf{w}^{n+1}); \quad (\text{P3.7})$$

3. The structure equations are solved to obtain the wall displacement and its velocity corresponding to the force that the fluid exerts on the structure, thus in particular the displacement of the fluid-structure interface:

$$\mathcal{S} = \mathcal{S}_2 \circ \mathcal{S}_1 : \begin{cases} \sigma = \mathcal{S}_1(\mathbf{u}^{n+1}, p^{n+1}), \\ (\mathbf{d}^{n+1}, \dot{\mathbf{d}}^{n+1}) = \mathcal{S}_2(\sigma). \end{cases} \quad (\text{P3.8})$$

In (P3.7), \mathcal{F} is the Navier–Stokes solution operator. In some circumstances, we might be interested to solve a linear flow problem in which the convective field \mathbf{u}^{n+1} is replaced by a suitable (given) approximation, say \mathbf{u}^* . In that case the corresponding solution operator will be noted by $\mathcal{F}_{\mathbf{u}^*}$. Note that $\mathcal{F} = \mathcal{F}_{\mathbf{u}^{n+1}}$.

In (P3.8) we have split the computation of \mathcal{S} into the computation of the forcing terms exerted by the fluid on the structure and the resolution of the structure problem. If the equation for the structure is non-linear, we use for example a Newton–Raphson method to solve P3.8. It is also possible to linearize the structure equation with an extrapolation, in which case some light modifications apply.

For the applications to blood flow simulations we are interested in strongly coupled algorithms. More precisely we look for a fixed-point on the interface displacement of the mapping $\mathcal{T} = \gamma_\Sigma \circ \mathcal{S} \circ \mathcal{F} \circ \mathcal{D}$ (where γ_Σ is the restriction operator that maps $(\mathbf{d}, \dot{\mathbf{d}})$ to \mathbf{d}_Σ) at every time step, i.e.,

$$\mathcal{T}(\mathbf{d}_\Sigma) = \mathbf{d}_\Sigma. \quad (3.10)$$

Note that \mathcal{T} is a nonlinear operator acting on the structure displacement at the fluid-structure interface. The nonlinearities come from: The inertial term in the Navier–Stokes equations, the displacements of the fluid domain and the (large) displacements in the structure.

In the sequel, we will look for strategies to solve efficiently the coupled problem and we will refer to the operators \mathcal{T} , \mathcal{S} , \mathcal{F} and \mathcal{D} defined here.

3.3 An abstract formulation

In this and the following sections we provide a general framework for a class of iterative methods which are widely used for the solution of the non-linear coupled problem P3.6-P3.7-P3.8. In this section we describe Newton based methods to solve the fluid-structure problem (3.10) and we consider how to compute the Jacobian of the fixed point problem and possible approximations leading to different algorithms.

In order to solve the fluid-structure interaction problem (3.10) by a Newton method, we make use of the formalism introduced in [FM04]. We denote by $u = (\mathbf{u}, p)$ the fluid state variables and $d = (\mathbf{d}, \dot{\mathbf{d}})$ the solid ones. Let $U^f \times U^s$ be the space of fluid and solid states and V^f and V^s the space of the fluid and solid test functions. Let \mathbb{F} be the fluid operator associated to the fluid variational formulation (3.7)-(3.8):

$$\begin{aligned} \mathbb{F} : U^f \times U^s &\rightarrow (V^f)' \\ (u, d) &\mapsto \mathbb{F}(u, d), \end{aligned}$$

Then, since \mathcal{F} and \mathcal{D} are the solvers associated to (3.7) and (3.8), for all d in U^s

$$\mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) = 0 \quad \text{in } (V^f)'. \quad (3.11)$$

Similarly, let \mathbb{S} be the solid operator associated to the solid variational formulation (3.9):

$$\begin{aligned} \mathbb{S} : U^f \times U^s &\rightarrow (V^s)' \\ (u, d) &\mapsto \mathbb{S}(u, d), \end{aligned}$$

hence for all u in U^f

$$\mathbb{S}(u, \mathcal{S}(u)) = 0, \quad \text{in } (V^s)'. \quad (3.12)$$

3.3. AN ABSTRACT FORMULATION

Since equations (3.11) and (3.12) characterize the solvers \mathcal{D} , \mathcal{F} and \mathcal{S} , using the above definitions, the coupled non-linear problem P3.6-P3.7-P3.8 can be reduced to:

Find (u, d) in $U^f \times U^s$ such that

$$\mathbb{F}(u, d) = 0, \quad (3.13)$$

$$\mathbb{S}(u, d) = 0. \quad (3.14)$$

Substituting u in (3.13) with $\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma)$ and replacing it in (3.14) leads to the fixed point problem (3.10), which we can write as: Find \mathbf{d}_Σ such that

$$\mathcal{R}(\mathbf{d}_\Sigma) \stackrel{\text{def}}{=} \mathcal{T}(\mathbf{d}_\Sigma) - \mathbf{d}_\Sigma = 0. \quad (3.15)$$

Although problems (3.13)-(3.14) and (3.15) are equivalent, the first one is set on the domain $\Omega^f \cup \Omega^s$, while the second one on the fluid-structure interface Σ .

3.3.1 Newton based algorithms for the solution of the fixed-point problem

A natural approach consists of defining fixed point iterations to solve (3.15), i.e., given \mathbf{d}_Σ^0 , find for $k > 0$, $\mathbf{d}_\Sigma^{k+1} = \mathcal{T}(\mathbf{d}_\Sigma^k)$. However, for the problem at hand in most cases these iterations diverge. The convergence may be recovered by relaxed fixed point method as described in chapter 4. Otherwise, we can formulate a non-linear GMRES algorithm to solve the non-linear problem (3.15) in a general form as follows. Let $J(\mathbf{d}_\Sigma)$ be an approximation of the Jacobian of \mathcal{R} in \mathbf{d}_Σ and ω^k a scalar to be chosen at each iteration.

We want to find an approximation \mathbf{d}_Σ^* of the root of \mathcal{R} , such that $\|\mathcal{R}(\mathbf{d}_\Sigma^*)\| < \text{Tol}$ for given tolerance and norm. The (quasi-)Newton algorithm (qN) associated to this problem reads:

1. define an initial guess \mathbf{d}_Σ^0 , set $k = 0$ and compute $\mathcal{R}(\mathbf{d}_\Sigma^0)$;
2. solve $J(\mathbf{d}_\Sigma^k)\delta\mathbf{d}_\Sigma = -\mathcal{R}(\mathbf{d}_\Sigma^k)$;
3. set $\mathbf{d}_\Sigma^{k+1} = \mathbf{d}_\Sigma^k + \omega^k\delta\mathbf{d}_\Sigma$;
4. compute the residual $\mathcal{R}(\mathbf{d}_\Sigma^{k+1})$;
5. if $\|\mathcal{R}(\mathbf{d}_\Sigma^{k+1})\| < \text{Tol}$, then stop, otherwise increase k and go to 2.

Step 2 can be carried out using an iterative free matrix method such as GMRES. In this case, we only need to evaluate several times the operator $J(\mathbf{d}_\Sigma)$ against solid state perturbations \mathbf{z}_Σ ,

$$J(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma = \gamma_\Sigma \cdot \mathcal{S}'(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma)) \cdot (\mathcal{F} \circ \mathcal{D})'(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma - \mathbf{z}_\Sigma, \quad (3.16)$$

where we recall that $\gamma_\Sigma \cdot d = \mathbf{d}_\Sigma$.

On step 3, the scalar ω^k is in general equal to one, save in the cases when the norm of the residual is not reduced. In fact, when using the exact Jacobian, the Newton algorithm guarantees that the norm is monotonically decreasing. However, this can be no longer true, when replacing the Jacobian by an inexact Jacobian. Then a strategy, for example line search, must be chosen in order to guarantee a smaller residual.

In the following chapters, we will propose three alternatives to approximate the Jacobian (see section 5.3) and one to approximate the residual (see chapter 4). We will also present a dynamic preconditioner (section 5.4) which can be used on step 2, as well as an extension to the vector case of the Aitken method to dynamically choose ω^k when the Jacobian is replaced by the identity.

3.3.2 Computation of the Jacobian against a given vector

We want to compute $J(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma$ exploiting the relationship between problems (3.13)-(3.14) and (3.15). With this aim, let d and z be extensions of \mathbf{d}_Σ and \mathbf{z}_Σ in Ω^s and D_u and D_d denote differentiation with respect to u and d . Assume that we are able to compute the four block of the Jacobian of $(\mathbb{F}, \mathbb{S})^T$

$$\begin{bmatrix} D_u \mathbb{F} & D_d \mathbb{F} \\ D_u \mathbb{S} & D_d \mathbb{S} \end{bmatrix}. \quad (3.17)$$

We do not apply a Newton method for the problem (3.13)-(3.14) on the whole domain $\Omega^f \cup \Omega^s$, but prefer to use the blocks in (3.17) to compute (3.16).

From the implicit differentiation theorem and identity (3.11), we have

$$\begin{aligned} 0 &= D_d (\mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d)) \cdot z \\ &= D_u \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot (\mathcal{F} \circ \mathcal{D})'(\mathbf{d}_\Sigma) \cdot \gamma_\Sigma \cdot z + D_d \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot z \\ &= D_u \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot (\mathcal{F} \circ \mathcal{D})'(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma + D_d \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot z, \end{aligned} \quad (3.18)$$

and from (3.12),

$$0 = D_u (\mathbb{S}(u, \mathcal{S}(u))) \cdot w = D_u \mathbb{S}(u, \mathcal{S}(u)) \cdot w + D_d \mathbb{S}(u, \mathcal{S}(u)) \cdot \mathcal{S}'(u) \cdot w, \quad (3.19)$$

In order to evaluate (3.16), we have first to compute

$$w = (\mathcal{F} \circ \mathcal{D})'(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma,$$

which thanks to (3.18) is equivalent to the solution w of the following problem,

$$D_u \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot w = -D_d \mathbb{F}(\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma), d) \cdot z. \quad (3.20)$$

Since \mathbf{d}_Σ is a given vector in (3.16), let us denote by u the value $\mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma)$. Then we must compute

$$y = \mathcal{S}'(u) \cdot w$$

If the structure operator is linear, than y can be computed straight-forwardly. Otherwise, from (3.19) we have that y can be obtained by solving the following problem,

$$D_d \mathbb{S}(u, \mathcal{S}(u)) \cdot y = -D_u \mathbb{S}(u, \mathcal{S}(u)) \cdot w. \quad (3.21)$$

To summarize, if \mathbf{d}_Σ is a given solid state vector, the computation of $J(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma$ for any \mathbf{z}_Σ can be achieved by the following steps:

1. Solve (3.20);
2. Solve (3.21);
3. compute $\gamma_\Sigma y - \mathbf{z}_\Sigma$.

The diagonal blocks of the Jacobian (3.17) are the natural derivatives of the fluid, respectively structure, problems, which implies the solution of the tangent problem of the fluid, respectively structure. The main difficulty (see [FM03, FM04]) relies on the computation of

3.3. AN ABSTRACT FORMULATION

the extra-diagonal blocks. For example, the right hand side of (3.20) involves the evaluation of the cross-derivative of the fluid operator:

$$D_d \mathbb{F}(u, d) \cdot z,$$

which corresponds to the directional derivative with respect to fluid-domain perturbations. In previous works, the evaluations of these cross-Jacobians were performed using finite difference approximations, that only require state operators evaluations [MS02, MS03, Tez01, Hei03]. However, the lack of *a priori* criteria for selecting optimal finite difference infinitesimal steps, may lead to a reduction of the overall convergence speed [GV03].

In order to speed up the convergence toward the solution of problem (3.15), in the next paragraphs we describe several techniques to perform (possibly using Jacobian approximations) the critical step (3.20) in the Newton's loop (qN).

Neglecting the cross derivative $D_d \mathbb{F}$

If $D_d \mathbb{F}$ is replaced by zero, then in expression (3.20) $w = 0$ and in (3.16) $J(\mathbf{d}_\Sigma) \cdot \mathbf{z}_\Sigma = -\mathbf{z}_\Sigma$ and the Newton algorithm (qN) reduces to a block Gauss-Seidel (BGS) algorithm in the variables u and d (see BGS, section 4.1). In this case ω^k is a relaxation parameter which must be chosen appropriately (see Aitken acceleration, section 4.2). Since this algorithm needs several iterations to converge, it is convenient to replace the computation of the residual by a simplified one at least for some iterations (see transpiration conditions, section 4.3).

Neglecting the volumic terms in $D_d \mathbb{F}$ and the convective and the diffusive terms in $D_u \mathbb{F}$

In [GV03] the authors provide an explicit expression for the case when $D_d \mathbb{F}$ is neglected on $\Omega^f \setminus \Sigma$ and $D_u \mathbb{F}$ is replaced by $D_u \tilde{\mathbb{F}}$: The nonlinear acceleration and viscous terms are neglected in Ω^f (see also section 5.3, FSI-QN1). Then the tangent problem for the fluid is replaced by a simpler one where the domain is frozen about its current state.

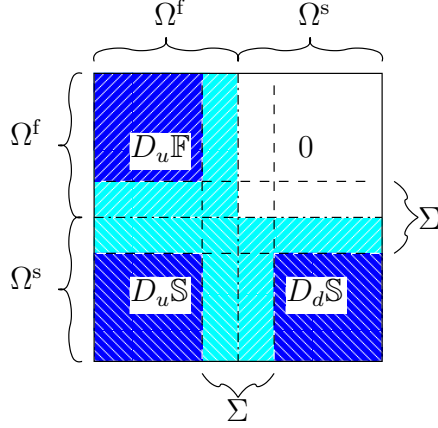
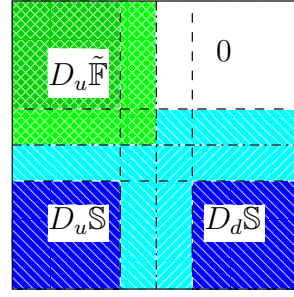
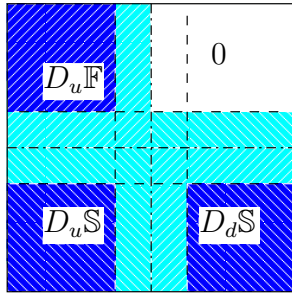
Neglecting the volumic terms in $D_d \mathbb{F}$

In section 5.3 (see FSI-QN2) we provide an explicit expression for the case when $D_d \mathbb{F}$ is neglected on $\Omega^f \setminus \Sigma$. Then the tangent problem for the fluid is replaced by a simpler one where the domain is frozen about its current state.

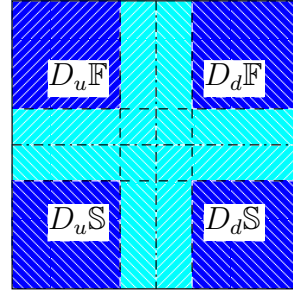
Exact Jacobian

In [FM04], the authors provide the explicit expression of the cross Jacobians using shape sensitivity calculus (see [SZ92]). In a block formulation, this can be viewed as performing Newton's iterations using the complete exact Jacobian of the coupled non-linear problem.

The approximations indicated above for problem (3.13)-(3.14) would lead to the following inexact Jacobian (3.17):

Neglecting the cross derivative $D_d F$

 Neglecting the volumic terms in $D_d F$ and the convective and the diffusive terms in $D_u F$

 Neglecting the volumic terms in $D_d F$


Exact Jacobian



Remark: Dynamic preconditioner and accelerated Newton algorithm

In chapter 5.4 we introduce a dynamic preconditioner that can be used in the GMRES resolution of step 2 which is well suited for the case of either the exact or the inexact Jacobian computation. The preconditioner is built up during the first Newton loop (at each time step).

In section 5.5 we propose a modified Newton algorithm which spares some residual's evaluations and Jacobian's inversions.

3.3.3 Strategies for the solution of the non-linear coupled system

To conclude this section, we give a few suggestions on the suitability of the various methods indicated above for the solution of the fluid-structure problem.

The BGS algorithm (chapter 4) is well suited when we want to re-use an already existing software for the fluid and the structure, separately, and we can not modify them to include the computation of the Jacobian. If the software allows to, a transpiration condition can be applied (section 4.3). However, a very restrictive condition on the time step applies (see [GV03]).

The exact computation of the Jacobian reduces dramatically the number of Newton iterations and the condition on the time step may be relaxed. However, the computation of the Jacobian is costly and requires the use of shape derivatives (see [FM04]).

The approximations FSI-QN1 and FSI-QN2 of the Jacobian, in particular FSI-QN1, seem to offer a good compromise (chapter 5). In fact, the codes can be modified with a relative effort and the computation of the approximated Jacobian, in particular for FSI-QN1, is cheap. However, numerical experience suggests that the time step must be smaller than in the case

of the exact Jacobian and a line-search at step 3 is necessary if the residual does not decrease (see [FM04]).

3.4 A domain decomposition formulation approach

In this section we reformulate the fluid-structure interaction problem in a domain decomposition framework, then we propose several splitting algorithms which are mutated from sub-domain iterative procedures (see also [DDQ]).

3.4.1 Stokes problem

We start by considering the Stokes problem that we formally write as

$$\text{Stokes}(\mathbf{u}, p) = \mathbf{f}. \quad (3.22)$$

Let Ω be split into two sub-domains Ω_1 and Ω_2 and let Σ be the interface between them. We assume that on each $\partial\Omega_j$ there is a set of positive measure where Neumann conditions are prescribed. We denote by λ the velocity $\mathbf{u}|_\Sigma$ on the interface and by u_j the fluid variables (\mathbf{u}, p) on the sub-domain Ω_j , $j = 1, 2$. We also denote the normal stress of u_j on Σ , $j = 1, 2$, by $\sigma_j(u_j)|_\Sigma = \mu \frac{\partial \mathbf{u}_j}{\partial \mathbf{n}_j} - p_j \mathbf{n}_j$ (no summation on repeated indices) and we introduce the following operators:

Dirichlet-to-Neumann map in Ω_j , $j = 1, 2$:

$$\begin{aligned} S_j : \lambda &\mapsto \sigma \\ \lambda &\mapsto \text{find } u_j : \begin{cases} \text{Stokes}(u_j) = 0 \\ u_j|_\Sigma = \lambda \end{cases} \mapsto \sigma = \sigma_j(u_j)|_\Sigma. \end{aligned}$$

Neumann-to-Dirichlet map (homogeneous) in Ω_j , $j = 1, 2$:

$$\begin{aligned} (\bar{S}_j)^{-1} : \sigma &\mapsto \lambda \\ \sigma &\mapsto \text{find } u_j : \begin{cases} \text{Stokes}(u_j) = 0 \\ \sigma_j(u_j)|_\Sigma = \sigma \end{cases} \mapsto \lambda = u_j|_\Sigma. \end{aligned}$$

Note that if $\mathbf{f}_j = 0$, then $(\bar{S}_j)^{-1}$ is the inverse of S_j . The Steklov-Poincaré problem associated to the Stokes problem (3.22) (see [QA99, §5] for details) reads: Find λ such that

$$S_1(\lambda) + S_2(\lambda) = \chi, \quad (3.23)$$

for a suitable right hand side χ that depends, among others, on \mathbf{f} .

Preconditioned Richardson method

The preconditioned Richardson method applied to the Steklov–Poincaré problem (3.23) reads: Start with λ^0 , $k = 0$.

Repeat until $\|\mu^k\| < \text{Tol}$

$$\begin{aligned} (D) \quad & \begin{cases} \sigma_1^k = S_1 \lambda^k \\ \sigma_2^k = S_2 \lambda^k \end{cases}, \\ & \sigma^k = \sigma_1^k + \sigma_2^k - \chi, \\ & \mu^k = P^{-1} \sigma^k, \\ & \lambda^{k+1} = \lambda^k + \omega^k \mu^k, \end{aligned}$$

with appropriate choices of the scalar ω^k and of the preconditioner P that maps the interface variables space onto the space of normal stresses, $P : \lambda \mapsto \sigma$ (for example $P : \mathbf{H}_{00}^{1/2}(\Sigma) \rightarrow \mathbf{H}^{-1/2}(\Sigma)$ when $\lambda = 0$ on $\Sigma \cap \partial\Omega$).

The parameter ω^k can be chosen by the same Aitken extrapolation technique described in section 4.2, i.e., for $k > 0$

$$\omega^k = -\frac{(\mu^k - \mu^{k-1}) \cdot (\lambda^k - \lambda^{k-1})}{\|\mu^k - \mu^{k-1}\|^2},$$

which minimizes

$$\left\| (\lambda^k - \lambda^{k-1}) + \omega^k (\mu^k - \mu^{k-1}) \right\|.$$

Special choices of P lead to well known sub-domain iterative procedures. More precisely, let us define:

$$P^{-1} = \alpha_1^k (\bar{S}_1)^{-1} + \alpha_2^k (\bar{S}_2)^{-1}.$$

We obtain:

- If $\alpha_1^k = 1$ and $\alpha_2^k = 0$ (or vice-versa), the Dirichlet-Neumann method;
- If $\alpha_1^k + \alpha_2^k = 1$, the Neumann-Neumann method.

Then $P^{-1} \sigma^k$ is computed in two steps,

$$\begin{aligned} (N) \quad & \begin{cases} \mu_1^k = (\bar{S}_1)^{-1} \sigma^k \\ \mu_2^k = (\bar{S}_2)^{-1} \sigma^k \end{cases}, \\ & \mu^k = \alpha_1^k \mu_1^k + \alpha_2^k \mu_2^k, \end{aligned}$$

Since the problem is linear, in the Dirichlet–Neumann case the computational effort may be reduced to only one Dirichlet solution in one sub-domain and one Neumann solution in the other.

In the Neumann–Neumann case, it can be interesting to choose α_1^k and α_2^k which minimize μ^k , i.e., $\alpha_1^k = 1 - \alpha_2^k$ and

$$\alpha_2^k = -\frac{(\mu_2^k - \mu_1^k) \cdot \mu_1^k}{\|\mu_2^k - \mu_1^k\|^2}.$$

3.4. A DOMAIN DECOMPOSITION FORMULATION APPROACH

Still in the Neumann–Neumann case, another possibility is to choose first ω_1^k and ω_2^k which minimize

$$\left\| \left(\lambda^k - \lambda^{k-1} \right) + \omega_1^k \left(\mu_1^k - \mu_1^{k-1} \right) + \omega_2^k \left(\mu_2^k - \mu_2^{k-1} \right) \right\|$$

then to set

$$\lambda^{k+1} = \lambda^k + \omega_1^k \mu_1^k + \omega_2^k \mu_2^k.$$

The parameters ω_1^k and ω_2^k are the solution of

$$A^T A \begin{pmatrix} \omega_1^k \\ \omega_2^k \end{pmatrix} = A^T \left(\lambda^k - \lambda^{k-1} \right),$$

where A is the two column matrix

$$A = \left(\left(\mu_1^k - \mu_1^{k-1} \right); \left(\mu_2^k - \mu_2^{k-1} \right) \right).$$

3.4.2 Fluid-structure interaction

In the case of the coupled fluid-structure problem, the domain is naturally split into the fluid domain Ω^f and the structure one Ω^s . Let Σ be the interface between them and suppose that the problem has already been discretized in time. We suppose to have solved the problem at time t^n and we look for the solution at time t^{n+1} . We neglect the superscript $n+1$ and denote by λ the interface variable $\mathbf{d}|_\Sigma$, \mathbf{u} the fluid variables (\mathbf{u}, p) while \mathbf{d} is $(\mathbf{d}, \dot{\mathbf{d}})$. We use the following shorthand notation for the fluid and structure problems P3.1 and P3.2:

$$\begin{aligned} \text{Fluid}(\mathbf{u}; \mathbf{g}) : & \begin{cases} \Delta \mathcal{A}_{t^{n+1}} = 0, \\ \Omega_{t^{n+1}}^f = \mathcal{A}_{t^{n+1}}(\hat{\Omega}^f), \\ \rho_f \left(\frac{\partial \mathbf{u}}{\partial t} \Big|_{\hat{\mathbf{x}}} + (\mathbf{u} - \mathbf{w}) \nabla \mathbf{u} \right) = \text{div}(2\mu \boldsymbol{\epsilon}(\mathbf{u})) - \nabla p & \text{on } \Omega_{t^{n+1}}^f, \\ \text{div } \mathbf{u} = 0 & \text{on } \Omega_{t^{n+1}}^f, \\ \boldsymbol{\sigma}_f \cdot \mathbf{n} = \mathbf{g} & \text{on } \Gamma_{t^{n+1}}^{\text{in}} \cup \Gamma_{t^{n+1}}^{\text{out}}, \end{cases} \\ \text{Str}(\mathbf{d}) : & \begin{cases} \rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} = \text{div}(\boldsymbol{\sigma}_s) & \text{on } \hat{\Omega}^s, \\ \mathbf{d} = 0 \text{ or } \boldsymbol{\sigma}_s \cdot \mathbf{n}_s = 0 & \text{on } \partial \hat{\Omega} \setminus \hat{\Sigma}, \end{cases} \end{aligned}$$

with coupling conditions on Σ

$$\begin{aligned} \mathcal{A}_{t^{n+1}}|_\Sigma &= \lambda, \\ \mathbf{u}|_\Sigma &= \frac{\lambda - \mathbf{d}_\Sigma^n}{\Delta t}, \\ \mathbf{d}|_\Sigma &= \lambda, \\ \boldsymbol{\sigma}_f(\mathbf{u}) \cdot \mathbf{n}_f + \boldsymbol{\sigma}_s(\mathbf{d}) \cdot \mathbf{n}_s &= 0, \end{aligned}$$

where we simplified the notations in P3.1 and P3.2, \mathbf{n}_f is the outward normal unit vector of $\partial \Omega^f$, \mathbf{n}_s is the outward normal unit vector of $\partial \Omega^s$, $\boldsymbol{\sigma}_f$ is the Cauchy stress tensor of the fluid and $\boldsymbol{\sigma}_s$ is the first Piola–Kirchoff stress tensor of the structure.

We note by Fluid'_λ and Str'_λ the tangent problem solvers associated to Fluid and Str. If the ALE mapping is known, than we denote by Fluid_A the solution of the fluid problem. It

will be clear from the context, whether we impose Dirichlet or Neumann boundary conditions on the interface Σ . On $\partial\Omega$, instead, the problems always have the same kind of boundary conditions.

We need to define the operators S_j and $(S_j)^{-1}$, $j = f, s$. We also formally introduce the tangent operators \bar{S}_j and $(\bar{S}_j)^{-1}$, which we may need in the definition of the preconditioner. Remark that in the case of Stokes, the linearity of the problem implies that the tangent operators are equal to the homogeneous operators noted by a “bar”.

Dirichlet-to-Neumann map in Ω^f ,

$$S_f : \lambda \mapsto \text{find } u : \begin{cases} \text{Fluid}(u; \mathbf{g}) \\ \mathbf{u}|_\Sigma = \frac{\lambda - \mathbf{d}_\Sigma^n}{\Delta t} \end{cases} \mapsto \sigma = \boldsymbol{\sigma}_f(u) \cdot \mathbf{n}_f.$$

Note that $u = (\mathbf{u}, p) = \mathcal{F} \circ \mathcal{D}(\mathbf{d}_\Sigma)$ for $\mathbf{d}_\Sigma = \lambda$ on page 65.

Neumann-to-Dirichlet map in Ω^f , with given $\mathcal{A}_{t^{n+1}}$,

$$(S_{f,\mathcal{A}})^{-1} : \sigma \mapsto \text{find } u : \begin{cases} \text{Fluid}_{\mathcal{A}}(u; \mathbf{g}) \\ \boldsymbol{\sigma}_f(u) \cdot \mathbf{n}_f = \sigma \end{cases} \mapsto \lambda = \Delta t \mathbf{u}|_\Sigma + \mathbf{d}_\Sigma^n.$$

Dirichlet-to-Neumann map in Ω^s ,

$$S_s : \lambda \mapsto \text{find } d : \begin{cases} \text{Str}(d) \\ \mathbf{d}_\Sigma = \lambda \end{cases} \mapsto \sigma = \boldsymbol{\sigma}_s(d) \cdot \mathbf{n}_s.$$

Neumann-to-Dirichlet map in Ω^s ,

$$(S_s)^{-1} : \sigma \mapsto \text{find } d : \begin{cases} \text{Str}(d) \\ \boldsymbol{\sigma}_s(d) \cdot \mathbf{n}_s = \sigma \end{cases} \mapsto \lambda = \mathbf{d}_\Sigma.$$

Note that $d = (\mathbf{d}, \dot{\mathbf{d}}) = S_2(-\sigma)$ on page 65.

The Steklov-Poincaré interface equation is

$$S_f(\lambda) + S_s(\lambda) = 0. \quad (3.24)$$

Note that the dependence on the data (χ in the right hand side of (3.23)) is hidden in the definition of the operators S_f and S_s .

