CENTRO INTERNACIONAL DE MÉTODOS NUMÉRICOS EN INGENIERÍA

Incompressible Lagrangian Fluid Flow with Thermal Coupling

R. Aubry
S.R. Idelsohn
E. Oñate

Monograph CIMNE Nº 95, May 2006

Incompressible Lagrangian Fluid Flow with Thermal Coupling

R. Aubry
S.R. Idelsohn
E. Oñate

Monograph CIMNE Nº-95, May 2006

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

INTERNACIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING
Edificio C1, Campus Norte UPC
Gran Capitán s/n
08034 Barcelona, Spain
www.cimne.upc.es

First edition: May 2006

Incomprensible Lagrangian Fluid Flow with Termal Coupling
Monograph CIMNE M95

ISBN: 84-95999-92-7
Depósito legal: B-14314-2006

Abstract

A method is presented for the solution of an incompressible viscous fluid flow with heat transfer and solidification using a fully Lagrangian description of the motion. The originality of this method consists in assembling various concepts and techniques which appear naturally due to the Lagrangian formulation.

First of all, the Navier-Stokes equations of motion coupled with the Boussinesq approximation must be reformulated in the Lagrangian framework, whereas they have been mostly derived in an Eulerian context. Secondly, the Lagrangian formulation implies to follow the material particles during their motion, which means to convect the mesh in the case of the Finite Element Method (FEM), the spatial discretisation method chosen in this work. This provokes various difficulties for the mesh generation, mainly in three dimensions, whereas it eliminates the classical numerical difficulty to deal with the convective term, as much in the Navier-Stokes equations as in the energy equation. Even without the discretization of the convective term, an efficient iterative solver, which constitutes the only viable alternative for three dimensional problems, must be designed for the class of Generalized Stokes Problems (GSP), which could be able to behave well independently of the mesh Reynolds number, as it can vary greatly for coupled fluid-thermal analysis.

Moreover, it offers a natural framework to treat free-surface problems like wave breaking and rough fluid-structure contact. On one hand, the convection of the mesh during one time step after the resolution of the non-linear system provides explicitly the locus of the domain to be considered. On the other hand, fluid-to-fluid and fluid-to-wall contact, as well as the update of the domain due to the remeshing, must be accurately and efficiently performed. Finally, the solidification of the fluid coupled with its motion through a variable viscosity is considered

An efficient overall algorithm must be designed to bring the method effective, particularly in a three dimensional context, which is the ambition of this monograph. Various numerical examples are included to validate and highlight the potential of the method.

Keywords: Updated Lagrangian Formulation, Coupled Fluid-Thermal problems, Thermal Convection, Rayleigh-Bénard Instability, Incompressible Fluid Flow, Delaunay Triangulation with size constraint, solidification.

Acknowledgments

First of all, I would like to express my gratitude to my first adviser Prof. Eugenio Oñate to accept myself in the CIMNE, firstly as an engineer in a moment where I was not completely sure about what I wanted to do professionally, and secondly to begin the Ph.D two years later with this exciting subject of Lagrangian flows. The fact that very different research domains such as fluid flow, iterative solver and mesh generation had to be interpenetrated was what really interested me, and decided me to join the investigation world thereafter. During the Ph.D, Prof. Oñate always encouraged me to follow new options and this open minded attitude gave me the freedom to try and test various different things that made grow my experience in the numerical field.

I would like to thank Prof. Sergio R. Idelsohn, my second adviser for the three years that we shared during this PhD. The experience and the sympathy of Sergio were necessary to achieve the goal proposed three years ago. PhD. Facundo Del Pin played also an important part in this work, as he introduced me to the method at the beginning, and was always a big support during the doctorate, particularly these last two years.

I would like to thank the Civil Engineering School of Cataluña and the Agency for Management of University and Research Grants (AGAUR) for the economic help received during these three years through the International Graduated School of Catalonia grant.

Next, I would like to thank Prof. Ramon Codina, who also played an important role in my numerical teaching, through various courses and mainly through having taken the time to kindly answer to numerous questions.

I would like to thank the professors of the Civil Engineering School of Cataluña, Profs. Carlos Agelet de Saracibar Bosch, Xavier Oliver Olivella and Michele Chiumenti for many discussion during and after the writing and implementation of the Lagrangian description applied to fluids.

I also would like to thank the fluid team of the first floor for many fruitful discussion, Guillaume Houzeaux and Javier Principe in general, and in particular Herbert Owen Coppola for the help given to understand the level set method in an Eulerian implementation.

Finally, I would like to thank all the C6 and C7 room for having contributed to make these three years as interesting in a scientific as in a personal viewpoint, particularly Pooyan Dadvan for all the help given during the learning of the C++ and Santiago Badia for many interesting fluid discussions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The present work will describe an extension of a recent numerical method, the *Particle Finite Element Method* (PFEM) first presented in [63], applied to fluid problems with thermal coupling. This work is the continuation of a precedent thesis by F. Del Pin [37]. Various different elements will be necessary to explain and prove the efficiency and robustness of the method. Each element will be presented in details, reflecting the work done during this doctorate. The Lagrangian formulation in fluid mechanics gives rise naturally to new problems, as much in a theoretical manner as in an implementation point of view. The necessity to join all these areas of research constitutes the difficulty of this work, and evidently its most interesting aspect.

1.1 State of the art

In this section, the actual state of the art is reviewed in the main domains that constitute this work, namely the Lagrangian formulation applied to the Navier-Stokes equations, the mesh generation, the iterative solvers applied to the Generalized Stokes Problem (GSP), the classical methods to treat free surface flows and the coupled thermal flows with solidification.

1.1.1 Lagrangian formulation

The classical Navier-Stokes equations have been studied for a long time, both from the theoretical [52, 106], and numerical [59, 72, 33, 91, 40] points of view. These equations were mostly associated with an Eulerian description of motion which led to the impressive results in computational fluid dynamics (CFD) during the last 20 years. At the opposite, the Lagrangian formulation happens to be the standard formulation in structural mechanics [12, 11], because of its natural ability to follow the media during the motion, and the ease of applying boundary conditions. Basically, the two main streams in the Lagrangian formulation applied to computational mechanics are the Total Lagrangian formulation, when

one keeps the reference configuration fixed during the motion, and the Updated Lagrangian formulation, where the reference configuration changes during the motion. Both are strictly equivalent at the continuum level, corresponding to different points of view and different implementations.

In fluid mechanics, the Lagrangian formulation has been applied much more punctually. Two early references are Hirt *et al.* [60], and Fritts and Boris [47], where an unstructured mesh of triangles is used with a finite difference-like method, and a rezoning technique. Muttin *et al.* [83] develop the Lagrangian formulation to metal casting flow using P2 isoparametrical elements and a penalty technique for incompressibility. Surface tension is taken into account and a special remeshing is presented to adapt the boundaries as the fluid fills the mould. Bach *et al.* [9] consider a free surface problem with squared elements, discontinuous pressure and surface tension, and Ramaswamy [95] studies the sloshing problem with a coupled Lagrangian-ALE approach. In [66], Kawahara *et al.* compare the numerical results obtained by their method to the analytical solution of a solitary wave propagation, and in [56] Hayashi *et al.* take advantage of the fractional step formulation to simulate free surface problems. However, all these papers do not introduce the material description, and iterate on the spatial description [9, 95, 66, 56], or even do not iterate [83], assuming small perturbations of the fluid movement during the time step. Radovitzky *et al.* [92] and Malcevic *et al.* [75] present a fully non linear model, using the Picard linearization in [75], and the Newton-Raphson linearization in [92]. Nevertheless, [92] stops the simulation when the wave breaks upon itself and [75] do not consider free surfaces. A different approach is proposed by Hansbo by using a space-time finite element method [53, 55, 54], where the space-time 'slabs' may or may not be aligned along the characterictics, providing a continuous bridge from the Eulerian to the Lagrangian formulation. However, all computations are also performed without merging and splashing. These authors use in one way or another a remeshing or 'rezoning' technique when the mesh becomes too distorded, and then interpolate the results through a certain interpolant operator. Moreover, they only present results in two dimensions, with simple geometry, and without breaking waves, the free-surface being smooth. It is also remarkable that, except the paper of Malcevic *et al.* [75], the Lagrangian formulation is only associated with free surface problems. Recently, [63] proposed a new method where free surfaces, breaking waves and fluid-structures interaction are fully considered. This model was adapted in Aubry *et al.* [6, 7] to thermal problems with a different reference configuration, which constitutes the basis of this work, and presented in [5, 4]. In [84, 85], a global presentation of this method is presented. In [88], various applications of the method are reported.

1.1.2 Mesh generation

Concerning the automatic mesh generation field, which will constitute an important part of the method, this area of research began only twenty years ago. If the mesh generation was the bottleneck of the computation in the past, this is definitively not true nowadays, producing good quality meshes of a few mil-

ions elements on a PC in a few minutes [46, 72, 49, 108]. This field can be divided into three main parts: the Delaunay-based mesh generation with the pioneering work of Hermeline [58], Bowyer [18], and Watson [113], the advancing front method with George [48] and more recently Löhner *et al.* [73, 71], and the octree-based methods with Yerry et al. [116]. For the Delaunay-based mesh generation in the finite element context, Weatherill *et al.* [115, 114] propose alternatives to the George method of boundary recovery. Joe develops a completely different approach trying to extend the results of Lawson [69] in two dimensions to the three-dimensional case. In [64], he conjectures that his algorithm constructs always a Delaunay triangulation, and he proves it in [65]. For the frontal method, apart from the work of Löhner, George *et al.* also propose their vision of a frontal method in [50]. Möller *et al.* [81] present an advancing front method to generate particularly strechted elements in specified directions. Finally, Frey *et al.* [45] couple both approaches to use the accurate placement of the points of the frontal method in the classical Delaunay triangulation in a Delaunay-advancing front fashion, by defining as a front all the elements that are considered as bad with respect to a given criterion, and Mavriplis [77], and more recently Radovitzky and Ortiz [93] use the connectivity efficiency of the Delaunay triangulation in a coupled frontal-Delaunay approach.

1.1.3 Generalized Stokes flow and fractional step

Iterative solvers are at the heart of the resolution of flow solvers. Iterative solvers are characterized by a low amount of storage requirement and good convergence properties for well conditioned systems [100, 8, 80]. Iterative solvers applied to the generalized Stokes flow where time discretization is taken into account is an active domain of research [43], mainly due to the saddle-point nature of the Generalized Stokes Problem (GSP). In order to solve a Stokes-like problem with iterative methods, two main approaches can be distinguished, whether the problem is solved as a whole, or 'coupled', or if it is solved in a segregated way, for the pressure and then for the velocity [14]. In the first category, two possibilities for a Krylov-based iterative solver which take into account the symmetry of the problem are the clever MINRES and SYMMLQ [89]. The Conjugate Gradient can break down in case of an indefinite matrix as a division by zero can occur. Page and Saunders noticed that the Conjugate Gradient was a Lanczos method where the coefficients of the Conjugate Gradient are obtained by a Choleski factorization of the coefficients created by the Lanczos process. For indefinite systems, the Choleski factorization may fail, so that SYMMLQ is based on an LQ factorization of matrix of the Lanczos coefficients, and MINRES minimizes the Euclidian norm of the residual of the equation using the least-square solution instead of minimising the A-norm of the error as in the Conjugate Gradient. In the second category, the Uzawa method [1], or Preconditioned Conjugate Gradient Uzawa (PCGU) variant, allows to replace a minimisation constrained problem due to incompressibility with a sequence of minimization problems without constrains, in fact the dual problem on the pressure [41, 44]. Nonetheless, it appears that in both approaches, the bulk

of the computation is concentrated on a good approximation of the Pressure Schur Complement (PSC) [109], which mimics the Uzawa operator at the discrete level. One of the main differences between both methods is due to the fact that the first only requires a spectrally equivalent operator of the velocity operator, whereas the second requires its accurate action. For example, as noted in [111], the PCGU does not converge for a fixed number of inner iterations for one multigrid V-cycle whereas the MINRES converges at a rate independent of h for a Stokes problem. In fact, four iterations are necessary to obtain convergence of the PCGU.

The first category gave rise to a vast literature on block preconditioning of the Stokes and generalized Stokes problems. The block preconditioning allows to taken advantage of the knowledge of the Partial Differential Equations (PDE) properties at hand. The first paper applying the MINRES solver to the saddle point problem with block preconditioners is from Rusten and Winther [99], which solves a mixed method for second-order elliptic problems and the Stokes problem. It also gives some results on the relationship between the eigenvalue distribution of each block, and the one of the whole system. In two papers by Wathen and Sylvester [112, 102], block diagonal and block preconditioners are considered and a solid theoretical basis is settled to show the importance of the preconditioning on the MINRES method. If preconditioning is important to decrease the condition number in the Conjugate Gradient method, it is more important to cluster the eigenvalues in case of the MINRES method, as the eigenvalues are distributed in both positive and negative part of the real axis, and it is easier to find a polynomial which cancels at the eigenvalues, and whose norm will give the rate of convergence of the method [111]. Then Ramage and Wathen [94] compare a two-level Uzawa-like iteration with the MINRES approach, implemented as a hybrid ORTHOMIN-ORTHODIR solver [2] with block diagonal preconditioners, reporting a comparable asymptotic rate for both methods but with a slightly lower constant for the MINRES method. Finally Bramble and Pasciak [19], and particularly Mardal and Winther [76] proved that using appropriate approximations of the PSC, a uniform preconditioner, namely a preconditioner whose rate of convergence is independent of the parameters, is obtained for the MINRES method.

The second category mainly consists in slight modifications of the basic Uzawa method with appropriate preconditioners. The Uzawa method relies on an inner loop where the velocity system is solved, and an outer loop where the pressure residual is decreased. The velocity system must be solved accurately as it is responsible of the orthogonality of the search directions, and then on the convergence of the whole system. Atanga and Sylvester [3] compare the performances of an iterated penalty method, a Conjugate Gradient Uzawa method and a PCG applied to the monolithic system, even if it is not positive definite, giving preference to the Uzawa approach. Bank *et al.* [10] reformulate the saddle point problem in order to approximate both the action of the inner system and of the outer system. Elman and Golub [42] consider the basic Uzawa method with a fixed parameter and propose to approximate the action of the inverse of the velocity system ..y using a tolerance on the inner system depending of the

residual of the outer system. In the paper from Bramble *et al.* [20], distinction is made if the approximation of the inner system is performed by a preconditioner or by a non linear approximation, varying the tolerance of the linear inner solver. It provides more general results on the convergence, supposing that the preconditioners for the inner and the outer iteration are uniforms. However, the second category really began to be considered as a serious option with the breakthrough article of Cahouet and Chabard [21], giving rise to all the next applications [15, 36, 68, 22, 117, 74].

Finally, another common way to solve the GSP in an approximate way is through the use of fractional step methods. Initially, the method was developped for time dependent problems by Témam [105] and Chorin [26], providing a formal first order in time fractional error. This error was latter improved by Van Kan [110], introducing the pressure of the last time step in the first equation, also known as the incremental pressure correction scheme. The fractional step has always given rise to polemic discussions on pressure boundary conditions [96, 107] and velocity boundary conditions for the intermediate velocity [67]. In order to bypass this delicate question, Blair Perot reinterpreted the fractional step as an incomplete block LU factorization, so that spatial discretization and boundary conditions are already discretized in the algebraic system. This point of view opened the way to recent publications like Quarteroni *et al.* [90], where the discrete and the continuous fractional step are compared, Codina [27], where a study of the pressure stability is performed for the first and second order algebraic splitting, and the stability properties of the classical fractional step are highlighted, Henriksen and Holmes [57], where high-order algebraic splittings are discussed, and Saleri and al. [101], where different approximations are tested, relying on the block LU decomposition. The book of Turek [109] was the first to really highlight the parallelism between the PCGU and the fractional step, which allows to understand boundary condition problems, and the real range of application of the classical fractional step. This is the point of view adopted in this work.

1.1.4 Free surface flow

The first problem dealing with a Lagrangian formulation for fluids is the need for a constant remeshing due to the severe distortion of the mesh as the nodes move in time. Particle type methods offer an appealing alternative to this remeshing and have been used extensively. A precursor in this field was Monoghan [51] for the treatment of astrophysical hydrodynamic problem with the so called *Smooth Particle Hydrodynamics Method* (SPH). Kernel approximations are used in the SPH method to interpolate the unknowns.

In order to avoid the frequent remeshing burden, other methods using finite elements have also been developped. The volume of fluid (VOF) method and its variants are commonly used for the analysis of free surfaces problems [61]. The idea is to introduce a scalar function that is transported by the velocity field. This function is defined on all the volume, and a certain isocontour determines the free-surface. It has been extensively used in mould filling by Ravindran *et al.*

[97]. The free surface tracking can also be performed by a Lagrangian interface technique as proposed in Cruchaga *et al.* [32] and [31]. Here, the idea consists in updating in a Lagrangian manner the interface position with the end-of-step velocity. The elements affected by the interface change are refined to describe accurately the free-surface, and a balance of mass is computed to guarantee the global mass conservation. Finally, the Arbitrary Lagrangian-Eulerian (ALE) method is another alternative to track the free-surface boundary, which aim is to take advantage of both the Lagrangian and Eulerian formulation, as presented in [39] and [62]. In the ALE approach, the domain changes its shape to accomodate the dynamic interfaces. The mesh follows the interface motion near the moving boundary, but is fixed on far-field boundaries. In between, the mesh movement is computed solving an elastic problem to minimise the deformation of the mesh. However, if the boundary motion is very large, frequent remeshings are also needed. Furthermore, as the mesh does not follow the particules, the convective term still exists, driven by the mesh velocity. Reference [38] presents the PFEM method coupled with an ALE approach to solve free surface and fluid-object interactions problems.

1.1.5 Coupled thermal flow with solidification

For the solidification problem or Stefan problem, basically two main streams of work appear, as is the case in free-surface problems, whether the mesh follows the solidification front, namely the front tracking methods, or the front goes through the mesh, namely front capturing methods. An early reference of a finite element method applied to front tracking is the one of Bonnerot *et al.* [16] extended later in [17], where a space-time finite element method is proposed. The velocity of the front is computed from the last known temperature and the front updated. Recently, Mc Daniel *et al.* [79] propose a Least-Square formulation including convection effects, moving the mesh with transfinite interpolation technique. The classical drawbacks of the moving mesh methods are the same as the Lagrangian formulation in a flow problem, namely intense remeshing, possible difficulties to deal with appearing and desappearing phases, and non-smooth interfaces.

In the front capturing context, which offers the largest literature, various attempts were first made to extend the thermal problem to latent heat effects so that only few modifications would have to be done in a nonlinear heat transfer code. The main difficulty is created by the fact that, in the case of an isothermal phase change, the enthalpy function presents a strong discontinuity near the phase change interface. In [82], Morgan *et al.* extend a previous work [28], and apply the enthapy method to incorporate the latent heat effect in the thermal capacity coefficient by applying a smoothing on the enthapy function. The main drawback is due to the fact that small time steps must be performed in order to capture the phase change. In [98], the latent heat effects are included in the residual of the linearized heat equation as an additional source term forcing the analysis to follow the H-T curve. From the point of view of the linearization, the Newton-Raphson 2^{nd} order is lost and it can be seen that the method is

not weakly conservative. In a serie of papers Crivelli, Storti and Idelsohn [29, 103, 104] advocate to direcly integrate numerically the discontinuous enthalpy function on the elements where the phase change occurs. The method is firstly applied to interface with straight sides in [29], and extended to curved interfaces in [103]. In [104] they derive a Jacobian matrix expression which reaches the 2^{nd} order convergence of the Newton method for materials with constant properties in each phase. Finally, in [25] and [23], Celentano *et al.* present a different way of taking into account the latent heat transfer by mainly approximating the tangent latent heat matrix by its secant version. It is the approach followed in the present work.

In the thermally coupled fluid simulation with phase change area of research with front capturing method, different approaches were presented, but the difficulty is more related to the tracking of the free-surface than to the phase change. Cruchaga *et al.* [30] and Celentano [24] couple the velocity field to the temperature field with solidification through the Boussinesq approximation. They solve a staggered system with a Newton-Raphson-type algorithm. However, the effect of the free surface is not taken into account. Lewis *et al.* [70] use the pseudo-concentration method to track the free-surface, and consider solidification effects during mould filling. In Mcbride *et al.* [78], the free-surface is neither considered but a porous medium law is introduced to model the dentric zone and the fluid satisfies the Boussinesq approximation. Dantzig [34] presents an enthalpy method, which provides a very good pattern of convection cells on the Gallion melting example. Once more, all these paper only present results in a two dimensional context.

1.2 Objectives

In this work, we aim at developing a robust, stable and efficient method in three dimensions which can be able to deal with actual industrial problems such as breaking waves and mould filling. The last problem involves the computation of the free surface with an intense heat transfer between the mould and the casting, and rough contact during the filling. Real problems all involve three dimensional computations now affordable with PC's. Nevertheless, the algorithms used should be taylored with special care and velocity and low storage are evidently the two main contradictorial goals.

The first ingredient to achieve this goal relies on a correct theoretical derivation of the classical equations of motion used in fluid mechanics, the Navier-Stokes equations, in a Lagrangian framework. In view of the application, the heat equation should also be recast in this context and the coupling between both sets of equation be highlighted. In a numerical point of view, we aim at treating in an original way the classical problem of convection, due to the non linear term in the Eulerian formulation of the Navier-Stokes equations, but equally present in the heat equation with transport, which has produced so many literature the last four decades. Moreover, we are especially interested in solving free-surface problems, which may present some difficulties when solved

in an Eulerian formulation.

Respect to the mesh generation, we will develop the Delaunay-based mesh generation, and we will also explain why this is the best choice regarding to the application in fluid mechanics. Moreover, we will present in three dimensions an isotropic controled mesh generation, with a given size distribution in space. Indeed, the resources needed by three dimensional computations can be very effectively reduced by an adequate distribution of the size of elements. However, we do not use in this work any a priori or a posteriori error estimate to construct this size map but we suppose that the user has an idea of the problem to be solved, and more practically of the location of the possible quick variations of quantities of interest.