In the Stokes problem, we have introduced the homogeneous operators \bar{S}_j . Since the Stokes problem is linear, (\bar{S}_j) coincides with the Stokes tangent operator. Hence we formally introduce also the tangent operators for the fluid and structure problems

Dirichlet-to-Neumann tangent map at a given point λ (homogeneous boundary conditions on $\partial\Omega^f \setminus \Sigma$ and zero body forces),

$$S'_f(\lambda) : \delta\lambda \mapsto \text{find } \delta u : \begin{cases} \text{Fluid}'_\lambda(\delta u) \\ \delta \mathbf{u}|_\Sigma = \frac{\delta\lambda}{\Delta t} \end{cases} \mapsto \delta\sigma = \boldsymbol{\sigma}_f(\delta u) \cdot \mathbf{n}_f.$$

Neumann-to-Dirichlet tangent map at a given point λ (homogeneous boundary conditions on $\partial\Omega^f \setminus \Sigma$ and zero body forces),

$$(S'_f(\lambda))^{-1} : \delta\sigma \mapsto \text{find } \delta u : \begin{cases} \text{Fluid}'_\lambda(\delta u) \\ \boldsymbol{\sigma}_f(\delta u) \cdot \mathbf{n}_f = \delta\sigma \end{cases} \mapsto \delta\lambda = \Delta t \delta \mathbf{u}|_\Sigma.$$

3.4. A DOMAIN DECOMPOSITION FORMULATION APPROACH

Dirichlet-to-Neumann tangent map at a given point λ (homogeneous boundary conditions on $\partial\Omega^s \setminus \Sigma$ and zero body forces),

$$S'_s(\lambda) : \delta\lambda \mapsto \text{find } \delta d : \begin{cases} \text{Str}'_\lambda(\delta d) \\ \delta d|_\Sigma = \delta\lambda \end{cases} \mapsto \delta\sigma = \sigma_s(\delta d) \cdot \mathbf{n}_s.$$

Neumann-to-Dirichlet tangent map at a given point λ (homogeneous boundary conditions on $\partial\Omega^s \setminus \Sigma$ and zero body forces),

$$(S'_s(\lambda))^{-1} : \delta\sigma \mapsto \text{find } \delta d : \begin{cases} \text{Str}'_\lambda(\delta d) \\ \sigma_s(\delta d) \cdot \mathbf{n}_s = \delta\sigma \end{cases} \mapsto \delta\lambda = \delta d|_\Sigma$$

Preconditioned Richardson method

Since the Steklov–Pioncaré problem (3.24) is non-linear, the preconditioned Richardson method must be interpreted in a slightly different way. For the sake of simplicity, even if S_f and S_s are non-linear, we use the notation for linear operators when they are applied to a vector, for example $S_s\lambda^k$ denotes $S_s(\lambda^k)$. A step of the iterative method reads: Start with λ^0 , $k = 0$.

$$(D) \begin{cases} \sigma_f^k = S_f\lambda^k \\ \sigma_s^k = S_s\lambda^k, \\ \sigma^k = \sigma_f^k + \sigma_s^k, \\ \mu^k = P^{-1}\sigma^k, \\ \lambda^{k+1} = \lambda^k + \omega^k\mu^k, \end{cases}$$

with appropriate choice of the preconditioner that maps the interface variables space onto the space of normal stresses, $P : \lambda \mapsto \sigma$.

For example in the Dirichlet–Neumann or Neumann–Neumann methods, P^{-1} can be defined as

$$P^{-1} = \alpha_f^k (S_{f,\mathcal{A}})^{-1} + \alpha_s^k (S_s)^{-1},$$

and as before P^{-1} can be computed in two steps.

Again, in the Dirichlet–Neumann method, the computational effort can be reduced to the resolution of a Dirichlet problem in one sub-domain and a Neumann one on the other. Actually, if the structure is solved with Dirichlet boundary conditions on Σ , the Dirichlet–Neumann method is equivalent to a fixed-point algorithm (see BGS, section 4.1).

An approach resembling the Newton method is to use the tangent operators. The simplest extension is a Dirichlet–Neumann or a Neumann–Neumann method with the preconditioner computed with S'_f and S'_s instead of S_f and S_s :

$$P^{-1} = \alpha_f^k \left(S'_f(\lambda^k) \right)^{-1} + \alpha_s^k \left(S'_s(\lambda^k) \right)^{-1}. \quad (3.25)$$

The pure Newton algorithm is retrieved by choosing P as

$$P^{-1} = \left(S'_f(\lambda^k) + S'_s(\lambda^k) \right)^{-1}.$$

Then one may approximate the tangent problems to accelerate the computations or use a preconditioner to invert P .

This strategy is not equivalent to a Newton algorithm on the fixed-point problem $\mathcal{T}(\mathbf{d}_\Sigma) - \mathbf{d}_\Sigma = 0$ that we consider in section 3.3.1, however, it could be interesting to explore the preconditioner (3.25). The analog of the Newton strategy on the fixed-point iterations is

$$\begin{aligned} \mu^k = P^{-1} \sigma^k &= \left(S'_s(\lambda^k)^{-1} S'_f(\lambda^k) + Id \right)^{-1} S_s^{-1} \sigma^k \\ &= \left(S'_f(\lambda^k) + S'_s(\lambda^k) \right)^{-1} S'_s(\lambda^k) S_s^{-1} \sigma^k, \end{aligned}$$

which we may write, neglecting the dependence on λ^k , as

$$P = S_s S_s'^{-1} (S'_f + S'_s).$$

In fact,

$$S_s^{-1} \sigma^k = S_s^{-1} S'_f \lambda^k + \lambda^k$$

is the residual of the fixed point problem and the Jacobian is

$$S'_s(\lambda^k)^{-1} S'_f(\lambda^k) + Id$$

and the difference in the signs comes from the choice in the unit outwards normal of $\partial\Omega^s$ on Σ .

The methods described here are the result of recent research and open new perspectives. In particular, in the domain decomposition of the Stokes equations, the Aitken like extrapolation must still be tested and in fluid-structure interaction new schemes may be deduced by exploiting the analogy with the domain decomposition. This investigation will be continued in [DDQ].

Chapter 4

Efficient solution of BGS iterations

Introduction

The nature of the problem suggests the use of a simple algorithm, namely fixed point sub-iterations, which in this case are equivalent to block Gauss Seidel (BGS) sub-iterations. It has the main advantage of coupling the fluid and the structure in an independent fashion, such that codes can be easily coupled through natural conditions.

Indeed, a pure fixed point algorithm can not be employed, since in most of the cases the sub-iterations do not converge. In the literature, for example in [Nob01], relaxed fixed point iterations have been successfully used. The algorithm may be easily modified to render it compatible with a factorization of the Navier–Stokes problem with a Yosida or a Chorin–Temam scheme as in section 2.3.4. In particular the fluid operator would depend also on an extrapolation of the pressure.

4.1 Block Gauss Seidel algorithm

The relaxed BGS algorithm reads

1. Set $k = 0$ and define an initial guess for the displacement of the wall and also for the fluid's unknowns

$$\begin{aligned} \mathbf{d}_{\Sigma}^{0,n+1} &= \left(\mathbf{d}_{\Sigma}^n + \frac{3\Delta t}{2} \dot{\mathbf{d}}_{\Sigma}^n - \frac{\Delta t}{2} \dot{\mathbf{d}}_{\Sigma}^{n-1} \right), \\ \mathbf{u}^{0,n+1} &= 2\mathbf{u}^n - \mathbf{u}^{n-1} \end{aligned}$$

and set $\mathbf{u}^* = \mathbf{u}^{0,n+1}$;

2. Solve straightforward

$$\begin{aligned} \left(\mathcal{A}_{t^{n+1}}^{k+1}, \mathbf{w}^{k+1,n+1} \right) &= \mathcal{D} \left(\mathbf{d}_{\Sigma}^{k,n+1} \right), \\ \left(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1} \right) &= \mathcal{F}_{\mathbf{u}^*} \left(\mathcal{A}_{t^{n+1}}^{k+1}, \mathbf{w}^{k+1,n+1} \right), \\ \left(\tilde{\mathbf{d}}^{k+1,n+1}, \dot{\tilde{\mathbf{d}}}^{k+1,n+1} \right) &= \mathcal{S} \left(\mathbf{u}^{n+1}, p^{n+1} \right); \end{aligned}$$

3. Chose $\omega \in (0, 1)$ and relax the interface displacement

$$\mathbf{d}_{\Sigma}^{k+1,n+1} = (1 - \omega) \mathbf{d}_{\Sigma}^{k,n+1} + \omega \tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1};$$

4. If convergence is achieved, i.e., for a given norm

$$\left\| \mathcal{T} \left(\mathbf{d}_{\Sigma}^{k,n+1} \right) - \mathbf{d}_{\Sigma}^{k,n+1} \right\| \leq \text{Tol}, \quad (4.1)$$

then the solution at time t^{n+1} is given by

$$(\mathbf{d}^{n+1}, \dot{\mathbf{d}}^{n+1}) = (\mathbf{d}^{k,n+1}, \dot{\mathbf{d}}^{k,n+1}), \quad (\mathbf{u}^{n+1}, p^{n+1}) = (\mathbf{u}^{k+1,n+1}, p^{k+1,n+1}).$$

Otherwise set

$$\mathbf{u}^* = \mathbf{u}^{k+1,n+1},$$

$k = k + 1$ and go to (2).

The following remarks are in order:

- The failure of the staggered algorithm to converge infers that the coupled problem must be solved with a small tolerance. We will show this with an example. Moreover, when testing the convergence on Δt , we suggest to replace the tolerance by $\Delta t \text{ Tol}$.
- The convergence test must be done on the residual of the coupled fluid-structure problem. In particular, if the fluid is solved by a factorization scheme, it involves also the error on the extrapolations \mathbf{u}^* and p^* . Moreover, the test on the displacement of the structure must be done on the “unrelaxed” quantities, otherwise the relaxation parameter influences the test. It is also important that the norm in step 4 involves also the first time derivative of the displacement.
- The choice of the relaxation parameter is crucial. Nobile in [Nob01] underlines that if ω is constant, than its value must be chosen as a function of the domain, otherwise the algorithm does not converge. In particular in the case of a straight tube, ω decreases to zero as the tube becomes longer and longer.

The problem with a relaxation parameter close to zero is that the convergence of the algorithm is slowed. This suggests a dynamic choice. This strategy is explored in [MW01] [MWR01] and [GV03] and is described in section 4.2.

- The use of a factorization scheme in the fluid resolution is interesting, since extrapolated quantities are naturally given by the previous sub-iteration. However, as already mentioned, this implies a convergence test also on these quantities.
- The algorithm is slowed down by the computation of the ALE mapping and of the matrices. In section 4.3 we propose a version where the geometry is frozen during some sub-iterations.

4.1.1 Residual of BGS

When testing the convergence, we have to define a norm for the residual and to evaluate the residual in inequality (4.1). The residual of the equation is equal to

$$\begin{aligned} \mathcal{T}(\mathbf{d}_{\Sigma}^{k,n+1}) - \mathbf{d}_{\Sigma}^{k,n+1} &= \mathcal{T}(\mathbf{d}_{\Sigma}^{k,n+1}) - \tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} + \tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1} \\ &= \mathcal{S} \circ (\mathcal{F} - \mathcal{F}_{\mathbf{u}^*}) \circ \mathcal{D}(\mathbf{d}_{\Sigma}^{k,n+1}) + \tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1}. \end{aligned}$$

We assume that there is a constant c such that we may bound the norm of the residual and we make the dependence on the norm of the residual on the first time derivative of the structure displacement explicit:

$$\begin{aligned} \left\| \mathcal{T}(\mathbf{d}_{\Sigma}^{k,n+1}) - \mathbf{d}_{\Sigma}^{k,n+1} \right\| &\leq c \left\| (\mathcal{F} - \mathcal{F}_{\mathbf{u}^*, \mathbf{p}^*}) \circ \mathcal{D}(\mathbf{d}_{\Sigma}^{k,n+1}) \right\| \\ &\quad + c \left\| (\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}, \dot{\tilde{\mathbf{d}}}_{\Sigma}^{k+1,n+1}) - (\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1}) \right\|. \end{aligned} \quad (4.2)$$

Indeed there are at least two ways to consider the first term on the right hand side. The first one, is that, assuming that the norm of the operator $(\mathcal{F} - \mathcal{F}_{\mathbf{u}^*}) \circ \mathcal{D}$ is uniformly bounded,

$$c \left\| (\mathcal{F} - \mathcal{F}_{\mathbf{u}^*}) \circ \mathcal{D}(\mathbf{d}_{\Sigma}^{k,n+1}) \right\| \leq c' \left\| (\mathbf{d}_{\Sigma}^{k-1,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k-1,n+1}) - (\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1}) \right\|,$$

In particular, if this method is used, the convergence must be checked on the fluid-structure interface displacement and velocity for the last two sub-iterations,

$$\begin{aligned} \left\| (\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}, \dot{\tilde{\mathbf{d}}}_{\Sigma}^{k+1,n+1}) - (\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1}) \right\| &\leq C_1 \text{ Tol}, \\ \left\| (\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1}) - (\mathbf{d}_{\Sigma}^{k-1,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k-1,n+1}) \right\| &\leq C_2 \text{ Tol}. \end{aligned}$$

The second, is to check the convergence on both the fluid and the structure, i.e.,

$$\begin{aligned} \left\| (\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}, \dot{\tilde{\mathbf{d}}}_{\Sigma}^{k+1,n+1}) - (\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1}) \right\| &\leq C_1 \text{ Tol}, \\ \left\| \mathbf{u}^{k+1,n+1} - \mathbf{u}^{k,n+1} \right\| &\leq C_3 \text{ Tol}. \end{aligned}$$

Then inequality (4.2) guarantees the convergence of the coupled problem.

The constants C_1 , C_2 and C_3 may be difficult to choose and they depend on the norm used. To avoid the evaluation of these constants, it is possible to use the relative error with for example a discrete maximum norm. The main advantage of the relative error is that it is adimensional. However, when the structure is almost at rest, this stopping criterion becomes too restrictive.

This is not the case if we can normalize the error with an absolute value. In this case it is necessary to find adequate reference values. For example, the test on the displacement can be normalized with a reference displacement d^{ref} for the fluid domain. For the study of blood flow in an artery a reasonable value is ten percent of the mean radius of the vessel, since this is the expected value of the displacement of big arteries.

We suggest a heterogeneous test of this kind: For a given norm, e.g. the discrete maximum norm, define the tolerance that we want to achieve for the normalized displacement. Suppose that we would like to achieve

$$\frac{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1}\|}{d^{\text{ref}}} \cong \text{Tol},$$

which is equivalent to

$$\frac{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1}\|}{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}\|} \cong \frac{d^{\text{ref}} \text{Tol}}{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}\|}.$$

Then we can perform the convergence test as

$$\begin{aligned} \|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1}\| &< d^{\text{ref}} \text{Tol}, & \frac{\|\dot{\tilde{\mathbf{d}}}_{\Sigma}^{k+1,n+1} - \dot{\mathbf{d}}_{\Sigma}^{k,n+1}\|}{\|\dot{\tilde{\mathbf{d}}}_{\Sigma}^{k+1,n+1}\|} &< \frac{d^{\text{ref}} \text{Tol}}{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}\|}, \\ \frac{\|\mathbf{u}^{k+1,n+1} - \mathbf{u}^{k,n+1}\|}{\|\mathbf{u}^{k+1,n+1}\|} &< \frac{d^{\text{ref}} \text{Tol}}{\|\tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1}\|}. \end{aligned} \quad (4.3)$$

This approach has two main advantages. The first one is that the test on the displacement is absolute, such that when the displacement is very small, there is no abnormal increase of the number of sub-iterations, and the convergence is performed on all variables with the same order of magnitude. In the applications we have verified the importance of the test on both the velocity of the structure and of the fluid.

The second one is that we can choose the norms independently for each variable. In our applications, we choose the discrete maximum norm for the structure variables and the M -norm based on the lumped mass matrix for the fluid variables.

4.1.2 Fluid's residual's norm

Let Tol be the precision that we require in the coupling. As in section 2.3.3, the tolerance for the residual must be proportional to the parameter h of the triangulations used to solve the fluid and the structure.

Moreover, when using a Yosida factorization scheme, we can use inequality (4.2) to dynamically adapt the precision in the resolution of the fluid. In fact, the resolution of the fluid must be more precise than the accuracy required at the end, but does not need to be “extremely” precise. In other words, we can easily code the linear solver of the fluid, such that the stopping criteria are, with the same matrices notation as in chapter 2,

$$\begin{aligned} \|A^* \tilde{\mathbf{U}} - \mathbf{b}_1 + D^T \mathbf{P}^*\| &\leq \max \left\{ \frac{c_1}{10} \|\mathbf{U}^* - \tilde{\mathbf{U}}\|, \frac{c_2}{10} \text{Tol} \right\}, \\ \|D H D^T \delta \mathbf{P} - \mathbf{b}_2 - D \tilde{\mathbf{U}}\| &\leq \max \left\{ \frac{c_3}{10} \|\delta \mathbf{P}\|, \frac{c_2}{10} \text{Tol} \right\}, \\ \|A^* \mathbf{U}^{k+1,n+1} - \mathbf{b}_1 - D^T \mathbf{P}^{k+1,n+1}\| &\leq \max \left\{ \frac{c_1}{10} \|\mathbf{U}^* - \mathbf{U}^{k+1,n+1}\|, \frac{c_2}{10} \text{Tol} \right\}. \end{aligned}$$

The constants c_1 , c_2 and c_3 depend on C_3 and on the norm used. Moreover, the criteria are divided by ten such that the residual of the fluid-structure problem is not affected by the residual of the fluid equations.

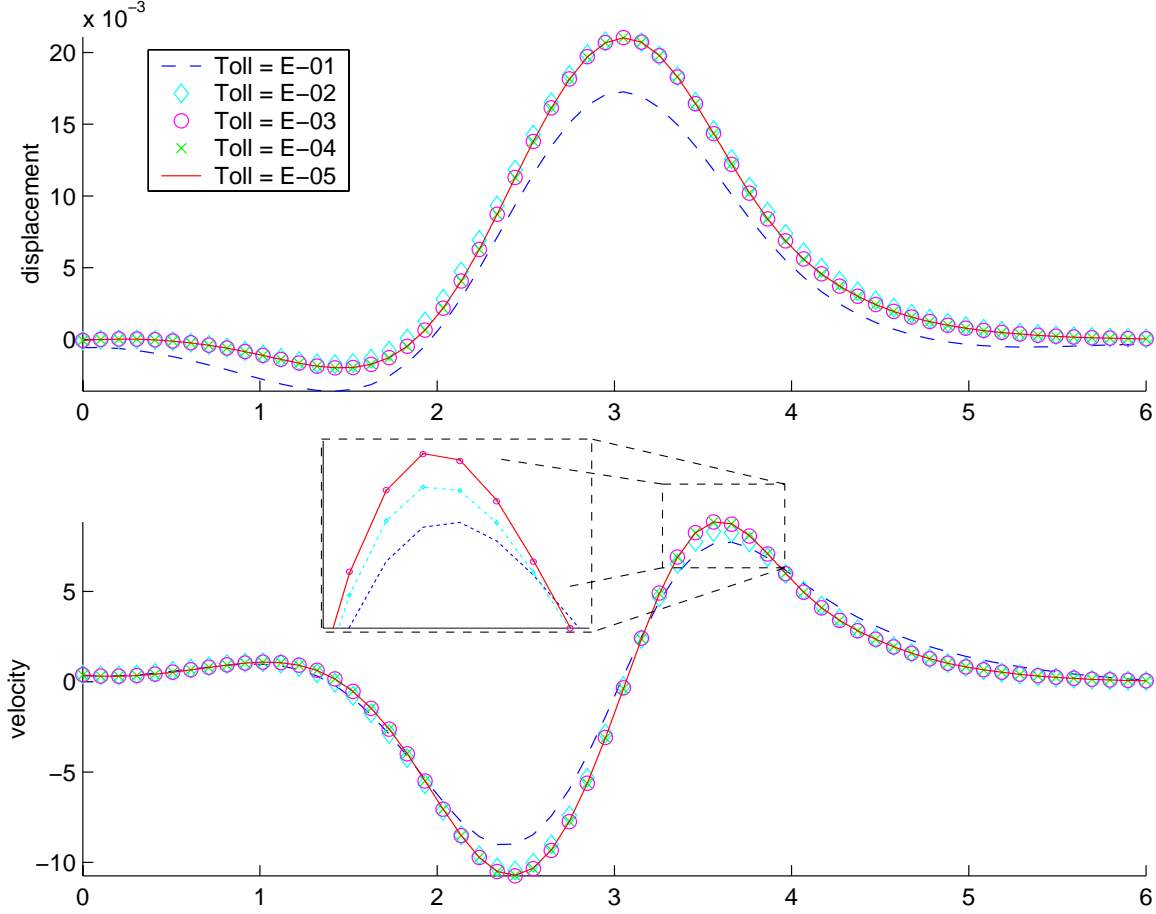


Figure 4.1: Wall displacement and velocity with a weak and very small tolerances.

Example. Let Ω be an axisymmetric tube, the fluid be axisymmetric, incompressible and Newtonian and the structure be modeled by a generalized string equation. The initial domain is a cylinder of radius $R = 0.5\text{cm}$ and length $L = 6\text{cm}$. The cylinder wall may deform only along the radial direction. The fluid and the structure are initially at rest.

The external pressure and the initial pressure of the fluid are both set equal to zero. The wall density is $\rho_s = 1.1\text{g/cm}^3$, its thickness $h = 0.1\text{cm}$, Young modulus $E = 0.75 \cdot 10^4\text{dyne/cm}^2$, Poisson coefficient $\nu = 0.5$, longitudinal stress $kGh = 2.5 \cdot 10^4\text{dyne}$ and viscoelastic constant $\gamma = 0.01$. The fluid viscosity is $\mu = 0.035\text{poise}$ and the fluid density $\rho_f = 1\text{g/cm}^3$.

On the outlet we impose $\sigma(\mathbf{u}, p)\mathbf{n} = 0$ and on the inlet one “pressure wave” of a period of 5 ms, i.e.,

$$\sigma(\mathbf{u}, p)\mathbf{n} = \begin{cases} -\frac{P_{\text{in}}}{2} \left[1 - \cos\left(\frac{2\pi t}{5}\right) \right] \mathbf{n}, & t \leq 5 \text{ ms}, \\ 0, & t > 5 \text{ ms}. \end{cases}$$

with $P_{\text{in}} = 2 \cdot 10^4 \text{ dyne/cm}^2$. We have adopted axisymmetric P1isoP2/P1 finite elements for the fluid and P1 for the structure such that the normal stresses on the structure are easily computed as the residual of the discretized fluid equations. The fluid mesh is uniformly

regular with $h = 1/20\text{cm}$, 5516 elements, 11527 velocity nodes and 3006 pressure nodes. The time is discretized by a mid-point scheme for the structure and implicit Euler for the fluid equations (see chapter 3).

We would like to show that the tolerance required for the coupled problem may be crucial. We adopt the convergence test as in (4.3) with $d^{\text{ref}} = 0.05$. First of all, we remark that without relaxation, the algorithm breaks down after few iterations, even with a very small Δt .

In figure 4.1 the displacement is plotted at different time steps with a too mild tolerance and a very small one. We can remark that the structure displacement and velocity are different for different tolerance. In particular, the velocity needs a more accurate coupling. Also, in the tests, the more difficult constraint to satisfy in the convergence test (4.3) is the one the wall velocity.

□

4.2 The role of Aitken extrapolation

In resolving a fixed-point problem using iterations techniques it is sometimes necessary to introduce a relaxation step. This is for example the case in fluid-structure interaction problems, as described in chapter 3. Fixing the relaxation parameter in advance has are two major drawbacks. Firstly, the choice depends on the problem under consideration and secondly, even if we can find an optimal value, it still involves too many fixed-point iterations.

In the one-dimensional problem, the parameter may be chosen at each iteration by the Aitken method (see [Tra64, A.D], [IK94, 3.1] or [QSS00, 6.6]), which may be derived from the assumption that the problem is linear. In n -dimensional problems, an extensions of the Aitken acceleration method has been presented in [MWR01] and [IT69]. Here we present an interpretation of the formula therein and some possible extensions. The interpretation is based on the linearization of the problem about the exact solution. This allows several extensions of the method, and we propose some of them. In one dimension there exists analytical results about the convergence order of the Aitken method, but to our knowledge, these results are no longer valid in n -dimensions.

4.2.1 Problem setting

Let \mathcal{T} be a vector map from \mathbb{R}^n to \mathbb{R}^n , which in our case is the discrete fluid structure mapping raised on the interface. We are looking at a solution of $\mathcal{T}(\mathbf{x}) = \mathbf{x}$. This problem can be transformed into a root's problem by defining $\mathcal{R}(\mathbf{x}) = \mathcal{T}(\mathbf{x}) - \mathbf{x}$. The Newton method (see also chapter 5) is based on the iterations

$$\mathbf{x}^{k+1} = \mathbf{x}^k - J_{\mathcal{R}}^{-1}(\mathbf{x}^k)\mathcal{R}(\mathbf{x}^k), \quad (4.4)$$

where $J_{\mathcal{R}}(\mathbf{x}^k)$ is the Jacobian of \mathcal{R} computed at point \mathbf{x}^k . Often the Jacobian is costly or difficult to compute. The quasi-Newton method is based on the idea of approximating $J_{\mathcal{R}}^{-1}(\mathbf{x}^k)$ by a suitable and efficient operator \tilde{J} . Here we seek an approximation of the Jacobian which is equal to a scalar times the identity matrix.

4.2.2 Scalar Aitken method

If the problem is scalar ($n = 1$) and linear, the Newton method converges (obviously) in only one iteration (to, say, x^*) independently from the initial guess x^0 . Moreover the Jacobian is

4.2. THE ROLE OF AITKEN EXTRAPOLATION

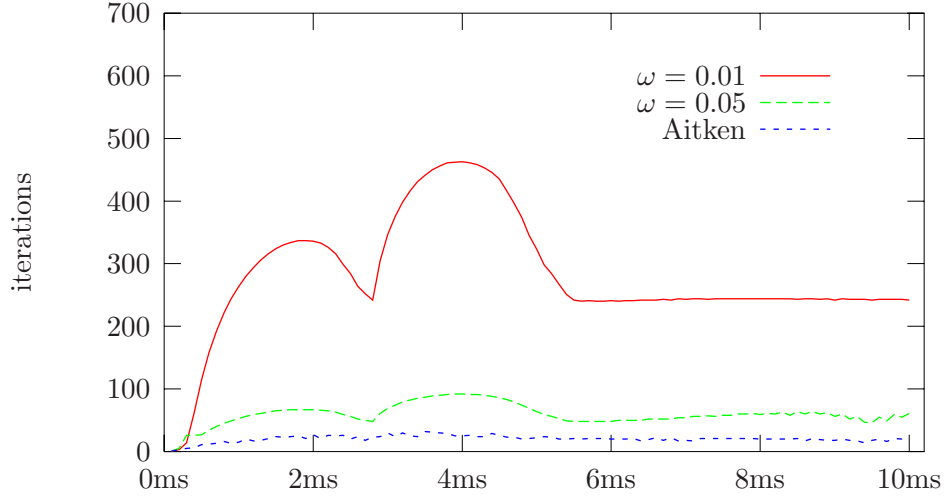


Figure 4.2: Number of iterations with a tolerance of 10^{-4} (ms stands for milliseconds). The results refers to the example in section 4.1.2 on page 81.

a constant (call ω its inverse), hence for two (possibly unrelated) points x^{k-1} and x^k it holds that

$$\begin{aligned} x^* &= x^{k-1} - \omega \mathcal{R}(x^{k-1}), \\ x^* &= x^k - \omega \mathcal{R}(x^k). \end{aligned} \tag{4.5}$$

This can be interpreted as a system of equations in (ω, x^*) , whose solution is

$$\begin{aligned} \omega &= \frac{x^k - x^{k-1}}{\mathcal{R}(x^k) - \mathcal{R}(x^{k-1})}, \\ x^* &= x^k - \omega \mathcal{R}(x^k). \end{aligned} \tag{4.6}$$

If our problem is still scalar but not linear, the last equality does not hold but can be used to find a new iterate x^{k+1} . Then for a scalar non-linear problem the Aitken acceleration method reads:

1. Choose an initial guess x^0 and an initial relaxation parameter ω^0 . Set $k = 0$ and compute $\mathcal{R}(x^0)$;
2. Set $x^{k+1} = x^k - \omega^k \mathcal{R}(x^k)$;
3. Compute $\mathcal{R}(x^{k+1})$;
4. If $\|\mathcal{R}(x^{k+1})\| < \text{Tol}$, then exit. Otherwise:
5. Compute $\omega^{k+1} = \frac{x^{k+1} - x^k}{\mathcal{R}(x^{k+1}) - \mathcal{R}(x^k)}$;
6. increase k and go to 2.

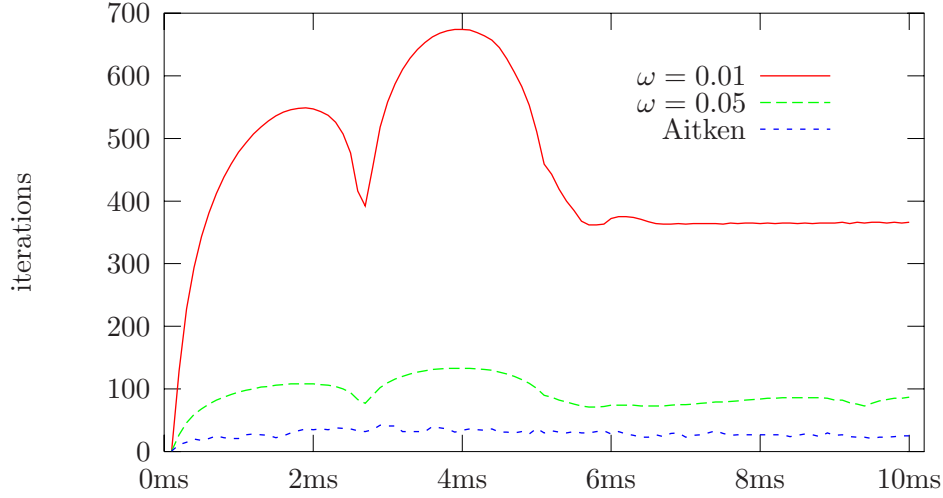


Figure 4.3: Number of iterations with a tolerance of 10^{-5} (ms stands for milliseconds). The results refers to the example in section 4.1.2 on page 81.

In [QSS00, 6.6] it is proved that if a given fixed point iteration is convergent of order $p \geq 1$ and the root has multiplicity 1, then the method 4-6 is of order $\max(2, 2p - 1)$ and in particular, if $p = 1$ (in which case the un-relaxed version may even not converge), this method is quadratically convergent. If the root has multiplicity $m \geq 2$ and the underlying fixed point iteration has order 1, then the Aitken method converges linearly with convergence factor $C = 1 - 1/m$.

4.2.3 Extension to the vector case

Suppose again that \mathcal{R} is linear. As in the scalar case the Newton method converges in only one iteration and the Jacobian is a constant matrix. We approximate the inverse of the Jacobian by a scalar times the identity, ωId , then equation (4.5) may be written as

$$\begin{aligned} \mathbf{x}^{*,k-1} &= \mathbf{x}^{k-1} - \omega \mathcal{R}(\mathbf{x}^{k-1}), \\ \mathbf{x}^{*,k} &= \mathbf{x}^k - \omega \mathcal{R}(\mathbf{x}^k). \end{aligned} \tag{4.7}$$

This system of equations is not well defined, hence ω must be recovered using another technique, for example by least squares:

$$\begin{aligned} \omega^k &= \arg \min_{\omega} \left\| \mathbf{x}^{*,k} - \mathbf{x}^{*,k-1} \right\|^2 \\ &= \arg \min_{\omega} \left\| \left(\mathbf{x}^k - \mathbf{x}^{k-1} \right) - \omega \left(\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1}) \right) \right\|^2 \end{aligned} \tag{4.8}$$

This problem can be written as

$$\omega^k = \arg \min_{\omega} \left\| \mathbf{b} - \mathbf{a}\omega \right\|^2, \tag{4.9}$$

where $\mathbf{b} = (\mathbf{x}^k - \mathbf{x}^{k-1})$ and $\mathbf{a} = (\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1}))$, and it is equivalent to solving

$$\mathbf{a} \cdot \mathbf{a}\omega = \mathbf{a} \cdot \mathbf{b} \tag{4.10}$$

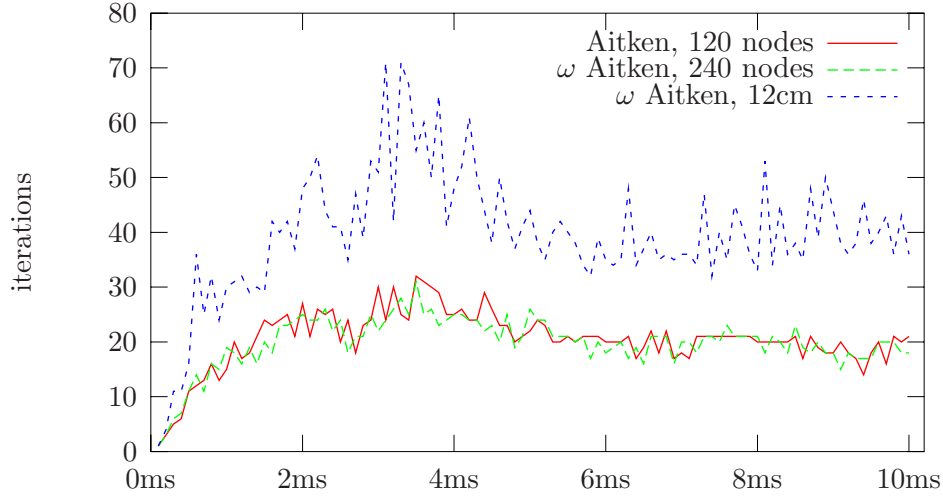


Figure 4.4: Number of iterations with a tolerance of 10^{-4} (ms stands for milliseconds). The results refers to the example in section 4.1.2 on page 81 and are the comparison between (in the same order as in the legend) 6cm tube length and 120 nodes on the interface, 6cm and 240 nodes, and 12cm and 240 nodes.

Hence the solution of (4.8) is given by

$$\omega^k = \frac{(\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})) \cdot (\mathbf{x}^k - \mathbf{x}^{k-1})}{\|\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})\|^2} \quad (4.11)$$

and the scalar-case algorithm may be applied with the new definition of ω^k given by (4.11) with the addition of a test to avoid that ω^k is too close to zero.

Example. We applied this method to the example in section 4.1.2 on page 81 and the number of iteration decreases as showed in figures 4.2 and 4.3. We remark that the effects on the number of iterations of a smaller tolerance are amplified in the constant relaxation approach. The number of iteration is affected by the length of the tube but not by the number of nodes (see figure 4.4). □

4.2.4 Minimizing on ω^{-1}

The nature of the minimization in (4.8) allows to change the unknown ω with ω^{-1} ,

$$(\omega^k)^{-1} = \arg \min_{\omega^{-1}} \left\| \omega^{-1} (\mathbf{x}^k - \mathbf{x}^{k-1}) - (\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})) \right\|^2,$$

which leads to the equation

$$\omega^k = \frac{\|\mathbf{x}^k - \mathbf{x}^{k-1}\|^2}{(\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})) \cdot (\mathbf{x}^k - \mathbf{x}^{k-1})} \quad (4.12)$$

and the algorithm may be applied with the new definition of ω^k given by (4.12). In our tests, the results are the same as those obtained by using (4.11).

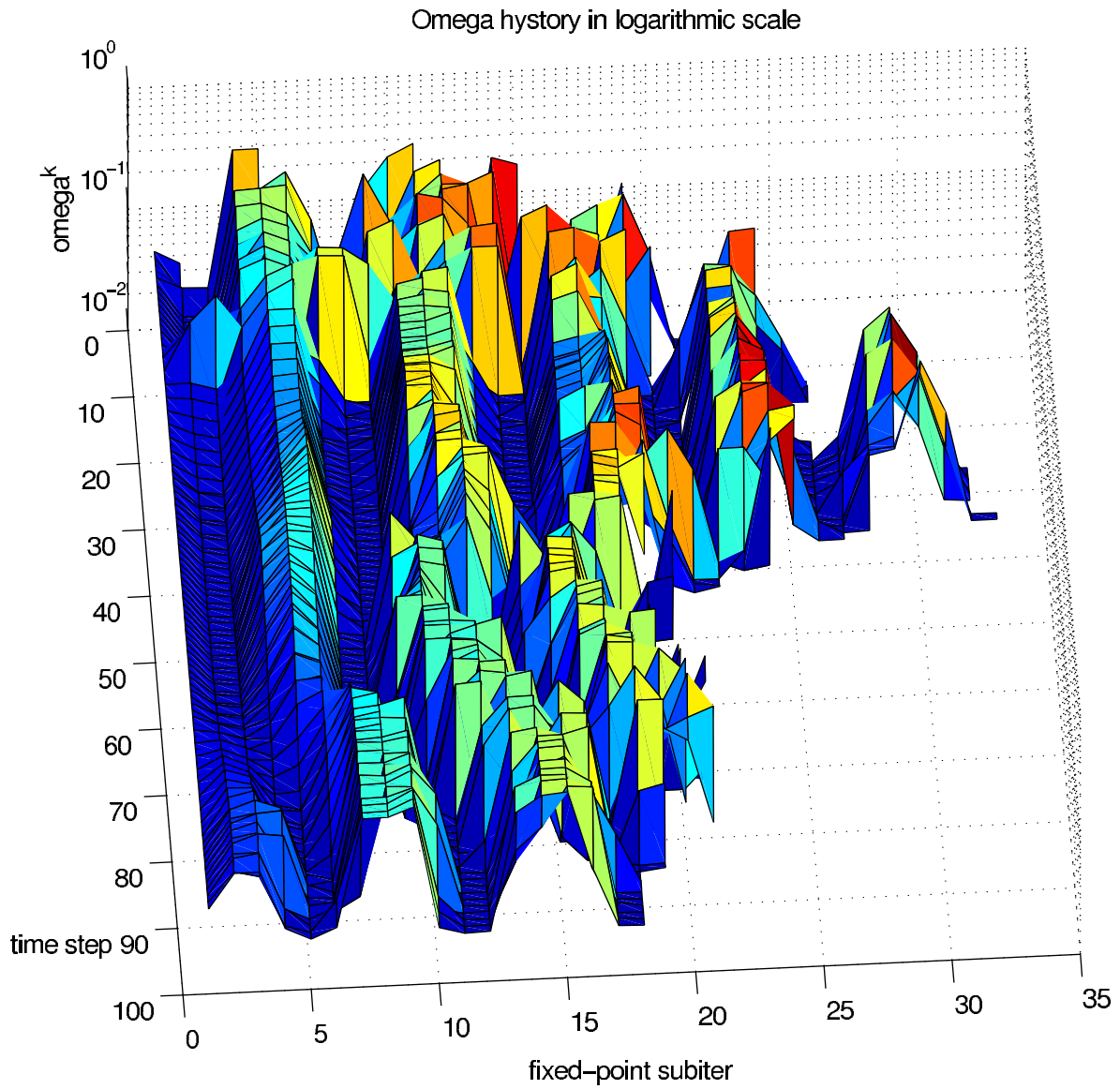


Figure 4.5: ω^k versus k with a tolerance of 10^{-5} . The results refers to the example in section 4.1.2 on page 81.

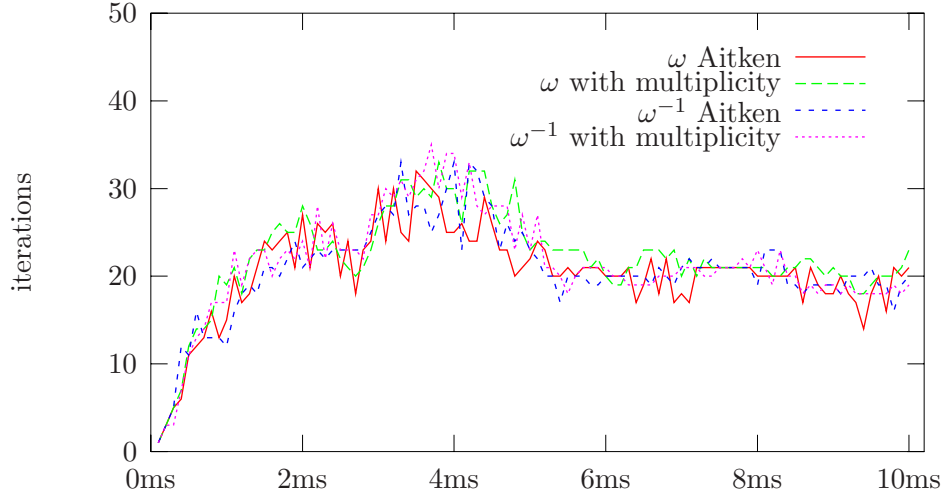


Figure 4.6: Number of iterations with a tolerance of 10^{-4} (ms stands for milliseconds). The total number of iterations are: 2054 for the original Aitken method, 2215 with the addition of the multiplicity, 2085 for the minimization on the inverse and 2153 with the addition of the multiplicity. The results refers to the example in section 4.1.2 on page 81 and the multiplicity of the root is 1.

4.2.5 Approximating the multiplicity

In the scalar case, we can use a guess of the multiplicity of a root defined as follows (see [QSS00, 6.6]),

$$m^k = \frac{\mathcal{T}(x^k) - \mathcal{T}(x^{k-1})}{\mathcal{R}(x^k) - \mathcal{R}(x^{k-1})} = 1 + \omega^k,$$

where we recall that $\mathcal{R}(x) = \mathcal{T}(x) + x$. In fact for $k \rightarrow \infty$, m^k converges to the multiplicity of the root. We can generalize this equation to n dimensions as

$$m^k = \frac{(\mathcal{T}(\mathbf{x}^k) - \mathcal{T}(\mathbf{x}^{k-1})) \cdot (\mathcal{T}(\mathbf{x}^k) - \mathcal{T}(\mathbf{x}^{k-1}))}{(\mathcal{T}(\mathbf{x}^k) - \mathcal{T}(\mathbf{x}^{k-1})) \cdot (\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1}))},$$

and we replace ω^k with $m^k \omega^k$ in the algorithm.

The multiplicity of the root in the example under consideration is one and is therefore not surprising that the number of sub-iterations needed by this variants is equivalent to that of the original one (see figures 4.6 and 4.7).

4.2.6 Variants

We have tested other simple variants to accelerate the convergence of our fluid-structure BGS algorithm, however in general the results are disappointing. Yet, we present them here to show that the choice of the relaxation parameter is quite delicate.

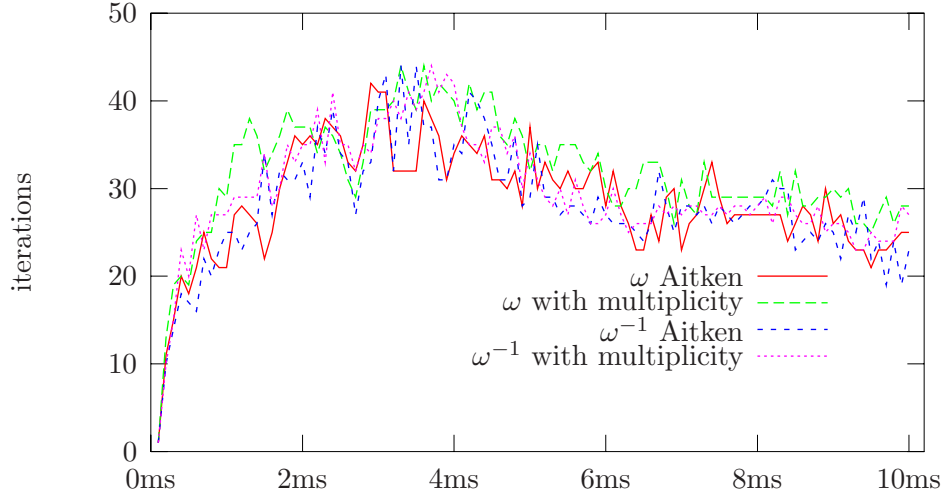


Figure 4.7: Number of iterations with a tolerance of 10^{-5} (ms stands for milliseconds). The total number of iterations are: 2864 for the original Aitken method, 3195 with the addition of the multiplicity, 2828 for the minimization on the inverse and 2972 with the addition of the multiplicity. The results refers to the example in section 4.1.2 on page 81 and the multiplicity of the root is 1.

Diagonal relaxation

The inverse of the Jacobian is approximated by a block diagonal matrix,

$$J_{\mathcal{R}}^{-1}(\mathbf{x}^k) \sim \begin{bmatrix} \omega_1 \text{Id} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_N \text{Id} \end{bmatrix},$$

where N is the number of blocks and is smaller than n (the number of nodes on the interface). Each block represents a piece of interface.

The resulting scheme splits each block into independent sections and the least square leads to

$$\boldsymbol{\omega}^k = \arg \min_{\boldsymbol{\omega}} \left\| \begin{pmatrix} \mathbf{x}_1^k - \mathbf{x}_1^{k-1} \\ \vdots \\ \mathbf{x}_1^k - \mathbf{x}_1^{k-1} \end{pmatrix} - \begin{bmatrix} \omega_1 \text{Id} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_N \text{Id} \end{bmatrix} \begin{pmatrix} \mathcal{R}_1(\mathbf{x}^k) - \mathcal{R}_1(\mathbf{x}^{k-1}) \\ \vdots \\ \mathcal{R}_1(\mathbf{x}^k) - \mathcal{R}_1(\mathbf{x}^{k-1}) \end{pmatrix} \right\|^2,$$

which can be solved component by component for $j = 1, \dots, N$

$$\omega_j = \frac{(\mathcal{R}_j(\mathbf{x}^k) - \mathcal{R}_j(\mathbf{x}^{k-1})) \cdot (\mathbf{x}_j^k - \mathbf{x}_j^{k-1})}{\|\mathcal{R}_j(\mathbf{x}^k) - \mathcal{R}_j(\mathbf{x}^{k-1})\|^2},$$

i.e., the minimizing parameters are given by equation (4.11), with the full vectors replaced by the block corresponding to the computed parameter.

If the blocks are reduced to single components, this method is equivalent to applying the scalar Aitken relaxation component by component.

We tried this scheme with a number of blocks going from 2 to n in the example presented in section 4.1.2 but the scheme diverges.

Block relaxation

Another option is to express the inverse of the Jacobian as a block matrix whose blocks are a scalar times the identity and each block has the same size. If we split the Jacobian into N blocks but the dimension n of the problem is not a multiple of N , then it is possible to replicate some of the components, e.g.,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_1 \end{pmatrix}.$$

The maximal number of blocks is \sqrt{n} . The inverse of the Jacobian is approximated in the following way,

$$J_{\mathcal{R}}^{-1}(\mathbf{x}^k) \sim \begin{bmatrix} \omega_{1,1} \text{Id} & \cdots & \omega_{1,N} \text{Id} \\ \vdots & \ddots & \vdots \\ \omega_{N,1} & \cdots & \omega_{N,N} \text{Id} \end{bmatrix}$$

and we have to solve

$$\omega^k = \arg \min_{\omega} \left\| \begin{pmatrix} \mathbf{x}_1^k - \mathbf{x}_1^{k-1} \\ \vdots \\ \mathbf{x}_1^k - \mathbf{x}_1^{k-1} \end{pmatrix} - \begin{bmatrix} \omega_{1,1} \text{Id} & \cdots & \omega_{1,N} \text{Id} \\ \vdots & \ddots & \vdots \\ \omega_{N,1} & \cdots & \omega_{N,N} \text{Id} \end{bmatrix} \begin{pmatrix} \mathcal{R}_1(\mathbf{x}^k) - \mathcal{R}_1(\mathbf{x}^{k-1}) \\ \vdots \\ \mathcal{R}_1(\mathbf{x}^k) - \mathcal{R}_1(\mathbf{x}^{k-1}) \end{pmatrix} \right\|^2.$$

To further describe this case, we need to express the vector \mathbf{b} and matrix A introduced before as block matrices:

$$\begin{aligned} \mathbf{b}_j &= \mathbf{x}_j^k - \mathbf{x}_j^{k-1} \quad j = 1, \dots, N, \\ A &= [\mathcal{R}_1(\mathbf{x}^k) - \mathcal{R}_1(\mathbf{x}^{k-1}), \dots, \mathcal{R}_N(\mathbf{x}^k) - \mathcal{R}_N(\mathbf{x}^{k-1})] \end{aligned}$$

where \mathbf{x}_j and $\mathcal{R}_j(\mathbf{x})$ are block column vectors. We can rewrite the minimization as

$$\omega^k = \arg \min_{\omega} \left\| \begin{pmatrix} \mathbf{b}_1 - A \begin{pmatrix} \omega_{1,1} \\ \vdots \\ \omega_{1,N} \end{pmatrix}, \dots, \mathbf{b}_N - A \begin{pmatrix} \omega_{N,1} \\ \vdots \\ \omega_{N,N} \end{pmatrix} \end{pmatrix} \right\|^2,$$

which leads to N independent problems: for $j = 1, \dots, N$ solve

$$A^T A \begin{pmatrix} \omega_{j,1} \\ \vdots \\ \omega_{j,N} \end{pmatrix} = A^T \mathbf{b}_j,$$

where in $A^T A$ and $A^T \mathbf{b}_j$ the blocks are multiplied with a scalar product. Since in some cases $A^T A$ may have some eigenvalues equal to zero, it is preferable to solve this problem by an iterative method such as conjugate gradient or MINRES [VdV03, 5, 6 and 10].

In contrast to the diagonal relaxation, this method preserves the dependence between each block.

Independent evaluations

We would like to mention two other strategies which however were not efficient at all in the BGS-iterations for fluid-structure interaction (with the hope that they might perform better on different non-linear problems).

The following variant has the advantage of having independent evaluations of \mathcal{R} which can be carried out in parallel. The algorithm is:

Choose two initial guesses \mathbf{x}^0 and \mathbf{x}^1 . Set $k = 1$ and repeat

$$\begin{aligned}
 & \text{compute } \mathcal{R}(\mathbf{x}^k), & \text{compute } \mathcal{R}(\mathbf{x}^{k-1}), \\
 & \text{if } \|\mathcal{R}(\mathbf{x}^k)\| < \text{Tol exit,} & \text{if } \|\mathcal{R}(\mathbf{x}^{k-1})\| < \text{Tol exit,} \\
 & \omega^{k+1} = \frac{(\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1}))^T (\mathbf{x}^k - \mathbf{x}^{k-1})}{\|\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})\|^2}, & (4.13) \\
 & \mathbf{x}^{k+2} = \mathbf{x}^k - \omega^{k+2} \mathcal{R}(\mathbf{x}^k), & \mathbf{x}^{k+1} = \mathbf{x}^{k-1} - \omega^{k+2} \mathcal{R}(\mathbf{x}^{k-1}), \\
 & k = k + 2.
 \end{aligned}$$

Non-constant relaxation

We replace ω by a linearization $\omega_m g(\mathbf{x}^k) + \omega_q$, where g is a real function to be chosen. For example define g as the difference of the evaluation index, then (4.7) becomes

$$\begin{aligned}
 \mathbf{x}^{*,k-1} &= \mathbf{x}^{k-1} - \omega_q \mathcal{R}(\mathbf{x}^{k-1}), \\
 \mathbf{x}^{*,k} &= \mathbf{x}^k - (\omega_m + \omega_q) \mathcal{R}(\mathbf{x}^k).
 \end{aligned} \tag{4.14}$$

and (5.7)

$$(\omega_m^k, \omega_q^k) = \arg \min_{(\omega_m, \omega_q)} \left\| (\mathbf{x}^k - \mathbf{x}^{k-1}) - \omega_m \mathcal{R}(\mathbf{x}^k) - \omega_q (\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})) \right\|^2. \tag{4.15}$$

Defining the matrix $A = (\mathcal{R}(\mathbf{x}^k), (\mathcal{R}(\mathbf{x}^k) - \mathcal{R}(\mathbf{x}^{k-1})))$, the relaxation parameters may be computed by solving

$$A^T A \begin{pmatrix} \omega_m \\ \omega_q \end{pmatrix} = A^T (\mathbf{x}^k - \mathbf{x}^{k-1}). \tag{4.16}$$

Then the new iterate is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\omega_m + \omega_q) \mathcal{R}(\mathbf{x}^k).$$

4.3 Transpiration interface conditions

In this section we focus on the issue accelerating the BGS iterations introduced in section 3 by saving some computations of the ALE mapping and of the fluid matrices. The standard Block-Jacobi or Block-Gauss-Seidel iterations are both CPU time consuming. Indeed, to the generally slow convergence of the algorithms we must add the cost of updating the fluid mesh, and the corresponding fluid matrices, at each iteration. We propose a modified fixed-point algorithm which combines the Block-Gauss-Seidel iterations with a transpiration formulation (see [HMY⁺93, RH93, Ren98, FFT00, Fer01, Med99]). The underlying idea of our approach relies on the fact that standard BGS iterations associated with moderate

4.3. TRANSPIRATION INTERFACE CONDITIONS

interface deformations can be treated through transpiration techniques. These formulations do not require updating of the fluid computational mesh and matrices. They only involve modifications of the interface boundary conditions.

The contents of this section have been already published in a paper by Deparis, Fernández and Formaggia, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions* [DFF03].

Each iteration of the standard BGS method (see section 4.1) involves an update of the fluid domain through the ALE mapping $\mathcal{A}_{t^{n+1}}^{k+1}$ and of its velocity $\mathbf{w}^{k+1,n+1}$. Consequently, the fluid matrices have to be recomputed in this new configuration. This is due to the circumstance that we are using an ALE formulation for the fluid (since large displacements are involved in the whole fluid-structure problem). However, between two successive BGS iterations the fluid-structure interface frequently features moderate variations.

In order to be able to solve a low cost fluid-structure problem featuring moderate deformation, aeronautical engineers have developed transpiration techniques, from an early idea of Lighthill [Lig58]. These formulations do not require to update the computational grid, but only involve modifications of the boundary conditions for the fluid at the fluid-structure interface.

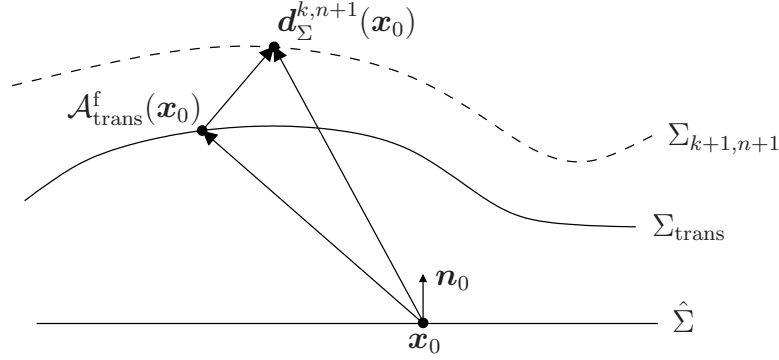


Figure 4.8: Taylor expansion of the fluid velocity.

Consider the BGS sub-iterations to solve the fluid-structure coupling at a time-step $n + 1$. We fix a (known) configuration for the fluid (for example one at a given BGS sub-iteration) and we denote it with Ω_{trans}^f . The fluid will be computed on this configuration for some BGS sub-iterations. To this configuration is associated a mapping

$$\mathcal{A}_{\text{trans}} : \hat{\Omega}^f \rightarrow \Omega_{\text{trans}}^f,$$

a velocity of the domain

$$\mathbf{w}_{\text{trans}} = \frac{\mathcal{A}_{\text{trans}} - \mathcal{A}_{t^n}}{\Delta t}$$

and an interface

$$\Sigma_{\text{trans}} = \mathcal{A}_{\text{trans}}(\hat{\Sigma}).$$

To this quantities, we associate a displacement of the interface

$$\mathbf{d}_{\Sigma}^{\text{trans}} = \mathcal{A}_{\text{trans}}|_{\hat{\Sigma}}$$

At a sub-iteration step $k + 1$, the fluid domain is given by $\Omega_{k+1,n+1}^f$ and the interface associated to it is denoted $\Sigma_{k+1,n+1}$; the ALE mapping $\mathcal{A}_{t_{n+1}}^{k+1}$ is given in terms of the structure displacement $\mathbf{d}^{k,n+1}|_{\hat{\Sigma}}$ at step k . The structure displacement and velocity $(\cdot \mathbf{d}^{k,n+1})$ are supposed to be known by the previous sub-iteration. As in the previous chapter, we denote $\mathbf{d}_{\Sigma}^{k,n+1} = \mathbf{d}^{k,n+1}|_{\hat{\Sigma}}$.

Since the two domains Ω_{trans}^f and $\Omega_{k+1,k+1}^f$ do not coincide and we want to compute the fluid solution on Ω_{trans}^f , it is necessary to derive the conditions on Σ_{trans} from the conditions imposed by the structure on Σ_{k+1} . This is possible using transpiration interface conditions, which can be derived in a heuristic way from a truncated Taylor expansion of the fluid velocity in the neighborhood of the reference fluid-structure interface Σ_{trans} (see figure 4.8). Then

$$\begin{aligned} \mathbf{u}^{k+1,n+1} \left(\mathcal{A}_{t_{n+1}}^{k+1}(\mathbf{x}_0) \right) &= \mathbf{u}^{k+1,n+1} \left(\mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \\ &+ \nabla \mathbf{u}^{k+1,n+1} \left(\mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \cdot \left(\mathcal{A}_{t_{n+1}}^{k+1}(\mathbf{x}_0) - \mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \\ &+ o \left(\left| \mathcal{A}_{t_{n+1}}^{k+1}(\mathbf{x}_0) - \mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right| \right) \end{aligned} \quad (4.17)$$

for \mathbf{x}_0 on $\hat{\Sigma}$. See [HMY⁺93, RH93, Ren98, Med99] and refer to [FFT00, Fer01, Mou02] for a more rigorous justification.