1.3 Contents

The three main ingredients of this work are the numerical algorithm to solve the Navier-Stokes and the heat equations in a Lagrangian formulation, the mesh generation, and the free-surface treatment. All are closely related; the mesh should evidently have good properties for the numerical resolution, but the velocity solution will be responsible of the placement of the nodes for the mesh generation and the locus of the free surface.

After this introduction, the basis of the Lagrangian and Eulerian descriptions are recalled and applied to the heat and Navier-Stokes equations and their weak expressions are derived in a form suitable for the non linear coupled problem. In Chapter 3, the mesh generation is tackled from the Delaunay triangulation, implemented through the Delaunay kernel which constitutes the heart of the connecting scheme, to the boundary conforming process, which is the critical point of the method, the node creation on a given size distribution, and the optimization algorithm. Chapter 4 presents the discretization of the considered equations in space and time, with particular emphasis in the numerical resolution of the Generalized Stokes Problem obtained, as no convective term appears in the discretization. Compatible spaces, mass conservation and discretization with different mixed elements are considered. The relationship and implementation of fractional steps, algebraic splitting and the preconditioned Uzawa method are discussed in details. Then, the description of the free surface is considered in Chapter 5. Various classical discretizations are discussed and a proposition is made which blends efficiency and accuracy. Chapter 6 deals with thermally coupled flows and the introduction of the solidification problem coupled with the fluid flow. The numerical method to treat the solidification discretization is not new but is applied in a new context. Finally, a conclusion closes the present work. Possible extensions of the method are then discussed, as new application fields are then opened.

The program used for all the proposed example is written in C++, but does not use all the complex inheritance of the C++ and is almost a C implementation. It consists in about 80 files and about 100 000 lines of code, one third being the mesh generator. Files are usually written for the two dimensional and

the three dimensional case separately, mainly because of the huge difference in the mesh generator in both cases, and are compiled independently. Three post processors were used in this work:

- the GiD pre and post processor: *www.gid.cimne.com*
- the medit software: *www.ann.jussieu.fr/~frey/logiciels/medit.html*
- the opendx postprocessor: *www.opendx.org*

1.4 Improvements respect to previous work

Being the logical continuation of the work realized by F. Del Pin, we present the various improvements performed during these three years:

- Introduction of the thermal coupled flow
- Introduction of the solidification problem
- Meshing performed with only tetrahedras
- Recovery of the wall boundaries mesh
- A complete second order in time iterative solver for the fluid flow
- Improvements of the mass conservation
- Improvement of the free surface
- Experience with mixed elements

These improvements and extensions were mainly motivated by problems encountered during the analysis of numerous examples during these three years, and by the extension of the method to new domains of applications.

Bibliography

[1] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming*. Stanford University Press, 1958.

[2] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor. A taxonomy for conjugate gradient methods. *SIAM J. Numer. Anal.*, 27(6):1542–1568, 1990.

[3] J. Atanga and D. Silvester. Iterative methods for stabilized mixed velocity-pressure finite elements. *Int. J. Num. Meth. Fluids*, 14:71–81, 1992.

[4] R. Aubry, S.R. Idelsohn, and E. Oñate. Particle Finite Element Method applied to mould filling. In R. Ohayon, J.P. Grellier, and A. Rassineux, editors, *Septième Colloque National en Calcul des Structures, Giens*, pages 385–390. Hermes, Lavoisier, 2004.

[5] R. Aubry, S.R. Idelsohn, and E. Oñate. Particle Finite Element Method with thermal coupling. In Z.H. Yao, M.W. Yuan, and W.X. Zhong, editors, *Proceedings of the Sixth World Congress on Computational Mechanics, Beijing, China*, pages 205–210. Tsinghua University Press and Springer Verlag, 2004.

[6] R. Aubry, S.R. Idelsohn, and E. Oñate. Particle Finite Element Method in fluid mechanics including thermal convection-diffusion. *Computer & Structures*, 83:1459–1475, 2005.

[7] R. Aubry, S.R. Idelsohn, and E. Oñate. Fractional step like schemes for free surface problems with thermal coupling using the Lagrangian PFEM. *Computational Mechanics*, 2006. submitted.

[8] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, Cambridge, 1994.

[9] P. Bach and O. Hassager. An algorithm for the use of the Lagrangian specification in newtonian fluid mechanics and applications to free-surface flow. *J.Fluid Mech.*, 152:173–190, 1985.

[10] R.E. Bank, B.D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Numer. Math.*, 56:645–666, 1990.

[11] K.J. Bathe. *Finite Element Procedures*. Prentice Hall, 1995.

[12] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[13] T. Belytschko, Y. Liu, and L. Gu. Element free Galerkin methods. *Int. J. Num. Meth. Eng.*, 37:229–256, 1994.

[14] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[15] F. Bertrand and P.A. Tanguy. Krylov-based Uzawa algorithms for the solution of the Stokes equations using discontinuous-pressure tetrahedral finite elements. *J. Comput. Phys.*, 181(2):617–638, 2002.

[16] R. Bonnerot and P. Jamet. A second order finite element method for the one-dimensional Stefan problem. *Int. J. Num. Meth. Eng.*, 8:811–820, 1974.

[17] R. Bonnerot and P. Jamet. Numerical computation of the free boundary for the two-dimensional Stefan problem with space-time finite elements. *J. Comp. Phys.*, 25:163–181, 1977.

[18] A. Bowyer. Computing Dirichlet tessellations. *Comput. J.*, 24:162–166, 1981.

[19] J.H. Bramble and J.E. Pasciak. Iterative techniques for time dependent Stokes problems. *Comp. Math. Applic.*, 33:13–30, 1997.

[20] J.H. Bramble, J.E. Pasciak, and A.T. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM J. Num. Anal.*, 34:1072–1092, 1997.

[21] J. Cahouet and J.P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. Num. Meth. Fluids*, 8:869–895, 1988.

[22] R. Carriere and D. Jeandel. A 3D finite element method for the simulation of thermoconvective flows and its performances on a vector-parallel computer. *Int. J. Num. Meth. Fluids*, 12:929–946, 1991.

[23] D. Celentano. *Un modelo thermomecánico para problemas de solidificación de metales.* PhD thesis, UPC, 1994.

[24] D. Celentano. A finite element formulation for phase-change problems with advective effects. *Comm. Num. Meth. Eng.*, 14:719–730, 1998.

[25] D Celentano, E. Oñate, and S. Oller. A temperature-based formulation for finite element analysis of generalized phase-change problems. *Int. J. Num. Meth. Eng.*, 37:3441–3465, 1994.

[26] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[27] R. Codina. Pressure stability in fractional step finite element methods for incompressible flows. *J. Comput. Phys.*, 190:1579–1599, 2000.

[28] G. Comini, D. Del Giudice, R.W. Lewis, and O.C. Zienkiewicz. Finite element solution of nonlinear heat conduction problems with special reference to phase change. *Int. J. Num. Meth. Eng.*, 8:613–624, 1974.

[29] L.A. Crivelli and S.R. Idelsohn. A temperature-based finite element solution for phase-change problems. *Int. J. Num. Meth. Eng.*, 23:99–119, 1986.

[30] M. Cruchaga and D. Celentano. A finite element thermally coupled flow formulation for phase-change problems. *Int. J. Num. Meth. Eng.*, 34:279–305, 2000.

[31] M. Cruchaga, D. Celentano, and T. Tezduyar. A moving Lagrangian interface technique for flow computations over fixed meshes. *Comp. Meth. Appl. Mech. Eng.*, 191:525–543, 2001.

[32] M. Cruchaga, D. Celentano, and T. Tezduyar. Computation of mould filling processes with a moving Lagrangian interface technique. *Comm. Num. Meth. Eng.*, 18:483–493, 2002.

[33] C. Cuvelier and A. Segal. *Finite Element Methods and Navier-Stokes Equations*. D. Reidel Publishing Company, 1988.

[34] J.A. Dantzig. Modelling liquid-solid phase changes with melt convection. *Int. J. Num. Meth. Eng.*, 28:1769–1785, 1989.

[35] S. De and K.J. Bathe. The method of finite sphere with improved numerical integration. *Computer and Structures*, 79:2183–2196, 2001.

[36] E.J. Dean and R. Glowinski. On some finite element methods for the numerical simulation of incompressible viscous flow. In M.D. Gunzburger and R.A. Nicolaides, editors, *Incompressible computational fluid dynamics*, pages 17–65. Cambridge University Press, Cambridge, 1993.

[37] F. Del Pin. *The Meshless Finite Element Method Applied to a Lagrangian Particle Formulation of fluid Flows*. PhD thesis, Santa Fe, 2003.

[38] F. Del Pin, S.R. Idelsohn, E. Oñate, and R. Aubry. The ALE/Lagrangian Particle Finite Element Method : a new approach to computation of free-surface flows and fluid-object interactions. *Computer and Fluids*, 2005. Accepted to publication.

[39] J. Donea. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions. *Comp. Meth. Appl. Mech. Eng.*, 33:689–723, 1982.

[40] J. Donea and A. Huerta. *Finite Element Methods for flow problems*. John Wiley& Sons, 2003.

[41] I. Ekeland and R. Témam. *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, 1999.

[42] H.C. Elman and G.H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Num. Anal.*, 31:1645–1661, 1994.

[43] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, 2005.

[44] M. Fortin and R. Glowinski. *Augmented Lagrangian Methods*. North-Holland, Amsterdam, 1983.

[45] P.J. Frey, H. Borouchaki, and P.L. George. 3D Delaunay mesh generation coupled with advancing-front approach. *Comp. Meth. Appl. Mech. Eng.*, 157:115–131, 1998.

[46] P.J. Frey and P.L. George. *Maillages, applications aux elements finis*. Hermes, 1999.

[47] M.J. Fritts and J.P. Boris. The Lagrangian solution of the transient problems in hydrodynamics using a triangular mesh. *J. Comput. Phys.*, 31:173–215, 1979.

[48] A. George. *Computer implementation of the finite element method.* PhD thesis, Dept. of Computer Science, Stanford Univ., 1971.

[49] P.L. George, editor. *Maillage et adaptation.* Lavoisier, 2001. In French.

[50] P.L. George and E. Seveno. The advancing-front mesh generation method revisited. *Int. J. Num. Meth. Eng.*, 37:3605–3619, 1994.

[51] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics, theory and application to non-spherical stars. *Mon. Nat. Roy. Astr. Soc.*, 181:375–389, 1977.

[52] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equation.* Springer, 1986.

[53] P. Hansbo. The characteristic streamline diffusion method for the time-dependent incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 99:171–186, 1992.

[54] P. Hansbo. Lagrangian incompressible flow computations in three dimensions by use of space time finite elements. *Int. J. Num. Meth. Fluids.*, 20:989–1001, 1995.

[55] P. Hansbo. A Crank-Nicolson type space-time finite element method for computing on moving meshes. *J. Comput. Phys.*, 159:274–289, 2000.

[56] M. Hayashi, K. Hatanaka, and M. Kawahara. Lagrangian finite element method for free surface Navier-Stokes flow using fractional step methods. *Int. J. Num. Meth. Fluids*, 13:805–840, 1991.

[57] M. Henriksen and J. Holmen. Algebraic splitting for Navier-Stokes equations. *J. Comput. Phys.*, 175:438–453, 2002.

[58] F. Hermeline. *Une méthode automatique de maillage en dimension n.* PhD thesis, Paris VI, 1980.

[59] C. Hirsch. *Numerical Computation of Internal and External Flows, Volume I & II.* Numerical Methods in Engineering, Wiley, New York, 1995.

[60] C.W. Hirt, J.L. Cook, and T.D. Butler. A Lagrangian method for calculation of the dynamics of an incompressible fluid with a free surface. *J. Comput. Phys.*, 5:103–124, 1970.

[61] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39:201–225, 1981.

[62] A. Huerta and F. Casadei. New ALE applications in non-linear fast transient solid dynamics. *Engng. Comput.*, 11:317–345, 1994.

[63] S.R. Idelsohn, E. Oñate, and F. Del Pin. The Particle Finite Element Method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Num. Meth. Eng.*, 61(7):964–989, 2004.

[64] B. Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Stat. Comput.*, 10:718–741, 1989.

[65] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.

[66] M. Kawahara and A. Anjyu. Lagrangian finite element method for solitary wave propagation. *Computational Mechanics*, 3:299–307, 1988.

[67] J. Kim and P. Moin. Application of the fractional-step method to the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.

[68] G.M. Kobelkov and M.A. Olshanskii. Effective preconditioning of Uzawa type schemes for a generalized Stokes problem. *Numer. Math.*, 86:443–470, 2000.

[69] C.L. Lawson. Transforming triangulations. *Discrete Math.*, 3:365–372, 1972.

[70] R.W. Lewis and K. Ravindran. Finite element modelling of solidification effects in mould filling. *Int. J. Num. Meth. Eng.*, 47:29–59, 2000.

[71] R. Löhner. Extensions and improvements of the advancing front grid generation technique. *Comm. Num. Meth. Eng.*, 12:683–702, 1996.

[72] R. Löhner. *Applied CFD Techniques*. Wiley, 2001.

[73] R. Löhner and P. Parikh. Three-dimensional grid generation by the advancing front method. *Int. J. Num. Meth. Fluids*, 8:1135–1149, 1988.

[74] Y. Maday, D. Meiron, A.T. Patera, and E.M. Rønquist. Analysis of iterative methods for the steady and unsteady Stokes problem: application to spectral element discretizations. *SIAM J. Sci. Comput.*, 14:310–337, 1993.

[75] I. Malcevic and O. Ghattas. Dynamic-mesh finite element method for Lagrangian computational fluid dynamics. *Finite Element in Analysis and Design*, 38:965–982, 2002.

[76] K.A. Mardal and R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numerische Mathematik*, 98(2):305–327, 2004.

[77] D.J. Mavriplis. An advancing front Delaunay triangulation algorithm designed for robustness. *J. Comput. Phys.*, 117:90–101, 1995.

[78] E. Mcbride, J.C. Heinrich, and D.R. Poirier. Numerical simulation of incompressible flow driven by density variations during phase change. *Int. J. Num. Meth. Fluids*, 31:787–800, 1999.

[79] D.J. McDaniel and N. Zabaras. A least-square front-tracking finite element method analysis of phase change with natural convection. *Int. J. Num. Meth. Eng.*, 37:2755–2777, 1994.

[80] G. Meurant. *Computer solution of large linear systems*, volume 28 of *Studies in Mathematics and Its Applications*. North-Holland, 1999.

[81] P. Möller and P. Hansbo. On advancing-front mesh generation in three dimensions. *Int. J. Num. Meth. Eng.*, 38:3551–3569, 1995.

[82] K. Morgan, R.W. Lewis, and O.C. Zienkiewicz. An improved algorithm for heat conduction problems with phase change. *Int. J. Num. Meth. Eng.*, 11:1191–1195, 1977.

[83] F. Muttin, T. Coupez, M. Bellet, and J.L. Chenot. Lagrangian finite-element analysis of time dependent viscous free-surface flow using an automatic remeshing technique:application to metal casting flow. *Int. J. Num. Meth. Eng.*, 36:2001–2015, 1993.

[84] E. Oñate, S.R. Idelsohn, F. Del Pin, and R. Aubry. The Particle Finite Element Method: an overview. *Int. J. Comp. Meth.*, 1(2):267–307, 2004.

[85] E. Oñate, S.R. Idelsohn, F. Del Pin, and R. Aubry. Possibilities of the Particle Finite Element Method for fluid structure interaction problems with free surface waves. *REEF*, 13:637–666, 2004.

[86] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, and R.L. Taylor. A finite point method in computational mechanics. aplication to convective transport and fluid flow. *Int. J. Num. Meth. Eng.*, 39:3839–3886, 1996.

[87] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, and C. Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Comp. Meth. Appl. Mech. Eng.*, 39:315–346, 1996.

[88] E. Oñate, J. Rojek, M. Chiumenti, S.R. Idelsohn, F. Del Pin, and R. Aubry. Advances in stabilized finite element and particule methods for bulk forming processes. *Comp. Meth. Appl. Mech. Eng.*, 2004. Submitted to.

[89] C.C. Page and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *Siam J. Numer. Anal.*, 12(4):617–629, 1975.

[90] A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 188:505–526, 2000.

[91] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin, 1997.

[92] R. Radovitzky and M. Ortiz. Lagrangian finite element analysis of Newtonian fluid flows. *Int. J. Num. Meth. Eng.*, 43:607–619, 1997.

[93] R. Radovitzky and M. Ortiz. Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front-Delaunay triangulation, 2000.

[94] A. Ramage and A. Wathen. Iterative solution techniques for the Stokes and Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, 19:67–83, 1994.

[95] B. Ramaswamy. Numerical simulation of unsteady viscous free surface flow. *J. Comput. Phys.*, 90(2):396–430, 1990.

[96] R. Rannacher. On Chorin's projection method for the incompressible Navier-Stokes equations. In *The Navier-Stokes Equations II. Theory and Numerical Methods, Lecture Notes in Mathematics*. Springer-Verlag, Berlin/New York, 1991.

[97] K. Ravindran and R.W. Lewis. Finite element simulation of metal casting. *Finite Element in Analysis and Design*, 31:99–116, 1998.

[98] W.D. Rolph III and K.J. Bathe. An efficient algorithm for analysis of nonlinear heat transfer with phase changes. *Int. J. Num. Meth. Eng.*, 18:119–134, 1982.

[99] T. Rusten and R. Winther. A preconditioned iterative method for saddle point problems. *SIAM J. Matrix Anal. Appl.*, 13:887–904, 1992.

[100] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003. Second edition.

[101] F. Saleri and A. Veneziani. Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations. *SIAM J. Num. Anal.*, 43:174–194, 2005.

[102] D.J. Silvester and A.J. Wathen. Fast iterative solution of stabilised Stokes systems Part II: Using general block preconditioners. *SIAM J. Numer Anal.*, 31:1352–1367, 1994.

[103] M. Storti, L.A. Crivelli, and S.R. Idelsohn. Making curved interfaces straight in phase-change problems. *Int. J. Num. Meth. Eng.*, 24:375–392, 1987.

[104] M. Storti, L.A. Crivelli, and S.R. Idelsohn. An efficient tangent scheme for solving phase-change problems. *Comp. Meth. App. Mech. Eng.*, 66:65–86, 1988.

[105] R. Témam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires. *Arch. Rat. Mech. Anal.*, 32:135–153, 1969.

[106] R. Témam. *Navier-Stokes Equations*. North-Holland, 1984.

[107] R. Témam. Remark on the pressure boundary condition for the projection method. *Theor. Comput. Fluid Dyn.*, 3:181–184, 1991.

[108] J.F. Thompson, S. Bharat, and N. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, 1999.

[109] S. Turek. *Efficient solvers for incompressible flow problems - an algorithmic and computational approach*. Springer, Berlin, Heidelberg, New York, 1999.

[110] J. Van Kan. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comput.*, 7:870–891, 1986.

[111] A. Wathen, B. Fischer, and D. Silvester. The convergence rate of the minimal residual method for the Stokes problem. *Numer. Math.*, 71(1):121–134, 1995.

[112] A.J. Wathen and D.J. Silvester. Fast iterative solution of stabilised Stokes systems Part I: Using simple diagonal preconditioners. *SIAM J. Numer Anal.*, 30:630–649, 1993.

[113] D.F. Watson. Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes. *Comput. J.*, 24:167–172, 1981.

[114] N.P. Weatherill. The integrity of geometrical boundaries in the two-dimensional Delaunay triangulation. *Comm. Appl. Num. Meth.*, 6:101–109, 1990.

[115] N.P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation. *Int. J. Num. Meth. Eng.*, 37:2005–2039, 1994.

[116] M.A. Yerry and M.S. Shephard. Automatic three dimensional mesh generation by the modified octree technique. *Int. J. Num. Meth. Eng.*, 20:1965–1990, 1984.

[117] R.Q.N. Zhou. Preconditioned finite element algorithms for 3D Stokes flow. *Int. J. Num. Meth. Fluids*, 17:667–685, 1993.

Chapter 2

The Lagrangian description applied to fluid mechanics

The purpose of this chapter is to give the classical tools to write the Navier-Stokes equations in a Lagrangian manner, both in the strong and in the variational form. The Navier-Stokes equations have been classically associated with an Eulerian description of motion, where the velocity is known on each spatial point of the studied domain. The Lagrangian formulation offers a different point of view, as each particle knows its velocity. The Lagrangian formulation has been extensively used in solid mechanics, as the movement of the whole structure is usually confined in a rather small space, so that all the initial material under consideration could be followed during the study. At the continuum level, both descriptions are strictly equivalent but give rise to different implementations and difficulties. The main difference is the inclusion of the convective term in the temporal or total derivative. On the other hand, all other spatial operators become highly non linear.

It must be pointed out that the discussion will stay at the classical calculus level. Furthermore, no boundary condition is considered as we are mainly concerned by the correct writing of the operators. Boundary conditions and the functional setting will be discussed in Chapter 4. These equations will be the foundations of the numerical method presented in this work. A similar approach to ours can be found in [11]. However, the discretization and the resolution is performed in a completely different way.

In the first part, a review of the basis of the Lagrangian and Eulerian formulation is discussed. Thereafter, these tools are applied to the heat equation firstly, and then to the full Navier-Stokes equations. Finally, the linearization process is mentioned, even if it maybe belongs more to Chapter 4, in order to compare the Navier-Stokes equations in a Lagrangian framework with a finite strain incompressible formulation, which is its nearest formulation in solid mechanics.

2.1 Lagrangian versus Eulerian formulation

In this section, the main features of the Lagrangian and Eulerian approaches are recalled, the notations are defined and the incompressibility condition is derived in both formulations.

2.1.1 Relations between Lagrangian and Eulerian tensor

In the Lagrangian formulation, the notions of reference and current configuration play a central part [9, 7, 3]. The Lagrangian formulation describes all the measured quantities with respect to the position of a particle on a previously chosen reference configuration and to time, as $M = M(\mathbf{X}, t)$ where M is whatever interesting quantity and $\mathbf{X}$ the reference position. On the other hand, the Eulerian description uses the position $\mathbf{x}$ of a given particle on the current configuration to define the quantity $M = m(\mathbf{x}, t)$. Quantities in the reference and current configuration will be written with capital and lower case letters, respectively. Obviously, both should describe the same property, which is noted introducing the configuration as a mapping of the location of a particle $\mathbf{X}$ of a body into $\mathcal{R}^n$ as $\mathbf{x} = \chi(\mathbf{X}, t)$, writing:

$$M = m(\mathbf{x}, t) = m(\chi(\mathbf{X}, t), t) = M(\mathbf{X}, t) \tag{2.1}$$

All tensor fields and their derivatives will be written with respect to a particular reference. A classical issue is to relate a derivative of a scalar tensor M in the Eulerian and Lagrangian descriptions. There, the deformation gradient $\mathbf{F}$, plays an important role:

$$\mathbf{F} = grad_X(\chi(\mathbf{X}, t) = grad_X(\mathbf{x}) \tag{2.2}$$

Using differential calculus, one gets:

$$\frac{\partial m(x,t)}{\partial x} = \frac{\partial M(X,t)}{\partial X}\frac{\partial X}{\partial x} \tag{2.3}$$

which leads, for a scalar quantity M, to:

$$grad_x(m) = \mathbf{F}^{-\mathbf{T}} grad_X(M) \tag{2.4}$$

and, for a first order tensor $\mathbf{V}$:

$$grad_x(\mathbf{v}) = grad_X(\mathbf{V})\,\mathbf{F}^{-\mathbf{1}} \tag{2.5}$$

The well-known Nanson's formula is helpful to get a similar relation for the divergence of a second order tensor:

$$\mathbf{n}\,dS = J\mathbf{F}^{-\mathbf{T}}\mathbf{N}\,dS \tag{2.6}$$

where $J = det\,\mathbf{F}$, $\mathbf{n}$ and $\mathbf{N}$ are the normals to a current and reference area element. Multiplicating both sides to the left by a first or second order tensor,

integrating on a closed surface using Gauss theorem and finally changing the variables in the right hand side, one gets for a vector $\mathbf{v}$:

$$div_x(\mathbf{v}) = \frac{1}{J}\, div_X(J \cdot \mathbf{F^{-1}V}) \tag{2.7}$$

and for a second order tensor $\boldsymbol{\sigma}$:

$$div_x(\boldsymbol{\sigma}) = \frac{1}{J}\, div_X(J\boldsymbol{\sigma} \cdot \mathbf{F^{-T}}) \tag{2.8}$$

These relations are classically known as the Piola transform of a tensor [8]. Finally, differentiating $\mathbf{V} = \mathbf{V}(\mathbf{X}, t) = \mathbf{v}(\chi(\mathbf{X}), t)$ with respect to time leads to:

$$\frac{D\,\mathbf{V}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot grad_x(\mathbf{v}) \tag{2.9}$$

The first term refers to the material derivative of a vector $\mathbf{V}$ and the second one to the spatial derivative. In a Lagrangian formulation the convective term is implicitely contained in the material derivative as the velocities are function of the material particles.

2.1.2 The incompressibility condition

The conservation of mass and the incompressibility condition are good examples of how to switch from the Lagrangian to the Eulerian formulation and viceversa. Writing the conservation of mass between two instants in two different configurations and changing the variables between both configurations, one obtains:

$$\rho_0 = \rho\, J \tag{2.10}$$

where ρ and ρ_0 are the mass densities in the current and reference configuration. The last equation describes the conservation of mass in the Lagrangian formulation. Differentiating (2.10) with respect to time leads to:

$$\dot{\rho} J = -\rho \dot{J} \tag{2.11}$$

Using the fact that:

$$\dot{J} = J\, div_x(\mathbf{v}) \tag{2.12}$$

one gets:

$$-\frac{\dot{\rho}}{\rho} = \frac{\dot{J}}{J} = div_x(\mathbf{v}) \tag{2.13}$$

which is the classical mass conservation in the Eulerian formulation. The incompressibility condition in a Lagrangian frame is thus derived:

$$\rho = \rho_0 \qquad \text{or} \qquad J = 1 \tag{2.14}$$

whereas, from an Eulerian standpoint, it is classically written:

$$div_x(\mathbf{v}) = 0 \qquad \dot{J} = 0 \tag{2.15}$$

From the expression of the divergence of a vector, it can also be written:

$$Tr(\, grad_X(\mathbf{V})\mathbf{F^{-1}}) = 0 \tag{2.16}$$

where Tr is the trace operator of a second order tensor. This expression will be used later on.

2.2 The heat equation

In this section, the strong form of the heat equation in a Lagrangian formulation will be obtained from the well known Eulerian formulation. Then, the variational form will be derived and the relationships between both formulations will be emphasized.

2.2.1 Strong form of the heat equation

There are several ways to introduce the Lagrangian formulation, either directly from the conservation principles or from the classical Eulerian equations. The second way will be chosen as these equations are very well-known. Writing the classical heat equation with convection in an Eulerian form, neglecting mechanical effects and volume changing, and using Fourier's law, one obtains [1]:

$$\rho C \partial_t T + \rho C(\mathbf{v} \cdot grad_x(T)) = div_x(\kappa\, grad_x(T)) \tag{2.17}$$

where T is the temperature, κ the thermal conductivity, ρ the density, C the heat capacity and $\mathbf{v}$ represents the convective velocity. As seen before, the term in the left hand side represents the spatial derivative. Using the material derivative and the Piola transform, the heat equation in the Lagrangian description reads:

$$\rho_0 C \frac{DT}{Dt} = div_X(\kappa\, J\, \mathbf{F^{-1}}\, \mathbf{F^{-T}} grad_X(T)) \tag{2.18}$$

Here, the same notation has been used for the representation of the temperature in an Eulerian and in a Lagrangian descriptions for the sake of simplicity.

Remark 2.2.1 No convective term appears in the last equation which has a linear appearance. However, the non linearity of this equation comes from the fact that $\mathbf{F}$, the deformation gradient, depends on the displacement $\mathbf{U}$. As noted in [12], the Lagrangian equations are highly non linear in the space coordinates.

2.2.2 Variational form of the heat equation

Multiplying equation (2.18) by a test function W and integrating on the whole domain Ω_0, the equation reads:

$$\int_{\Omega_0} \rho_0 C \frac{DT}{Dt}\, W\, dV_0 = \int_{\Omega_0} div_X(\kappa\, J\, \mathbf{F^{-1}}\, \mathbf{F^{-T}} grad_X(T))\, W\, dV_0 \tag{2.19}$$

Integrating by parts the right term of the equation (2.19) leads to:

$$\int_{\Omega_0} \rho_0 C \frac{DT}{Dt} W \, dV_0 = - \int_{\Omega_0} \kappa \, J \mathbf{F^{-1}} \, \mathbf{F^{-T}} grad_X(T) \cdot grad_X(W) \, dV_0 \quad (2.20)$$

Remark 2.2.2 The first $\mathbf{F^{-1}}$ of the right hand side of equation (2.20) could be transposed and be written just after the dot, before the spatial gradient of the weight function. Using equation (2.4), the expression of the gradient in the current configuration then appears. If the volume of integration is changed from the reference to the current configuration, J disappears and the classical variational formulation is obtained.

2.3 The Navier-Stokes equations

The same structure as the preceeding part is followed to derive the Lagrangian form of the Navier-Stokes equations. First, the strong formulation is presented through the introduction of the first Piola-Kirchhoff stress tensor. Then, the variational formulation is derived and the link between Eulerian and Lagrangian formulation is also highlighted.

2.3.1 Strong form of the Navier-Stokes equations

In the Eulerian framework, the classical equation of momentum conservation reads [6]:

$$\rho \partial_t \mathbf{v} + \rho \mathbf{v} \cdot grad_x(\mathbf{v}) = div_x(\boldsymbol{\sigma}) + \rho \mathbf{f} \quad (2.21)$$

where $\mathbf{v}$ is the velocity, $\mathbf{f}$ an external force per unit of mass, and the stresses $\boldsymbol{\sigma}$ are related to the pressure and the velocities by:

$$\boldsymbol{\sigma} = -p \, \mathbf{I} + 2\mu \, \mathbf{D} \quad (2.22)$$

for a Newtonian fluid where, in equation (2.22), μ is the fluid viscosity and $\mathbf{D}$ is the symmetric part of the gradient velocity, refered to the deformed configuration. Furthermore, considering the incompressibility condition, and supposing a constant viscosity, the classical Navier-Stokes equations read:

$$\rho(\partial_t \mathbf{v} + \mathbf{v} \cdot grad_x(\mathbf{v})) = -grad_x(p) + \mu \Delta \mathbf{v} + \rho \mathbf{f} \quad (2.23)$$

$$div_x(\mathbf{v}) = 0 \quad (2.24)$$

In the reference configuration, using the Piola transform for the second order stress tensor $\boldsymbol{\sigma}$ leads to:

$$\rho \frac{D\mathbf{V}}{Dt} = \frac{1}{J} \, div_X(J \boldsymbol{\sigma} \mathbf{F^{-T}}) + \rho \mathbf{f} \quad (2.25)$$

or:

$$\rho_0 \frac{D\mathbf{V}}{Dt} = div_X(\boldsymbol{\Pi}) + \rho_0 \mathbf{f} \quad (2.26)$$

where $\mathbf{\Pi}$ is the first Piola-Kirchhoff stress tensor defined as:

$$\mathbf{\Pi} = J\boldsymbol{\sigma}\mathbf{F}^{-\mathbf{T}} \tag{2.27}$$

Using the constitutive equation (2.22), the Lagrangian equations of motion for an incompressible fluid read:

$$\rho_0 \frac{D\mathbf{V}}{Dt} = -div_X(Jp\,\mathbf{F}^{-\mathbf{T}}) + \mu\, div_X(Jgrad_X(\mathbf{V})\mathbf{F}^{-\mathbf{1}}\mathbf{F}^{-\mathbf{T}}) + \rho_0\mathbf{f} \tag{2.28}$$

$$J = 1 \qquad \text{or} \qquad Tr(\,grad_X(\mathbf{V})\mathbf{F}^{-\mathbf{1}}) = 0 \tag{2.29}$$

Remark 2.3.1 As in the heat equation, the non linearity of these equations appears through the deformation gradient. Both the velocity and the displacement appear in these equations, coupled with the pressure.

Remark 2.3.2 In equation (2.28), we have used the property that as the flow is incompressible and the viscosity constant, to derive:

$$div_x(-p\mathbf{I} + 2\mu\mathbf{D}) = -grad_x(p) + \mu\Delta\mathbf{v} \tag{2.30}$$

However, we will see later that, in order to include various boundary conditions in the variational form in a natural way [10] and to possibly take into account a variable viscosity considered in the coupled thermal flow, the precedent simplification is not performed so that the momentum conservation reads:

$$\begin{aligned}\rho_0 \frac{D\mathbf{V}}{Dt} &= \rho_0\mathbf{f} - div_X(Jp\,\mathbf{F}^{-T}) + \\ &div_X(\mu J(\mathbf{grad}_X(\mathbf{V})\mathbf{F}^{-1} + \mathbf{F}^{-T}\mathbf{grad}_X^T(\mathbf{V}))\mathbf{F}^{-T})\end{aligned} \tag{2.31}$$

2.3.2 Variational form of the Navier-Stokes equations

Equations (2.28) and (2.29) are multiplied by test functions $\mathbf{W}$ and q respectively. The variational form of the Lagrangian Navier-Stokes equations reads, after integration by parts of these equations:

$$\begin{aligned}\int_{\Omega_0} \rho_0 \frac{D\mathbf{V}}{Dt}\cdot\mathbf{W}\,dV_0 = \int_{\Omega_0} Jp\,\mathbf{F}^{-\mathbf{T}} : grad_X(\mathbf{W})\,dV_0 - \\ \textstyle\int_{\Omega_0} \mu\, J\, grad_X(\mathbf{V})\mathbf{F}^{-\mathbf{1}}\mathbf{F}^{-\mathbf{T}} : grad_X(\mathbf{W})\,dV_0\end{aligned} \tag{2.32}$$

$$\int_{\Omega_0} J\,Tr(\,grad_X(\mathbf{V})\mathbf{F}^{-\mathbf{1}})q\,dV_0 = 0 \tag{2.33}$$

Remark 2.3.3 As in the variational form of the heat equation, it is worth noting that the terms $\mathbf{F}^{-\mathbf{T}}$ in equation (2.32) can be transposed and written just after the gradient terms of the shape function, so that, using equation (2.5), the gradients in the deformed configuration can be recovered. Changing the variables in the integral and integrating on Ω leads to the classical Eulerian weak form, as expected.

Remark 2.3.4 To take into account different boundary conditions and a variable viscosity, the variational form of equation (2.31) reads:

$$\int_{\Omega_0} \rho_0 \frac{D\mathbf{V}}{Dt} \cdot \mathbf{W}\, dV_0 = \int_{\Omega_0} Jp\, \mathbf{F^{-T}} : grad_X(\mathbf{W})\, dV_0 - \int_{\Omega_0} \mu\, J\, (grad_X(\mathbf{V})\mathbf{F^{-1}} + \mathbf{F^{-T}} grad_X^T(\mathbf{V}))\mathbf{F^{-T}} : grad_X(\mathbf{W})\, dV_0 \quad (2.34)$$

In all previous equations appears the volume of the reference configuration Ω_0. This reference configuration can vary during the analysis if appropriate changes are performed. The non-linearity of the spatial operators are due to the fact that the final spatial configuration is not known and must be found by the analysis.

2.3.3 Linearization of the equations

The Picard method has been chosen to linearize equations (2.32)-(2.33), as the Newton-Raphson implies too many extra storage for a Newtonian fluid. As a matter of fact, the Picard algorithm allows to perform the calculation in the deformed configuration taking into account the fully non linear model with large rotations and finite strains. Linearizing with a Newton-Raphson method, as described in [11], produces many extra terms due to the fact that the constitutive equation is given *in the deformed configuration* as we will try to show.

For hyperelastic incompressible materials [5, 4] used in finite strains, problems are usually solved by the Newton or a Newton-like method, which involves the linearized form of balance equations. Considering an hyperelastic material, where the fundamental assumption is:

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} \quad (2.35)$$

where $\mathbf{S}$ is the second Piola-Kirchhoff stress tensor, $\mathbf{E}$ the Green-Lagrange strain tensor and Ψ the stored strain energy function, the linearization of the Principle of Virtual Work with respect to the displacement will give [2], p 338:

$$\dot{f}^{int} = \int_{\Omega_0} \dot{\mathbf{P}} :\ grad_X(\mathbf{N})\, d\Omega_0 \quad (2.36)$$

with $\mathbf{S}$ the first Piola-Kirchhoff stress tensor and $\mathbf{N}$ a test function. Using the relationship between the first and the second Piola-Kirchhoff stress tensor, the last equation reads:

$$\dot{f}^{int} = \int_{\Omega_0} (\dot{\mathbf{S}}\mathbf{F^T} + \mathbf{S}\dot{\mathbf{F}}\mathbf{^T}) :\ grad_X(\mathbf{N})\, d\Omega_0 \quad (2.37)$$

Introducing the elastic tensor $\mathbf{C}$:

$$\mathbf{C} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} = \frac{\partial^2 \Psi}{\partial \mathbf{E}^2} \quad (2.38)$$

equation (2.37) becomes:

$$\dot{f}^{int} = \int_{\Omega_0} (\mathbf{C}\dot{\mathbf{E}}\mathbf{F}^{\mathbf{T}} + \mathbf{S}\dot{\mathbf{F}}^{\mathbf{T}}) : grad_X(\mathbf{N})\, d\Omega_0 \tag{2.39}$$

It finally appears that the first term of the last equation, the material non linearity, has been simplified thanks to the fundamental asumption of equation (2.35), for which the elastic tangent tensor puts in relation two quantities *in the reference configuration*. However, in the case of a fluid, the second Piola-Kirchhoff tensor is only known through its spatial counterpart $\boldsymbol{\sigma}$, the Cauchy stress tensor. As:

$$\mathbf{S} = J\mathbf{F}^{-\mathbf{1}}\sigma\mathbf{F}^{-\mathbf{T}} \tag{2.40}$$

the linearization of $\mathbf{S}$ will involve the linearization of J, $\mathbf{F}^{-\mathbf{1}}$, σ, and $\mathbf{F}^{-\mathbf{T}}$ whereas the geometric non linearity and all the other terms involving the pressure linearization will present the same form. This difference is the reason why the linearization of the Navier-Stokes equations in a Lagrangian formulation implies so many extra terms compared with the Picard linearization and an hyperelastic law formulation linearized with the Newton method.

2.4 Conclusion

In this chapter, the main tools to derive the Navier-Stokes and the heat transport equations have been presented. One possibility to obtain these equations consists in deriving directly these equations from the physics, the second is to use the Eulerian versions of these equations and then to switch to the reference configuration. Their weak form has also been highlighted. The convective term has been eliminated at the expense of the spatial operators expressed in the reference configuration. At the continuous level, the elimination of the convective term is the most significant difference. However, at the numerical level, various other differences arise, mainly due to the convection of the mesh, such as the superposition of particles due to the crossing of the characteristics of these particles, and the remeshing procedure, which will be detailed in the next chapter. It is also interesting to note that the mass conservation equation is expressed through the Jacobian of the transformation, whereas it involves the divergence of the velocity in the Eulerian formulation. The strict equivalence between both equations appears at the continuous level, but not at the discrete level, as the time derivative is only approximately performed, and will give rise to the difficult problem of mass conservation at the discrete level. The variational form of the heat equation and the Navier-Stokes equations are the basis of the numerical scheme presented in Chapter 4.

Bibliography

[1] G.K. Batchelor. *An Introduction to fluid dynamics*. Cambridge University Press, Cambridge, 2000.

[2] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[3] J. Bonnet and R.D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Cambridge, 1997.

[4] U. Brink and R. Stein. On some mixed finite element methods for incompressible and nearly incompressible finite elasticity. *Computational Mechanics*, 19:105–119, 1996.

[5] S. De and K.J. Bathe. The method of finite sphere with improved numerical integration. *Computer and Structures*, 79:2183–2196, 2001.

[6] E. Guyon, J.P. Hulin, and L. Petit. *Hydrodynamique physique*. EDP Sciences, 2001.

[7] G.A. Holzapfel. *Nonlinear Solid Mechanics, A Continuum Approach for Engineering*. Wiley, 2000.

[8] J.E. Marsden and T.J.R. Hughes. *Mathematical Foundations of Elasticity*. Dover, 1983.

[9] R.W. Ogden. *Non-linear Elasticity*. Dover, 1997.

[10] O. Pironneau. *Méthodes des Eléments finis pour les fluides*. Masson, Paris, 1988.

[11] R. Radovitzky and M. Ortiz. Lagrangian finite element analysis of Newtonian fluid flows. *Int. J. Num. Meth. Eng.*, 43:607–619, 1997.

[12] R. Témam. *Navier-Stokes Equations and Nonlinear Functional Analysis*. SIAM, Philadelphia, 1995.

Chapter 3

Mesh generation

In this chapter, we will describe in details the whole mesh generation process in its now classical form, and the particularities due to the convection of the nodes in the present method. The approach developped in this work follows closely all the articles from the INRIA Projet Gamma team, which is exclusively dedicated since twenty years to the mesh generation area.

The mesh generation research is closely related to the computational geometry, a word which comes from the work of Shamos and Preparata [89]. This relatively new science mixes the theoretical results of abstract geometry with the algorithmic and the difficulty of the numerical approximation to implement these results on computers as described in O'Rourke [84]. First of all, we will present the Delaunay triangulation and its properties, and we will quickly review the different implementations of this special triangulation. Then, we will describe the incremental algorithm, which is the method used in this work, and which is the basis of the mesh generator. In a third part, we will present a method to develop a complete mesh generator. Finally, some numerical example illustrating the depicted method conclude this chapter.

3.1 The Delaunay triangulation

In this section, we introduce a few geometrical definition and a particular triangulation, the Delaunay triangulation. We consider an ensemble of points S of $\mathcal{R}^d$ in general position, which means that in two dimensions there are no three points in the same line or four points on the same circle, and in three dimensions, there are no four points on the same plane or five points on the same sphere.

3.1.1 Statement of the problem

In order to define a triangulation we need to introduce the concept of simplicial covering and simplex. Following George *et al.* [46]:

Definition 3.1.1 *In $\mathcal{R}^d$, a d-simplex is the convex hull of $d+1$ points not in the same hyperplane.*

For example, a 2-simplex is a triangle, a 3-simplex is a tetrahedron.

Definition 3.1.2 *A simplicial covering $\mathcal{T}_r$ is a space covering made of simplices which verify:*

- *The set of the vertices of $\mathcal{T}_r$ is exactly S*
- $\Omega = \bigcup_{K \in \mathcal{T}_r} K$.
- *Every element K in $\mathcal{T}_r$ has a non empty interior.*
- *The intersection of the interior of two elements is an empty set.*

Definition 3.1.3 *$\mathcal{T}_r$ is a triangulation if $\mathcal{T}_r$ is a simplicial covering and:*

- *The intersection of two elements of $\mathcal{T}_r$ is :*
  - *the empty set,*
  - *a vertex,*
  - *an edge,*
  - *a face (if d=3).*

Definition 3.1.2 prevents any non overlapping of the simplices, as oposed to the kernel functions of the meshless methods. Definition 3.1.3 is added to Definition 3.1.2 in order to deal with the notion of conformity of the triangulation necessary to derive error estimates for the Finite Element Method in conforming approaches [23].