Thus, from the (semi-implicit) kinematic condition on $\hat{\Sigma}$

$$\mathbf{u}^{k+1,n+1} \circ \mathcal{A}_{t_{n+1}}^{k+1} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1},$$

we get the following transpiration condition of first order in $|\mathcal{A}_{t_{n+1}}^{k+1}(\mathbf{x}_0) - \mathcal{A}^{\text{trans}}(\mathbf{x}_0)|$ on the reference interface $\hat{\Sigma}$

$$\mathbf{u}^{k+1,n+1} \circ \mathcal{A}^{\text{trans}} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1} - \nabla \mathbf{u}^{k+1,n+1} \circ \mathcal{A}^{\text{trans}} \cdot \left(\mathbf{d}_{\Sigma}^{k,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}} \right),$$

or equivalently on the fixed interface Σ_{trans}

$$\mathbf{u}^{k+1,n+1} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1} \circ (\mathcal{A}^{\text{trans}})^{-1} - \nabla \mathbf{u}^{k+1,n+1} \cdot \left(\mathbf{d}_{\Sigma}^{k,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}} \right) \circ (\mathcal{A}^{\text{trans}})^{-1}.$$

The implicit dependence on the gradient $\nabla \mathbf{u}^{k+1,n+1}$ can be made explicit by modifying the relation into

$$\mathbf{u}^{k+1,n+1} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1} \circ (\mathcal{A}^{\text{trans}})^{-1} - \nabla \mathbf{u}^{k,n+1} \cdot \left(\mathbf{d}_{\Sigma}^{k,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}} \right) \circ (\mathcal{A}^{\text{trans}})^{-1} \quad (4.18)$$

on Σ_{trans}

This latter condition can now be used to approximate the fluid subproblem P3.7.

Moreover, we have alternative options for the convective term \mathbf{u}^* as well. Indeed, it can be updated at every sub-iteration as in the classic BGS algorithm, or frozen. This last choice is more appropriate in the frame of the transpiration strategy, as it prevents from recomputing the fluid matrices.

4.3. TRANSPIRATION INTERFACE CONDITIONS

P4.1 (Transpired fluid) Find $(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1})$ in $V(\Omega_{\text{trans}}^f)Q(\Omega_{\text{trans}}^f)$ such that

$$\left\{ \begin{array}{l} \frac{1}{\Delta t} \int_{\Omega_{\text{trans}}^f} \rho_f \mathbf{u}^{k+1,n+1} \cdot \mathbf{v} + \int_{\Omega_{\text{trans}}^f} \rho_f (\mathbf{u}^* - \mathbf{w}^{\text{trans}}) \cdot \nabla \mathbf{u}^{k+1,n+1} \cdot \mathbf{v} \\ - \int_{\Omega_{\text{trans}}^f} \rho_f \mathbf{u}^{k+1,n+1} \cdot \mathbf{v} \operatorname{div} \mathbf{w}^{\text{trans}} - \int_{\Omega_{\text{trans}}^f} p^{k+1,n+1} \operatorname{div} \mathbf{v} + \int_{\Omega_{\text{trans}}^f} 2\mu \epsilon(\mathbf{u}^{k+1,n+1}) \cdot \epsilon(\mathbf{v}) \\ = \frac{1}{\Delta t} \int_{\Omega_{t^n}^f} \rho_f \mathbf{u}^n \cdot \mathbf{v} + \int_{\Gamma_{\text{trans}}^{\text{in}} \cup \Gamma_{\text{trans}}^{\text{out}}} \mathbf{g} \cdot \mathbf{v} ds, \\ \int_{\Omega_{\text{trans}}^f} q \operatorname{div} \mathbf{u}^{k+1,n+1} = 0, \end{array} \right. \quad (4.19)$$

for all (\mathbf{v}, q) in $V_{\Sigma}(\Omega_{\text{trans}}^f)Q(\Omega_{\text{trans}}^f)$, with the following boundary condition for $\mathbf{u}^{k+1,n+1}$ on Σ_{trans} :

$$\mathbf{u}^{k+1,n+1} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1} \circ (\mathcal{A}^{\text{trans}})^{-1} - \nabla \mathbf{u}^{k,n+1} \cdot \left(\mathbf{d}_{\Sigma}^{k,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}} \right) \circ (\mathcal{A}^{\text{trans}})^{-1}. \quad (4.20)$$

The spaces V , \mathbf{v}_{Σ} and Q are defined in (3.1), (3.2) and (3.3) on page 63.

We formally denote the solver related to this problem by

$$\left(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1} \right) = \mathcal{F}_{\mathbf{u}^*}^{\text{trans}} \left(\mathbf{d}_{\Sigma}^{k+1,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k+1,n+1} \right).$$

The obtained fluid-subproblem allow us to take into account the interface motion, while keeping a fixed fluid domain. This is achieved by using non-standard boundary conditions on the fixed reference interface Σ_{trans} without the need of updating the mesh.

In the same way, the fluid stress at the moving interface can be recovered from a similar Taylor expansion. Let $\boldsymbol{\sigma}_f = p\mathbf{n} - 2\mu\epsilon(\mathbf{u})$, then

$$\begin{aligned} \boldsymbol{\sigma}_f^{k+1,n+1} \left(\mathcal{A}_{t^{n+1}}^{k+1}(\mathbf{x}_0) \right) &= \boldsymbol{\sigma}_f^{k+1,n+1} \left(\mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \\ &+ \nabla \boldsymbol{\sigma}_f^{k+1,n+1} \left(\mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \cdot \left(\mathcal{A}_{t^{n+1}}^{k+1}(\mathbf{x}_0) - \mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right) \\ &+ o \left(\left| \mathcal{A}_{t^{n+1}}^{k+1}(\mathbf{x}_0) - \mathcal{A}^{\text{trans}}(\mathbf{x}_0) \right| \right) \end{aligned} \quad (4.21)$$

on $\hat{\Sigma}$. Thus the subproblem P3.8 for the vessel structure can be replaced by the following one,

P4.2 (Transpired structure) Find $(\mathbf{d}^{k+1,n+1}, \dot{\mathbf{d}}^{k+1,n+1})$ in $X(\hat{\Omega}^s)L^2(\hat{\Omega}^s)$ such that for all $\hat{\varphi}$ in $X(\hat{\Omega}^s)$,

$$\begin{aligned} \frac{1}{\Delta t} \int_{\hat{\Omega}^s} \rho_s \left(\dot{\mathbf{d}}^{k+1,n+1} - \dot{\mathbf{d}}^n \right) \cdot \hat{\varphi} d\hat{\mathbf{x}} &+ \frac{1}{2} \left(a_s \left(\mathbf{d}^{k+1,n+1}, \hat{\mathbf{v}} \right) + a_s \left(\mathbf{d}^n, \hat{\varphi} \right) \right) = \\ \int_{\Sigma_{\text{trans}}} \left[\left(\boldsymbol{\sigma}_f^{k+1,n+1} + \nabla \boldsymbol{\sigma}_f^{k+1,n+1} \cdot \left(\mathbf{d}^{k,n+1} - \mathbf{d}^{\text{trans}} \right) \circ (\mathcal{A}^{\text{trans}})^{-1} \right) \cdot \mathbf{n} \right] \cdot \hat{\varphi} \circ (\mathcal{A}^{\text{trans}})^{-1} ds, \\ \frac{\mathbf{d}^{k+1,n+1} - \mathbf{d}^n}{\Delta t} &= \frac{\dot{\mathbf{d}}^{k+1,n+1} + \dot{\mathbf{d}}^n}{2}. \end{aligned}$$

The bilinear form $a_s(\cdot, \cdot)$ is defined in problem P3.5 on page 63. We formally denote the solver related to this problem by

$$\left(\mathbf{d}^{k+1,n+1}, \dot{\mathbf{d}}^{k+1,n+1}\right) = \mathcal{S}^{\text{trans}}\left(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1}\right).$$

A simpler approximation, see [Med99, FFT00], can be obtained by replacing the first order Taylor expansions (4.17) and (4.21) by zeroth order expressions. In that case, the interface transpiration condition in (4.19) reduces to

$$\mathbf{u}^{k+1,n+1} = \dot{\mathbf{d}}_{\Sigma}^{k,n+1} \circ (\mathcal{A}^{\text{trans}})^{-1} \quad (4.18\text{bis})$$

and the fluid interface stress in (4.21) to

$$\boldsymbol{\sigma}_f^{k+1,n+1}\left(\mathcal{A}_{t^{n+1}}^{k+1}(\mathbf{x}_0)\right) = \boldsymbol{\sigma}_f^{k+1,n+1}\left(\mathcal{A}^{\text{trans}}(\mathbf{x}_0)\right). \quad (4.21\text{bis})$$

on $\hat{\Sigma}$.

By exploiting the previous considerations we have derived the modified BGS algorithm reported in figure 4.9. The boxes on the right column of figure 4.9 represent the transpiration loop, which is also described at the end of this section. Here, instead of updating fluid mesh and matrices, we just enforce the transpiration velocity

$$\dot{\mathbf{d}}_{\Sigma}^{k,n+1} \circ (\mathcal{A}^{\text{trans}})^{-1} - \nabla \mathbf{u}^{k,n+1} \cdot \left(\mathbf{d}_{\Sigma}^{k,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}}\right) \circ (\mathcal{A}^{\text{trans}})^{-1}$$

at the interface. Tolerances $\text{Tol}_{\text{trans}}^{\text{in}}$ and $\text{Tol}_{\text{trans}}^{\text{out}}$ define the range of relative interface displacements where the transpiration formulation will be used. The convergence test of the whole algorithm is always made after two standard BGS iterations, (2x in the figure), in order to ensure the convergence to the original coupled problem P3.7-P3.7-P3.8. This also implies that the algorithm terminates with standard BGS iterations and with an updated mesh.

4.3.1 Confidence interval

In order to test whether to activate the transpiration part of the algorithm, the relative error described in section 4.1.1 on page 79 is useless. Indeed, what we have to measure in this case is how much the computational fluid domain is distant from the actual fluid domain. The transpiration may be adopted only when this distance is small. Hence, the condition that has to be satisfied is

$$\frac{\left\|\mathbf{d}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}}\right\|}{L_k^{\text{ref}}} < \text{Tol}_{\text{trans}}, \quad (4.22)$$

where L_k^{ref} is a characteristic “length” of the fluid domain at the k -th iteration. For blood fluid dynamics L_k^{ref} can be taken as the mean of $|R + \mathbf{d}^{k+1,n+1}|$ over the interface points.

4.3.2 Description of the algorithm

The part of the algorithm concerning pure BGS iterations is presented in section 4.1. At the end of one BGS iteration, if convergence is not achieved, we proceed as follows:

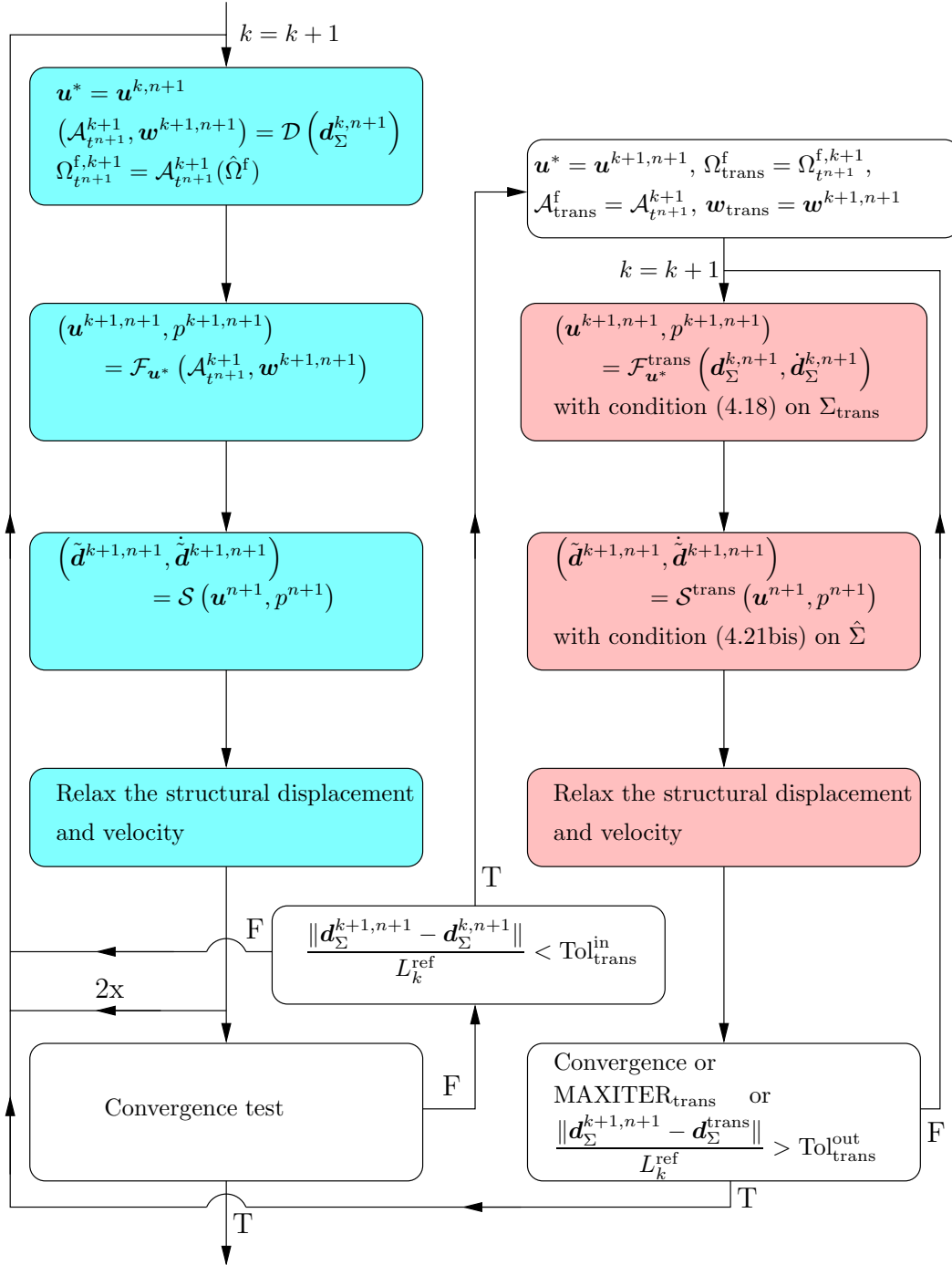


Figure 4.9: Diagram of the proposed algorithm. On the left the BGS part, on the right the transpiration steps.

1. If the displacement of the domain is small enough,

$$\frac{\|\mathbf{d}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{k,n+1}\|}{L_k^{\text{ref}}} < \text{Tol}_{\text{trans}}^{\text{in}},$$

we can enter the transpiration loops,

2. Freeze the domain and the linearization \mathbf{u}^* ,

$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}^{k+1,n+1}, \\ \Omega_{\text{trans}}^f &= \Omega_{t^{n+1}}^{f,k+1}, \quad \Sigma_{\text{trans}} = \Sigma_{t^{n+1}}^{k+1}, \\ \mathcal{A}_{\text{trans}} &= \mathcal{A}_{t^{n+1}}^{k+1}, \quad \mathbf{w}_{\text{trans}} = \mathbf{w}^{k+1,n+1}; \end{aligned}$$

3. Solve the problem P4.1 with condition (4.17) on Σ_{trans} ,

$$\left(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1} \right) = \mathcal{F}_{\mathbf{u}^*}^{\text{trans}} \left(\mathbf{d}_{\Sigma}^{k,n+1}, \dot{\mathbf{d}}_{\Sigma}^{k,n+1} \right);$$

4. Solve the structure problem P4.2 with condition (4.21) on Σ_{trans} ,

$$\left(\tilde{\mathbf{d}}^{k+1,n+1}, \dot{\tilde{\mathbf{d}}}^{k+1,n+1} \right) = \mathcal{S}^{\text{trans}} \left(\mathbf{u}^{k+1,n+1}, p^{k+1,n+1} \right)$$

5. Chose $\omega \in (0, 1)$ and relax the structure's displacement and velocity

$$\mathbf{d}_{\Sigma}^{k+1,n+1} = (1 - \omega) \mathbf{d}_{\Sigma}^{k,n+1} + \omega \tilde{\mathbf{d}}_{\Sigma}^{k+1,n+1},$$

6. If convergence is achieved, exit the transpiration loop, go to the standard BGS iterations and if this has also converged, impose an extra loop to the BGS. If the maximum number of transpiration loops is reached or if the reference domain Ω^{trans} is not accurate enough, i.e.,

$$\frac{\|\mathbf{d}_{\Sigma}^{k+1,n+1} - \mathbf{d}_{\Sigma}^{\text{trans}}\|}{L_k^{\text{ref}}} > \text{Tol}_{\text{trans}}^{\text{out}},$$

return to the standard BGS iterations. Otherwise set $k = k + 1$ and go to (3).

4.4 Numerical experiments

4.4.1 Two-dimensional test

We have applied the above algorithm to a fluid-structure problem arising in the modeling of blood flow on large arteries, precisely on a thin elastic tube conveying an incompressible viscous fluid. In order to simplify the problem we considered the axisymmetric incompressible Navier-Stokes equations without rotation (see chapter 2) combined with a generalized string model (see [Nob01]) for the structure. The test is the same carried out on the example on page 81.

We have adopted axisymmetric P1isoP2/P1 finite elements for the fluid and P1 for the structure. The time is discretized by a mid-point scheme for the structure and implicit Euler

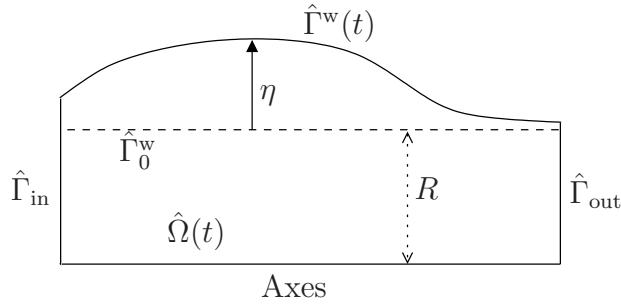


Figure 4.10: The computational domain.

Standard	1
Order 0	0.65
Order 1	0.63

Table 4.1: Normalized CPU time w.r.t. standard BGS , zeroth or first order transpiration schemes.

for the fluid equations (see chapter 3), with a time step of $\Delta t = 0.1$ ms, a tolerance of 10^{-4} and a fixed point with Aitken relaxation.

We have used the simplified form (4.21bis) for the forcing term and the following values for the tolerances in the proposed numerical scheme: $\text{Tol} = 10^{-6}$ for the fixed point with reference displacement equal to 10% of the initial radius of the artery, $\text{Tol}_{\text{trans}}^{\text{in}} = 0.05$, $\text{Tol}_{\text{trans}}^{\text{out}} = 0.1$ and $\text{MAXITER}_{\text{trans}} = 50$. We take as characteristic length of the domain the initial radius of the artery, $L_k^{\text{ref}} = R$.

In figure 4.11 we report the number of sub-iterations per time step required by the standard BGS method compared with the one obtained using the modified BGS scheme with transpiration. The number of BGS iterations is strongly reduced in the transpiration version (see figure 4.11, *1st order: BGS sub-iter*). Let us notice that at each time step, the number of outer iterations is almost equal in the two schemes. However, the computing time is greatly reduced: a gain of 40% over 240 time steps. The proposed algorithm does not introduce any loss of accuracy. Indeed, as mentioned above, the converged solution provided by our algorithm coincides with one iteration of the standard BGS method.

We have also tested the zeroth order formulation with (4.18bis).

The CPU time and the number of iterations are of the same order (see table 4.4.1). The slight difference in CPU time derives from the computation of the fluid velocity gradients. The fact that the convergence obtained with the two alternatives (zeroth and first order approximations) is similar, is due to the limited contribution of the velocity gradients for this test case. Indeed, the additional contribution given by the first order scheme is only 10^{-7} times the zeroth order term. This follows from the very little variations in the wall displacement between the first two BGS iterations and the following transpiration ones.

Remark

It is possible to improve the efficiency (in terms of CPU time) of the transpiration algorithm by updating the fluid domain only once per time steps, at the beginning or after the first time

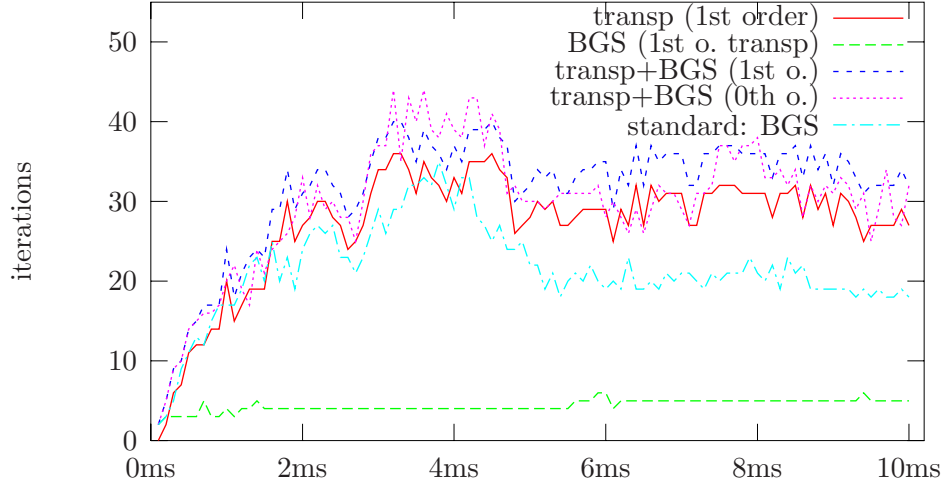


Figure 4.11: Iterations history.

that convergence is achieved. It is also possible to update the domain only every n time steps, provided that the displacement is in the confidence interval defined by (4.22). However, the gain in time of this approach is very small and does not justify the loss of accuracy introduced. In figure 4.12 it is possible to see the error introduced in the wall displacement in the test case.

The superiority of the first order transpiration condition with respect to the zeroth one can be better appreciated when the difference between the transpiration domain and the actual fluid domain are more significant. For example, if the fluid mesh is updated only at the beginning of the time step or every 5 time steps (see figure 4.12), the wall displacement is better reproduced by the first order condition.

4.4.2 Three-dimensional test

Here we present another test in three dimension to show that the performance of algorithm presented in this chapter is independent from the dimension of the problem. We couple the incompressible three-dimensional Navier-Stokes equations (for the fluid) with an independent ring model (for the wall displacement) in a cylindrical domain (see figure 4.13).

On the inlet we impose a “pressure pulse” of period of 5 ms,

$$\sigma(u, p)n = -\frac{P_{\text{in}}}{2} \left[1 - \cos\left(\frac{2\pi t}{5}\right) \right] n,$$

with $P_{\text{in}} = 1310^3$ dynes/cm².

The independent ring model reads

$$\rho_w h \frac{\partial \eta^2}{\partial t^2} + \frac{Eh}{(1 - \nu^2) R^2} \eta = p - p_0.$$

We use the following parameters: $R = 0.5\text{cm}$, $L = 5\text{cm}$, $\rho = 1\text{gr/cm}^3$, $\mu = 0.03\text{poise}$, $\rho_w = 1.1\text{gr/cm}^3$, $h = 0.1\text{cm}$, $E = 310^6\text{dyne/cm}^2$ and $\nu = 0.3$.

4.4. NUMERICAL EXPERIMENTS

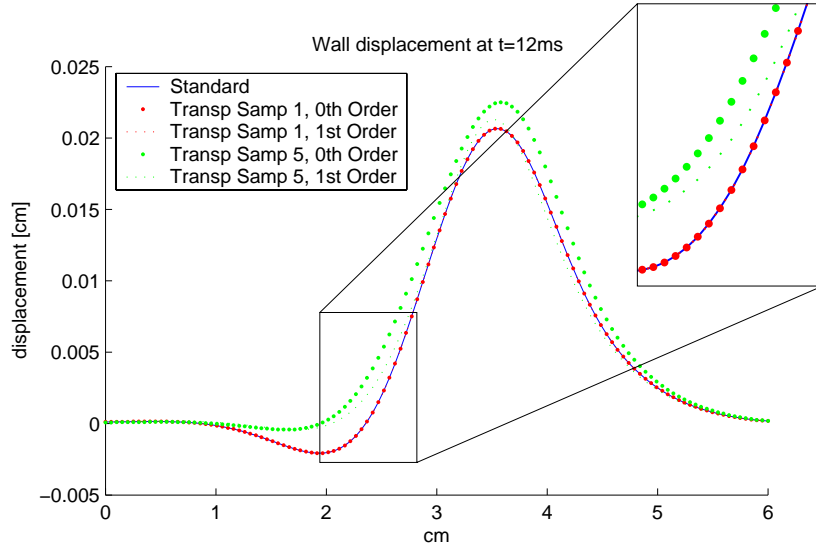


Figure 4.12: Comparison between zeroth and first order transpiration conditions. The domain is updated only at the beginning of the BGS iterations (Samp 1) or every 5 time steps (Samp 5). We have zoomed a portion of the picture to better emphasize the different curves. The enhanced accuracy of the second order transpiration condition is appreciated when the domain is updated less often.

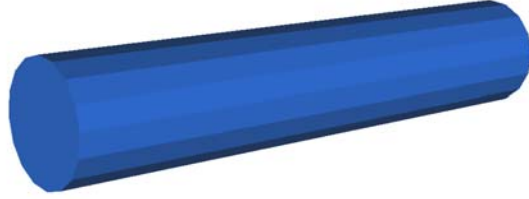


Figure 4.13: The three-dimensional cylinder.

The fluid is discretized with P1bubble/P1finite elements in space and implicit Euler in time. The structure is discretized with a mid-point scheme. We perform 150 time steps with $\Delta t = 0.1\text{ms}$. The parameters for the transpiration loop are: $\text{Tol}_{\text{trans}}^{\text{in}} = 0.05$, $\text{Tol}_{\text{trans}}^{\text{out}} = 0.1$ and $\text{MAXITER}_{\text{trans}} = 50$, with Aitken's relaxation parameter.

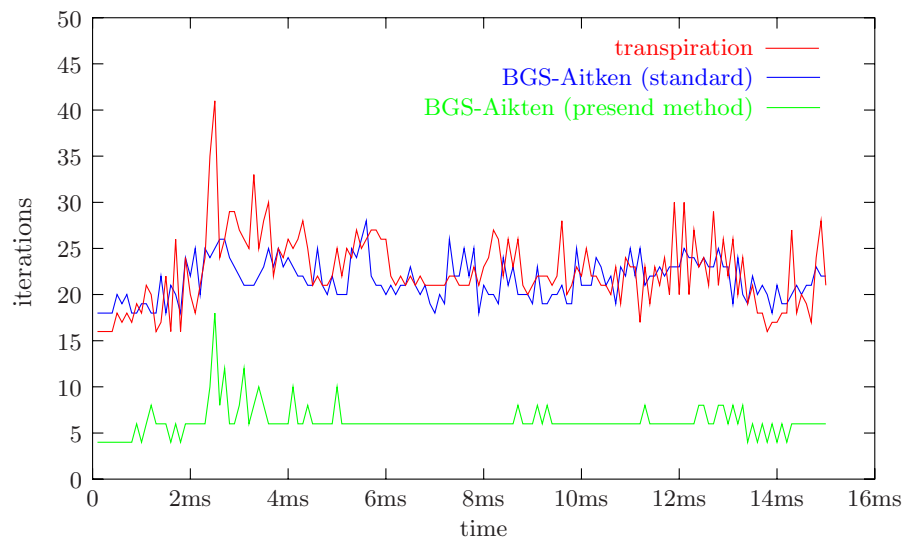


Figure 4.14: Iterations history for the three-dimensional experiment.

Chapter 5

Preconditioning of Newton and quasi-Newton algorithms for the solution of the fully implicit problem

Introduction

In this chapter, we investigate some acceleration techniques for the Newton or quasi-Newton algorithms.

Newton–Krylov [BS94] or Jacobian-free Newton–Krylov methods [KK03] are popular solution strategies for nonlinear problems in applied mathematics and computational physics. They rely on a combination of Newton-type methods with super-linearly convergence rates and Krylov subspace methods for solving the Newton correction equations. Their main core requires to solve a sequence of linear systems of type:

$$A_i \mathbf{x}_i = \mathbf{b}_i \quad (i=1,2, \dots) \quad (5.1)$$

where the coefficient matrices A_i and the right-hand sides \mathbf{b}_i are different. In this work, we investigate some linear and nonlinear acceleration techniques in the framework of Newton–Krylov or quasi-Newton–Krylov algorithms to derive an efficient and robust nonlinear solver.

Our first purpose is to accelerate the convergence of a given linear system by reusing information built during previous resolution processes. While most papers in the literature (e.g. [CW97, Par80, Saa97, SG95, SPM89]) consider the case of multiple linear systems only with different right-hand sides, at our knowledge few attempts [CN99, RR98] have addressed the general case (5.1). Moreover both approaches are restricted to a sequence of linear systems with symmetric positive definite matrices. Here the coefficient matrices A_i are only assumed to be regular and each linear system will be solved with the GMRES method [SS86]. The proposed dynamic preconditioner consists of exploiting information that is related to the sequence of Hessenberg matrices built during the successive orthogonalization procedures. We show that this new preconditioner is non-singular and well defined and we describe how to nestle more than one preconditioner.