After having defined what is a triangulation, we will focus on a particular triangulation, namely the Delaunay triangulation. Once more, we need to introduce another concept, the Voronoi diagram:

Definition 3.1.4 *Given a cloud of points S of $P_i, i = 1, \cdots, n$ the Voronoi diagram is the ensemble of cells V_i defined by:*

$$V_i = \{P \text{ such that } d(P, P_i) \leq d(P, P_j) \quad \forall j \neq i\} \tag{3.1}$$

The Delaunay triangulation is the dual of the Voronoi diagram in the sense that the Delaunay triangulation is obtained by joining the vertices belonging to two adjacent cells from the Voronoi diagram. An example of the Delaunay triangulation and its Voronoi diagram with a degenerated case is given in Figure 3.1. If the points are in general position, the Delaunay triangulation is unique, whereas it is not the case if d+2 points are cocyclic. This will not represent any problem as the construction method does not rely on the Voronoi Diagram. The fundamental property of this triangulation is the following:

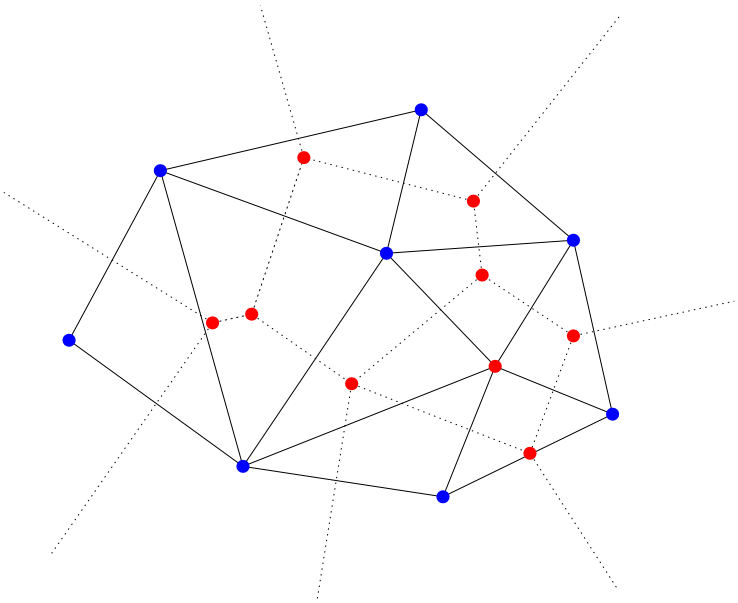


Figure 3.1: A Delaunay triangulation in plain lines and its dual, the Voronoi diagram in dotted lines

Lemma 3.1.1 *Given $\mathcal{T}$ any triangulation of the convex hull of a cloud of points, if $\mathcal{T}$ verifies the empty circle property for every two adjacent elements of $\mathcal{T}$, then $\mathcal{T}$ is the Delaunay triangulation.*

This very important lemma was demonstrated by B. Delaunay in 1934 [30]. This will be the fundamentals of the method. It is also remarkable that a local property implies a global property. Various extensions of the Voronoi diagram are high order Voronoi diagrams [98], diagrams for other objects than points [67], and particularly the extension due to the introduction of a metric, isotropic in [4], and anisotropic in [69, 72, 9].

3.1.2 Properties of the Delaunay triangulation

The Delaunay triangulation verifies some nice properties which contribute to its popularity as a basis for the mesh generator. We mention the main ones in two dimensions:

Theorem 3.1.1 *The Delaunay triangulation maximizes lexicographically the minimum angle between two edges.*

Theorem 3.1.2 *The Delaunay triangulation minimizes lexicographically the maximum incircle radio of an element.*

Unfortunately, these results are not true in three dimensions and let guess the problems that we will have to face. However, some other properties are met for any dimensions [90]:

- The maximum min-containment radius of the Delaunay triangulation of a point set in $\mathcal{R}^d$ is less than the maximum min-containment radius of any other triangulation of the point set.

- The union of the circumballs of the triangles incident on an interior point in the Delaunay triangulation of a point set lies inside the union of the circumballs of triangles incident on the same point in anyother triangulation of the point set. For a point on the convex hull of the point set, this result is true provided only the portion of the union which lies in the interior cone of the convex hull at the point is considered.

- The weighted sum of squares of the edge lengths is the smallest for the Delaunay triangulations, where the weight is proportional to the sum of the volumes of the triangles incident on the edges. The sum of these weights is independent of the triangulation, therefore the weights can be normalized.

- If the triangulation is non-obtuse then it is the Delaunay triangulation.

To sum up, the Delaunay triangulation is well appreciated in the computational geometry area as it possesses a sound mathematic basis and nice properties in two dimensions. However, these properties are rarely met and enjoyed in a practical implementation and others goals have to be reached.

3.1.3 Algorithm for the Delaunay triangulation

There exists a lot of different methods to implement the Delaunay triangulation. All rely on the empty sphere property, but important differences are met during the numerical treatment of the algorithm as round-off errors are not propagated in the same way.

The incremental algorithm

A first method consists in the incremental algorithm, which we will detail thereafter. The idea is to introduce the points one by one, and by supposing that the triangulation which contains the already inserted points is Delaunay, do what is necessary so that, by the introduction of the new point, the resulting triangulation remains Delaunay. Therefore, the elements that contain the new points are eliminated and a new local triangulation is then performed. The main feature of this algorithm is its extension theoretically to any dimension and was proposed independently in the same journal volume by Bowyer [13] and Watson [103].

The flip algorithm

Another important algorithm due to Lawson [70] lays on the following theorem:

Theorem 3.1.3 *Given a triangulation $\mathcal{T}_r$ of the convex hull of a cloud of points S, it is possible in two dimensions to obtain the Delaunay triangulation by applying edges sweeping for two adjacent element which do not verify the incircle property.*

Therefore these sweeping can be performed incrementally or not, considering an initial triangulation obtained by one or another method. This theorem has been later extended partially to the three dimensional case by Joe [61], relying on another work of Lawson [71]. An important difference between the incremental and the flip algorithm is due to the fact that the flip algorithm only relies on predefined patterns as basic transformations. In three dimensions, these are the $2 \rightarrow 3$ and $3 \rightarrow 2$ flips that will be seen in the optimization stage. So that, at every moment, the algorithm deals with a valid triangulation. At the oposite, the incremental algorithm deletes elements and must triangulate an empty space, so that it may give rise to a topologycaly invalid triangulation with overlapping tetrahedrons and gasps, as reported in [62].

Divide and conquer

An important algorithm comes from the divide and conquer methodology. The idea is to divide the problem in two other simpler problems and to apply it recursively until having elementary problems, and then to assembly the results. The difficulty relies on the assembly procedure but the whole algorithm is of order $O(n\,log(n))$ in the worst case [55].

The lifting transformation

There exist also a stretch relationship between the convex hull algorithms and the Delaunay triangulation. This link has been discovered by Edelsbrunner and Seidel [35]. Indeed, the Delaunay triangulation can be obtained by the convex hull in a space of one dimension higher that the one of the points under studies, by assigning as additional component for each point the sum of the square of the other components, indeed the value of the paraboloid associated to the other components. After having performed the convex hull in the $d+1$ dimension space, this hull is projected through the hyperplane orthogonal to the last component, and this projection is the Delaunay triangulation. All the different algorithms in $O(n\,log(n))$ in the worst case to compute the convex hull are then directly usefull for the Delaunay triangulation.

The sweeping algorithm

A sweeping algorithm in $O(n\,log(n))$ has been also proposed by Fortune in the two dimensional case [37]. A sweepline algorithm will construct the Voronoi

diagram in the upper part of the line dividing the plane. The difficulty is due to the fact that the Voronoi edges will be encountered by this sweep line before the Voronoi edges responsible of these regions. The idea of Fortune relies on applying an appropriate mapping to these sites so that they are encountered first.

3.2 The incremental algorithm

As mentionned before, the idea of the incremental algorithm consists in introducing the points one at a time in an existing Delaunay triangulation. Various stages are performed to construct a Delaunay triangulation due to the insertion of the new point, as described in this section.

3.2.1 Description of the method

In order to simplify the problem, the algorithm begins with a bounding box which contains all the points to be inserted and which is triangulated by two triangles in two dimensions and by 5 or 6 tetrahedra in three dimensions. This initial triangulation is a Delaunay triangulation. The points are then inserted one by one in this first mesh. The algorithm consists in the following:

- Find all the elements whose circumsphere contains the newly inserted point and delete these elements
- Find the faces of the polygon (polyhedron) formed by the deleted elements
- Join these faces to the new point to create the new triangulation

This is sumed up in George *et al.* [46] by:

$$\mathcal{T}_{i+1} = \mathcal{T}_i - \mathcal{C}_P + \mathcal{B}_P \tag{3.2}$$

where $\mathcal{T}_i$ is the triangulation before the insertion of the new point, $\mathcal{C}_P$ is the cavity of the new point constituted by the old elements whose sphere contains the new point, and $\mathcal{B}_P$ is the ball of the new point, which represents all the new elements that contain the new point. Furthermore the cavity possesses the important theoretical property to be star-shaped with respect to the newly added point if the points are in general position, property that will be checked explicitly in the numerical algorithm. Figures 3.2 and 3.3 illustrate the algorithm for the newly inserted point P. It must be noted that the aforementioned algorithm is the theoretical basis of the method. It represents however only a crude version of the numerical implementation, fully described in two dimensions in [45] and in three dimensions in [43].

3.2.2 Asymptotical complexity

Each of these operations could be costly. For example, finding which element whose sphere contains the new point could be trivially done by examining all

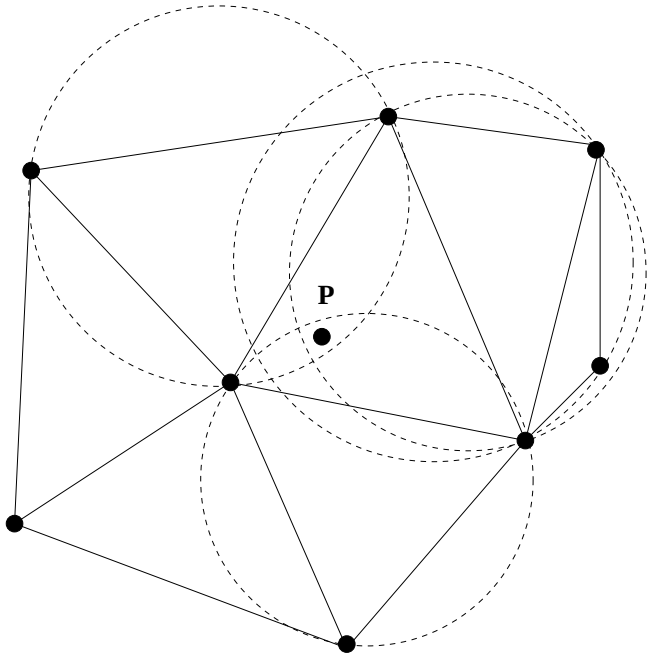


Figure 3.2: The triangulation before the insertion of point P

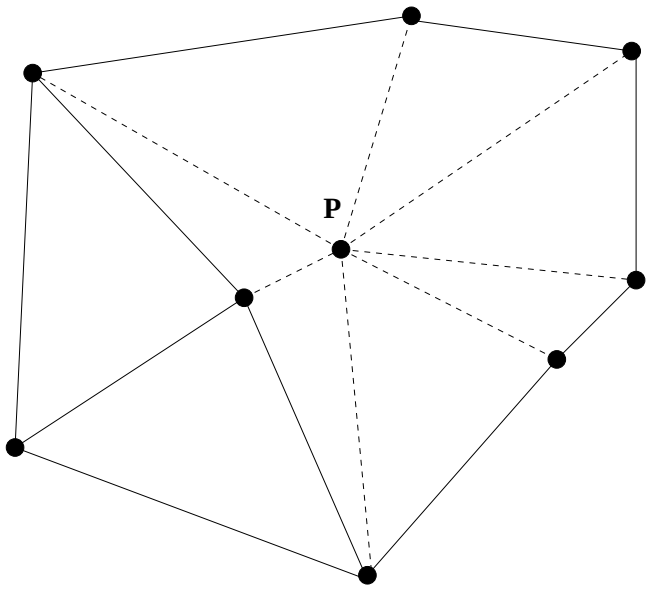


Figure 3.3: The triangulation after the insertion of point P

the elements in the list of elements each time a point is inserted. For n elements and m points, this represents n*m tests where $m = O(n)$ usually, so that the final number is of order $O(n^2)$. This is evidently not acceptable. Therefore, special algorithms are designed to reduce all these operations, relying on an appropriate use of data structures, as described in [26, 77, 95, 2, 1, 25].

Globally, the Delaunay triangulation with the incremental algorithm is of order $O(n^2)$ in the worst case, as in the worst case, we can destroy $O(n)$ elements at the introduction of each new point (see [89] and [20] for such examples). To avoid in average the worst case, a randomized version is usually used, where the points are introduced in random order to achieve a $O(n\,log(n))$ in average. However, for usual real geometries and with appropriated data structures, the incremental algorithm is of order $O(n)$, including the corrections applied in case of a wrong numerical decisions, as will be proved experimentally later on. This transforms the Delaunay-based mesh generator as the quickest one of the different methods, particularly for huge meshes thanks to its asymptotical linear rate.

3.2.3 Implementation

The three main tasks of section 3.2.1 require different tools. For the first task, an adjacent search is performed from the element(s) containing the newly inserted point, called the base elements. Firstly, supposing that each point knows one element to which it belongs, an additional spatial data structure provides an efficient localisation of a point near the newly inserted point to find this basis. Then a walk through the mesh [80] is performed to find one or various elements where the point falls. Classical structures are octrees, binary trees, k-d tree [95] or the Alternate Digital Tree (ADT) [10], which are well adapted for large variations of point distribution but could not impeed a $O(n\,log(n))$ complexity due to the tree depth for each point. Bins, which are another alternative, are the equivalent of hashing in multidimensions [2], providing a linear complexity in average, but which can behave badly for very large variations on point distributions. The latter has been used in this work. The second task is straightforward knowing which are the elements in the cavity. The third task requires the update of the neighbours of each tetrahedron, which is not a necessity in the database but represents a valuable knowledge for many search operations. It could be done by hashing [43], but advantage can be taken of the inheritance of the volumic configuration from the cavity surface, as described in [45] for the two-dimensional case and in [12] in three dimensions.

In Watson's paper, the Delaunay criterion is tested for each newly inserted point against all the elements in the mesh. At the oposite, the adjacent search is already present in the Bowyer's paper, with the acceleration procedure to find the first element(s) which contains the newly inserted point. In both previous work, the Delaunay criterion, is checked on the circumcircle of the actual element of the mesh against the distance between the new point and the circumcenter of theses elements. As the Delaunay criterion is symmetric, it is the same, apart from round-off errors, as checking the Delaunay criterion of the

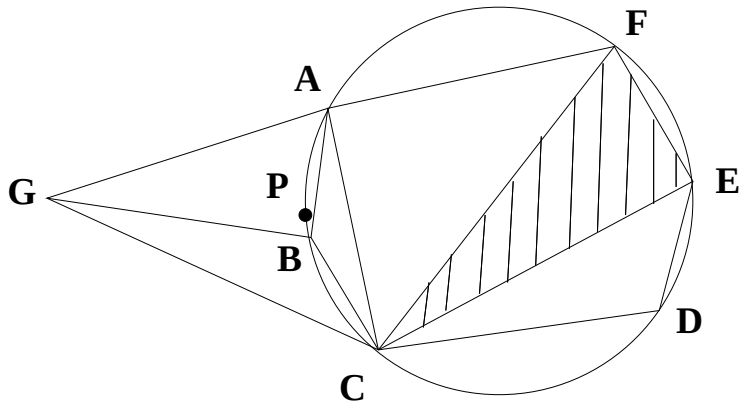


Figure 3.4: Numerical errors in the triangulations

new elements against the old points. As the radii and the circumcenter could be stored, the second choice is however more expensive. Nevertheless, Shenton and Cendes [100], and Wright and Jack [106] proceed from the base by expanding the insertion polyhedron by testing with the new spheres against the old points.

The last implementation accelerates the whole process but other solutions have to be found to avoid difficulties of the method itself. Indeed, the implementation could not be straightforward as the Incircle test requires to calculate distances to the power $d + 1$. The use of floating numbers could produces bad decisions during the construction of the cavity. Using integer numbers guarantees an exact computation of volumes in the range specified by the machine. As this range is fixed, the resolution of the method is of one order lower in three dimensions than in two dimensions. Two cases are possibles, which are illustrated in Figure 3.4.

- Either the cavity contains a point of the old mesh which has been disconnected of the old mesh because of the proximity with the recently inserted mesh. In this example, B is very close from the point P so that during the new triangulation of the cavity, B could stay without elements.

- Or the cavity is not connex, due to a problem of cosphericity, as mentionned in [53], which is shown for the triangle CEF which does not belong to the cavity.

A correction algorithm has been presented in [43], where the idea is to build a possibly wrong cavity in a first step, and then to check explicitly the star-shaped property of the cavity and to correct it if this property is not verified. This algorithm is the fundamental part of the mesh generator. We will come back to another well-known degeneracy due to the method itself in the optimization part. It must be noted that other approaches have been proposed to extend the accuracy of the floating point operations [101, 24, 38].

3.3 Mesh generation

Until then, only triangulation algorithms were described, but the mesh generation problem is slightly different as we have a domain with boundaries to mesh, and we do not have a cloud of points already available, so that we have to decide where to locate the points and how much points do we need. Furthermore, the mesh must verify some quality requirements depending on the application, in our case the Finite Element Method. These three points will be presented thereafter, but we firstly give the definition of a mesh following [41]:

Definition 3.3.1 *Given a closed bounded domain Ω, $\mathcal{T}_r$ is a mesh of Ω if:*

- $\Omega = \bigcup_{K \in \mathcal{T}_r} K$.
- *Every element K in $\mathcal{T}_r$ has a non empty interior.*
- *The intersection of two inside of two elements is empty.*
- *The intersection of two elements of $\mathcal{T}_r$ is :*
  - *the empty set,*
  - *a vertex,*
  - *an edge,*
  - *a face (if d=3).*

It is equivalent to construct a conforming triangulation of Ω, as we will see next with the difference that the cloud of points is not given, and the boundaries belong to the triangulation.

3.3.1 Boundary regeneration

In a Delaunay mesh-based generator, the aim is to produce a mesh of a spatial domain described by its boundary in the form of a list of its edges in two dimensions, and faces and edges in three dimensions. The input of the algorithm is therefore such a list plus a few possible imposed points, with a possible size map describing the desired distribution of points. The ensemble of the boundary points is then inserted and triangulated with the Delaunay kernel. However, at this moment, the boundary elements could be only partially present in the resulting mesh, depending of the Delaunay conformity of the boundary [87], so that special operators should be applied to recover these boundary elements. We define thereafter two usefull notions for the boundary recovery:

Definition 3.3.2 *$\mathcal{T}_r$ is a constrained triangulation of Ω if all the elements of $\mathcal{C}onst$ are present in $\mathcal{T}_r$ where $\mathcal{C}onst$ is the set of possible imposed edges and faces.*

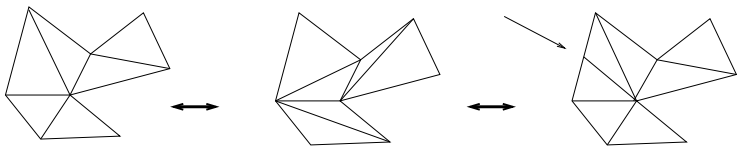


Figure 3.5: Exact and weak satisfability of the constraints

The definition does not take into acount the points as they are trivially included in the mesh. We need also to make a difference between a constrained and a conforming mesh ([74, 49] and references therein), or what George calls strong or weak equivalency between two meshes:

Definition 3.3.3 *A given triangulation $\mathcal{T}_r$ satisfies exactly a given set of constraints if any member of $\mathcal{C}onst$ is an entity of $\mathcal{T}_r$.*

Definition 3.3.4 *A given triangulation $\mathcal{T}_r$ satisfies weakly a given set of constraints if any member of $\mathcal{C}onst$ is an entity of $\mathcal{T}_r$, exactly or as a partition.*

The two first meshes of Figure 3.5 satisfy strongly the set of constraints whereas the third mesh satisfies it only weakly. The two first meshes are equivalents whereas the third is weakly equivalent to the two others.

To sum up, we have to apply modifications on the mesh to recover the lost boundaries. Here there is a huge difference between the two dimensional case and the three dimensional. Indeed, in two dimensions, we have:

Theorem 3.3.1 *There exists a triangulation without internal vertex covering any domain with a non self-intersecting polygonal boundary [46].*

Moreover, we add a very interesting result from Chazelle [21], which concludes years of research:

Theorem 3.3.2 *It is possible to construct a triangulation of a polygon with n vertices in O(n).*

Unfortunately, no one of these results is extendible to the three dimensional case. For example, the Schönhardt polyhedron on Figure 3.6 is the smallest polyhedron in terms of number of faces which could not be triangulated without adding an interior point. This type of points is called a Steiner point. However in this case, this point could be added everywhere inside the polyhedron as all the faces are visible from this point. In general, the situation is much worse, as proved by two results from Ruppert *et al.* [94]:

Theorem 3.3.3 *It is NP-complete to decide whether a given three dimensional polyhedron can be triangulated without adding Steiner points.*

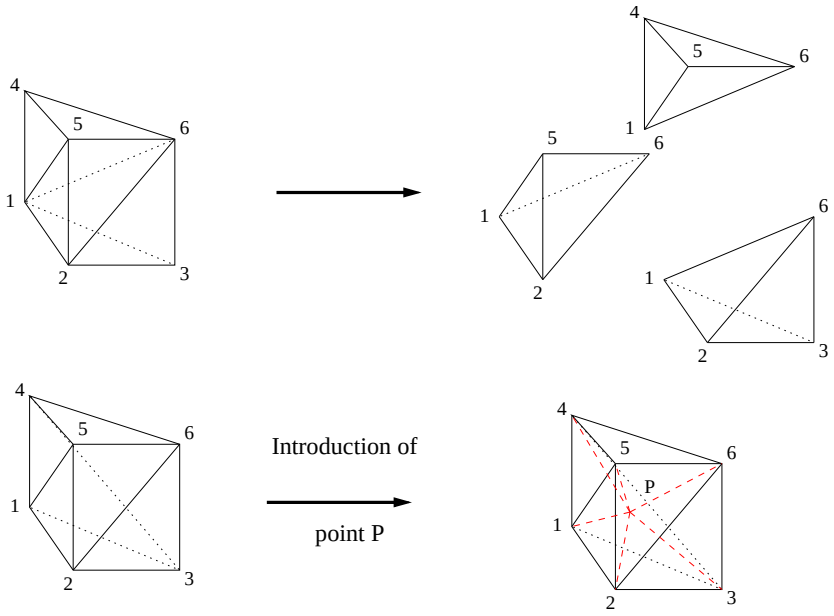


Figure 3.6: The Schönhardt polyhedron

Theorem 3.3.4 *For any fixed integer $k > 0$, it is a NP-hard problem to determine whether a given polyhedron can be triangulated with at most k Steiner points.*

We recall that a NP problem (Non-deterministic Polynomial problem) is a problem for which, given a possible answer, this solution can be checked to be correct in polynomial time respect to the size of the input. A NP-hard problem is a problem for which every other problem in NP is reducible to it. A NP-complete problem is a NP problem and a NP-hard problem, which means that it is at least as hard as a NP problem [42]. Evidently, these results announce the difficulty that we will have to face in the three dimensional case.

Various methods have been proposed in the literature, mainly depending if the boundary recovery process is allowed to modify the boundary surface representation by inserting points and swapping the triangles defining the surface or not, or by trying to extract the surface as faces of the tetrahedra of the Delaunay triangulation of the boundary points. In the latter category, an early method is proposed by Baker [6], where specific points are added inside the volume to be meshed, and tetrahedras are considered as inside the domain if they have at least one such node in their connectivity. The approach is simple but very restrictive as it depends on the placement of the specified nodes, and on the complexity of the surface boundary. Later, Baker [7] improves the method by a better placement of the inserted point but no guarantee on the recovery exists and faces are rather extracted as a byproduct of the Delaunay triangulation.

In the conforming approach, three subcategories arise. In the first case, intersection points between missing edges and missing faces, and the tetrahedras of the actual triangulation are computed and inserted directly. Weatherill and Hassan [104] modify the boundary representation to have a partition of the triangle faces into subfaces by firstly subdividing edges, faces and elements cuts by the boundary faces and edges. Once performed, they try to take out the introduced points from the mesh. However, there is no guarantee that these points could be taken out from the mesh. These points are introduced in the triangular faces, so that the geometry can not be improved by these insertions, contrarily to [73]. The same method is used in Karamete *et al.* [66], but considering also the case of the remeshing of a small part of the mesh, which they call cavity. This is a difficult task in the incremental algorithm as points are inserted in an already triangulated volume, but no empty space is created. It leads them to study complex splits to merge this cavity with the original volume mesh. Recently, Du and Wang [32] modify slightly the method by considering the recovery of the faces directly, and by using the constrained Delaunay kernel to insert the Steiner points, a strategy shared also by George *et al.* [49] for the constrained case as seen below. In the second case, Delaunay conforming boundaries are created a priori, or a posteriori so that they will undoubtely appear in the Delaunay mesh, by an appropriate presubdivision of the boundary. In this group, Cavalcanti and Mello [17] perform an edge then a surface recovery by inserting points on midedges until the edge appears in the triangulation. A similar idea is used in [104] in the two dimensional case. This process could not converge but solutions exist to remedy to this problem, by using the notion of Delaunay admissibility of the edges and faces. This notion is extensively used in the work of Pébay [87] to recover the constrains a priori and to keep a Delaunay triangulation during all the process. For the faces, no proof of convergence exists and heuristics must be employed. The same technique is used in [83]. In the third case, swapping on the triangle surfaces is performed to try to give rise to a more or less equivalent geometrical surface triangulation which includes the surfaces already created in the tetrahedrization. Liu and Baida [74] try to reduce the inserted points on the surface as much as possible by testing combinaisons of flips on the surfaces that are intersected by the missing edges, but also on neighboring faces, by hopping that blocked situations will be unlocked by neighboring flippings. Flip are allowed on the planar surface triangles. Sharov and Nakahashi [99] propose two original algorithms to recover the boundary. In the first algorithm, triangle swapping are allowed for small dihedral angles. Then, flipping in the volume mesh are performed if the previous procedure failed to recover all the constrain. Finally, points are directly inserted if the second procedure also partially failed. It is interesting to note that the insertion procedure has been greatly simplified respect to the work of Hassan and Weatherill [104], by recursively using only two operators. The same operators are used in the recent work of George *et al.* [49], and were already present in the work of Borouchaki [11]. In the second algorithm, boundary points are inserted in an advancing front manner, so that a maximum part of the constrain will be recovered during the insertion, mainly by flippings and constrained Delaunay

insertion. If this insertion fails to recover all the constrain, a similar approach as before is used.

Finally, in the constrained approach, which does not modify the input boundary data, Joe [59] gives an interesting but expensive solution, extending the method he proposed in the two dimensional case in [65] by first dividing the volume to be meshed in convex polyhedras. It is a simple thing then to introduce one Steiner point or not to triangulate the interior of the convex polyhedras. The work of George *et al.* [52, 51, 50] is based on local modifications also to strictly verify the constrains. The reason of using a constrained approach could be, for example, that the surface discretization is common to different parts, and the possible solver behind the process may not be aware of the changes in the surface. It must be noted that it is the unique constrained approach which is proved to converge, at least apart from round-off errors and for the edge recovery. The key point is that local operators may fail because of a non convex local configuration as simplices are used. Points will be inserted to recover a local convex subdomain and obtain the solution. As already mentioned, George *et al.* [49] improved recently their method by also considering intersection cases, but taking out the introduced points by others special tools. The same idea is also proposed independently by Qu and Wang [33], by considering first a conforming approach of the already cited reference [32], and then taking the intersection points out of the recovered but partitioned face with different operators than George *et al.* [49]. The method implemented in this work follows the work of George with the slight difference that face swapping could be allowed for planar boundary faces as an option. This can greatly reduce the amount of work as noted in [104]. As a remark, this kind of problem is not only special to the Delaunay methodology, but happens also in the advancing front method, when various fronts concur in a cavity, creating an arbitrary polyhedron which must also be triangulated.

From the algorithmic point of view, the method consists in the following:

- identify the missing faces and edges in the present mesh
- for each edge:
  - form the set of elements intersected by this edge or pipe
  - apply local operators to regenerate the missing edge
  - if local operators fail, introduce point to unlock the situation
- for each face (if $d = 3$):
  - form the set of elements whose edges intersect the missing face
  - apply local operators to regenerate the missing face
  - if local operators fail, introduce point to unlock the situation
- try to collapse the Steiner points
- optimize the position of Steiner points by relocation

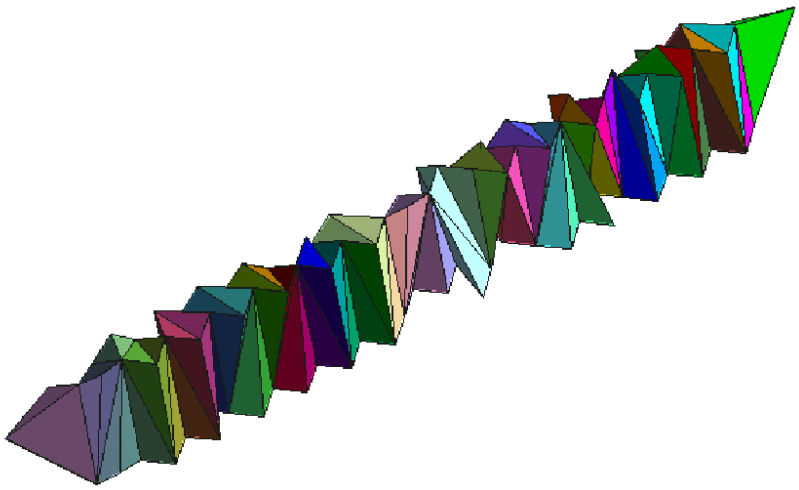


Figure 3.7: A pipe : the set of elements intersected by an edge. Here 115 elements are intersected by the edge.

An example of a pipe is given in figure 3.7. In the three dimensional case it must be noted that a given face could be missing in the mesh whereas its three edges are present, because it is intersected by edges of other elements. The operators used consist in various type of face and edge swappings as described in [52], [51], and [50]. Figure 3.8 illustrates one of this operators, the transformation $Tr_{2->3}$. Once more, the last algorithm is oversimplified due to numerous subcases, and this algorithm was by far the most difficult encountered during this work.

3.3.2 Node creation

At this point, we hopefully dispose of a mesh which contains the imposed faces and edges, composed only of the boundary points plus few other Steiner points. This mesh does not have good quality properties as the domain is not discretized finely enough. As mentionned before, the Delaunay triangulation does not give any indication on the number of points to be used and where to locate them. Three main cases are possible:

- There is no information available on a mesh distribution map.
- The mesh distribution map possesses information on size specifications depending on each point but not on any direction, i.e. a scalar value.
- The mesh distribution map possesses information on size specifications depending on each point and on a direction at this point i.e. a tensorial value.

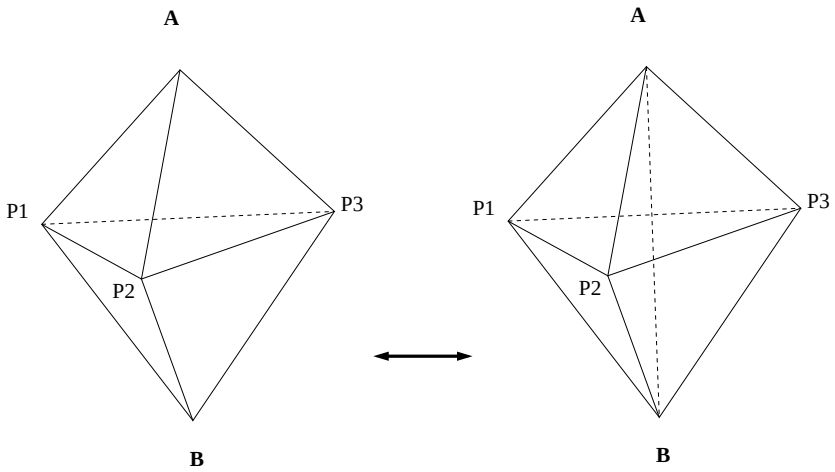


Figure 3.8: The transformation $Tr_{2->3}$

The first case correponds to the classical mesh generation scheme where the map distribution should be constructed from one method or another. The second case corresponds to the isotropic case with size specification, and the third to the anisotropic case. Each case is a particular case of the following case. In this work, results are presented for the three cases. For a finite element computation, the mesh should comply some fundamental requisites from which depend the convergence properties and the quality of the final result. Particularly, the interpolant error estimates [91] rely on the hypothesis of regular mesh which corresponds to the fact that there exists a constant σ such that:

$$\frac{h_k}{\rho_k} \leq \sigma \quad \forall k \in \mathcal{T}_r \tag{3.3}$$

where h_k is the diameter of the element, the largest side, and ρ_k the diameter of the sphere inscribed in the element. We will discuss the quality properties in the next section, but we already see that the quality notion is undoubtly related to the notion of distance. This notion and its evaluation at the implementation level will be the basis of the node creation.

In this node creation process, the literature is also rich of proposals. Weatherill *et al.* [104] use the barycenter of the element as prospective point. Chew [22] and Ruppert [93] uses the circumcenter of the element instead of the barycenter. Frey *et al.* [40] and Mavriplis [82] use the advancing front methodology to create points from boundary faces, where boundary face are boundaries of a volume defined on specific criterias. The points are almost optimaly placed but this construction is expensive. After having performed the convex polyhedra descomposition, and further subdivided each polyhedron to respect a given variation of the mesh size distribution, Joe [63] generates the points on a quasi-uniform grid oriented on the diameter of the polyhedron. To specify the size of the element desired in a region to be meshed, distribution of sources in the

domain can also be used as well as background grids [105, 79]. Recently, Du and Wang [31] introduced the centroidal Voronoi tesselation, where the inserted nodes are furthermore the center of mass of their Voronoi cell. They extend this approach to anisotropic metrics in [34].

As already said, the quality of a mesh is related to the distance between their nodes, among others things. That is why in this work, the method of George *et al.* [47] has been chosen, as it relies on the edges of the mesh, which implicitely contains the notion of distance. Furthermore, the method extends directly to anisotropic meshes. We consider the general case, i.e. a Riemanian space where we know in each point a $d \times d$ symmetric definite positive tensor $\mathcal{M}(X)$ called the metric. Given two points A and B and γ a parametrisation of the geodesic Γ at least of class C^1 joining A and B, we have:

$$d_{\mathcal{M}}(A,B) = l(\Gamma) = \int_0^1 \sqrt{\gamma'(t)^T \cdot \mathcal{M} \cdot \gamma'(t)}dt \tag{3.4}$$

in the Euclidian case, where the goedesic are straight lines, we recover:

$$d_{\mathcal{M}}(A,B) = \sqrt{\overrightarrow{AB}^T \mathcal{M} \overrightarrow{AB}} \tag{3.5}$$

The geometric locus of the points which are to a given distance to a given point is an ellipsoid $\mathcal{E}$ whose equation verifies:

$$\overrightarrow{AB}^T \mathcal{M}(X) \overrightarrow{AB} = 1 \tag{3.6}$$

so that given a metric $\mathcal{M}(X)$, the mesh is conform to this metric if the edges have an average unit mesh in the Riemanian structure. Knowing the size map, it is possible to construct a metric in which the edges will have an average unit, so that they will respect the specifications. A mesh with this property is called a unit mesh by George *et al.* [47].

At the implementation level, the fundamental property is to evaluate correctly the length of each edge of the mesh. The process is iterative:

- Given a mesh, we examine all the internal edges of this mesh.
- For each edge, we have to approach the integral computation. Therefore, we use a quadrature formula as:

$$l_{\mathcal{M}}(AB) = \frac{\sqrt{\overrightarrow{AB}^T \mathcal{M}(\mathcal{A}) \overrightarrow{AB}} + \sqrt{\overrightarrow{AB}^T \mathcal{M}(\mathcal{B}) \overrightarrow{AB}}}{2} \tag{3.7}$$

  - if the length of the edge is greater than a given tolerance, we introduce the middle of the edge in the real space and compute its metric with respect to the size map.
  - we dispose then of a number of points which divide the edge, each part having a length smaller than a given tolerance in the Riemanian space, but which are used only to accurately measure the length

of the edge. Begining from one extremity of the edge, we search for the interval so that the distance between the new point and its predecessor is equal to one respect to the given metric.

- The new points are well-distributed along the edge but not globally. We filter this cloud of points so that no points could be closer than the given distance in the Riemanian context and we insert them randomly through the constrained Delaunay kernel, as the boundaries must be protected when constructing the insertion polyhedron [106].

The iteration stops when all the edges have been refined. The domain is then saturated by points. The node insertion is different in the isotropic and anisotropic case as the Delaunay criterion must be accounted for, as described in [44].

3.3.3 Optimization process

As already mentioned, the quality of the mesh will influence the properties of the numerical solution. A well-known condition is the minimum angle condition of Bramble and Zlamal in two dimensions [14], generalized by Ciarlet in higher dimensions [23], which states that the angles of a triangle in two dimensions must be bounded away from zero, or that the ratio of the diameter of the element by the incircle radius must be bounded away from zero. It is a sufficient but not necessary condition, as stated by Babuska and Aziz [5] and Jamet [60], where the maximal angle condition appears to be more essential than the minimum angle condition. More recently, Rippa [92], and D'Azevedo and Simpson [29] insist on the fact that, for linear interpolation, the stretching should be aligned with the direction where the magnitude of the second directional derivative is small, so that it could be thin in directions where the magnitude of the second directional derivative is large, as the norm of second order derivative also appears in the error bound. These results are particularly important in the anisotropic case.

Various quality factors have been proposed. The paper from Parthasarathy *et al.* [86] compares different classical values used in mesh generation. An extensive discussion on the quality criterion can be found in [68] and references therein. In this work, we have chosen one of the quality measure proposed by the Gamma project [15]:

$$Q_k = \alpha \frac{h_k}{\rho_k} \tag{3.8}$$

for the isotropic case in two and three dimensions and the anisotropic two dimensional case, and:

$$Q_k = \beta \max_{0<i<dim+1} \frac{\sum_{j=0}^{6} l^2_{\mathcal{M}_i}}{V_{\mathcal{M}_i}} \tag{3.9}$$

for the three dimensional anisotropic case, where α and β are parameters such that the quality of a regular simplex be equal to 1. Criterion (3.8) is much cheaper than criterion (3.9). As in the anisotropic case the criterion must be

evaluated respect to the metric defined at each point, it represents a non negligeable saving. The isotropic version of the quality measure (3.9) is proposed in [75]. Both criteria characterize the quality of an element by a value in the range $[1, +\infty]$, where a higher value means a worse quality. The quality of the whole mesh is given by:

$$Q_M = \max_{K \in \mathcal{T}} Q_k \tag{3.10}$$

It is also possible to take the inverse of the proposed quality criterias to have quality values in the range $[0, 1]$. A very interesting representation of various quality measures is proposed in [97].

Once more, the difference between the two and the three dimensional case is drastical. The Delaunay triangulation is based on the evaluation of the distance between the center of an element and a point, and the radio of the sphere which contains this element. In two dimensions, a flat element will have an infinite quality as the radio of the innercircle goes to zero, so that there are few probabilities that this element would be constructed during the process. However, only to show that a Delaunay mesh is not necessarily synonym of a good quality mesh, a very simple example is depicted in Figure 3.9 to show that triangle ABP could be as degenerated as one wants by moving P in the direction of edge AB, the mesh being always Delaunay. Nevertheless, it is a simplified explanation of the fact that the Delaunay triangulation comply min-max properties in two dimensions. However, the element which has four points on the same sphere, with the four coplanar or quasi-coplanar points does not have an infinite inner ratio, so that it is completely valid respect to the Delaunay criterion. This well-known element, called "sliver", is the cause of artificially high gradients of the shape functions derivatives and deteriorates dramatically the quality of the numerical solution. It is depicted in Figure 3.10. Furthermore, the whole implementation process is in danger as all the local operations of location will give very poor results. The quality criterion reflects a min-max property that the Delaunay triangulation does not meet in three dimensions. It is the reason why controling the distance in two dimensions is a guarantee of controling the quality, whereas it is not true in the three dimentional case. We could have presented the sliver in the Delaunay kernel which is responsible of its elimination but we prefer to include it here to emphasize the quality aspect and the degeneration produced by the Delaunay triangulation, most of all in three dimensions. The remedy for the kernel module is presented in the already cited reference [43]. However, even if the very bad elements have been eliminated, bad elements are still presents in the mesh. That is why the optimization process is stricly necessary in three dimensions, particularly for the kind of application that we have in mind as we will see later on.

Once more, local operators or sequences of local operators [73, 64] are applied, but this time the process is driven by the quality function; the mesh modification is simulated, and if the quality measure is increased, the operation is applied on the mesh. Here, two general types of operators are applied:

- The topological operators which change the connections of the mesh without changing the position of the nodes. In two dimensions, the unique

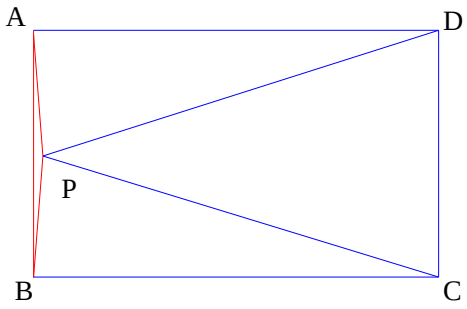


Figure 3.9: A Delaunay mesh with a degenerated triangle

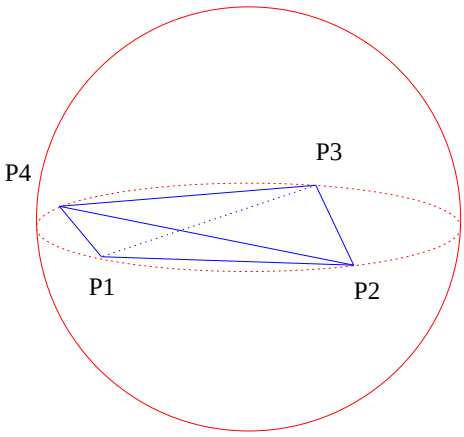


Figure 3.10: A sliver element. The four points are cospherical and coplanar.

topological operator is the edge swapping which consists for two adjacent elements in deleting the common edge and creating the other diagonal of this quadrangle. In three dimensions, this operator is much more complex and terribly effective. In three dimensions, a shell is the set of element sharing a same edge. The shell transformation operator is the generalization of the edge swapping in two dimensions. The points of this set which do not belong to this edge form a polygon in the three dimensional space. The target is to find a triangulation of this polygon, even if the shell is not convex, that increases the quality of the shell, choosing as quality the quality of its worst element. As the number of possible triangulations corresponds to the Catalan number

$$Cat(n) = \frac{(2n-2)!}{n!(n-1)!} \tag{3.11}$$

which has an exponential asymptotic behaviour, an efficient coding must be designed [48, 85, 39, 80]. Classical optimizations consists in coding at hand all the possibilities, applying this operator in an ordered way on the elements with the worst quality, rejecting as quick as possible the configurations with negative volume or with worst quality in function of their frequency, marking at each pass only the modified edges, and storing the quality and volume for the already met tetrahedras. This is the operator implemented in the present work and illustrated in Figure 3.11. The possible triangulations for the case of five element sharing the edge are illustrated in Figure 3.12. The same operator could be applied to relax the valence of a node, defined as the number of edges emanating from this point, where 6 is an ideal number in two dimensions and 12 in three dimensions [8].