Some aspects of our approach are similar to [Cho95, §4] and [CE96]. These authors consider a restarted GMRES(m) method and build a preconditioner for the next perturbed

system using the Krylov subspace and the Hessenberg matrix stored during the previous GMRES(m) application. They also apply the preconditioners in a nested way to solve the Navier–Stokes equations within a Newton algorithm. Although connected, the two approaches differ in the iterative method used (restarted against full GMRES) and in the stored data: even if they nestle the preconditioners, they store only one Krylov subspace, hence some informations may be lost. Here we propose to store some Krylov subspaces and the related basis associated to benefit from all the information already collected. The drawback of our approach is the amount of memory needed, however nowadays this item is loosing importance. Moreover, in the fluid-structure interaction problem presented here, the degrees of freedom concerned by the Newton–Krylov algorithm are only those located on the interface; this leads to small sized problems in practice.

We then detail a nonlinear acceleration technique for the resulting preconditioned Newton–Krylov algorithm. Previous work was done in this respect. For instance Washio et al. [WO97] and Fokkema et al. [FSvdV98] have proposed to store both the iterates and the residuals and to search for a better iterate in the affine subspace generated during the previous iterations. Washio et al. [WO97] have proposed to find the new iterate by minimizing the norm of the linearized residual. Following this approach, we remark that if the linearization of the residual is accurate enough, a new (quasi-)Newton step can be carried out even without evaluating the new residual. Indeed, we are able to define an entire acceleration step, where neither the residual nor the inverse of the Jacobian are *explicitly* computed. They are replaced by a linearization of the residual and an application of the dynamic preconditioner respectively.

The resulting algorithm is well suited to problems where the functional or its Jacobian are very expensive to compute. In particular, to build the preconditioner there is no need to explicitly build the matrices A_i (in other words, the Jacobians), but only to evaluate the matrices against given vectors in the GMRES algorithm.

The resulting solution method is then tested in the framework of fluid-structure interaction problems in haemodynamics. The numerical results for two and three-dimensional problems are very satisfactory, with a total computational gain of up to 50% in CPU time versus a standard Newton method.

5.1 Newton

For the applications we have in mind – blood flow simulations – we are interested in strongly coupled algorithms. This means that a fixed point of the map $\mathcal{T} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{D}$ presented in chapter 3 has to be found at each time step, which is equivalent as finding the root of the operator $\mathcal{R} = \mathcal{T} - Id$. Note that \mathcal{R} is a nonlinear operator restricted to the fluid-structure interface. The nonlinearities come from: The inertial term in the Navier-Stokes equations, the displacements of the fluid domain, the (large) displacements in the structure. For the test cases presented here the constitutive law of the structure is linear.

We can formulate the Newton algorithm in a general form as follows. Given a vector field on \mathbb{R}^N , we want to find an approximation \mathbf{x}^* in \mathbb{R}^N of the root of \mathcal{R} , such that $\|\mathcal{R}(\mathbf{x}^*)\| < \text{Tol}$ for a given tolerance and for the Euclidean norm.

- 1) define an initial guess \mathbf{x}_0 and set $k = 0$;
- 2) compute $\mathcal{R}(\mathbf{x}_0)$;
- 3) solve $J(\mathbf{x}_k)\delta\mathbf{x} = -\mathcal{R}(\mathbf{x}_k)$, where $J(\mathbf{x}_k)$ is the Jacobian of \mathcal{R} in \mathbf{x}_k ;

5.2. HOW TO COMPUTE THE JACOBIAN

- 4) set $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta \mathbf{x}$;
- 5) compute the residual $\mathcal{R}(\mathbf{x}_{k+1})$;
- 6) if $\|\mathcal{R}(\mathbf{x}_{k+1})\| < \text{Tol}$, then stop, otherwise increase k and go to 3.

The computational challenge is represented by the inversion of the Jacobian in step 3. When a Krylov subspace method is used, it requires many evaluations of $J(\mathbf{x}_k)$ at different points. Moreover the evaluation itself can be costly, even if sometimes it is possible to solve the system inexactly (with a relaxed accuracy). In many cases this cost can be reduced by replacing the Jacobian by a simpler and cheaper operator, as in quasi-Newton algorithms. The dynamic preconditioner defined in section 5.4 can be used to reduce the number of iterations in GMRES iterations. In section 5.5, we will propose a modified Newton algorithm that can be used to further speed up the convergence. In other cases, it is possible or even mandatory to compute the Jacobian in an exact way (see [FM04]).

5.2 How to compute the Jacobian

In the Newton algorithm, we have to solve the problem

$$J_{\mathcal{R}}(\mathbf{x}^k) \delta \mathbf{x} = -\mathcal{R}(\mathbf{x}^k), \quad (5.2)$$

Its resolution can be carried out with a GMRES method, which implies the evaluation of the Jacobian multiplied with the successive residuals.

In the fluid-structure problem introduced in chapter 3, the unknown \mathbf{x}^* must be replaced by the interface displacement $\mathbf{d}_{\Sigma}^{n+1}$ at time t^{n+1} . We note by $\mathbf{d}_{\Sigma}^{k,n+1}$ the intermediate values of the displacement given by the Newton algorithm. In the sequel we neglect the time step t^{n+1} and its index $n+1$ to simplify the notations. The exact computation of $J(\mathbf{d}^k) \mathbf{r}$ is treated in [FM03] and [FM04] and its implementation implies shape derivatives (see also section 3.3). A more simple approach is proposed in [MS00] with finite difference methods. However, this technique implies a choice of a parameter, which depends on the situation and whose choice is critical. This technique is computationally expensive, since it implies the additional computation of the residual \mathcal{R} in the direction of the derivative. An efficient solution is to approximate the Jacobian with simplified models based on the physical meaning of the problem at hand. In the following section we propose two simplified models.

In section 5.4 we introduce a preconditioner for the inversion of the Jacobian which is based on the properties of the GMRES method. Its advantage is that it is independent from the choice for the Jacobian, i.e., it can be used with an approximate Jacobian or with the exact one. At each time step, a new preconditioner with the first GMRES iterations is built and it is used from the second Newton iteration on.

5.3 How to compute approximate Jacobian

We now propose two approximate models used to evaluate the Jacobian.

FSI-QN 1

The first simplified model, which has been proposed by Gerbeau and Vidrascu [GV03], is based on the following assumptions:

- (i) The fluid domain is frozen about its current state.
- (ii) The structure equation is linearized about its current state.
- (iii) Non-linear inertial and viscous terms are neglected in the fluid. The fluid equation therefore reduces to a Poisson problem on the pressure.

FSI-QN 2

The second simplified model, which is derived from a zeroth order transpiration, is based on the following assumptions:

- (i) The fluid domain is frozen about its current state.
- (ii) The structure equation is linearized about its current state.
- (iii) The fluid equation is linearized about its current state. The fluid problem therefore reduces to an Oseen equation with a reaction term.

In both cases, we can split the computation of $J(\mathbf{d}^k)\mathbf{r}$ in two steps, which derive from the following equality,

$$J(\mathbf{d}^k)\mathbf{r} = J_S(\mathcal{F} \circ \mathcal{D}(\mathbf{d}^k)) J_{\mathcal{F} \circ \mathcal{D}}(\mathbf{d}^k) \mathbf{r},$$

where $J_{\mathcal{F} \circ \mathcal{D}}$ is the Jacobian of the fluid problem in a moving domain defined by \mathbf{d}^k and $J_S(\mathcal{F} \circ \mathcal{D}(\mathbf{d}^k))$ is the Jacobian of the structure problem related to $\mathcal{F} \circ \mathcal{D}(\mathbf{d}^k)$. Note that since the residual $\mathcal{R}(\mathbf{d}^k)$ has already been computed, so has $\mathcal{F} \circ \mathcal{D}(\mathbf{d}^k)$. Moreover, if the structure is linear we have already computed the related matrix (which we denote by DK). Otherwise, we can choose to solve the structure with a Newton–Raphson algorithm ([Ode72, CCG96, Cha85, MMV99]) and we denote by DK the tangent operator of the structure problem which is computed during this step (see also[GV03]).

FSI-QN 1

The quasi-Newton algorithm related with FSI-QN 1 reads:

Suppose that \mathcal{A}_{t^n} , (\mathbf{u}^n, p^n) , \mathbf{d}^n and $\dot{\mathbf{d}}^n$ are known at time t^n ; their update at time t^{n+1} is obtained as follows

- 1) Set $k = 0$ and extrapolate the position of the interface,

$$\mathbf{d}_{\Sigma}^k = \mathbf{d}_{\Sigma}^n + \frac{3\Delta t}{2} \dot{\mathbf{d}}_{\Sigma}^n - \frac{\Delta t}{2} \dot{\mathbf{d}}_{\Sigma}^{n-1},$$

as well as the fluid velocity $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ (and pressure $p^* = 2p^n - p^{n-1}$, in case of a fractional step);

- 2) Solve the the genuine fluid-structure problem with

$$\begin{aligned} (\mathcal{A}^{k+1}, \mathbf{w}^{k+1}) &= \mathcal{D}(\mathbf{d}_{\Sigma}^k), \\ (\mathbf{u}^{k+1}, p^{k+1}) &= \mathcal{F}_{\mathbf{u}^*}(\mathcal{A}^{k+1}, \mathbf{w}^{k+1}), \\ (\tilde{\mathbf{d}}^{k+1}, \dot{\tilde{\mathbf{d}}}^{k+1}) &= \mathcal{S}(\mathbf{u}^{k+1}, p^{k+1}); \end{aligned}$$

5.3. HOW TO COMPUTE APPROXIMATE JACOBIAN

3) Evaluation of the residual,

$$\mathcal{R}_k = \tilde{\mathbf{d}}_{\Sigma}^{k+1} - \mathbf{d}_{\Sigma}^k;$$

4) If converged (see section 4.1.1 on page 79), go to the next time step. Otherwise

5) Compute $\delta \mathbf{d}_{\Sigma}^{k+1}$ by solving the approximate tangent problem with GMRES:

$$\mathcal{R}' \delta \mathbf{d}_{\Sigma}^{k+1} = -\mathcal{R}_k.$$

The matrix \mathcal{R}' is not explicitly computed. The evaluation of the product of \mathcal{R}' by a vector \mathbf{z} (one time per GMRES iteration) is performed as follows

(a) Solve

$$\begin{aligned} \Delta \delta p &= 0 && \text{in } \Omega^{f,k+1}, \\ \frac{\partial \delta p}{\partial \mathbf{n}} &= -\frac{\rho_f}{\Delta t^2} \mathbf{z} \circ (\mathcal{A}^{k+1})^{-1} \cdot \mathbf{n} && \text{on } \Sigma^{k+1}, \\ \delta p &= 0 && \text{on } \Gamma^{\text{in},k+1} \cup \Gamma^{\text{out},k+1}; \end{aligned}$$

(b) Find $\delta \mathbf{f}$ in $H^{1/2}(\hat{\Sigma})^3$ such that for all $\hat{\varphi}$ in $X(\hat{\Omega}^s)$ (see also equation (3.6))

$$\int_{\hat{\Sigma}} \delta \mathbf{f} \cdot \hat{\varphi} d\hat{s} = \int_{\Sigma^{k+1}} \delta p \mathbf{n} \cdot \hat{\varphi} \circ \mathcal{A}^{k+1} ds; \quad (5.3)$$

(c) Solve

$$DK^k \delta \mathbf{z} = \delta \mathbf{f};$$

(d) The product \mathcal{R}' by \mathbf{z} is given by $\mathbf{z} - \delta \mathbf{z}$;

The GMRES iterations are stopped as soon as the norm of the linear residual is lower than ϵ_{lin} times the norm of the nonlinear residual;

6) Compute the new displacement of the interface as

$$\mathbf{d}_{\Sigma}^{k+1} = \mathbf{d}_{\Sigma}^k + \omega^k \delta \mathbf{d}_{\Sigma}^{k+1},$$

where ω^k is computed, if necessary (i.e., if the increment with $\omega^k = 1$ does not decrease the norm of the residual), by a line-search strategy;

7) Set $\mathbf{u}^* = \mathbf{u}^{k+1}$, $p^* = p^{k+1}$ and go to step 2;

8) Go to next time step with

$$\begin{aligned} \mathcal{A}_{t^{n+1}} &= \mathcal{A}^{k+1}, \mathbf{u}^{n+1} = \mathbf{u}^{k+1}, p^{n+1} = p^{k+1} \\ \mathbf{d}^{n+1} &= \mathbf{d}^k, \dot{\mathbf{d}}^{n+1} = \dot{\mathbf{d}}^k. \end{aligned}$$

FSI-QN 2

With the second simplified model, we have only to replace steps 5a and 5b. The evaluation of the product of \mathcal{F}' by a vector \mathbf{z} (one time per GMRES iteration) is performed as follows: Find $(\delta \mathbf{u}, \delta p)$ in $V(\Omega^{f,k+1}) \times Q(\Omega^{f,k+1})$ such that for all (\mathbf{v}, q) in $V_\Sigma(\Omega^f) \times Q(\Omega^f)$,

$$\begin{cases} \frac{1}{\Delta t} \int_{\Omega^{f,k+1}} \rho_f \delta \mathbf{u} \cdot \mathbf{v} + \int_{\Omega^{f,k+1}} \rho_f (\mathbf{u}^* - \mathbf{w}^{k+1}) \cdot \nabla \delta \mathbf{u} \cdot \mathbf{v} + \int_{\Omega^{f,k+1}} \rho_f \delta \mathbf{u} \cdot \nabla \mathbf{u}^* \cdot \mathbf{v} \\ \quad - \int_{\Omega^{f,k+1}} \rho_f \delta \mathbf{u} \cdot \mathbf{v} \operatorname{div} \mathbf{w}^{k+1} - \int_{\Omega^{f,k+1}} \delta p \operatorname{div} \mathbf{v} + \int_{\Omega^{f,k+1}} 2\mu \epsilon(\delta \mathbf{u}) \cdot \epsilon(\mathbf{v}) = 0 \\ \int_{\Omega^{f,k+1}} q \operatorname{div} \delta \mathbf{u} = 0, \end{cases} \quad (5.4)$$

$$\delta \mathbf{u} = \frac{\mathbf{z}}{\Delta t} \circ (\mathcal{A}^{k+1})^{-1} \quad \text{on } \Sigma^{k+1}.$$

Step 5b reads: Find $\delta \mathbf{f}$ in $H^{1/2}(\hat{\Sigma})^3$ such that for all $\hat{\varphi}$ in $X(\hat{\Omega}^s)$,

$$\int_{\hat{\Sigma}} \delta \mathbf{f} \cdot \hat{\varphi} d\hat{s} = \int_{\Sigma^{k+1}} (\delta p \mathbf{n} - 2\mu \epsilon(\delta \mathbf{u}) \cdot \mathbf{n}) \mathbf{n} \cdot \hat{\varphi} \circ \mathcal{A}^{k+1} ds. \quad (5.5)$$

As in problem P3.5, the right hand side of (5.5) can be computed as the residual of (5.4).

Clearly, FSI-QN 1 (that has been proposed in [Ger03, GV03]) approximates more roughly the nonlinear problem. It is therefore not surprising that it requires generally more Newton iterations to converge than FSI-QN 2. Nevertheless, in all the test cases we have performed, the CPU time was less with FSI-QN 1 than with FSI-QN 2, each evaluation of FSI-QN 1 being much cheaper than that of FSI-QN 2. This conclusion certainly depends on the physical situation considered, and it is possible that in certain circumstances, FSI-QN 2 leads to better results. This is the reason why we present both models.

Regardless the fluid-structure interaction context, the reader may consider that FSI-QN 1 corresponds to a situation where the Jacobian of the nonlinear problem is roughly approximated, the convergence therefore requires several, but cheap, Newton iterations, whereas FSI-QN 2 corresponds to a better Jacobian approximation, which leads to less numerous, but more expensive, Newton iterations.

5.4 Preconditioned Krylov iterations for the Jacobian system

In this section we present a preconditioner which can be used whenever a sequence of similar linear problems has to be solved. This is the case when inverting the Jacobian in the Newton algorithm. We use generic notations to keep the method presented here independent from the fluid-structure interaction problem.

5.4.1 The GMRES iterative method

Consider the following linear problem: Find \mathbf{x} in V such that

$$A\mathbf{x} = \mathbf{b}, \quad (5.6)$$

where A is a linear operator from and onto a generic vector space V with scalar product (\cdot, \cdot) and \mathbf{b} is an element of V . The GMRES iterative method builds an orthonormal sequence

$(\mathbf{v}_j)_{j=1,\dots,k+1}$ in V , and two matrices, one Q_k , orthogonal of size $k+1 \times k+1$, and the other R_k , upper triangular of size $k+1 \times k$, such that if $\mathbf{x} = \sum_{j=1}^k \xi_j \mathbf{v}_j$, then $A\mathbf{x} = \sum_{j=1}^{k+1} \zeta_j \mathbf{v}_j$ where $\boldsymbol{\zeta} = Q_k R_k \boldsymbol{\xi}$.

The sequence $(\mathbf{v}_j)_{j=1,\dots,k+1}$ is an orthonormal basis of the Krylov subspace

$$\mathcal{L}_{k+1} = K^{k+1}(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^k \mathbf{r}_0\},$$

where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$. Moreover Q_k and R_k are the QR-decomposition of the Hessenberg matrix H , defined by $H_{jl} = (\mathbf{v}_j, A\mathbf{v}_l)$ (see for example Saad and Schultz [SS86], Van der Vorst [VdV03, §4 and §6]).

The residual of (5.6) is minimized on the linear subspace \mathcal{L}_k and the approximated solution is given by

$$\mathbf{x} = \arg \min_{\mathbf{x} \in \mathcal{L}_k} \|A\mathbf{x} - \mathbf{b}\|. \quad (5.7)$$

The standard implementation of the GMRES algorithm does not solve this problem explicitly, since the residual is built up dynamically, and the solution is only computed at the end.

Suppose that instead of building the decomposition of the Hessenberg matrix and the basis of the Krylov subspace, these are already given and we would like to solve (5.7) for a given datum \mathbf{b} (for example if one need to solve equation (5.6) for two different left hand sides). Then problem (5.7) is equivalent to

$$R_k \boldsymbol{\xi} = \Pi^* Q_k^T \boldsymbol{\beta}, \quad (5.8)$$

where $\beta_j = (\mathbf{b}, \mathbf{v}_j)$ for $j = 1, \dots, k+1$, and Π^* is the projection that sets the last component of a vector to zero (note that the last line of R_k has all components equal to zero). Note that the error is equal to the last component of $Q_k^T \boldsymbol{\beta}$.

In exact algebra, the GMRES method ends after a finite number of iterations, namely the dimension of the vector space V . Usually a tolerance is given for the relative error on the residual, such that it is in general not necessary to build a full basis of V . A drawback of this method is the need to store the sequence $(\mathbf{v}_j)_{j=1,\dots,k+1}$ and the matrices Q_k and R_k . This can require a large amount of memory. In the literature, to avoid memory problems, a restart is proposed but in this work we consider only full GMRES. If storing the sequence of vectors and the matrices is not a major problem, the sequence and the matrices can be exploited to build an efficient preconditioner for the solution of another problem close to the first one (see section 5.4.3).

5.4.2 Dynamic initial guess

Assume that we have two problems of type (5.6) on the same space V : Find \mathbf{x}_1 and \mathbf{x}_2 in V such that

$$A_1 \mathbf{x}_1 = \mathbf{b}_1 \text{ and } A_2 \mathbf{x}_2 = \mathbf{b}_2, \quad (5.9)$$

where the operators A_1, A_2 are close enough, e.g., the first one is a good preconditioner of the second one, but are both expensive to invert.

Then it is possible to generate an initial guess for the second problem using the iterates built during the solution of the first one. More precisely, let the first full (i.e., not restarted) GMRES method produce the sequence $(\mathbf{v}_j^{(1)})_{j=1,\dots,k+1}$ and the matrices $Q_{1,k}$ and $R_{1,k}$, where $k+1$ is the number of iterations required to converge for the first problem (up to a given tolerance).

The classical way to define the initial guess for the solution of the second equation, is to take the solution \mathbf{x}_1 of the first one which is in fact $\arg \min_{\mathbf{x} \in \mathcal{L}_{1,k}} \|A_1 \mathbf{x} - \mathbf{b}_1\|$. As pointed out previously, it is possible to use the Hessenberg decomposition and the Krylov subspace of the first GMRES resolution. This means that we can find at low computational costs an initial guess as

$$\arg \min_{\mathbf{x} \in \mathcal{L}_{1,k}} \|A_1 \mathbf{x} - \mathbf{b}_2\|. \quad (5.10)$$

Intuitively, this choice is more appropriate as a first approximation of the solution of the second equation in (5.9).

The minimization in (5.10) can be carried out without any evaluation of A_1 , since it is equivalent to solve as in (5.8) the problem reduced to the first k components,

$$R_{1,k} \boldsymbol{\xi} = \Pi^* Q_{1,k}^T \boldsymbol{\beta}_2, \quad (5.11)$$

where the components of $\boldsymbol{\beta}_2$ are now given by $(\mathbf{b}_2, \mathbf{v}_j^{(1)})$. To be efficient, the two operators have to be close to each other. See section 5.6.

5.4.3 Dynamic preconditioner

We would like to extend this idea and to use the information stored in the sequence and the matrices of the previous GMRES iteration to build a preconditioner. Indeed this is based on the idea that A_1 can be a good preconditioner for A_2 .

The basic step is to take the orthogonal projection of \mathbf{b}_2 on the image of $A_1|_{\mathcal{L}_{1,k}}$ (noted $\Pi_{\text{Im}(A_1|_{\mathcal{L}_{1,k}})} \mathbf{b}_2$) and then to apply the inverse of $Q_{1,k} R_{1,k}$. In fact $Q_{1,k}$ and $R_{1,k}$ are the QR factorization of the restriction of A_1 on $\mathcal{L}_{1,k}$

$$A_1|_{\mathcal{L}_{1,k}} : \mathcal{L}_{1,k} \rightarrow \text{Im}(A_1|_{\mathcal{L}_{1,k}}) \subset \mathcal{L}_{1,k+1}.$$

Then, to avoid a singular preconditioner, add $\lambda^{-1}(\mathbf{b}_2 - \Pi_{\text{Im}(A_1|_{\mathcal{L}_{1,k}})} \mathbf{b}_2)$, where λ is a scalar to be chosen.

The corresponding preconditioner is

$$P_1^{-1} = (A_1|_{\mathcal{L}_{1,k}})^{-1} \Pi_{\text{Im}(A_1|_{\mathcal{L}_{1,k}})} + \frac{1}{\lambda} (Id - \Pi_{\text{Im}(A_1|_{\mathcal{L}_{1,k}})}). \quad (5.12)$$

Its application to a vector \mathbf{y} in V can be carried out in five steps:

$$\zeta_j = (\mathbf{y}, \mathbf{v}_j^{(1)}) \quad j = 1, \dots, k+1, \quad (5.13)$$

$$\boldsymbol{\chi} = Q_{1,k}^T \boldsymbol{\zeta}, \quad (5.14)$$

$$\boldsymbol{\tau} = Q_{1,k} (Id - \Pi^*) \boldsymbol{\chi}, \quad (5.15)$$

$$\boldsymbol{\xi} : \text{solve } R_{1,k} \boldsymbol{\xi} = \Pi^* \boldsymbol{\chi}, \quad (5.16)$$

$$\mathbf{z} = \sum_{j=1}^k \xi_j \mathbf{v}_j^{(1)} + \frac{1}{\lambda} \left(\mathbf{y} - \sum_{j=1}^{k+1} (\zeta_j - \tau_j) \mathbf{v}_j^{(1)} \right), \quad (5.17)$$

where $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_{k+1})^T$, etc., and Π^* is the projection that sets the last component of a vector to zero and is equivalent to the projection $\Pi_{\text{Im}(A_1|_{\mathcal{L}_{1,k}})}$ on the image set of $A_1|_{\mathcal{L}_{1,k}}$

(see lemma 5.4.1). Since the last line of $R_{1,k}$ is zero, equation (5.16) is well defined and its resolution is straight-forward by backward substitution.

We propose to set the scalar λ equal to the last diagonal element of $R_{1,k}$, i.e., $R_{1,k}(k, k)$. Other choices can be derived from an Aitken relaxation parameter from the previous iterates, see section 5.6, or from the mean of the diagonal elements of $R_{1,k}$.

5.4.4 Invertibility of the preconditioner

To simplify the notation, in this section we refer to A , Q , R , etc., without indices whenever there are no ambiguities.

In the following we prove that for a non singular operator A on a finite dimensional vector field V with scalar product (\cdot, \cdot) , the preconditioner defined in (5.12) has full rank and can be computed following steps (5.13) to (5.17).

Theorem 5.4.1 *If A is not singular, then the operator*

$$M = (A|_{\mathcal{L}_k})^{-1} \Pi_{\text{Im}(A|_{\mathcal{L}_k})} + \frac{1}{\lambda} \left(Id - \Pi_{\text{Im}(A|_{\mathcal{L}_k})} \right) \quad (5.18)$$

is invertible on V .

Proof The GMRES algorithm computes an orthonormal sequence $(\mathbf{v}_j)_{j=1, \dots, k+1}$ such that $\text{Im}(A|_{\mathcal{L}_k})$ is included in \mathcal{L}_{k+1} . Recall that $\Pi_{\text{Im}(A|_{\mathcal{L}_k})}$ is the orthogonal projection on $\text{Im}(A|_{\mathcal{L}_k})$. Since A is regular, $A|_{\mathcal{L}_k}$ is invertible on $\text{Im}(A|_{\mathcal{L}_k})$ and M is well defined.

Let \mathbf{y} be an element in V such that $M\mathbf{y} = 0$. We need to show that \mathbf{y} is equal to zero. Let $\mathbf{y}_2 = \Pi_{\text{Im}(A|_{\mathcal{L}_k})}\mathbf{y}$ and $\mathbf{y}_1 = \mathbf{y} - \mathbf{y}_2$. Then

$$\frac{1}{\lambda} \mathbf{y}_1 = - (A|_{\mathcal{L}_k})^{-1} \mathbf{y}_2. \quad (5.19)$$

This implies that \mathbf{y}_1 is also in \mathcal{L}_k , i.e., $\mathbf{y}_1 = \sum_{j=1}^k \varphi_j \mathbf{v}_j$. But since \mathbf{y}_1 is in the orthogonal hull of $\text{Im}(A|_{\mathcal{L}_k})$, $(\mathbf{y}_1, A\mathbf{v}_l) = 0$ for all $l = 1, \dots, k$, hence

$$0 = \sum_{j=1}^k \varphi_j (\mathbf{v}_j, A\mathbf{v}_l) = \sum_{j=1}^k \varphi_j H_{jl},$$

where H is the Hessenberg matrix of A with respect to $(\mathbf{v}_j)_{j=1, \dots, k}$. This can be written in vector form as $\sum_{j=1}^k \varphi_j \mathbf{h}_j = 0$, where \mathbf{h}_j is equal to $(H_{j1}, \dots, H_{jk})^T$. Since A is regular and the vectors \mathbf{v}_j , $j = 1, \dots, k$, are linearly independent, the vectors \mathbf{h}_j are also linearly independent.

Hence $\varphi_j = 0$ for $j = 1, \dots, k$ and $\mathbf{y}_1 = 0$. From equation (5.19) also $\mathbf{y}_2 = 0$, which means that $\mathbf{y} = 0$ and that M is invertible. ■

Lemma 5.4.1 *The operator $Q \Pi^* Q^T: \mathbb{R}^{k+1} \rightarrow \mathbb{R}^{k+1}$ is an orthogonal projector with respect to the Euclidean scalar product. Moreover, its image is equal to the image of $QR: \mathbb{R}^k \rightarrow \mathbb{R}^{k+1}$.*

Proof The operator is a projection, since

$$(Q \Pi^* Q^T)^2 = Q \Pi^* \Pi^* Q^T = Q \Pi^* Q^T.$$

To prove the orthogonality first note that $Id - Q\Pi^*Q^T = QQ^T - Q\Pi^*Q^T = Q(Id - \Pi^*)Q^T$. Then for ξ and ζ in \mathbb{R}^{k+1}

$$(Q(Id - \Pi^*)Q^T\zeta, Q\Pi^*Q^T\xi) = ((Id - \Pi^*)Q^T\zeta, \Pi^*Q^T\xi) = 0,$$

since Q is orthogonal. Hence Π^* is an orthogonal projection with respect to the Euclidean scalar product.

Moreover we have that $\text{Im } QR = \text{Im } Q\Pi^*Q^T$. Firstly, $\text{Im } QR \subset \text{Im } Q\Pi^*Q^T$, since the last line of R has only zeros and

$$Q\Pi^*Q^TQR = Q\Pi^*R = QR.$$

Then, let $\xi \in \mathbb{R}^k$ and $\zeta \in \mathbb{R}^{k+1}$. Equation $QR\xi = Q\Pi^*Q^T\zeta$ is equivalent to $R\xi = \Pi^*Q^T\zeta$, which has a unique solution since the last line of R has only zeros, R is upper diagonal and all elements of the diagonal are different from zero (else A is singular). Hence $\text{Im } Q\Pi^*Q^T \subset \text{Im } QR$. ■

Proposition 5.4.1 *Let y be in V and set $\zeta_j = (y, v_j^{(1)})$, $j = 1, \dots, k+1$ and let $\xi = Q\Pi^*Q^T\zeta$. Then*

$$\Pi_{\text{Im}(A|_{\mathcal{L}_k})}y = \sum_{j=1}^{k+1} \xi_j v_j. \quad (5.20)$$

Proof Firstly note that for $x = \sum_{j=1}^k \chi_j v_j$, $Ax = \sum_{j=1}^{k+1} (QR\chi)_j v_j$. Then, the proof follows from lemma 5.4.1. ■

5.4.5 Application to a sequence of problems

In this section we apply the preconditioner defined in the previous section in a nested way to a sequence of problems.

The preconditioner may be applied as a left or a right preconditioner. Anyway, we present it in its right form, since a left preconditioned GMRES changes the norm used in computing the residual, while it is not the case in the right preconditioned version. Moreover, it is possible to apply this method to an already existing left preconditioned GMRES, such that the resulting GMRES has both left and right preconditioners (see also [VdV03, §10]).

Suppose that we have solved $A_1x = b_1$ with GMRES and that we define P_1 according to formula (5.18), then the second right preconditioned problem reads

$$A_2P_1^{-1}y = b_2 \quad x = P_1^{-1}y. \quad (5.21)$$

Suppose now that there is a third equations to solve. We can use P_1 as a preconditioner, but if operator A_3 is closer to A_2 than to A_1 , as in the case of the Newton algorithm, it is more interesting to use P_2 . Unfortunately, since we suppose that A_2 (not $A_2P_1^{-1}$!) is a good preconditioner for A_3 , this is not possible.

Storing the intermediate vectors

A first solution is to use a technique taken from Flexible GMRES (see [Saa93] or [VdV03]). FGMRES, in addition to the storage of the matrices and the basis of the Krylov subspace, stores the basis of the intermediate evaluations of P_1^{-1} .

More precisely, the GMRES resolution with right preconditioning, at a given step j , computes $AP_1^{-1}\mathbf{v}_j$. We perform this operation in two steps and we store the intermediate vector $\mathbf{w}_j = P_1^{-1}\mathbf{v}_j$.

This allows to build a preconditioner P_2 which is well suited for A_3 and its application to a vector \mathbf{y} in V can be carried out in five steps similarly to steps (5.13) to (5.17)

$$\begin{aligned}\zeta_j &= (\mathbf{y}, \mathbf{v}_j^{(2)}) \quad j = 1, \dots, k+1, \\ \boldsymbol{\chi} &= Q_{2,k}^T \boldsymbol{\zeta}, \\ \boldsymbol{\tau} &= Q_{2,k} (Id - \Pi^*) \boldsymbol{\chi}, \\ \boldsymbol{\xi} &: \text{solve } R_{2,k} \boldsymbol{\xi} = \Pi^* \boldsymbol{\chi}, \\ \mathbf{z} &= \sum_{j=1}^k \xi_j \mathbf{w}_j^{(2)} + \frac{1}{\lambda} \left(\mathbf{y} - \sum_{j=1}^{k+1} (\zeta_j - \tau_j) \mathbf{v}_j^{(2)} \right).\end{aligned}$$

In the last line, the term

$$\mathbf{y} - \sum_{j=1}^{k+1} (\zeta_j - \tau_j) \mathbf{v}_j^{(2)}$$

accounts for the complement of the projection on $\text{Im}(A_2|_{\mathcal{L}_{2,k}})$ of \mathbf{y} , i.e.,

$$(Id - \Pi_{\text{Im}(A_2|_{\mathcal{L}_{2,k}})}) \mathbf{y},$$

and it is important that there are involved the vectors $\mathbf{v}_j^{(2)}$ and not $\mathbf{w}_j^{(2)}$.

The coding of these steps requires only few modifications to the subroutines required for steps (5.13) to (5.17). Then the third preconditioned problem reads

$$A_3 P_2^{-1} \mathbf{y} = \mathbf{b}_3 \quad \mathbf{x} = P_2^{-1} \mathbf{y}.$$

The drawback of this method is that if for example P_1 works fine on A_2 , then the basis $\{\mathbf{v}_j^{(2)}\}$ has few members and P_2 is less rich than P_1 is. As a consequence, the basis $\{\mathbf{v}_j^{(3)}\}$ has more members and the preconditioner build from is richer. As a result, the number of GMRES iterations oscillates (see figure 5.15). We would like to avoid these oscillations and to inherit the richness of P_1 for A_3 and at the same time use also P_2 . This is possible with the following method.

Nested preconditioners

Instead, we choose to nest the preconditioners. For example, the third preconditioned problem reads

$$A_3 P_1^{-1} P_2^{-1} \mathbf{y} = \mathbf{b}_3 \quad \mathbf{x} = P_1^{-1} P_2^{-1} \mathbf{y}. \quad (5.22)$$

In fact, P_2 is a preconditioner derived from (5.21) and is therefore well suited for $A_3 P_1^{-1}$. It is mandatory to keep the number of preconditioners small, so it is recommended to do a restart

of the preconditioners. In principle one can either delete all the preconditioners or just keep P_1 .

As mentioned above, it is also possible to use an already defined left preconditioner M , such that the problems are

$$M^{-1}A_1 \mathbf{x} = M^{-1}\mathbf{b}_1, \quad (5.23)$$

$$M^{-1}A_2P_1^{-1} \mathbf{y} = M^{-1}\mathbf{b}_2 \quad \mathbf{x} = P_1^{-1}\mathbf{y}, \quad (5.24)$$

$$M^{-1}A_3P_1^{-1}P_2^{-1} \mathbf{y} = M^{-1}\mathbf{b}_3 \quad \mathbf{x} = P_1^{-1}P_2^{-1}\mathbf{y}, \quad (5.25)$$

$$\text{etc.} \quad (5.26)$$

Indeed, it is possible that the left preconditioner depends on the problem index, as long as the original preconditioned problems, i.e., $M_i^{-1}A_i$, are close to each other.

5.5 Nonlinear acceleration of the Newton–Krylov algorithm

We present in this section our global procedure for solving nonlinear problems. We first recall the solution method based on Newton–Krylov methods [BS94]. Then we describe an accelerating procedure that aims to improve its computational efficiency. Finally we present the global framework in an algorithmic fashion.

The main computational challenge is the inversion of the Jacobian matrix (step 3 on page 102) with Krylov-type methods. One drawback of this procedure is that in every linearization step the Jacobian must be evaluated in many directions. Nevertheless this potentially huge cost can be reduced by replacing the Jacobian matrix by a simpler and cheaper operator, as done in the quasi-Newton algorithm.

In this work a preconditioned GMRES method is used to solve the Jacobian problem stated in step 3. As a preconditioner for GMRES, the dynamic preconditioner presented in section 5.4 is investigated. An accelerating procedure is presented in the following for enhancing the efficiency of this preconditioned Newton-GMRES procedure.

We use generic notations to keep the method presented here independent from the fluid-structure interaction problem.

5.5.1 Acceleration strategy

We apply here a strategy proposed by Washio et al. [WO97] aiming to build a nonlinear subspace acceleration for general nonlinear solvers combined with the replacement of the Jacobian by the preconditioner defined in section 5.4.3 (noted \bar{J}).

The idea in [WO97] was to store the iterates and the residuals in two different subspaces (of dimension m) which represent a basis for an approximated linear problem. Then they proposed to find a new iterate \mathbf{x}_{new} on the affine subspace $\mathbf{x}_k + \sum_{j=0}^{m-1} \alpha_j (\mathbf{x}_j - \mathbf{x}_k)$ by minimizing the residual. With this goal in mind, the residual is considered affine in this subspace, such that the minimum can be found *without* new evaluation of \mathcal{R} .

A simplified acceleration scheme deduced from [WO97] has been adopted here. We refer to [WO97] for a complete description of the nonlinear convergence acceleration strategy. This simplified scheme basically requires two parameters: m , the dimension of the minimization subspace and ε_B , a parameter needed to control the nonlinear convergence. We keep in the minimization subspaces the m latest iterates and corresponding residuals when available. In the following, we only detail the procedure to be carried out after a Newton step if convergence

is not achieved. Indeed, this can be repeated as long as the approximation \bar{J} of the Jacobian is satisfactory.

Likely, the simplest approximation of the Jacobian is to use the extension of the Aitken relaxation method which can be seen as a rough approximation of the Jacobian. In this case, $\bar{J}^{-1} = \omega Id$, where ω is a scalar equal to

$$\omega = \frac{[\mathcal{R}(\mathbf{x}_k) - \mathcal{R}(\mathbf{x}_{k-1})]^T (\mathbf{x}_k - \mathbf{x}_{k-1})}{\|\mathcal{R}(\mathbf{x}_k) - \mathcal{R}(\mathbf{x}_{k-1})\|^2} \quad (5.27)$$

If we make this choice, it is preferable to do only one acceleration step, since otherwise we would be barely performing an accelerated Aitken algorithm.

A better choice is the preconditioner defined in section 5.4.3, $\bar{J} = P_k \cdots P_1$. This is motivated from the good properties as preconditioner in the resolution of the following Jacobian.

1) Find the solution of the minimization problem:

$$\alpha = \arg \min_{\alpha \in \mathbb{R}^m} \left\| \mathcal{R}(\mathbf{x}_k) + \sum_{j=0}^{m-1} \alpha_j (\mathcal{R}(\mathbf{x}_j) - \mathcal{R}(\mathbf{x}_k)) \right\|;$$

2) Set the candidate iterate as: $\mathbf{x}_{\text{new}} = \mathbf{x}_k + \sum_{j=0}^{m-1} \alpha_j (\mathbf{x}_j - \mathbf{x}_k)$;

3) Control of the acceleration strategy:

$$\text{if } \varepsilon_B \|\mathbf{x}_{\text{new}} - \mathbf{x}_k\| > \min_{j < m} \|\mathbf{x}_j - \mathbf{x}_k\|$$

$$\mathbf{x}_{\text{new}} = \mathbf{x}_k;$$

$$\bar{\mathcal{R}}_{\text{new}} = \mathcal{R}(\mathbf{x}_k);$$

else

$$\bar{\mathcal{R}}_{\text{new}} = \mathcal{R}(\mathbf{x}_k) + \sum_{j=0}^{m-1} \alpha_j (\mathcal{R}(\mathbf{x}_j) - \mathcal{R}(\mathbf{x}_k));$$

end;

4) Solve $\bar{J}\delta\mathbf{x} = -\bar{\mathcal{R}}_{\text{new}}$ by a Krylov subspace solver;

5) Set the next iterate as: $\mathbf{x}_{k+1} = \mathbf{x}_{\text{new}} + \delta\mathbf{x}$;

6) Compute the next residual: $\mathcal{R}(\mathbf{x}_{k+1})$;

7) if $\|\mathcal{R}(\mathbf{x}_{k+1})\| < \text{tol}$, then stop;

8) Control: if $\|\mathcal{R}(\mathbf{x}_{k+1})\| > \min_{j \leq m} \|\mathcal{R}(\mathbf{x}_j)\|$ then

(a) Perform a line-search and exit the acceleration step; **or**

(b) Do not store \mathbf{x}_{k+1} and exit the acceleration step;

9) If \bar{J} is old, exit the acceleration step, otherwise $k = k + 1$ and go to 1.

Steps 1 to 3 are related to the acceleration scheme and are briefly detailed in the following. To find the minimizing α (item 1), store the points \mathbf{x}_j of the Newton algorithm in a matrix V and the values of the residuals $\mathcal{R}(\mathbf{x}_j)$ in a matrix F , such that

$$\begin{aligned} V &= [\mathbf{x}_0, \dots, \mathbf{x}_{k-1}] \quad \text{and} \\ F &= [\mathcal{R}(\mathbf{x}_0) - \mathcal{R}(\mathbf{x}_k), \dots, \mathcal{R}(\mathbf{x}_{k-1}) - \mathcal{R}(\mathbf{x}_k)] \end{aligned} \quad (5.28)$$

Then α in \mathbb{R}^k can be found by solving the linear problem

$$F^T F \alpha = -F^T \mathcal{R}(\mathbf{x}_k). \quad (5.29)$$

Since this problem may be singular, there are two choices: solve the problem with an iterative method (we use conjugate gradient), or, as proposed by Washio et al. in [WO97], use a direct method (Cholesky factorization) and, to ensure the invertibility of problem (5.29), add a term $\epsilon_F Id$ for a small ϵ_F to the matrix $F^T F$. They propose $\epsilon_F = 10^{-16} \max(\text{diag}(F^T F))$.

Since this scheme is based on a linearization of the residual, this algorithm must be "securised". Thus a control step (step 3) is introduced to avoid the occurrence of too close iterates leading to stagnation in nonlinear convergence. In case of stagnation, both latest available iterate \mathbf{x}_k and residual $\mathcal{R}(\mathbf{x}_k)$ are retained. This control criterion can be replaced by a more efficient one, namely $\sum_{j=0}^{m-1} |\alpha_j| > \varepsilon_B$ or $\|\alpha\|_2 > \varepsilon_B$, which in addition can be carried out before step 2.

The positive control parameter ε_B can be chosen depending on the nonlinearity of the problem. If the problem is highly nonlinear, it is crucial to keep ε_B small, since the algorithm is based on the good approximation of the residual in \mathbf{x}_{new} given by the linearization.

Note that this accelerating procedure is rather cheap: a solution of a small system of size $m \times m$, $2m + 2$ inner products and m vector updates and the evaluation of nonlinear residual at most.

Finally a control procedure for the global Newton process (step 8) has been added. Indeed it may happen that the new residual norm $\|\mathcal{R}(\mathbf{x}_{k+1})\|$ is larger than the minimal residual norm of the intermediate solutions. There are potentially two reasons for this behavior: either the frozen Jacobian \bar{J} is no more accurate enough or the linearized residual $\bar{\mathcal{R}}_{\text{new}}$ is really different from the true one $\mathcal{R}(\mathbf{x}_{\text{new}})$. A control procedure is therefore needed to cure these two bottlenecks. In practice however this situation has not been experienced numerically.

In a Newton framework, this acceleration scheme offers two main advantages. First the new residual is expressed as a linear combination of residuals that belong to the minimization subspace of residuals. Thus an expensive operation (evaluation of the residual) is avoided when building the right-hand side for the new Jacobian system. Second, the frozen Jacobian \bar{J} can be reused again and again as will be shown in the numerical results. These two points explain the improved efficiency of the preconditioned Newton-GMRES algorithm.

5.5.2 Global procedure

Finally the global procedure is a preconditioned Newton-GMRES method accelerated by the nonlinear strategy presented in section 5.5.1. The linear preconditioner is the dynamic preconditioner presented in section 5.4. This relatively complex algorithm is shown in Figure 5.1. It is meant to be robust and efficient when treating general (maybe highly) nonlinear problems. As a first evaluation, it has been investigated in the framework of fluid-structure interaction in haemodynamics as described next.

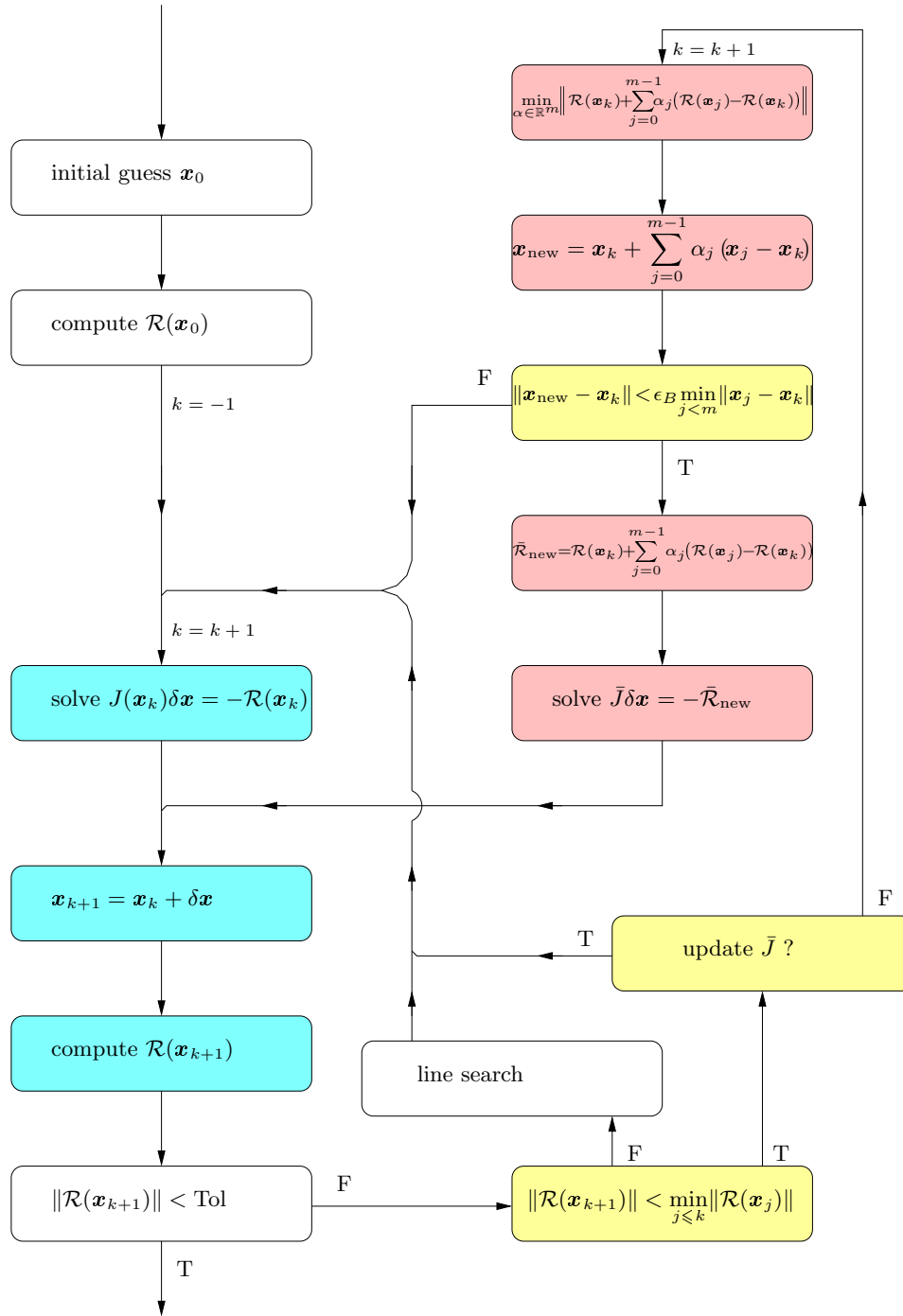


Figure 5.1: Accelerated Newton algorithm. On the left, the standard Newton algorithm at top, to the right, the acceleration step.

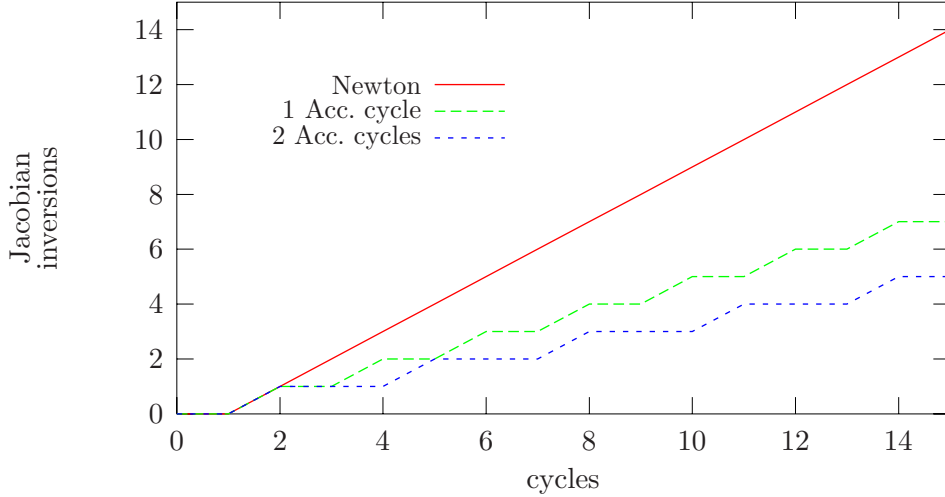


Figure 5.2: Number of Jacobian inversions per cycle (= Newton or accelerated).

5.6 Application to fluid-structure interaction

We present in this section an application of the previous algorithms to fluid-structure interaction problems.

5.6.1 Settings

The linear preconditioner is used in a nested way as explained in section 5.4.5, and is fully restarted at every time step. A reason for this complete restart is that at a new time step the underlying system can be quite different from the one at the previous time step. As we pointed out, the key idea for the efficiency of this dynamic preconditioner is that the problems in the sequence must be close to each other.

The nonlinear convergence acceleration scheme is used with minimization subspaces of dimension 5 ($m = 5$) and the control parameter ε_B is set to 0.5. The frozen Jacobian \bar{J} corresponds to the previous Jacobian matrix.

In the accelerating step, as Jacobian's approximation we use the preconditioner built from the previous resolution of the Jacobian and we keep that approximation for two acceleration steps. As already mentioned, we keep only the last 5 residual's evaluations. We apply the accelerating step even after the first evaluation of the residual, which means that \mathbf{x}_{new} is equal to \mathbf{x}_k , but that the Jacobian is not evaluated for two Newton steps. This choice is arbitrary and in some cases may be negative. Another choice is to apply the accelerating step only when there are at least two or three evaluations of the residual. We suggest to verify which choice fits at best from problem to problem.

5.6.2 Three-dimensional test case

Test case: Pressure wave in a bent cylinder, 100 time iterations ($\delta t = 2 \cdot 10^{-4}$). The discretized fluid domain has 5874 nodes while the interface and the structure have 1056 nodes.

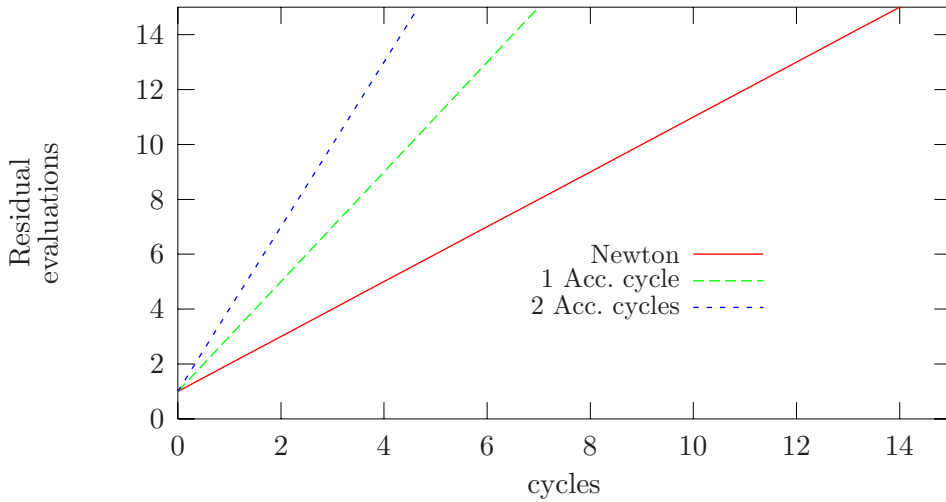


Figure 5.3: Number of residual evaluations per cycle (= Newton or accelerated).

Algorithm	CPU Time	Normalized CPU Time
No Prec.	6h16'	1
Prec.	5h56'	0.94
Prec. & Acc.	5h31'	0.88

Table 5.1: CPU time for the FSI-QN 1 approach.

The nested preconditioners applied to the quasi-Newton method to solve the fluid-structure interaction problem highly reduces the number of GMRES iterations in both experiments (see figures 5.5 and 5.8). Note that its use reduces the CPU time of only 6 % for the first approximation of the tangent problem (see table 5.1). In this case the evaluation of the Jacobian is a small computational task in comparison to the evaluation of the residual. Using the nested preconditioner reduces the CPU time of 29 % in the second tangent problem (see table 5.2). Here the Jacobian evaluation is more involved, hence saving GMRES iterations has a bigger impact on the computing time. In contrast, since the number of Newton iteration is highly reduced (2 or 3 iterations), the accelerated algorithm slows down the computations. In fact, since the approximate Jacobian \bar{J} is not precise enough (with respect to the exact one), the accelerated part of the algorithm is not as efficient as the standard part. Hence the the additional cost of the acceleration (i.e., extra residual evaluations see figure 5.3) slows down the computations.

In Figures 5.5 and 5.8 the number of GMRES iterations against the successive Newton resolutions is shown. At each new time step the stored preconditioners are deleted, hence the number of GMRES iterations at the first Newton iteration for the “no prec” and the “prec” approaches are obviously the same. As expected, in both cases the number of Newton iterations is not reduced.

As shown in Figure 5.2, the acceleration algorithm can be effective when there is at least a relative number of Newton iterations to be performed. In fact, in the second experiment, there are only few (2 or 3) Newton iterations at each time step (see figure 5.9). Consequently the acceleration approach slows down the computations (see table 5.2). On the other hand,

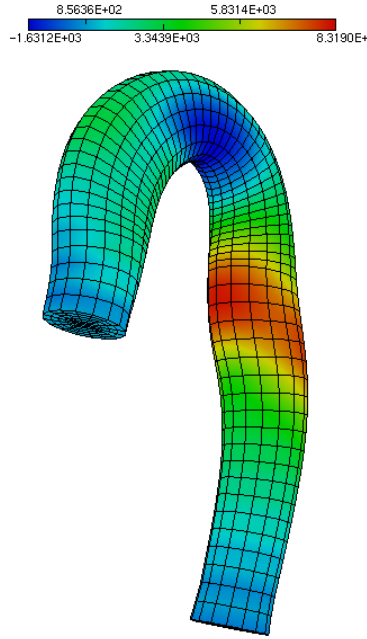


Figure 5.4: Propagation of a pressure wave in a bent cylinder.

Algorithm	CPU Time	Normalized CPU Time
No Prec.	12h36'	1
Prec.	8h57'	0.71
Prec. & Acc.	9h13'	0.73

Table 5.2: CPU time for the FSI-QN 2 approach.

in the first experiment the number of Newton iterations to be performed is sufficient for the accelerated algorithm to show its benefits (see Figure 5.6). In this case the number of genuine Newton iterations is reduced from 8-9 to 3-4. The CPU time is reduced of another 6 % for a total gain of 12 %.

To compare these results with a different problem's size, we have tested the same approach to a two-dimensional case. We report the results in the next section.

5.6.3 Two-dimensional test case

In this section we show the results of a two-dimensional experiment with the same parameters for the proposed algorithm considered in the three-dimensional one. The computational domain is a straight tube, the fluid is modeled by the two-dimensional incompressible Navier–Stokes equations and the structure by a generalized string model. The discretized fluid domain has 779 nodes while the interface and the structure have 41 nodes.

The tolerance for the solution of the coupled system is 10^{-6} and 10^{-8} respectively, in order to show the differences when the number of Newton iterations increase.

The results are essentially the same as in three dimensions, except the fact that in this case the approximated Jacobian FSI-QN 2 is only slightly more expensive than FSI-QN 1

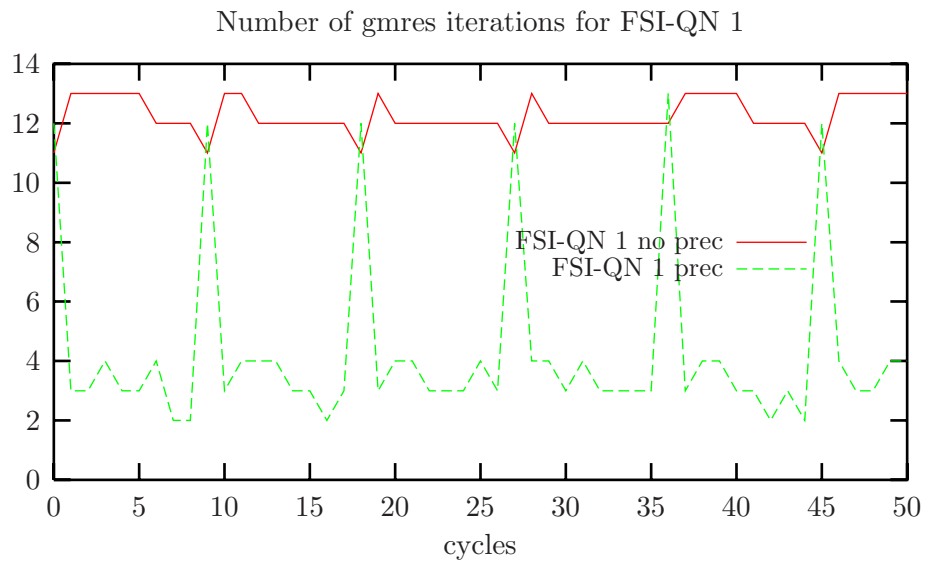


Figure 5.5: Number of GMRES iterations for FSI-QN 1.