- The geometric operators which change the position of the nodes without changing the connectivity. Various Laplacian smoothing have been investigated in the literature [36, 41, 57, 27, 76] but are used only for Steiner points in this work, as already seen.

By modifying the quality criterion from isotropic to anisotropic, the same operators can be used to extend the optimization process to anisotropic considerations [15, 28, 73]. A completely different approach is proposed in [16, 107]. As the mesh quality function for the whole triangulation is not differentiable, the optimization process is considered as a whole, and minimization methods of classical optimization are applied to the mesh.

As a final remark of this section, all that was said on quality optimization is correct if the volume of each simplex of the triangulation is positive. If this is not the case, as for example the collapsing operator for two points that can be deleted, and for which an initial point in the union of both balls must be tried as first candidate for the collapse, it is the minimization of the volume that plays the role of the quality measurement. A first possibility is to find the

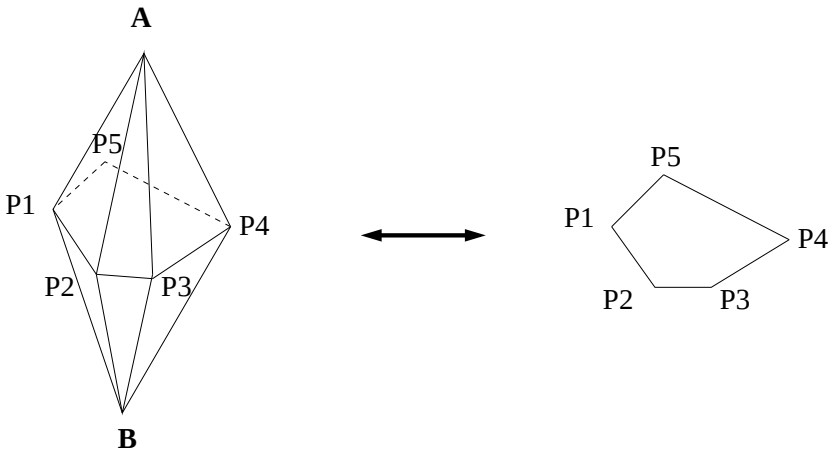


Figure 3.11: Shell AB consisting of 5 elements

minimal volume of the absolute volume of all the tetrahedras [54]:

$$\min_{K \in \mathcal{T}} |V_K| \tag{3.12}$$

A second strategy is to minimize the absolute volume of the elements of negative volume only, as proposed in [49]:

$$\min_{K \in \mathcal{T} | V_K < 0} |V_K| \tag{3.13}$$

Once all the elements have positive volume, the optimization criterion becomes again the classical one.

3.3.4 Interpolation

When frequent remeshings are performed, the solution must be transfered from the old mesh to the new mesh. If new points are inserted in the mesh, their unknown values must be deduced from the old mesh. If no new point is inserted but the topology of the old mesh is changed, variation in the unknowns, and particularly in the unknown spatial derivatives are changed. The interpolation process must at least guarantee the same order of interpolation as the one used in the spatial discretization. Various methods have been proposed in the literature, mainly depending on the desired accuracy.

The most simple process is to interpolate a C^0 field defined on the domain by the nodal values. If the values are elemental or discontinuous, a L^2 projection is required to obtain the least-square solution. This is the case for discontinuous pressures, as will be seen in the next chapter. It appears also for elemental and historical values, as met in plasticity problems as in [88]. It could also be the case for fluid-structure interaction, for an accurate transfer of the pressure on the solid structure [18, 19]. The interpolation can also take restrictions and

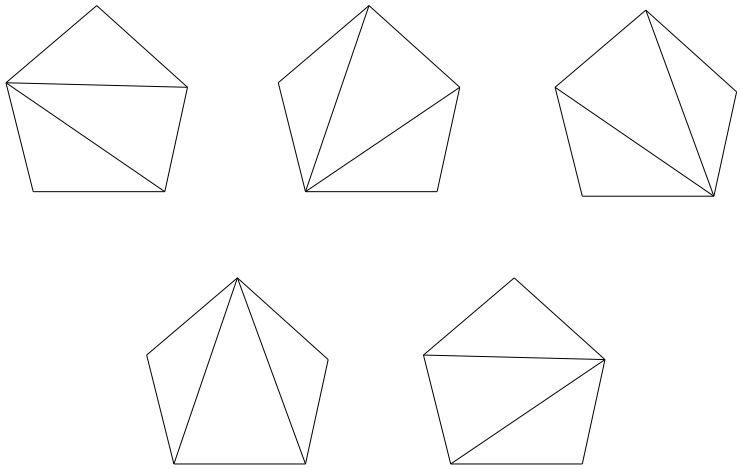


Figure 3.12: The five triangulations of a planar pentagon

constrains into account [58, 56], or must verify given criterias [102]. An alternative is proposed in [56], where only local remeshing techniques are performed, so that the interpolation process can also take advantage of the local aspect in order to avoid the appearently more costly described below. Finally the error in time due to the interpolation is studied in [96, 3].

However, whatever scheme used, the implementation consists in finding the host element of some points, which means the element where each point falls. These points could be nodal or integration points. This search could be as costly as a $O(n^2)$ process, by scanning for each point all the list of elements, the brute force option. In this work, remeshings are performed every time step. The interpolation procedure must therefore be accurate and very fast. However, few points are introduced from one time step to another, usually less than five percent, so that the interpolation error is somehow minimized. The whole algorithm must nevertheless update the few introduced points. In this work, the algorithm proposed in [78] has been implemented. It mainly relies on a greedy approach to begin the search, coupled with an efficient walk-through search, as seen in the Delaunay kernel section, and a spatial data structure, coupled with brute force for difficult cases. The algorithm is amazingly fast and robust. As a remark, the walk-through search is a potentially $O(n^2)$ process. It is however the fastest process to walk through a mesh [80], so that care must be taken with the asymptotical complexities.

3.3.5 Special features

Until then, we have presented the classical approach of a Delaunay-based mesh generator. However, the whole mesh generation will be affected by the particularities of the Lagrangian formulation. We briefly review each presented point emphasizing the particularities encountered in our application.

The Delaunay kernel

During the construction of the Delaunay triangulation, we already mentioned a way of avoiding flat elements in three dimensions. This process is based on the computation of the degeneracy of the volume of the new elements. If this degeneracy is too high, the new element is not formed. However, the first elements that contain the new inserted points can not be discarded, so that a high value of tolerance will produce a mesh where the new elements are only formed by joining the new points with the faces of the element containing the new point. This mesh is evidently not Delaunay, and has poor properties. At this opposite, a low tolerance will allow the formation of slivers, but the mesh will be much closer to a Delaunay mesh. As will be seen in Chapter 5, a method used to determine the fluid locus is the alpha-shape method which relies on the properties of a Delaunay mesh. A high tolerance will provoke a bad surface description by the alpha shape algorithm. An appropriate tolerance must be used to balance these two contradictory phenomenons if the alpha-shape method is chosen.

Boundary regeneration

The boundary regeneration is very important for a particularity of the Lagrangian flow. Even if we suppose that we verify exactly the incompressibility condition, which is not the case, this condition can not avoid that two particles get closer and closer, but much worst, this can not avoid a particle to get closer and closer from the wall boundaries, so that a degenerated element is formed or the particle goes through the wall. Such a particle has to be eliminated when the normal distance of the wall is less than a given tolerance. This can only be achieved by knowing explicitly the boundaries of the solid part.

Node creation

A very interesting feature of the Delaunay-based mesh appears in the fact that the node creation and insertion are performed separately so that the nodes of the old iteration in time can be used to compute the new mesh. Furthermore, the interpolation error is greatly reduced. The interpolation search is also greatly facilited by taking advantage of the convex context of the incremental algorithm.

Quality optimization

This is the most important special feature of the Lagrangian formulation. Negative Jacobian are evidently not acceptable, reflecting the fact that elements reverse. To avoid this phenomenon, the time step is limited by the distance of the normal projection of a node to the opposite face of an element, weighted by the respective velocities. If a first order fractional step method is chosen as fluid solver, only well-shaped elements will produce a larger time step, which is fundamental as much for the whole computation process as for the pressure stability, whose stabilization term is of order $O(\delta t)$. However, as the velocity of

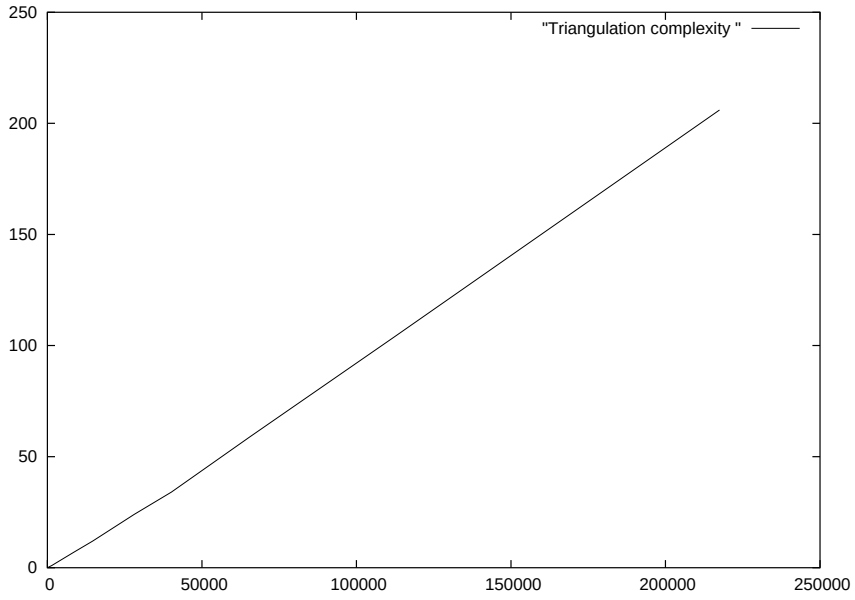


Figure 3.13: Linear relationship between nodes and time

the next step could only be predicted, a special procedure has been implemented to come back at the precedent instant with a shorter time step, in a backtrace manner. Evidently, this mechanism only should be for emergency use as the whole non linear time step computation must be reinitialized.

3.4 Numerical examples

The goal of this section is to illustrate numericaly all the themes and implementations discussed in this section.

3.4.1 Complexity of the Delaunay kernel

As a first result, we are interested in demonstrating the linear behaviour of the Delaunay incremental algorithm. In Figure 3.13, we have reported the number of nodes in abscisse versus the time to generate the triangulation process. The exemple is a cube discretized by a different number of points. Here, the time refers only to the triangulation algorithm. The points were previously created, the boundary was not regenerated, and no optimization procedure was applied. We clearly see the linear behaviour of the algorithm for a number of nodes until $22 \cdot 10^4$, and 1 million of elements.

3.4.2 An industrial part

Figure 3.14 presents two industrial parts to reach one milion elements. A detail of the surface mesh is also presented. This particular surface defintion has been obtained by the alpha shape method as the main goal was to study the Delaunay kernel efficiency.

3.4.3 An isotropic case with size specification

Figure 3.15 illustrates the complete capability of the mesh in three dimensions. This exemple is a cube centered on zero with 1 as side length. Here, a size map of the form:

$$h(x,y) = \begin{cases} 0.995d + 0.5 & \text{if } d < 0.5 \\ 0.005 & \text{elsewhere} \end{cases} \tag{3.14}$$

where $d = \sqrt{x^2 + y^2 + z^2}$ as the cube is centered on 0. The mesh consists in $4 \cdot 10^6$ nodes and 2 millions elements. The variation is rather important between the smallest and largest edge. However a quality of 9.38 is obtained after the optimization process, which corresponds to an excellent mesh as this is the quality of the worst element. It was not so obvious to achieve such a quality as a quick change of size will degrade the quality of the mesh.

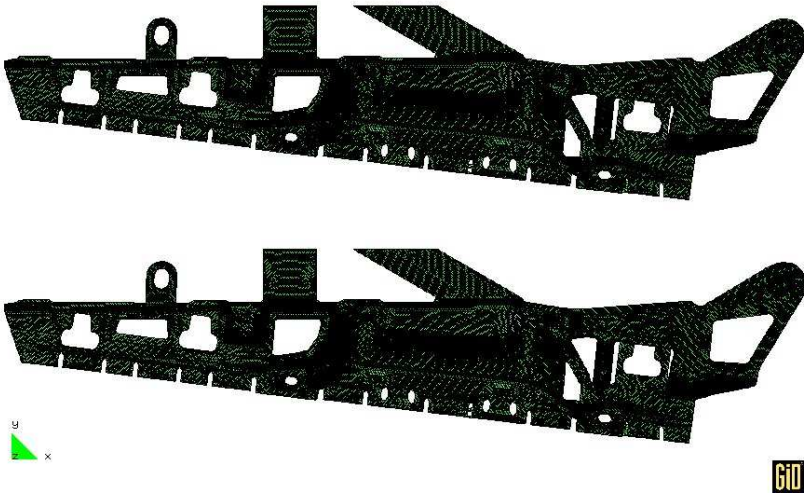
3.4.4 Anisotropic examples

Figure 3.16 illustrates a two dimensional unit cavity meshed with an anisotropic metric. The metric is radial, with 0.1 and 1 as principal values in the radial and tangential direction outside a circle of radio 0.4, and the tangential component changes linearly until being equal to the radial component the center of the cercle. The colors represent the temperature distribution a time $t = 0$ and allows to see that the colour goes further in the domain in the most anisotropic triangles.

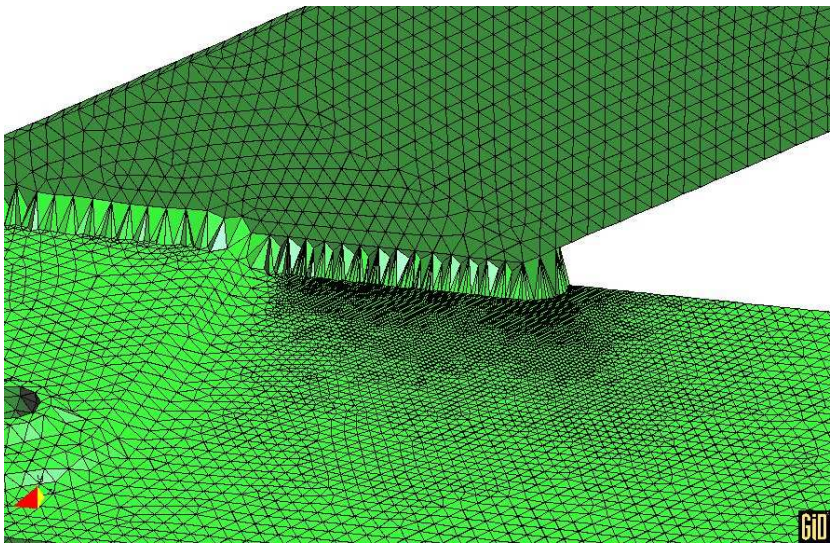
Figure 3.17 illustrates a simple three dimensional anisotropic example. The domain is paralelepipedic, and one dimension is ten times smaller than the two others. The first picture represents the surface mesh, the second a cut through the volume. Three iterations of optimization without moving the points increase the quality from 33 to 11 to 5 to 3. The mesh is particularly simple but the filter, the node creation and the optimization part work well. The point placement is obtained by the edge cutting process, and cycles of adaptation will surely improve their placement, but here only the original placement is shown to emphasize the acceptable distribution of points.

3.5 Conclusion

In this chapter, the mesh generation part of this work by a Delaunay based mesh generator has been presented. Various Delaunay triangulations algorithms have been reviewed before highlighting the incremental version, which is at

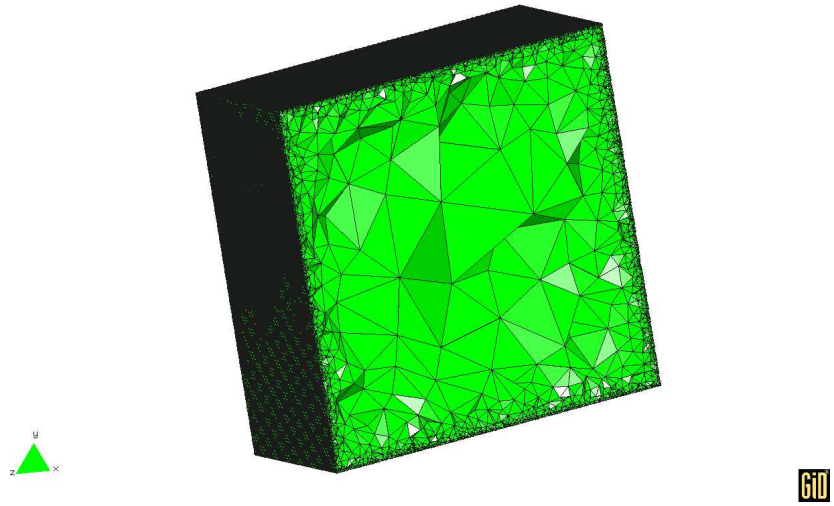


(a) Mesh of both parts

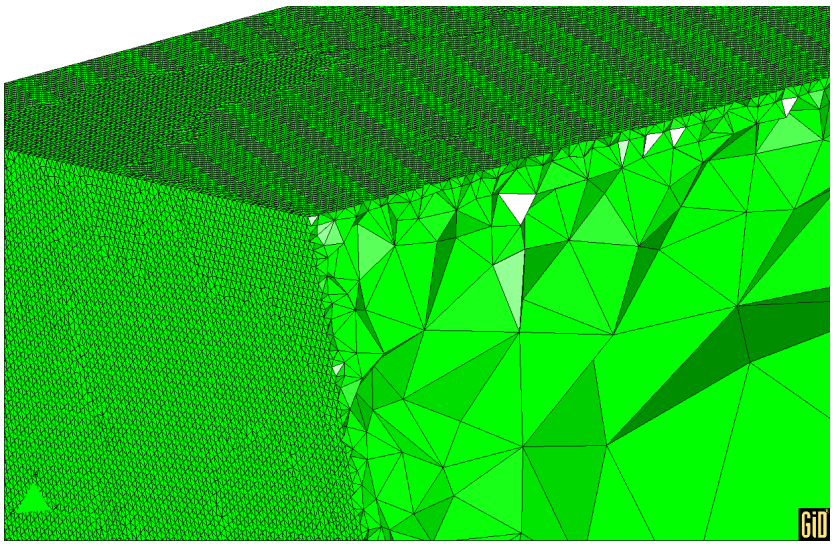


(b) Detail of the mesh

Figure 3.14: Industrial part



(a) Cut through the mesh



(b) Detail of the corner

Figure 3.15: Cube with h varying from 0.005 to 0.5

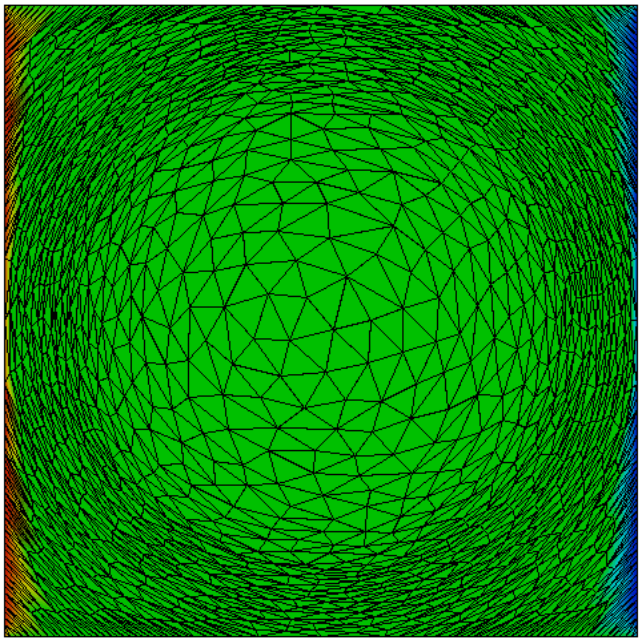
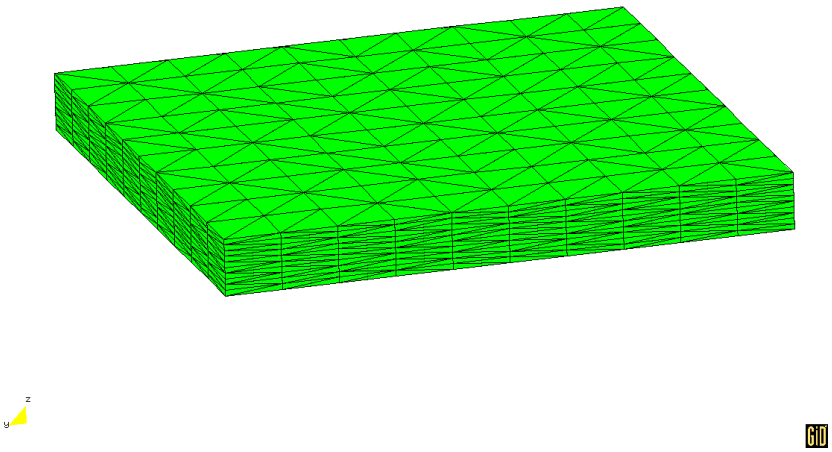


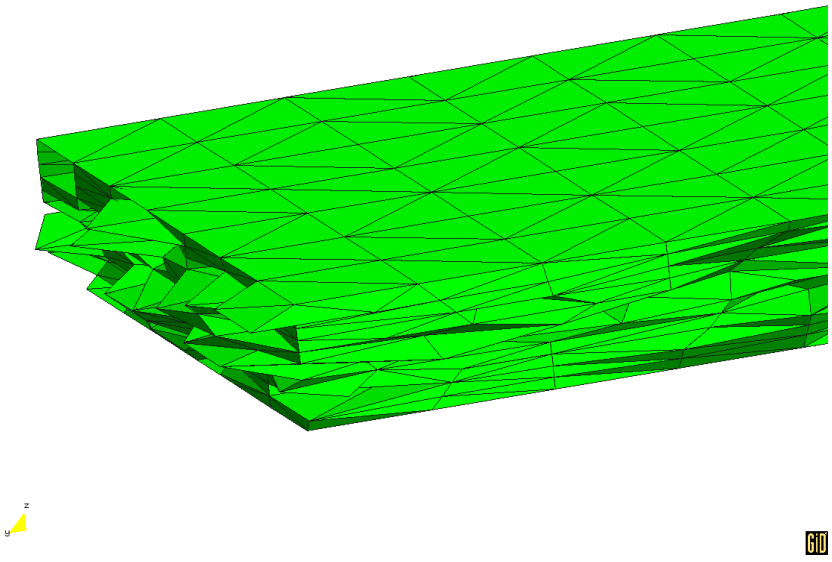
Figure 3.16: A two dimensional anisotropic cavity

the heart of the mesh generator. Various aspects of mesh generation such as boundary recovery, node creation and optimization process have been described. Anisotropy have been slightly studied, but is one of the most promising aspects of the method. Some numerical examples have also been given to show the efficiency and the robustness of the method in 2D and particularly in 3D. The linear behavior of the Delaunay triangulation has been clearly shown in 3D and meshes up to one million elements have been used for computation as seen in the next chapters. The efficiency of the method is mainly due to the global nature of the Delaunay triangulation. However, if this global nature offers an appealing context to find, interpolate and generate, it also creates its weak point. The maximum dimension of the object to be meshed being given, it is a simple matter to decrease the local size of the mesh so that the ratio between the shortest and the longest length tends to something as small as desired, producing numerical difficulties. Thinking to large meshes, this could become surely a limitation for the boundary regeneration part and for the Delaunay kernel insertion, which as noted in [49], gives the numerical resolution of the whole method. A coupling between the Delaunay triangulation and another technique such as [81] could be an interesting option to force the local context to be favorable. This point will constitute the basis of future works.