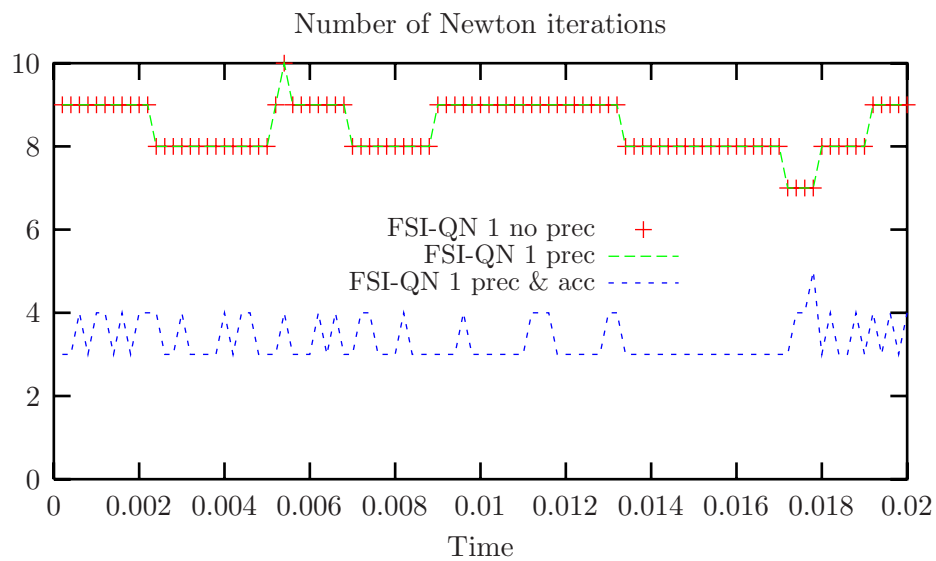


Figure 5.6: Number of genuine Newton iterations for FSI-QN 1.

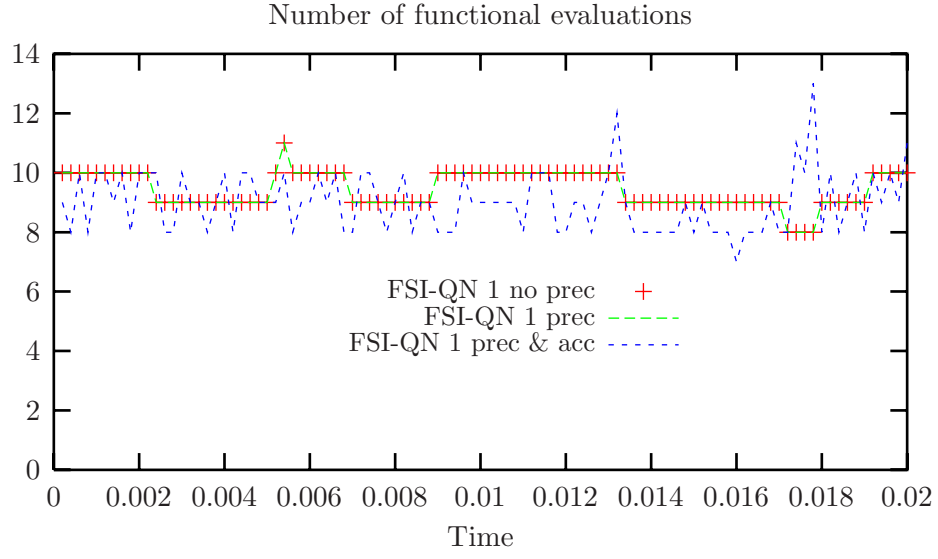


Figure 5.7: Number of functional \mathcal{T} evaluations for FSI-QN 1.

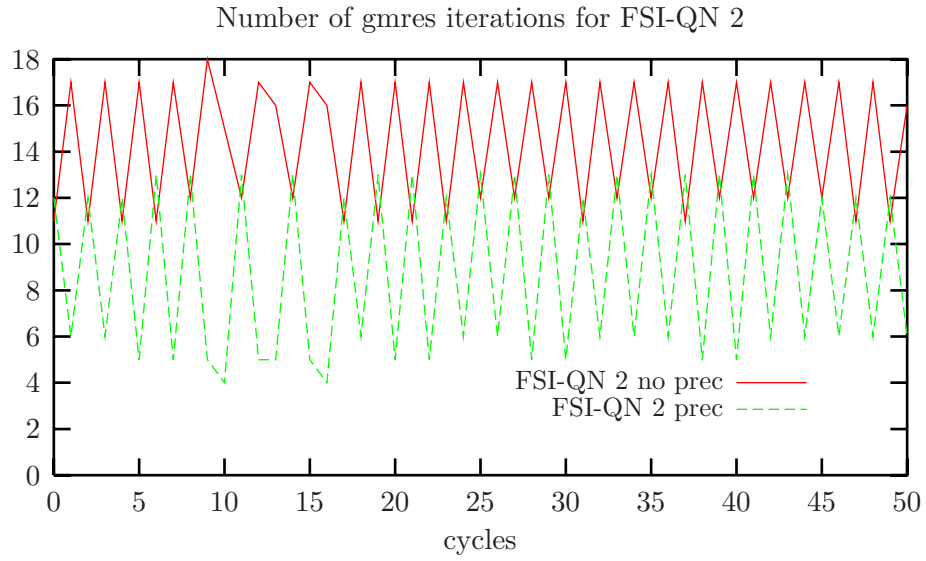


Figure 5.8: Number of GMRES iterations for FSI-QN 2.

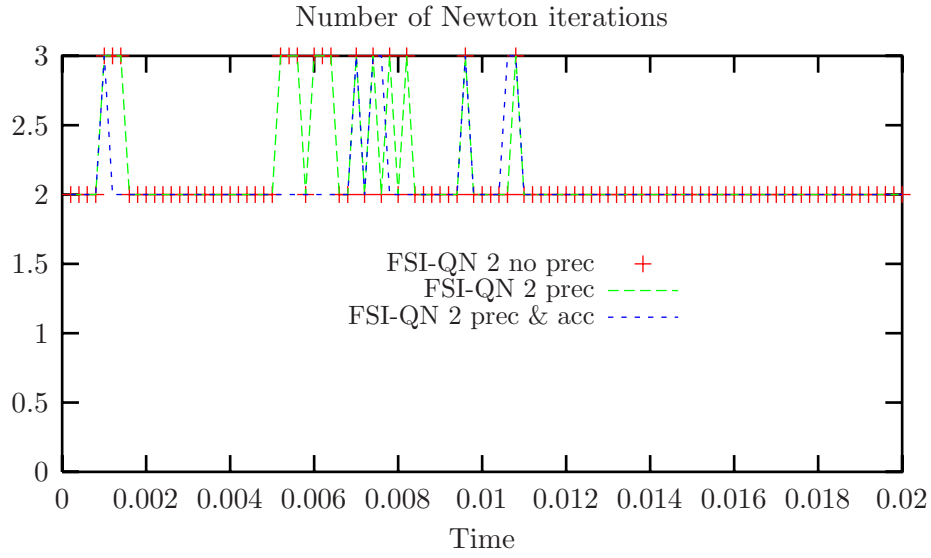
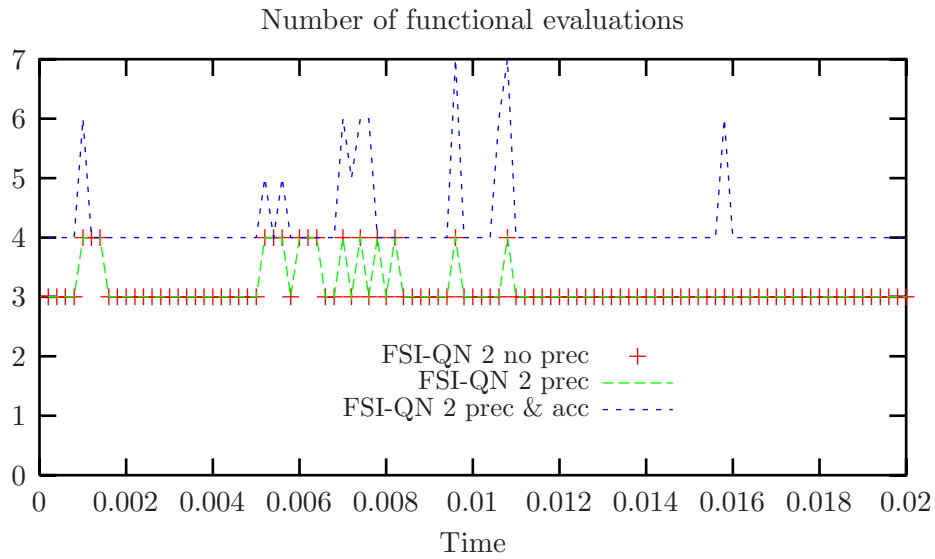


Figure 5.9: Number of genuine Newton iterations for FSI-QN 2.

Figure 5.10: Number of functional \mathcal{T} evaluations for FSI-QN 2.

and that the improvements of the preconditioner and of the accelerated algorithm in the FSI-QN 1 case are pronounced (see tables 5.3 and 5.4). Remark also that when using a more restrictive tolerance, the approximation with FSI-QN 2 and the accelerated algorithm shows a performance nearer to the FSI-QN 1 case.

As we can see in figures 5.5, 5.8, 5.11-5.14, the benefits of the presented preconditioner are equivalent in both experiments.

In addition to the tests carried out in the previous section, we consider a mixed use of the approximate Jacobians FSI-QN 1 and FSI-QN 2. In this case we do not consider the accelerated algorithm. At each time step, we proceed as follows: At the first Newton iteration, we solve the tangent problem using FSI-QN 1. Among others, this allows to compute a preconditioner P_1 at low cost. Then at the next Newton iterations we use FSI-QN 2. We would like to remark that the results in term of number of iteration is the same as when using only FSI-QN 2, but that the CPU time when the tolerance is smaller (see tables 5.3 and 5.4 and figure 5.14).

For testing purposes, we applied also the FGMRES approach described in section 5.4.5, where we do not need to nestle the preconditioners. As expected, the number of GMRES iterations oscillates because when a preconditioner is built with many basis, the GMRES needs less iterations and the new preconditioner is built with less basis, and vice-versa (see figure 5.15).

FSI-QN 1	CPU Time	GMRES iter.	residual eval.
No Prec.	1 = 10'09"	1 = 8158	1 = 1219
Prec. P_1	0.78	0.42	1
Nested Prec.	0.77	0.40	1
Prec. & Acc.	0.61	0.21	0.93
FSI-QN 2			
No Prec.	1.13=11'20"	0.39 = 3173	0.49 = 600
Nested Prec.	0.88	0.24	0.49
Prec. & Acc.	0.94	0.24	0.64
FSI-QN 1+2			
Prec	0.87	0.29	0.64

Table 5.3: Two-dimensional experiment with tolerance 10^{-6} for 200 time steps. The values are normalized with respect to the “no prec.” algorithm.

5.7 Conclusion

In this chapter we have dealt with the definition of a dynamic preconditioner to be applied to a sequence of problems and with the acceleration of a (quasi-)Newton algorithm.

We have shown that the preconditioner is well defined and that its application is straight forward. A great advantage of the defined preconditioner, is that there is no need to build the matrix A explicitly. In fact, A represents a generic linear operator, and in the numerical experiments the corresponding matrix is never built.

We then applied the preconditioner as an approximation of the Jacobian in the acceleration of the Newton algorithm.

5.7. CONCLUSION

FSI-QN 1	CPU Time	GMRES iter.	residual eval.
No Prec.	1 = 16'35"	1 = 14339	1 = 1942
Prec. P_1	0.72	0.39	0.98
Nested Prec.	0.70	0.34	0.98
Prec. & Acc.	0.50	0.19	0.79
FSI-QN 2			
No Prec.	1.11 = 18'25"	0.39 = 5624	0.41 = 790
Nested Prec.	0.72	0.29	0.41
Prec. & Acc.	0.64	0.15	0.51
FSI-QN 1+2			
Prec.	0.65	0.19	0.46

Table 5.4: Two-dimensional experiment with tolerance 10^{-8} for 200 time steps. The values are normalized with respect to the “no prec” algorithm.

Both have been tested in two fluid-structure simulations with a gain of up to 29% in CPU time in a three-dimensional experiment and up to 50% in a three dimensional one.

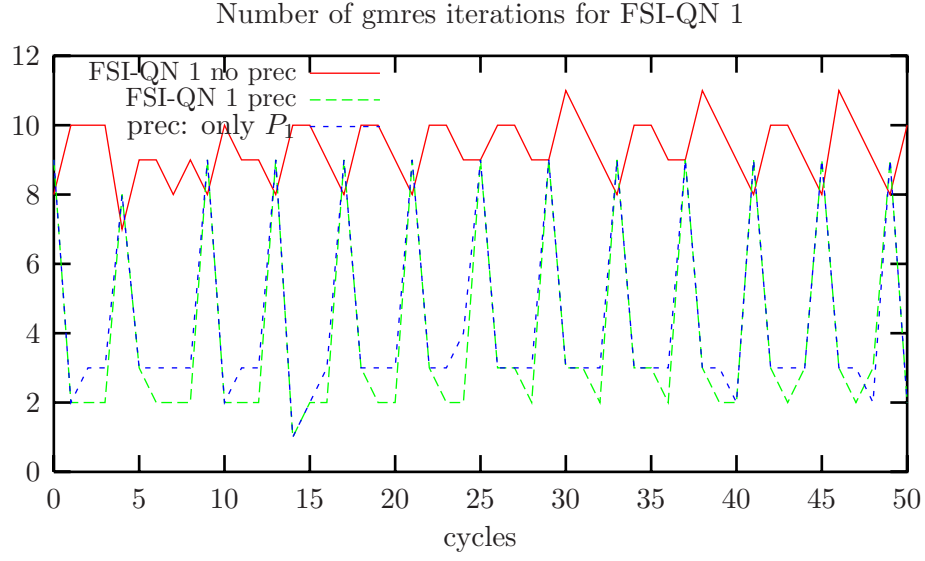


Figure 5.11: 2D: Number of GMRES iterations for FSI-QN 1, tolerance 10^{-6} .

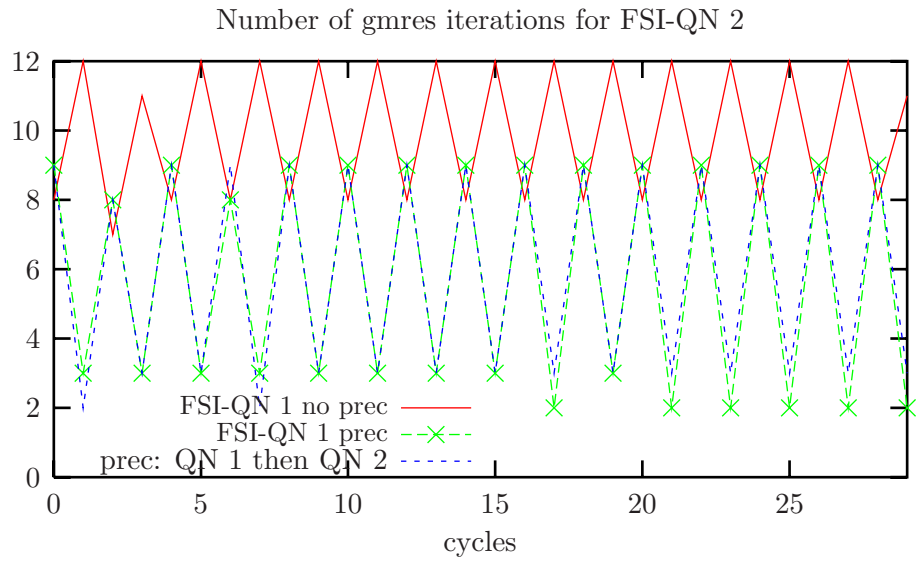
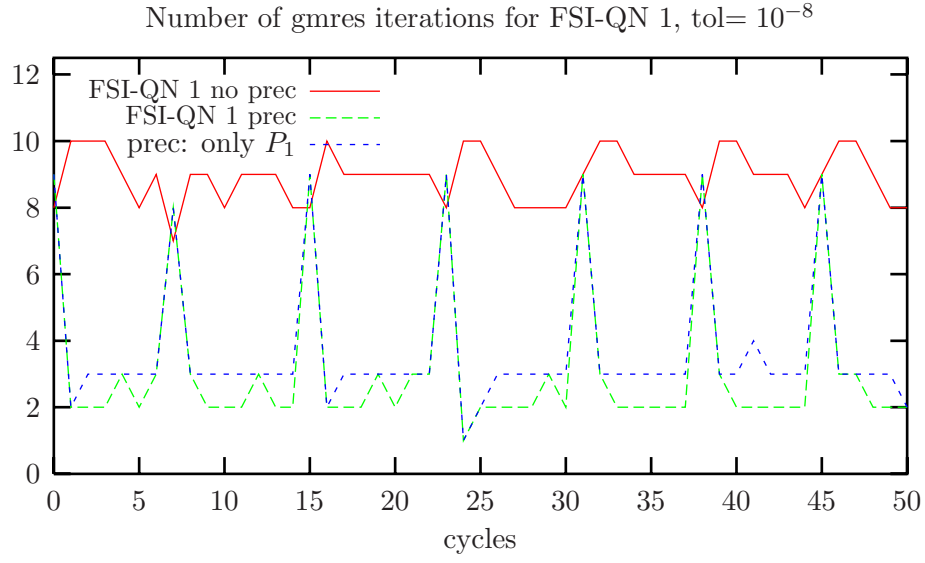
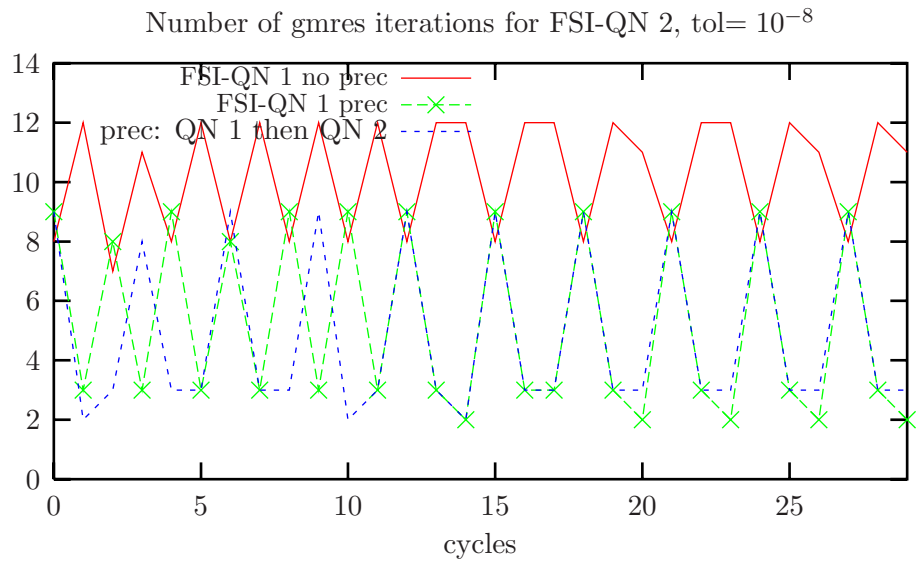


Figure 5.12: 2D: Number of GMRES iterations for FSI-QN 2, tolerance 10^{-6} .

Figure 5.13: 2D: Number of GMRES iterations for FSI-QN 1, tolerance 10^{-8} .Figure 5.14: 2D: Number of GMRES iterations for FSI-QN 2, tolerance 10^{-8} .

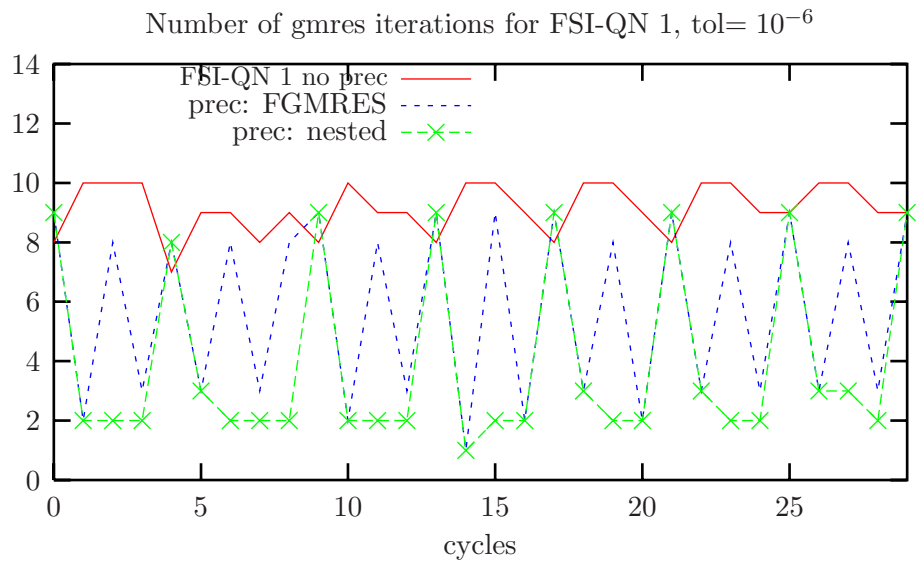


Figure 5.15: 2D: Comparison between the nested preconditioners and the FGMRES approach.

Conclusions

In this thesis we addressed the mathematical modeling of blood flow in arterial vessels. The numerical problem that is generated after space and time discretization of the mathematical equations is computationally complex. Hence a judicious choice of solution algorithms is mandatory.

In the first part, we proposed an axisymmetric model for the blood flow in vessels featuring a symmetry with respect to a straight axis. Moreover, should the angular component of the data be zero, the three-dimensional problem reduces to a two dimensional one on a half section of the domain under consideration. With this aim, the integrals in the weak formulation of the three-dimensional equations are simplified by a change of variable (from Cartesian to cylindrical coordinates) and integration with respect to the angular coordinate.

In the case of the steady axisymmetric Stokes equations, we formulated a finite element discretization and shown stability and optimal a priori error estimates. For the unsteady axisymmetric Navier–Stokes equations in moving domains, we showed stability and provided a stabilization for the semi-discretized formulation.

In the second part, we described the fluid-structure interaction problem in an abstract setting that is suited in a broad variety of situations. We reviewed classical fixed-point and Newton like algorithms and proposed convenient techniques for accelerating the convergence.

On one hand, should the programs for the solution of the fluid and the structure be originated from independent sources, the use of a relaxed fixed-point iteration strategy is advisable. In order to choose the relaxation parameter we used the Aitken extrapolation method. In addition, by using the transpiration boundary conditions technique we strongly reduced the computational time.

On the other hand, when the codes of the fluid and structure solvers are accessible, Newton-based methods are generally more effective. Depending upon the specific situation (strength of the non-linearity, time step constraints, ...), it is possible to approximate the Jacobian by physically inspired less sophisticated models. We proposed an approximation of the Jacobian (in addition to the exact form and to another approximation already present in the literature) and a preconditioner to invert the Jacobian, which have the advantage of being simple and computationally cheap.

Besides, we also proposed to use, at each new time step, a rough approximation of the Jacobian based on a very simple model at the first Newton iteration in order to build the preconditioner. Then in the subsequent Newton iterations we use a more appropriate approximation of the Jacobian (or else the exact one) in conjunction with the preconditioner based on the first Newton iteration. Finally, we exploited a Newton-Krylov technique to provide an accelerated Newton algorithm.

All the algorithms above have been tested on two and three-dimensional test cases with a consistent reduction of computational time with respect to the existing algorithms.

Appendix A

Useful properties needed in the analysis of the axisymmetric Stokes problem

A.1 Lemmas needed by proposition 1.5.2

In this section we prove some lemmas needed by proposition 1.5.2. For T_k in \mathcal{T}_h and x^{kj} a midpoint of T_k not on $\Gamma \cup \Gamma_0$, define D_k as the union of the three subtriangles of T_k members of $\mathcal{T}_{h/2}$ which have x^{kj} as a vertex. Let $|D_k|$ be the area of D_k and t_k, d_k be the weighted measures of T_k, D_k respectively. Let φ^{kj} be a basis function of $V_{h/2}$ which is one at x^{kj} and zero at every other node of $\mathcal{T}_{h/2}$.

Lemma A.1.1 *There is a scalar ρ_{kj} with $\frac{9}{11} \leq \rho_{kj} < \frac{9}{8}$, such that*

$$\int_{D_k} \varphi^{kj} r d\mathbf{x} = \frac{1}{3} d_k \rho_{kj}.$$

Proof Define x^* as the radial coordinate of the midpoint between x^{kj} and the vertex of T_k which is not on the same edge as x^{kj} . Let r^{kj} and r^* be the radial coordinates of x^{kj} and x^* respectively. Define \bar{r}^{kj} and \tilde{r}^{kj} as the radial coordinates of

$$\bar{x}_{kj} = \frac{1}{6}(5x^{kj} + x_{kj}^*) \text{ and } \tilde{x}_{kj} = \frac{1}{9}(7x^{kj} + 2x_{kj}^*). \quad (\text{A.1})$$

Notice that

$$\int_{D_k} \varphi^{kj} r d\mathbf{x} = \frac{1}{3} |D_k| \bar{x}^{kj}$$

and

$$d_k = |D_k| \tilde{x}^{kj}.$$

Let ρ be equal to $\frac{\tilde{r}^{kj}}{\bar{r}^{kj}}$. If $0 < r^{kj} \leq r^*$, then $\bar{r}_{kj} < \tilde{r}_{kj}$ (and $\rho_{kj} \leq 1$) and $r^* \leq 2r^{kj}$. Therefore $\bar{r}_{kj} \geq r^{kj}$ and $\tilde{r}_{kj} \leq r^{kj} + \frac{2}{9}r^{kj}$, hence $\rho_{kj} \geq r^{kj} / (r^{kj} + \frac{2}{9}r^{kj}) = \frac{9}{11}$.

If $0 < r^* < r^{kj}$, then $\rho_{kj} > 1$ and $r^{kj} \leq 2r^*$, therefore $\bar{r}_{kj} < r^{kj}$ and $\tilde{r}_{kj} > \frac{1}{9}(7r^{kj} + 2\frac{1}{2}r^{kj})$ hence $\rho_{kj} < \frac{9}{8}$. ■

Lemma A.1.2 *Let \mathbf{a} and \mathbf{b} be two vectors in \mathbb{R}^2 and note $\angle(\mathbf{a}, \mathbf{b}) = \theta$ the angle between them. If $\theta_0 < \theta \leq \pi - 2\theta_0$ and $\theta_0 \leq \frac{\pi}{3}$, then for all \mathbf{c} in \mathbb{R}^2*

$$|\mathbf{c}|^2 \leq \frac{1}{\sin \frac{\theta_0}{2}} \left[\left(\frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}|} \right)^2 + \left(\frac{\mathbf{b} \cdot \mathbf{c}}{|\mathbf{b}|} \right)^2 \right].$$

Proof The minimum of $\{\angle(\mathbf{a}, \mathbf{c}), \angle(\mathbf{b}, \mathbf{c})\}$ is bigger than $\frac{\pi}{2} - \frac{\theta_0}{2}$, hence

$$\max \left\{ \frac{|\mathbf{a} \cdot \mathbf{c}|}{|\mathbf{a}|}, \frac{|\mathbf{b} \cdot \mathbf{c}|}{|\mathbf{b}|} \right\} \geq |\mathbf{c}| \cos \left(\frac{\pi}{2} - \frac{\theta_0}{2} \right) = |\mathbf{c}| \sin \frac{\theta_0}{2}.$$

■

Lemma A.1.3 *Let $(\mathcal{T}_h)_h$ be a regular family of triangulations. Then there exists a $\theta_0 \leq \frac{\pi}{3}$ independent of h , such that every couple of sides \mathbf{a} and \mathbf{b} of any triangle in \mathcal{T}_h , satisfies the hypothesis of lemma A.1.2.*

Proof Let T be a triangle in \mathcal{T}_h , note θ_i , $i = 1, 2, 3$ its angles and ρ_T the radius of the inscribed circle. Since $\theta_i < \pi$ for $i = 1, 2, 3$, $\tan \frac{\theta_i}{2} \geq \frac{\rho_T}{h_T} \geq \sigma$. Define θ_0 as $2 \arctan \sigma$. Then $\theta_i \geq \theta_0$, $3\theta_0 \leq \theta_1 + \theta_2 + \theta_3 = \pi$ and $\theta_1 = \pi - \theta_2 - \theta_3 \leq \pi - 2\theta_0$.