It must also be noted that only the volumic meshing problem has been considered, so that, even if the domain is remeshed, the boundaries of the domain stay the same. The boundary meshing constitues a topic by itself, which has not



(a) Surface mesh



(b) Volume mesh

Figure 3.17: A three dimensional anisotropic example

been tackled here. The limitation due to the fact of not adapting the boundary of the volume will constitute the next step of research.

In the Lagrangian formulation in particular, and in the Finite Element Method in general, mesh generation plays a fundamental part on the quality of the numerical results. In this work, the mesh generation is fully coupled to the numerical scheme: the solver provides the position of most of the nodes and even defines its boundaries, whereas the mesh gives the necessary spatial discretization to the numerical scheme. Black boxes for both of them is a rather limited approach, which tends to underestimate the potential of their coupling.

Bibliography

[1] A.V. Aho and J.E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1974.

[2] A.V. Aho, J.E. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1983.

[3] F. Alauzet. *Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en mécanique des fluides*. PhD thesis, Université Montpellier II, 2003.

[4] F. Aurenhammer and H. Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition*, 17:251–257, 1984.

[5] I. Babuska and A.K. Aziz. On the angle condition in the finite element method. *SIAM J. Num. Anal.*, 13:214–226, 1976.

[6] T.J. Baker. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Eng. Comp.*, 5:161–175, 1989.

[7] T.J. Baker. Shape reconstruction and volume meshing for complex solids. *Int. J. Num. Meth. Eng.*, 32:665–677, 1991.

[8] E. Boender. Reliable Delaunay-based mesh generation and mesh improvement. *Comm. Num. Meth. Eng.*, 10:773–783, 1994.

[9] J.D. Boissonnat, C. Wormser, and M. Yvinec. Anisotropic diagrams: Labelle shewchuk approach revisited. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05)*, pages 266–269, 2005.

[10] J. Bonet and J. Peraire. An Alternating Digital Tree (ADT) algorithm for 3D geometric searching and intersection problems. *Int. J. Num. Meth. Eng.*, 31:1–17, 1991.

[11] H. Borouchaki. *Triangulation sous contraintes en dimension quelconque.* PhD thesis, Paris 6, 1997.

[12] H. Borouchaki and S.H. Lo. Fast Delaunay triangulation in three dimensions. *Comp. Meth. Appl. Mech. Eng.*, 128:153–167, 1995.

[13] A. Bowyer. Computing Dirichlet tessellations. *Comput. J.*, 24:162–166, 1981.

[14] J.H. Bramble and M. Zlamal. Triangular elements in the finite element method. *Math. Comp.*, 24(112):809–820, 1970.

[15] E Brière De L'Isle and P.L. George. Optimisation de maillages tridimensionnels. *Rapport de recherche de l'INRIA*, 1993. In french.

[16] J. Cabello, R. Löhner, and O.P. Jacquotte. Recent improvements of a variational method for the optimization of directionally stretched elements. In *Proc. 4^{th} Int. Conf. on Numerical Grid Generation in CFD and Related Fields*, Swansea, UK, 1994.

[17] P. Cavalcanti and U. Mello. Three-dimensional constrained Delaunay triangulation: A minimalist approach. In *Proc. 8th International Meshing Roundtable, Albuquerque, NM, Sandia National Laboratories*, 1999.

[18] J.R. Cebral and R. Löhner. Conservative load projection and tracking for fluid-structure problems. *AIAA Journal*, 35:687–692, 1997.

[19] J.R. Cebral and R. Löhner. Fluid-structure coupling: Extensions and improvements. *AIAA-97-0858*, 1997.

[20] B. Chazelle. Convex partition of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal of Computing*, 13(3):488–507, 1984.

[21] B. Chazelle. Triangulating a simple polygon in linear time. *Disc. Comput. Geom.*, 6:485–524, 1991.

[22] P.L. Chew. Guaranteed-quality mesh generation for curved surfaces. In *SCG '93: Proceedings of the ninth annual symposium on Computational Geometry*, pages 274–280, 1993.

[23] P. Ciarlet. Basic error estimates for elliptic problems. In P. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume 2, pages 17–351. North-Holland, Amsterdam, 1991.

[24] K.L. Clarkson. Safe and effective determinant evaluation. In *Proc. 31st IEEE Symposium on Foundations of Computer Science*, pages 387–395, Pittsburgh, PA, 1992.

[25] T.H. Cormen, C. Stein, R.L. Rivest, and C.E. Leiserson. *Introduction to Algorithms.* McGraw-Hill Higher Education, 2001.

[26] H. Dannelongue and P. Tanguy. Efiicient data structure for adaptive remeshing with the fem. *J. Comput. Phys.*, 91:94–109, 1990.

[27] E. Dari and G. Buscaglia. Mesh optimization: How to obtain good unstructured 3D finite element meshes with not so good mesh generators. *Structural Optimization*, 8:181–188, 1994.

[28] E. Dari and G. Buscaglia. Anisotropic mesh optimization and its application in adaptivity. *Int. J. Num. Meth. Eng.*, 40(22):4119–4136, 1997.

[29] E.F. D'Azevedo and R.B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.*, 10(6):1063–1075, 1989.

[30] B. Delaunay. Sur la sphère vide. *Bull. Acad. Sci. USSR*, 7:793–800, 1934.

[31] Q. Du and D. Wang. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations. *Int J. Num. Meth. Eng.*, 56(9):1355–1373, 2003.

[32] Q. Du and D. Wang. Boundary recovery for three dimensional conforming Delaunay triangulation. *Comp. Meth. Appl. Mech. Eng.*, 193:2547–2563, 2004.

[33] Q. Du and D. Wang. Constrained boundary recovery for three dimensional Delaunay triangulation. *Int. J. Num. Meth. Eng.*, 61:1471–1500, 2004.

[34] Q. Du and D. Wang. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM J. Sci. Comput.*, 26(3):737–761, 2005.

[35] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete Comput. Geom.*, 1:25–40, 1986.

[36] D.A. Field. Laplacian smoothing and Delaunay triangulations. *Comm. Appl. Num. Meth.*, 4:709–712, 1988.

[37] S. Fortune. A sweepline algorithm for Voronoi diagrams. In *Proceedings of the second annual symposium on Computational Geometry*, pages 313–322, 1986.

[38] S. Fortune and C.J. Van Wyk. Efficient exact arithmetic for computational geometry. In *SCG '93: Proceedings of the ninth annual symposium on Computational Geometry*, pages 163–172, 1993.

[39] L.A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Num. Meth. Eng.*, 40:3979–4002, 1997.

[40] P.J. Frey, H. Borouchaki, and P.L. George. 3D Delaunay mesh generation coupled with advancing-front approach. *Comp. Meth. Appl. Mech. Eng.*, 157:115–131, 1998.

[41] P.J. Frey and P.L. George. *Maillages, applications aux elements finis*. Hermes, 1999.

[42] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1983.

[43] P.L. George. Improvements on Delaunay-based three dimensional automatic mesh generator. *Finite Element in Analysis and Design*, 25:297–317, 1997.

[44] P.L. George. Premières expériences de maillage automatique par une méthode de Delaunay anisotrope en trois dimensions. *Rapport de recherche de l'INRIA*, 2002. In french.

[45] P.L. George and H. Borouchaki. Aspects of 2D Delaunay mesh generation. *Int. J. Num. Meth. Eng.*, 40:1957–1975, 1997.

[46] P.L. George and H. Borouchaki. *Triangulation de Delaunay et maillage*. Hermes, 1998.

[47] P.L. George and H. Borouchaki. Quality mesh generation. *C. R. Acad. Sci. Paris*, 328:505–518, 2000.

[48] P.L. George and H. Borouchaki. Back to edge flips in 3 dimensions. In *Proc. 12th International Meshing Roundtable, Albuquerque, NM, Sandia National Laboratories*, 2003.

[49] P.L. George, H. Borouchaki, and E. Saltel. 'Ultimate' robustness in meshing an arbitrary polyhedron. *Int. J. Num. Meth. Eng.*, 58:1061–1089, 2003.

[50] P.L. George, F. Hecht, and E. Saltel. Automatic 3D mesh generation with prescribed meshed boundaries. *IEEE transactions on magnetics*, 26:771–774, 1990.

[51] P.L. George, F. Hecht, and E. Saltel. Fully automatic mesh generator for 3D domains of any shape. *Impact of computing in science and engineering*, 2:198–218, 1990.

[52] P.L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundaries. *Comp. Meth. Appl. Mech. Eng.*, 92:269–288, 1991.

[53] P.L. George and F. Hermeline. Delaunay's mesh of a convex polyhedron in dimension d. application to arbitrary polyhedra. *Int. J. Num. Meth. Eng.*, 33:975–995, 1992.

[54] C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Comput. Meth. Appl. Mech. Engrg.*, 194(48):4951–4976, 1998.

[55] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACT Transactions on Graphics*, 4:74–123, 1985.

[56] P. Hansbo. Aspects of conservation in finite element flow computations. *Comp. Meth. Appl. Mech. Eng.*, 117:423–437, 1994.

[57] P. Hansbo. Generalized laplacian smoothing of unstructured grids. *Comm. Num. Meth. Eng.*, 11:455–464, 1995.

[58] G. Houzeaux and R. Codina. Transmission conditions with constraints in finite element domain decomposition methods for flow problems. *Comm. Num. Meth. Eng.*, 17:179–190, 2001.

[59] T.J.R. Hughes. Three dimensional boundary constrained triangulations. In E. N. Houstis and J. R. Rice, editors, *Proceedings of the 13th IMACS World Congress*, pages 215–222. Elsevier Science, 1992.

[60] P. Jamet. Estimations d'erreur pour des lments finis droits presque dgnrs. *R.A.I.R.O. Anal. Numr.*, 10:43–60, 1976.

[61] B. Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Stat. Comput.*, 10:718–741, 1989.

[62] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.

[63] B. Joe. Delaunay versus max-min solid angle triangulations for three dimensional mesh generation. *Int. J. Num. Meth. Eng.*, 31:987–997, 1991.

[64] B. Joe. Costruction of three-dimensional improved-quality triangulation using local transformations. *SIAM J. Sci. Comput.*, 16(6):1292–1307, 1995.

[65] B. Joe and R.B. Simpson. Triangular meshes for regions of complicated shape. *Int. J. Num. Meth. Eng.*, 23:751–778, 1991.

[66] K.B. Kaan, M.W. Beall, and M.S. Shephard. Triangulation of arbitrary polyhedra to support automatic mesh generators. *Int. J. Num. Meth. Eng.*, 49:167–191, 2000.

[67] D.G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 18–27, 1979.

[68] P. Labbé, J. Dompierre, F. Guibault, and R. Camarero. Critères de qualité pour les maillages simpliciaux. In P.L. George, editor, *Maillage et adaptation*. Lavoisier, 2001. In French.

[69] F. Labelle and J.R. Shewchuk. Anisotropic Voronoi diagrams and guaranteed quality anisotropic mesh generation. In *Proceedings of the nineteenth annual symposium on Computational Geometry*, pages 191–200. ACM Press, 2003.

[70] C.L. Lawson. Transforming triangulations. *Discrete Math.*, 3:365–372, 1972.

[71] C.L. Lawson. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3:231–246, 1986.

[72] G. Leibon and D. Letscher. Delaunay triangulations and Voronoi diagrams for riemannian manifolds. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational Geometry*, pages 341–349, 2000.

[73] X. Li. *Mesh modification procedures for general 3D non-manifold domains.* PhD thesis, Rensselaer Polytechnic Institute, 2003.

[74] A. Liu and M. Baida. How far flipping can go towards conforming/constrained triangulation. In *Proceedings of the 9th International Meshing Roundtable*, pages 307–315. New-Orleans, 2000.

[75] S.H. Lo. Automatic mesh generation and adaptation by using contours. *Int. J. Num. Meth. Eng.*, 31:689–707, 1991.

[76] S.H. Lo. Optimization of tetrahedral meshes based on element shape measures. *Computer and Structures*, 63:951–961, 1997.

[77] R. Löhner. Some useful data structures for the generation of unstructured grids. *Comm. Appl. Num. Meth.*, pages 123–135, 1988.

[78] R. Löhner. Robust, vectorized search algorithm for interpolation on unstructured grids. *J. Comput. Phys.*, 118:380–387, 1995.

[79] R. Löhner. Extensions and improvements of the advancing front grid generation technique. *Comm. Num. Meth. Eng.*, 12:683–702, 1996.

[80] R. Löhner. *Applied CFD Techniques.* Wiley, 2001.

[81] R. Löhner and E. Oñate. An advancing front point generation technique. *Comm. Num. Meth. Eng.*, 14:1097–1108, 1998.

[82] D.J. Mavriplis. An advancing front Delaunay triangulation algorithm designed for robustness. *J. Comput. Phys.*, 117:90–101, 1995.

[83] M. Murphy, D.M Mount, and C.W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *Int. J. Comp. Geom. App.*, 11(6):669–682, 2001.

[84] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, Cambridge, 1998.

[85] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira, and A.J.H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady state and transient finite element calculations. *Int. J. Num. Meth. Eng.*, 190:3771–3796, 2001.

[86] V.N. Pathasarathy, C.M. Graichen, and A.F. Hathaway. A comparison of tetrahedron quality measures. *Finite elements in Analysis and Design*, 15:255–261, 1993.

[87] P. Pébay. *Delaunay admissibilité a priori en dimensions 2 et 3.* PhD thesis, Université Pierre et Marie Curie - Paris VI, 2000.

[88] D. Peric, C. Hochard, M. Dutko, and D.R.J. Owen. Transfer operators for evolving meshes in small strain elasto-plasticity. *Comp. Meth. Appl. Mech. Engng.*, 137:331–344, 1996.

[89] F.P. Preparata and M.I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag New York, Inc., 1985.

[90] V.T. Rajan. Optimality of the Delaunay triangulation in R^d. *Discrete & Computational Geometry*, 12:189–202, 1994.

[91] P.A. Raviart and J.M. Thomas. *Introduction à l'Analyse Numérique des Equations aux dérivées Partielles.* Masson, 1988.

[92] S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Numer. Anal.*, 29(1):257–270, 1992.

[93] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. In *SODA '93: Selected papers from the fourth annual ACM SIAM symposium on Discrete algorithms*, pages 548–585, 1995.

[94] J Ruppert and R. Seidel. On the difficulty of triangulating three-dimensional non convex polyhedra. *Disc. Comput. Geom.*, 7:227–253, 1992.

[95] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.

[96] P.A.B. Sampaio, P.R.M. Lyra, K. Morgan, and N.P. Weatherill. Petrov-Galerkin solutions of incompressible Navier-Stokes equations in primitive variables with adaptative remeshing. *Comp. Meth. Appl. Mech. Eng.*, 106:143–178, 1993.

[97] J. Sarrate, J. Palau, and A. Huerta. Numerical representation of the quality measures of triangles and triangular meshes. *Comm. Num. Meth. Eng.*, 19(7):551–561, 2003.

[98] M.I. Shamos and D. Hoey. Closest-point problem. In *Proceedings of the 16th IEEE Symposium on Foundations of Computer Science*, pages 151–162, 1975.

[99] D. Sharov and K. Nakahashi. A boundary recovery algorithm for Delaunay tetrahedral meshing. In *5th International Conference on Numerical Grid Generation on Computational Field Simulations*, pages 229–238. Mississippi State University, 1996.

[100] D.N. Shenton and Z.J. Cendes. Three-dimensional finite element mesh generation using Delaunay tesselation. *IEEE Transactions on Magnetics, IEEE*, 21:2535–2538, 1985.

[101] J.R. Shewchuk. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. *Discrete & Computational Geometry*, 18(3):305–363, 1997.

[102] P. Villon, H. Borouchaki, and K. Saanouni. Transfert de champs plastiquement admissibles. *Comptes Rendus Mecanique*, 330:313–318, 2002.

[103] D.F. Watson. Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes. *Comput. J.*, 24:167–172, 1981.

[104] N.P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation. *Int. J. Num. Meth. Eng.*, 37:2005–2039, 1994.

[105] N.P. Weatherill, M.J. Marchant, O. Hassan, and D.L. Marcum. Grid adaptation using a distribution of sources applied to inviscid compressable flow simulations. *Int. J. Num. Meth. Fluids*, 19:739–764, 1994.

[106] J.P. Wright and A.G. Jack. Aspects of three-dimensional constrained Delaunay meshing. *Int. J. Num. Meth. Eng.*, 37:1841–1861, 1994.

[107] P. D. Zavattieri, E. Dari, and G. Buscaglia. Optimization strategies in unstructured mesh generation. *Int. J. Num. Meth. Eng.*, 12:2055–2071, 1996.

Chapter 4

The Generalized Stokes problem

This chapter deals with the numerical procedure used to solve the Navier-Stokes equations in the Lagrangian description. Due to the absence of the convective term, and after the linearization, a generalized Stokes problem has to be solved at each non linear iteration. It differs from a standard Stokes problem by the presence of the inertial term in the equation. The main difficulty of these equations is the incompressibility constrain, which makes the system indefinite. An efficient procedure is mandatory as it represents the heart of the numerical solver. The notations in this chapter are the ones used for standard Eulerian notations as they are clearer than the Lagrangian formalism of Chapter 2.

The Generalized Stokes system and its weak form is firstly presented without taking into account the Lagrangian formulation. The discretization of the weak equations is then performed with a Finite Element Method (FEM) in space and a Finite Difference Method (FDM) in time. Various exact and approximate solvers of the algebraic system are then commented, mainly based on the Uzawa and fractional step like schemes. The mass lumping procedure is highlighted, mainly at high mesh Reynolds number, because of the wiggles due to the inversion of the consistent mass matrix. Finally, numerical results illustrate the theoretical considerations and efficiency of the various proposed schemes, and validate them on the cylinder example.

4.1 The Generalized Stokes equations

In this section, the model of the Stokes system for a transient flow is presented. Its weak form and functional properties such as existence and uniqueness are highlighted. They will be of utmost importance for the spatial discretization presented in the next section.

4.1.1 The incompressible Newtonian fluid

For an incompressible transient Newtonian fluid flow where the convective term has been neglected due to high viscosities or low velocities, the so called creeping flow model, the Generalized Stokes system reads:

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \mathbf{f}_V \tag{4.1}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{4.2}$$

where $\mathbf{D}$ is the symmetric part of the gradient velocity. Boundary conditions must be added to the system. Consider Ω a domain of $\mathcal{R}^n$ and Γ its boundary. Various boundary conditions are:

$$\Gamma_D = \{\partial\Omega \,|\, v_{|\Gamma} = v_\Gamma\} \tag{4.3}$$

$$\Gamma_N = \{\partial\Omega \,|\, \sigma(\mathbf{n}) = \mathbf{0}, \mathbf{n} \text{ exterior normal to } \Gamma_N\} \tag{4.4}$$

where classically Γ_D is the Dirichlet part of the boundary and Γ_N is the Neumann part of the boundary. More boundary conditions can be found in [50]. Also, an initial condition which fulfills the mass conservation equation must be provided.

4.1.2 Weak form and functional setting

Multiplying equations (4.1) and (4.2) by test functions $\mathbf{N}$ and ψ, summing over the whole domain Ω, and integrating by parts some terms, the system reads:

$$\int_\Omega \rho \frac{d\mathbf{v}}{dt} \cdot \mathbf{N}\, d\Omega + \int_\Omega 2\mu(\mathbf{D} : \mathbf{D})\, d\Omega - \int_\Omega p\, \nabla \mathbf{N}\, d\Omega = \int_\Omega \mathbf{f}_V \cdot \mathbf{N}\, d\Omega \tag{4.5}$$

$$-\int_\Omega \psi\, \nabla \cdot \mathbf{v}\, d\Omega = 0 \tag{4.6}$$

It must be noted that, with the weak form of the viscous term used here, the Neumann boundary condition becomes a natural boundary condition. Another form of the viscous term using the Laplacian velocity operator for each component gives rise to the classical scalar product of the gradient of the velocity in the weak form, but the boundary condition is modified. Furthermore, a minus sign has been added in the second equation to ensure the symmetry of the system.