■

Lemma A.1.4 *For T_k and D_k defined as before and without the restriction that x^{kj} is not on $\Gamma \cup \Gamma_0$,*

$$\frac{d_k}{t_k} \geq \frac{3}{8}. \quad (\text{A.2})$$

Proof Divide the triangle T_k into four sub-triangles A_1, \dots, A_4 belonging to $\mathcal{T}_{h/2}$ and order them such that the first has the smallest weighted measure. Let r_1, \dots, r_4 be the radial coordinates of the center of gravity of the sub-triangles. The weighted measure of each sub-triangle is $a_i = 2\pi r_i |A_i|$, where $|A_i|$ is the area of A_i . Since $d_k \geq 3a_1$ and the center of gravity of T_k has its radial component less or equal to $2r_1$, $t_k \leq 2\pi(2r_1)(4|A_1|) = 8a_1$. Therefore

$$\frac{d_k}{t_k} \geq \frac{3a_1}{8a_1} \geq \frac{3}{8}.$$

■

A.2 A result on the divergence operator

Here we present the proof of a result in [BDM99]. The proof is based on the same result on classic three dimensional Sobolev spaces that the reader can find in [GR86]. In this section we use the notation

$$X = \left\{ \mathbf{v} \in H_0^1(\check{\Omega})^3 : \operatorname{div} \mathbf{v} = 0 \text{ in } \check{\Omega} \right\},$$

$$M = \left\{ q \in L^2(\check{\Omega}) : \int_{\check{\Omega}} q \, d\mathbf{x} = 0 \right\}$$

A.2. A RESULT ON THE DIVERGENCE OPERATOR

and let X^\perp be the orthogonal complement of X in $H_0^1(\check{\Omega})^3$ w.r.t. $a(\cdot, \cdot)$. Note also

$$X_a = \left\{ \mathbf{v} \in V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega) : \operatorname{div} \mathbf{v} + \frac{1}{r} v_r = 0 \text{ in } \Omega \right\},$$

$$M_a = \left\{ q \in L_1^2(\Omega) : \int_{\Omega} q r d\mathbf{x} = 0 \right\}$$

and similarly let X_a^\perp be the orthogonal complement of X_a in $V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)$ w.r.t. $a(\cdot, \cdot)$.

Proposition A.2.1 *There exists a positive constant c , such that for all p in $L_1^2(\Omega)$ with $\int_{\Omega} r p = 0$, there exists a \mathbf{u} in $V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)$ such that*

$$\operatorname{div} \mathbf{u} + \frac{1}{r} u_r = p \text{ and } \|\mathbf{u}\|_{V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)} \leq c \|p\|_{L_1^2(\Omega)}.$$

Proof Lemma 3.2 in [GR86] states that the divergence operator is an isomorphism from X^\perp onto M . This implies that for any p in M_a , there is a unique $(\operatorname{div}^{-1})\check{p}$ in X^\perp and that there exists a constant C such that

$$\frac{1}{C} \|\check{p}\|_{L^2(\check{\Omega})} \leq \|(\operatorname{div}^{-1})\check{p}\|_{H^1(\check{\Omega})^3} \leq C \|\check{p}\|_{L^2(\check{\Omega})} = C \|p\|_{L_1^2(\Omega)}.$$

Since the divergence operator is axisymmetric, $(\operatorname{div}^{-1})\check{p}$ is axisymmetric. Write $(\check{v}_r, \check{v}_\theta, \check{v}_z)$ for it and define $\check{\mathbf{v}} = (\check{v}_r, 0, \check{v}_z)$ (and \mathbf{v} consequently). Then the norm of \mathbf{v} is bounded by the norm of p , since

$$\|\mathbf{v}\|_{V_{1\Diamond}^1(\Omega) \times H_{1\Diamond}^1(\Omega)} = \|\check{\mathbf{v}}\|_{H^1(\check{\Omega})^3} \leq \|(\operatorname{div}^{-1})\check{p}\|_{H^1(\check{\Omega})^3} \leq C \|p\|_{L_1^2(\Omega)},$$

and $\operatorname{div} \check{\mathbf{v}} = \check{p} - \frac{1}{r} \partial_\theta \check{v}_\theta = \check{p}$, since v_θ is axisymmetric, which implies that its derivative in θ is zero.

This means that $\operatorname{div} \mathbf{v} + \frac{1}{r} v_r = p$, which proves proposition A.2.1. ■

Bibliography

- [AL99] S. Anicic and A. Lger, *Formulation bidimensionnelle exacte du modèle de coque 3D de Kirchhoff-Love*, C. R. Acad. Sci. Paris Sér. I Math. **329** (1999), no. 8, 741–746. MR 2000h:74049
- [ALDR03] S. Anicic, H. Le Dret, and A. Raoult, *Lemme du mouvement rigide infinitésimal en coordonnées lipschitziennes et application aux coques de régularité minimale*, C. R. Math. Acad. Sci. Paris **336** (2003), no. 4, 365–370. MR 2004c:74049
- [ALT99] R. Araya and P. Le Tallec, *An a posteriori error estimate of hierarchical type*, Rapport de recherche de l'INRIA (1999).
- [ALT01] ———, *Hierarchical a posteriori error estimates for heterogeneous incompressible elasticity*, Computational fluid and solid mechanics, Vol. 1, 2 (Cambridge, MA, 2001), Elsevier, Amsterdam, 2001, pp. 39–42. MR 1 875 872
- [Ani02] S. Anicic, *Mesure des variations infinitésimales des courbures principales d'une surface*, C. R. Math. Acad. Sci. Paris **335** (2002), no. 3, 301–306. MR 1 933 678
- [AS03] N. Arada and A. Sequeira, *A note on non-newtonian modelling of blood flow in small arteries*, Proceedings of International Conference on Modelling and Numerical Simulation in Continuum Mechanics (CIM, ed.), 2003.
- [Bab73] I. Babuška, *The finite element method with Lagrangian multipliers*, Numer. Math. **20** (1972/73), 179–192. MR 50 #11806
- [BBD] Z. Belhachmi, C. Bernardi, and S. Deparis, *Weighted Clément operator and application to the finite element discretization of the axisymmetric Stokes problem*, Submitted to Numer. Math.
- [BDM] C. Bernardi, M. Dauge, and Y. Maday, *Polynomials in Sobolev spaces, and applications*.
- [BDM99] ———, *Spectral methods for axisymmetric domains*, Gauthier-Villars, Éditions Scientifiques et Médicales Elsevier, Paris, 1999, Numerical algorithms and tests due to Mejdi Azaïez. MR 2000h:65002
- [BdV04] H. Beiro da Veiga, *On the existence of strong solutions to a coupled fluid-structure evolution problem*, J.Math.Fluid.Mechanics **21-52** (2004).
- [BF91] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, 1991. MR 92d:65187

- [BG98] C. Bernardi and V. Girault, *A local regularization operator for triangular and quadrilateral finite elements*, SIAM J. Numer. Anal. **35** (1998), no. 5, 1893–1916. MR 99g:65107
- [BMR04] C. Bernardi, Y. Maday, and F. Rapetti, *Discrétisations variationnelles de problèmes aux limites elliptiques*, Mathématiques et Applications, no. 45, Springer, 2004.
- [BP79] M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the Stokes problem in the primitive variables*, Numer. Math. **33** (1979), no. 2, 211–224. MR 81g:65145
- [Bre74] F. Brezzi, *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers*, RAIRO Anal. Numér. **8** (1974), no. R-2, 129–151. MR 51 #1540
- [BS94] P. Brown and Y. Saad, *Convergence theory of nonlinear Newton-Krylov algorithms*, SIAM Journal on Optimization **4** (1994), 297–330.
- [CB03] D. Chapelle and K. Bathe, *The finite element analysis of shells - fundamentals*, Springer Verlag, 2003.
- [CC96] R. Codina and M. Cervera, *Block iterative algorithms for non-linear coupled problems*, Advanced Computational Methods in Structural Mechanics, CIMNE, Barcelona, 1996.
- [CCG96] M. Cervera, R. Codina, and M. Galindo, *On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems*, Engrg. Comput. **13** (1996), no. 6, 4–30.
- [CE96] R. Choquet and J. Erhel, *Newton-GMRES algorithm applied to compressible flows*, Internat. J. Numer. Methods Fluids **23** (1996), no. 2, 177–190. MR 97c:76041
- [Cha85] T. F. Chan, *An approximate Newton method for coupled nonlinear systems*, SIAM J. Numer. Anal. **22** (1985), no. 5, 904–913. MR 87a:65089
- [Cho95] R. Choquet, *Etude de la méthode de Newton-GMRES. application aux équations de Navier-Stokes compressible*, Ph.D. thesis, Université de Rennes 1, 1995.
- [Cia88] P. G. Ciarlet, *Mathematical elasticity. Volume 1: three dimensional elasticity*, Studies in Mathematics and its Applications, vol. 20, North Holland, 1988.
- [Clé75] P. Clément, *Approximation by finite element functions using local regularization*, RAIRO Anal. Numér. **9** (1975), no. R-2, 77–84. MR 53 #4569
- [CN99] T. Chan and M. Ng, *Galerkin projection methods for solving multiple linear systems*, SIAM Journal on Scientific Computing **21-3** (1999), 836–850.
- [CSB⁺03] A. Corno, N. Sekarski, M.-A. Bernath, M. Payot, P. Tozzi, and L. K. von Segesser, *Pulmonary artery banding: long-term telemetric adjustment*, European Journal of Cardio-thoracic Surgery **23** (2003), 317–322.

- [CW97] T. Chan and W. Wan, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM Journal on Scientific Computing **18**- (1997), 1698–1721.
- [DDQ] S. Deparis, M. Discacciati, and A. Quarteroni, *A domain decomposition framework for fluid-structure interaction problems*, in preparation.
- [DE99] B. Desjardins and M. J. Esteban, *Existence of weak solutions for the motion of rigid bodies in a viscous fluid*, Arch. Ration. Mech. Anal. **146** (1999), no. 1, 59–71. MR 2000b:76024
- [DE00] ———, *On weak solutions for fluid-rigid structure interaction: compressible and incompressible models*, Comm. Partial Differential Equations **25** (2000), no. 7-8, 1399–1413. MR 2001g:35216
- [DEGL01] B. Desjardins, M. J. Esteban, C. Grandmont, and P. Le Tallec, *Weak solutions for a fluid-elastic structure interaction model*, Rev. Mat. Complut. **14** (2001), no. 2, 523–538. MR 2002h:76035
- [DFF03] S. Deparis, M. A. Fernández, and L. Formaggia, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions*, M2AN **37** (2003), no. 4, 601–616.
- [DS80] T. Dupont and R. Scott, *Polynomial approximation of functions in Sobolev spaces*, Math. Comp. **34** (1980), no. 150, 441–463. MR 81h:65014
- [Fer01] M. A. Fernández, *Modèles simplifiés d’interaction fluide-structure*, Ph.D. thesis, Université de Paris IX, France, 2001.
- [FFT00] T. Fanion, M. Fernández, and P. L. Tallec, *Deriving adequate formulations for fluid structure interaction problems: from ale to transpiration*, Rv. Européenne Im. Finis **9** (2000), no. 6-7, 681–708.
- [FGG01] C. Farhat, P. Geuzaine, and C. Grandmont, *The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids*, J. Comput. Phys. **174** (2001), no. 2, 669–694. MR 2002h:76104
- [FGNQ00] L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni, *Numerical treatment of defective boundary conditions for the navier-stokes equations*, Report EPFL-DMA 20.2000 and INRIA-4093 (2000).
- [FGNQ01] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni, *On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels*, Comput. Methods Appl. Mech. Engrg. **191** (2001), no. 6-7, 561–582. MR 2002h:74020
- [FL00] C. Farhat and M. Lesoinne, *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*, Comput. Methods Appl. Mech. Engrg. **182** (2000), 499–515.
- [FLL98] C. Farhat, M. Lesoinne, and P. Le Tallec, *Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity*, Comput. Methods Appl. Mech. Engrg. **157** (1998), 95–114.

- [FM03] M. A. Fernández and M. Moubachir, *An exact block-Newton algorithm for the solution of implicit time discretized coupled systems involved in fluid-structure interaction problems*, Computational fluid and solid mechanics, Elsevier, Amsterdam, 2003, pp. 1337–1341.
- [FM04] ———, *A newton method using exact Jacobians for solving fluid-structure coupling*, INRIA, Rapport de recherche num 5085 (2004).
- [FNQ02] L. Formaggia, F. Nobile, and A. Quarteroni, *A one dimensional model for blood flow: application to vascular prosthesis*, Mathematical modeling and numerical simulation in continuum mechanics (Yamaguchi, 2000), Lect. Notes Comput. Sci. Eng., vol. 19, Springer, Berlin, 2002, pp. 137–153. MR 2003c:76140
- [FSvdV98] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Accelerated inexact Newton schemes for large systems of nonlinear equations*, SIAM J. Sci. Comput. **19** (1998), no. 2, 657–674.
- [Fun55] Y. C. Fung, *An introduction to the theory of aeroelasticity*, John Wiley & Sons Inc., New York, 1955. MR 17,101g
- [FV03] L. Formaggia and A. Veneziani, *Reduced and multiscale models for the human cardiovascular system*, ch. Lecture Notes of the 7th VKI Lecture Series, “Biological Fluid Dynamics”, Von Karman Institute, 2003.
- [Ger03] J.-F. Gerbeau, *A quasi-newton method for a fluid-structure problem arising in blood flows*, Proceedings of the second M.I.T. Conference on Computational Fluid and Solid Mechanics (K. Bathe, ed.), Elsevier, 2003, pp. 1355–1357.
- [GM00] C. Grandmont and Y. Maday, *Existence for an unsteady fluid-structure interaction problem*, M2AN Math. Model. Numer. Anal. **34** (2000), no. 3, 609–636. MR 2001e:76035
- [GM01] C. Grandmont and Y. Maday, *Fluid structure interaction: A theoretical point of view*, submitted to Revue Européenne des éléments finis, 2001.
- [GR86] V. Girault and P.-A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, Berlin, 1986, Theory and algorithms. MR 88b:65129
- [Gra98] C. Grandmont, *Analyse mathématique et numérique de quelques problèmes d’interaction fluide-structure*, Ph.D. thesis, Univ. Paris 6, 1998.
- [GV03] J.-F. Gerbeau and M. Vidrascu, *A quasi-Newton algorithm based on a reduced model for fluid structure problems in blood flows*, M2AN **37** (2003), no. 4, 631–648.
- [GVF03] J.-F. Gerbeau, V. Vidrascu, and P. Frey, *Fluid-structure interaction in blood flows on geometries coming from medical imaging*, Tech. report, INRIA No 5052, 2003.
- [Han94] P. Hansbo, *Aspects of conservation in finite element flow computations*, Comput. Methods Appl. Mech. Engrg. **117** (1994), no. 3-4, 423–437. MR 96b:76065

- [Hei03] M. Heil, *An efficient solver for the fully-coupled solution of large-displacement fluid-structure interaction problems*, Comput. Methods Appl. Mech. Engrg. (2003), In press.
- [HMY⁺93] W. Huffman, R. Melvin, D. Young, F. Johnson, J. Bussoletti, M. Bieterman, and C. Hilmes, *Practical design and optimisation in computational fluids dynamics*, proceedings of the AIAA 24th Fluid Dynamics Conference, Orlando, Florida, 1993.
- [IK94] E. Isaacson and H. B. Keller, *Analysis of numerical methods*, Dover Publications Inc., New York, 1994, Corrected reprint of the 1966 original [Wiley, New York; MR **34** #924]. MR 1 280 462
- [IT69] B. Irons and R. Tuck, *A version of the Aitken accelerator for computer iteration*, Int. J. Numer. Methods Eng. **1** (1969), 275–277.
- [KK03] D. Knoll and D. Keyes, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, Journal of Computational Physics (2003).
- [Kuf80] A. Kufner, *Weighted Sobolev spaces*, Teubner-Texte zur Mathematik [Teubner Texts in Mathematics], vol. 31, BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1980, With German, French and Russian summaries. MR 84e:46029
- [Lam04] D. Lamponi, *One dimensional models for blood circulation and application to multiscale modelling*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2004.
- [LDM⁺02] K. Laganà, G. Dubini, F. Migliavacca, R. Pietrabissa, G. Pennati, A. Veneziani, and A. Quarteroni, *Multiscale modelling as a tool to prescribe realistic boundary conditions for the study of surgical procedures*, Biorheology **39** (2002), no. 3-4, 359–364.
- [LF95] M. Lesoinne and C. Farhat, *Geometric conservation laws for aeroelastic computations using unstructured dynamics meshes*, AIAA-95-1709, Presented at the 12th AIAA Computational Fluid Dynamics Conference, San Diego, june 1995.
- [Lig58] M. Lighthill, *On displacement thickness*, J. Comput. Phys. **4** (1958), 383–392.
- [LL75] L. Landau and E. Lifschitz, *Elasticity theory*, Pergamon Press, Oxford, 1975.
- [LM01] P. Le Tallec and J. Mouro, *Fluid structure interaction with large structural displacements*, Comput. Methods Appl. Mech. Engrg. **190** (2001), no. 24-25, 3039–3067.
- [LMP91] P. Luchini, M. Lupo, and A. Pozzi, *Unsteady stokes flow in a distensible pipe*, ZAMM - Z. angew. Math. Mech. **71** (1991), 367–378.
- [LT94] P. Le Tallec, *Numerical methods for nonlinear three-dimensional elasticity*, Handbook of numerical analysis, Vol. III, North-Holland, Amsterdam, 1994, pp. 465–622. MR 96b:73093

- [Med99] G. Medic, *tude mathématique des modles aux tensions de reynolds et simulation numrique d'coulements turbulents sur parois fixes et mobiles*, Ph.D. thesis, University of Paris VI, France, 1999.
- [MLL92] X. Ma, G. C. Lee, and S. G. Lu, *Numerical simulation of the propagation of nonlinear pulsatile waves in arteries*, ASME Journal of Biomechanical Engineering (1992), no. 114, 490–496.
- [MMV99] W. Mackens, J. Menck, and H. Voss, *Coupling iterative subsystem solvers*, Scientific Computing in Chemical Engineering II. Simulation, Image Processing, Optimization and Control (F. Keil, H. Mackens, W. Voss, and J. Wertherm, eds.), Springer Verlag, Berlin, Heidelberg, 1999, pp. 184–191.
- [Mou02] M. Moubachir, *Mathematical and numerical analysis of inverse and control problems for fluid-structure interaction systems*, Ph.D. thesis, cole Nationale des Ponts et Chausses, France, 2002.
- [MPD⁺01] F. Migliavacca, G. Pennati, G. Dubini, R. Fumero, R. Pietrabissa, G. Urcelay, E. L. Bove, T.-Y. Hsia, and M. R. De leval, *Modeling of the Norwood circulation: effects of shunt size, vascular resistance, and heart rate*, Am. J. Physiol Heart Circ. Physil **280** (2001).
- [MR82] B. Mercier and G. Raugel, *Résolution d'un problème aux limites dans un ouvert axisymétrique par éléments finis en r, z et séries de Fourier en θ* , RAIRO Anal. Numér. **16** (1982), no. 4, 405–461. MR 84g:65154
- [MS00] H. Matthies and J. Steindorf, *Numerical efficiency of different partitioned methods for fluid-structure interaction*, Z. Angew. Math. Mech. **2** (2000), no. 80, 557–558.
- [MS02] ———, *Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction*, Computer & Structures **80** (2002), 1991–1999.
- [MS03] ———, *Partitioned strong coupling algorithms for fluid-structure interaction*, Computer & Structures **81** (2003), 805–812.
- [MW01] D. P. Mok and W. A. Wall, *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures.*, Trends in computational structural mechanics. (K. S. W. Wall, ed.), K.U. Bletzing, CIMNE, Barcelona, 2001.
- [MWR01] D. P. Mok, W. A. Wall, and E. Ramm, *Accelerated iterative substructuring schemes for instationary uid-structure interaction*, Computational Fluid and Solid Mechanics (K. Bathe, ed.), Elsevier, 2001, pp. 1325–1328.
- [NO98] W. W. Nichols and M. F. O'Rourke, *Mcdonald's blood flow in arteries theoretical, experimental, and clinical principles*, Arnold, 1998, With a contribution from Craig Hartley.
- [Nob01] F. Nobile, *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*, Thesis n. 2548, École Polytechnique Fédérale de Lausanne (EPFL), 2001.

- [NV99] F. Nobile and A. Veneziani, *Fluid structure interaction in blood flow problems*, (1999) *SIAM Z. Angew. Math. Mech.* (1999), no. 79, 255–258.
- [Ode72] J. T. Oden, *Finite elements for nonlinear continua*, New York, 1972.
- [Par80] B. Parlett, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, *Linear Algebra Appl.* **29** (1980), 323–346.
- [PFL95] S. Piperno, C. Farhat, and B. Larrouturou, *Partitioned procedures for the transient solution of coupled aeroelastic problems.*, *Comput. Methods Appl. Mech. Engrg.* **124** (1995), no. 1-2, 79–112. MR 96c:73051
- [PS98] M. Padula and A. Sequeira, *A note on a vector transport equation with applications to non-Newtonian fluids*, *Theory of the Navier-Stokes equations*, Ser. Adv. Math. Appl. Sci., vol. 47, World Sci. Publishing, River Edge, NJ, 1998, pp. 121–126. MR 99e:35188
- [QA99] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Oxford University Press, Oxford, 1999.
- [QF02] A. Quarteroni and L. Formaggia, *Modelling of living systems*, *Handbook of Numerical Analysis*, ch. Mathematical Modelling and Numerical Simulation of the Cardiovascular System, Elsevier Science, Amsterdam, 2002, submitted.
- [QSS00] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical mathematics*, *Texts in Applied Mathematics*, vol. 37, Springer-Verlag, New York, 2000. MR 2000m:65001
- [QSV99] A. Quarteroni, F. Saleri, and A. Veneziani, *Analysis of the Yosida method for the incompressible Navier-Stokes equations*, *J. Math. Pures Appl.* (9) **78** (1999), no. 5, 473–503. MR 2000c:35190
- [QSV00] ———, *Factorization methods for the numerical approximation of Navier-Stokes equations*, *Comput. Methods Appl. Mech. Engrg.* **188** (2000), no. 1-3, 505–526. MR 2001h:76091
- [QTV00] A. Quarteroni, M. Tuveri, and A. Veneziani, *Computational Vascular Fluid Dynamics: Problems, Models and Methods*, *Comp. Vis. Science* **2** (2000), 163–197.
- [Qua93] L. Quartapelle, *Numerical solution of the incompressible Navier-Stokes equations*, *International Series of Numerical Mathematics*, vol. 113, Birkhäuser Verlag, Basel, 1993. MR 95i:76060
- [Qua02] A. Quarteroni, *Mathematical modelling of the cardiovascular system*, *Proceedings of the International Congress of Mathematicians (Beijing)* (L. Tatsien, ed.), vol. III, Higher Education Press, 2002.
- [QV94] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, *Springer Series in Computational Mathematics*, vol. 23, Springer-Verlag, Berlin, 1994. MR 95i:65005
- [Ren98] J. Renou, *Une methode eulrienne pour le calcul de forces fluide-lastiques*, Ph.D. thesis, University of Paris VI, France, 1998.

- [RH93] P. Raj and B. Harri, *Using surface transpiration with an euler method for cost-effective aerodynamic analysis*, proceedings of the AIAA 24th Fluid Dynamics Conference, Monterey, Canada, 1993.
- [RR98] C. Rey and F. Risler, *A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems*, Numerical Algorithms **17** (1998), 279–311.
- [RS03] A. M. Robertson and A. Sequeira, *A director theory approach for modeling blood flow in the arterial system*, Annals of Biomedical Engineering (2003), submitted.
- [Saa93] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput. **14** (1993), no. 2, 461–469. MR 1 204 241
- [Saa97] Y. Saad, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Math. Comp. **48** (1997), 651–662.
- [SFP01] S. Sherwin, L. Formaggia, and J. Peiró, *Computational modelling of 1d blood flow with variable mechanical properties*, Proceedings of ECCOMAS CFD 2001, ECCOMAS, September 2001, CD-ROM Edition.
- [SFPP03] S. Sherwin, V. Franke, J. Peiró, and K. Parker, *One-dimensional modelling of a vascular network in space-time variables*, J. Eng. Math **47** (2003), 217–250.
- [SG95] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM Journal on Scientific Computing **16-** (1995), 917–933.
- [SPM89] C. Smith, A. Peterson, and R. Mittra, *A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields*, IEEE Trans. Antennas and Propagation **37** (1989), 1490–1493.
- [SS86] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput. **7** (1986), 856–869.
- [SZ92] J. Sokoowski and J.-P. Zolsio, *Introduction to shape optimization*, Springer Series in Computational Mathematics, vol. 16, Springer-Verlag, Berlin, 1992.
- [Tab96] M. Tabata, *Finite element analysis of axisymmetric flow problems*, Z. Angew. Math. Mech. **76** (1996), no. suppl. 1, 171–174, ICIAM/GAMM 95 (Hamburg, 1995). MR 1 444 389
- [Tez01] T. Tezduyar, *Finite element methods for fluid dynamics with moving boundaries and interfaces*, Arch. Comput. Methods Engrg. **8** (2001), 83–130.
- [Tra64] J. F. Traub, *Iterative methods for the solution of equations*, Prentice-Hall Series in Automatic Computation, Prentice-Hall Inc., Englewood Cliffs, N.J., 1964. MR 29 #6607
- [VdV03] H. A. Van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics, vol. 13, Cambridge University Press, Cambridge, 2003. MR 1 990 752

- [Ven98] A. Veneziani, *Mathematical and numerical modeling of blood flow problems*, Ph.D. thesis, Università degli Studi di Milano, 1998.
- [Ver] C. Vergara, *Defective boundary conditions and multiscale modeling with application to blood flow simulation*, Ph.D. thesis, Politecnico di Milano, in preparation.
- [Ver84] R. Verfürth, *Error estimates for a mixed finite element approximation of the Stokes equations*, RAIRO Anal. Numér. **18** (1984), no. 2, 175–182. MR 85i:65156
- [WO97] T. Washio and C. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, ETNA **6** (1997), 271–290.
- [Yin86] L. A. Ying, *Finite element approximation to axial symmetric Stokes flow*, J. Comput. Math. **4** (1986), no. 1, 38–49. MR 88f:65211
- [Zie67] O. C. Zienkiewicz, *The finite element method in structural and continuum mechanics*, McGraw-Hill, London, 1967.
- [ZT89] O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, fourth ed., vol. 1, London, 1989.

Curriculum Vitae

Born in Sorengo (TI) on October 30th 1973.

Married, two children.

Swiss and Italian citizen.

TITLES: Dipl. Math. ETH; Mastère Spécialisé et Diplôme d'Etudes Postgrades en Ingénierie Mathématique de l'Ecole Polytechnique de Paris et de l'EPF de Lausanne.

Graduate Studies

2001-2004 PhD student in scientific computing,
1998-1999 Masters in mathematical engineering,
at the EPFL and the Ecole Polytechnique de Paris,
1997 Diploma in mathematics at the ETHZ,
1992-1997 Studies in mathematics at the ETHZ.

Languages

Italian: mother tongue,
French: fluent in speaking, good in writing,
English: fluent in speaking, good in writing,
German: fluent in speaking, good in writing,
Swiss-German: good in speaking.

Work History

1999-present Teaching assistant in mathematics at the EPFL,
Summer 1999 Internship at Olsen & Associates, Zürich,
1997-1998 Teaching assistant in mathematics at the ETHZ,
1995-1997 Student teaching assistant at the ETHZ.

Refereed Journals Publications

- Z. Belhachmi, C. Bernardi and S. Deparis, *Weighted Clément operator and application to the finite element discretization of the axisymmetric Stokes problem*, Submitted to Numer. Math.
- S. Deparis, M. Discacciati, and A. Quarteroni, *A domain decomposition framework for fluid-structure interaction problems*, in preparation.
- S. Deparis, M. A. Fernández and L. Formaggia, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions*, M2AN 37(4) 2003, pp. 601–616.
- S. Deparis, M. A. Fernández, L. Formaggia and F. Nobile, *Modified fixed point algorithm in fluid-structure interaction*, Compte Rendue de Mecanique 331 (2003) pp. 525–530.
- S. Deparis, J.-F. Gerbeau and X. Vasseur, *GMRES Preconditioning and Accelerated Quasi-Newton Algorithm and Application to Fluid Structure Interaction*, in preparation.

Refereed Conference Proceedings

- S. Deparis, Miguel A. Fernández, L. Formaggia and F. Nobile, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions*, Computational Fluid and Solid Mechanics 2003, Bathe (Editor), Elsevier Science Ltd.
- S. Deparis, J.-F. Gerbeau and X. Vasseur, *Dynamic GMRES-based preconditioner with application to fluid-structure interaction problems*, Proceedings of the Oxford 2004 ICFD conference, submitted.
- S. Deparis and C. Martini, *Superhedging Strategies and Balayage in Discrete Time*, accepted in Proceedings of the 4th Ascona Conference on Stochastic Analysis, Random Fields and Applications, Progress in Probability series, Birkhäuser Verlag 2004.

Scientific Skills

MATHEMATICS: I am working on scientific computing and numerical analysis applied to haemodynamics. Previously I was working on stochastic optimal control applied to finance.

COMPUTER. Operating systems: Linux, Unix and Windows. Programming languages: C++, Fortran, Matlab, Splus and MapleV. System manager: Linux.

Interests

My interests are mainly focused on my family, mathematics and my friends. I enjoy playing many kind of sports, in particular soccer and skiing.