At this point the classical functional setting can be introduced:

Problem 4.1.1 *Given* $\mathbf{v}_0$ *with* $\nabla \cdot \mathbf{v}_0 \in \mathbf{L}^2(\Omega)^d$ *and* $\nabla \cdot \mathbf{v}_0 = 0$, *and* $\mathbf{f_V} \in \mathbf{L}^2(0,T;\mathbf{H}^{-1}(\Omega)^d)$, *find* $(\mathbf{v}, p)$ *belonging to* $L^2(0,T;\mathbf{H}^1_\Gamma(\Omega)^d) \times L^1(0,T;L^2_0(\Omega))$ *solution of equations* (4.5) *and* (4.6) *for all* $(\mathbf{N}, \psi)$ *belonging to* $(\mathbf{H}^1_0(\Omega)^d \times L^2(\Omega))$, *and* $\mathbf{v}$ *satisfying the initial condition.*

where, classically:

$$L^p(\Omega) = \{\psi : \Omega \to \mathcal{R} \,|\, \int_\Omega |\psi|^p \, d\Omega < \infty\} \tag{4.7}$$

$$L^2(0,T;L^p(\Omega)) = \{\psi \in L^p(\Omega) \,|\, \int_0^T \|\psi\|_{L^p}^2 \, dt < \infty\} \tag{4.8}$$

$$L_0^2(\Omega) = \{\psi \in L^2(\Omega) \,|\, \int_\Omega \psi \, d\Omega = 0 \text{ if } \Gamma_N = \emptyset\} \tag{4.9}$$

$$\mathbf{H}^1_\Gamma(\Omega)^d = \{\mathbf{v} \in L^2(\Omega)^d \,|\, \frac{\partial \mathbf{v}}{\partial x_i} \in L^2(\Omega)^d, \mathbf{v}_{|\Gamma_D} = \mathbf{v}_{\Gamma_D}\} \tag{4.10}$$

$$L^2(0,T;\mathbf{H}^1_\Gamma(\Omega)^d) = \{\mathbf{v} \in \mathbf{H}^1_\Gamma(\Omega)^d \,|\, \int_0^T \|v\|_{H^1}^2 \, dt < \infty \ \} \tag{4.11}$$

It is also classical to introduce the following bilinear forms :

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega \rho \frac{d\mathbf{u}}{dt} \cdot \mathbf{v} \, d\Omega + \int_\Omega 2\mu(\mathbf{D} : \mathbf{D}) \, d\Omega \tag{4.12}$$

$$b(\psi,\mathbf{u}) = \int_\Omega \psi \, \nabla \cdot \mathbf{u} \, d\Omega \tag{4.13}$$

so that the final system reads, with $(\cdot)$ the scalar product on $\mathbf{V}$:

$$a(\mathbf{u},\mathbf{v}) - b(p,\mathbf{v}) = (\mathbf{f}_V,\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{H}^1(\Omega)^d \tag{4.14}$$

$$-b(\psi,\mathbf{u}) = 0 \quad \forall \psi \in L^2(\Omega) \tag{4.15}$$

4.1.3 Existence and uniqueness

The problem considered here is linear. For the transient Stokes problem, a way to show existence and uniqueness for the underlying parabolic problem is first to consider the stationary Stokes problem, and then to use a development in series in time by a Faedo-Galerkin method [53, 61]. In this way, existence and uniqueness of the transient Stokes problem rely on existence and uniqueness of the stationary Stokes problem.

In the stationary case, there are various ways to prove them. One possibility is to find the velocity solution in a divergence free space, as done in [61]. The coupling with the pressure disappears and the coercivity in this space guarantees the existence and unicity of the velocity thanks to the Lax-Milgram theorem. The pressure is then recovered afterwards due to a classical theorem of De Rham [23]. Another possibility is to introduce the inf-sup condition at the continuous level and show that this condition and the coercivity of the bilinear form $a(\cdot,\cdot)$ are necessary and sufficient conditions to show the existence of the pair $(\mathbf{u},p)$ as done in [34]. Here the inf-sup mainly guarantees the injectivity of the transposed of the linear operator associated with the bilinear form $b(\cdot,\cdot)$. Also, it is possible to see the Stokes problem as an optimization problem, where the solution is a

saddle point of the functional considered. Finally, existence and uniqueness can be proved by considering a regularized problem where the mass conservation equation is relaxed by the introduction of a parameter multiplied by an elliptic form, and then by considering the limit as the parameter tends to zero.

Another possibility is proposed in [40], where special norms are used to take into account the inertial term arising in the non stationary case, and prove coercivity, continuity and the inf-sup condition in these particular norms, extending the results of the stationary case to the non stationary case, and will be very usefull in the preconditioner section.

4.2 Discretization

In this section, the weak form of the equations derived in the last section are discretized in space and time. The first step of the discretization is the introduction of the discrete spatial counterpart of problem 4.1.1. Here, given $\mathbf{V}_{ht}(\Omega) \subset L^2(0,T;\mathbf{H}^1_\Gamma(\Omega)^d)$ and $Q_{ht}(\Omega) \subset L^2(0,T;L^2(\Omega))$, and $\mathbf{V}_h(\Omega) \subset \mathbf{H}^1_0(\Omega)^d$ and $Q_h(\Omega) \subset L^2(\Omega)$, the discretized in space problem reads:

Problem 4.2.1 *Given $\mathbf{v}_{h,0} \in Q_h(\Omega)$ with $\nabla\cdot\mathbf{v}_{h,0} = 0$ and $\mathbf{f}_\mathbf{V} \in Q_{ht,0}(\Omega)^d$, find $(\mathbf{v},p)$ belonging to $\mathbf{V}_{ht}(\Omega)\times Q_{ht}(\Omega))$ solution of equations (4.5) and (4.6) for all $(\mathbf{N},\psi)$ belonging to $\mathbf{V}_h(\Omega)\times Q_h(\Omega)$, and $\mathbf{v}$ satisfying the initial condition.*

Here, a spatial partition $\{K^e\}$ of finite elements has been previously constructed. The discussion is furthermore limited to simplices, so that:

$$\mathbf{V}_h(\Omega) = \{\mathbf{v}\in C^0(\Omega)^d \mid \mathbf{v}_K \in P_k(K)^d\} \tag{4.16}$$

$$Q_h(\Omega) = \{\psi \mid \psi_K \in P_k(K)\} \tag{4.17}$$

$$Q_{ht,0}(\Omega) = \{\psi \in Q_{ht} \mid \int_\Omega \psi \; d\Omega = 0 \text{ if } \Gamma_N = \emptyset\} \tag{4.18}$$

where:

$$P_k(K) = \{\text{Space of polynomials of degree k on } K^e\} \tag{4.19}$$

Here, only the conforming approximation is considered as $\mathbf{V}_h(\Omega) \subset \mathbf{H}^1_\Gamma(\Omega)^d$ and $Q_h(\Omega) \subset L^2(\Omega)$ but non conforming approaches are also used and could be of great interest [22]. The discrete system then reads:

$$a(\mathbf{u}_h,\mathbf{v}_h) - b(p_h,\mathbf{v}_h) = (\mathbf{f}_{v,h},\mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h \tag{4.20}$$

$$-b(\psi_h,\mathbf{u}_h) = 0 \quad \forall \psi_h \in Q_h \tag{4.21}$$

4.2.1 Compatible pressure/velocity spaces

As seen above at the continuous level, the uniqueness of the pressure is guaranteed by the inf-sup condition. At the discrete level, a discrete inf-sup condition will also imply the invertibility of the whole system, along with the discrete coercivity. But this condition is not inherited from the continuous one, and must

be explicitly verified for the pair of velocity-pressure space under study. The discrete LBB condition then reads:

$$\sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{\mathbf{V}_h}} \geq \beta \|q_h\|_{Q_h} \qquad \forall q_h \in Q_h, \beta > 0 \tag{4.22}$$

Here also, various demonstrations are possible to prove the discrete inf-sup condition for a pair of discrete spaces. Discontinous pressure elements offer the ease to explicitly construct local operators which guarantee that the discrete problem inherits the properties of the continuous system [22, 29], whereas continous pressures are slightly more difficult to tackle due to the non local construction of divergence free basis [1, 65, 59]. We refer to [16, 54] for theoretical details and to [30, 31, 12] for various stable spatial approximations for tetrahedras and hexahedras.

In this work, three kinds of elements have been considered:

- a continuous $P1/P1$ element which does not satisfy the inf-sup but is commonly used with a first-order fractional step, giving some stability properties [19].
- a continuous $P1+/P1$ element belonging to the bubble family [1], where the bubble has been discretized by four tetrahedra, also called the "hat-function", allowing an exact integration [46, 47].
- a $P1++/P1$ element with bubble and linear discontinuous pressure [12].

Only the last two elements are div-stable.

4.2.2 The algebraic system

A classical θ-scheme is used to march in time. For the spatial discretization, a standard mixed finite element method [22] is performed so that the system reads, with $\mathbf{U}$ the displacement and $\mathbf{V}$ the velocity:

$$\frac{\rho}{\delta t}\mathbf{M}_V(\mathbf{V}^{n+1} - \mathbf{V}^n) + \mu \mathbf{K}_V(\mathbf{U}^{n+\theta})\mathbf{V}^{n+\theta} + \mathbf{B}^{\mathbf{T}}(\mathbf{U}^{n+\theta})\mathbf{P}^{n+1} = \mathbf{F}_V^{n+\theta} \tag{4.23}$$

$$\mathbf{B}(\mathbf{U}^{n+\theta})\mathbf{V}^{n+1} = 0 \tag{4.24}$$

where the previous matrices are:

$$\mathbf{M}_{Vij}^{ab} = \delta_{ab} \int_\Omega \mathbf{N}_i \mathbf{N}_j \, dV \tag{4.25}$$

$$\mathbf{K}_{Vij} = \int_\Omega (\nabla \mathbf{N}_i + \nabla^T \mathbf{N}_i) \cdot \nabla \mathbf{N}_j dV \tag{4.26}$$

$$\mathbf{B}_{ij} = -\int_\Omega \psi_i \, \nabla \cdot \mathbf{N}_j \, dV \tag{4.27}$$

where the subscripts refer to the node indices and the superscripts to the space indices, the $\mathbf{N}_i$ and the ψ_i are the velocity and pressure shape functions. The non-linear dependence of the matrices to the displacement variable has been emphasized to denote the Lagrangian description of the discretization. Writing the system (4.23)-(4.24) as a whole system, and including all the known terms in the right hand side, the algebraic system reads:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{F}}_V \\ \mathbf{0} \end{pmatrix} \tag{4.28}$$

Here, (4.28) represents a classical generalized Stokes operator as there is no convective term and $\mathbf{A}$ has the form:

$$\mathbf{A} = \frac{\rho}{\delta t}\mathbf{M}_V + \theta\mu\mathbf{K}_V \tag{4.29}$$

4.3 The Generalized Stokes solver

The previous equations are non linear in the displacement variable as the final domain is not known. After Picard linearization, a generalized Stokes problem needs to be solved at each non linear iteration. This section presents the solver used in this work, by highlighting a variant of the Uzawa method, and its relationships with the algebraic splitting methods.

4.3.1 Approximation of the Pressure Schur Complement

The main difficulty of the Stokes-like problems is the indefiniteness of the algebraic system due to the saddle-point nature of the mixed discretization, and is clearly seen by applying Sylvester Law of Inertia to the congruence transform of the Stokes discretization [27]. As seen in Chapter 1, two main approaches can be distinguished in order to solve a Stokes-like problem with iterative methods: whether the problem is solved as a whole, or 'coupled', or it is solved in a segregated way, for the pressure and then for the velocity [9]. In the first category, two possibilities for a Krylov-based iterative solver which take into account the symmetry of the problem are the MINRES and SYMMLQ solvers [44]. In the second category, the Uzawa method [2] allows to replace a minimisation constrained problem with a sequence of minimization problems without constrains. However, it appears that in both approaches, the bulk of the computation is concentrated on a good approximation of the Pressure Schur Complement (PSC), which mimics the Uzawa operator at the discrete level. Applying block Gaussian elimination to (4.28), the PSC for the generalized Stokes problem reads:

$$\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\tilde{\mathbf{F}}_V \tag{4.30}$$

Matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is SPD up to the hydrostatic pressure mode. A conjugate gradient applied to (4.30) is the basis of the Preconditioned Conjugate Gradient Uzawa (PCGU) preconised in [17, 64, 13] and references therein. Originaly, the

Uzawa method is a stationary Richardson method at the continuous level [55]. The Uzawa operator reads [24]:

$$U_z(p) = -\nabla \cdot (\frac{\rho}{\delta t} I_V - \theta\mu\Delta_V)^{-1}\nabla p = -\nabla \cdot (\frac{\rho}{\delta t} I_V - \theta\mu\Delta_V)^{-1} f \tag{4.31}$$

where $\mathbf{I}_V$ is the identity operator and Δ_V the Laplace operator in the velocity space. This is the equivalent of (4.30) at the continuous level. For a conjugate gradient algorithm, the convergence analysis dictates that the error between the iterate number k and the exact solution in the $\mathbf{U}$-norm of the matrix, with $\mathbf{U} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$, reads [27, 55]:

$$\|\mathbf{e}^k\|_{\mathbf{U}} \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)\|\mathbf{e}^0\|_{\mathbf{U}} \tag{4.32}$$

where κ is the condition number of matrix $\mathbf{U}$, where the U stands for Uzawa. The role of the preconditioner will be to lower κ, or ideally to bring κ independent of the discretization size, to accelerate convergence. It could also cluster the eigenvalues to have a small polynomial approximation error, from which bound (4.32) is obtained [27]. This is the motivation of the following.

The idea of [17] consists in distinguishing two separate behaviours of the Uzawa operator, whether the mesh Reynolds number defined as:

$$Re_m = \frac{\rho\, h_K^2}{\mu\, \theta\, \delta t} \tag{4.33}$$

is high or not, where h_K is an element reference length. In the first case this gives:

$$\mathbf{A} \approx \frac{\rho}{\delta t}\mathbf{I}_V \tag{4.34}$$

as the inertial part is preponderant over the diffusive part, so that:

$$U_z(p) \approx -\nabla \cdot (\frac{\rho}{\delta t} I_V)^{-1}\nabla p \approx -\frac{\delta t}{\rho}\Delta p \tag{4.35}$$

And in the second case:

$$\mathbf{A} \approx -\theta\mu\Delta_V \tag{4.36}$$

so that:

$$U_z(p) \approx -\nabla \cdot (-\theta\mu\Delta_V)^{-1}\nabla p \approx \frac{1}{\theta\mu} p \tag{4.37}$$

Finally, the proposed preconditioner of the Uzawa operator at the continuous level is:

$$\mathbf{C}^{-1} = \theta\mu\mathbf{I}_p^{-1} - \frac{\rho}{\delta t}\Delta_p^{-1} \tag{4.38}$$

At the discrete level, the preconditioner writes:

$$\mathbf{C}^{-1} = \theta\mu\mathbf{M}_p^{-1} + \frac{\rho}{\delta t}\mathbf{K}_p^{-1} \tag{4.39}$$

where $\mathbf{M}_p$ and $\mathbf{K}_p$ are the mass and Laplacian matrix in the pressure space. This idea relies on a Fourier analysis to simulate the multigrid efficiency at all frequencies. Later, [15] and [41] proved the uniform boundedness of the condition number of the preconditioned system with respect to the spatial discretization h and the time step δt. For the classical Stokes problem, the use of the inf-sup condition and the boundedness of the operator $\mathbf{B}$ give the well-known bounds:

$$\exists\, c_1, c_2 > 0, \qquad c_1 \leq \frac{(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T q, q)}{(\frac{1}{\mu}\mathbf{M}_p\, q, q)} \leq c_2 \quad \forall q \in \mathcal{R}^{n_p} \tag{4.40}$$

proving the nice behaviour of the Uzawa operator independently of the mesh size h, and the spectral equivalence between the Uzawa operator and the finite element pressure mass matrix. For the non stationary case, the inf-sup with the $\mathbf{H}_0^1$ norm for the velocity and L_0^2 norm for the pressure still holds but do not reflect the behaviour of the Uzawa operator. The idea of [41] is to use a norm depending on the temporal discretization parameter, the other parameters such as density and viscosity being included in this parameter, to extend equation (4.40). Proving the inf-sup condition and the boundedness of operator $\mathbf{B}$ in this particular norm extends equation (4.40) to the non stationary case and gives:

$$\exists\, c_3, c_4 > 0, \quad c_3 \leq \frac{(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T q, q)}{((\theta\mu\mathbf{M}_p^{-1} + \frac{\rho}{\delta t}\mathbf{K}_p^{-1})^{-1} q, q)} \leq c_4 \;\forall q \in \mathcal{R}^{n_p} \tag{4.41}$$

which reflects the spectral equivalence between the above mentioned preconditioner and the Uzawa operator or, what is the same, the independence of the Uzawa operator respect to the mesh size and the physical and temporal parameters, when used with the preconditioner.

A variant of the last preconditioner, already described in [17], is the use of the so-called compatible or discrete Laplacian [52], which reads:

$$\tilde{\mathbf{C}}^{-1} = \theta\mu\mathbf{M}_p^{-1} + \frac{\rho}{\delta t}(\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T)^{-1} \tag{4.42}$$

An appealing advantage of this discretization appears to be that, as it will be seen later, it opens the way to a fractional step with discontinuous pressure elements. As a direct extension, the PCGU can now be applied to discontinuous pressure elements, as in [13]. The use of the compatible Laplacian reflects the discretization of the Laplacian by a mixed formulation:

$$\begin{pmatrix} \mathbf{M}_V & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{F_p} \end{pmatrix} \tag{4.43}$$

where the natural boundary condition is $p = 0$ on the Neumann boundary of the velocity, due to the integration by parts performed in the mixed formulation (4.43). So that, as noted in [64], with Neuman boundaries for the velocities, matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ is SPD as is matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ with the same boundary conditions, and not semidefinite positive as is the standard Laplacian. On Dirichlet boundaries for the velocity, matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ does not provide any wrong boundary

condition. To sum up, the most important property is due to the fact that the pressure does not need a Dirichlet condition on the free surface as it must be with the standard Laplacian discretization. The superiority of this approach will be proved in the last section, but was already noticed in [17], particularly when the mesh Reynolds number tends to zero. It must be emphasized that the examples provided in [17, 41, 15] do not take into account Neumann boundary conditions for the velocity. In case of free surfaces, Dirichlet conditions must be applied on the standard pressure Laplacian so that if only this preconditioner is used, *the PCGU scheme can not converge* as the imposed pressure degrees of freedom can not reach the mass conservation, which is the residual of the PCGU. With the addition of the pressure mass matrix, the scheme converges, but much slowly than with the compatible Laplacian. However, the bandwith is roughly three times the bandwith of the standard Laplacian in 2D, and five times in 3D on a structured grid, as it involves the neighbours of the neighbours of each nodes in case of continuous pressures and the neighbouring elements for each element in case of discontinuous pressure. A good preconditioner is then mandatory. Furthermore, it could be possible to take advantage of the symmetric form of the matrix to require only the action of $\mathbf{B}$ or $\mathbf{B}^T$ which is already needed for the Stokes solver, as will be shown later. Evidently, the mass matrix must be lumped to have an efficient discretization.

4.3.2 Algebraic splitting and fractional step methods

As noted in [15], the PCGU requires the action of A^{-1} whereas the global approach only requires a spectrally equivalent preconditioner. However, both methods will need to iterate and, in particular, to apply the action of the above preconditioner to each iteration. On the other hand, algebraic splitting methods offer a good compromise between accuracy and efficiency [56]. In this work, fractional step methods are encompassed in the more general algebraic splitting schemes, even if discretization in time is performed before spatial discretization in the fractional step methods, as opposed to algebraic splitting, which rely on the algebraic monolithic system. This point of view is motivated by only considering the final algebraic system of equations. By fractional step, we mean that a standard Laplacian is used for the pressure Poisson equation, and no final update of the pressure is performed. In [64], the equivalence between the continuous fractional step method and one iteration of the PSC preconditioned by the standard Laplacian was already noticed. Due to the previous discussion on the behaviour of the Uzawa operator and on the imposition of the pressure degrees on the free surface, it seems natural to replace the standard Laplacian by the compatible one and to use a pressure mass matrix to approximate the Uzawa operator at low mesh Reynolds number.

A very interesting scheme already proposed in [64] balances the accuracy of the PSC and the efficiency of a second order algebraic splitting. One solve reads:

$$\left(\frac{\rho}{\delta t}\mathbf{M}_V + \theta\mu\mathbf{K}_V\right)\tilde{\mathbf{V}} = \mathbf{F}_V - \mathbf{B}^T\mathbf{P}^n + \left(\frac{\rho}{\delta t}\mathbf{M}_V - (1-\theta)\mu\mathbf{K}_V\right)\mathbf{V}^n \quad (4.44)$$

$$\frac{\delta t}{\rho}\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T\Delta\mathbf{P}^\star = \mathbf{B}\tilde{\mathbf{V}} \quad (4.45)$$

$$\mathbf{V}^{n+1} = \tilde{\mathbf{V}} + \frac{\delta t}{\rho}\mathbf{M}_V^{-1}\mathbf{B}^T\Delta\mathbf{P}^\star \quad (4.46)$$

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \Delta\mathbf{P}^\star + \theta\mu\mathbf{M}_p^{-1}\mathbf{B}\tilde{\mathbf{V}} \quad (4.47)$$

where $\Delta P^\star$ could be considered as the pressure increment due to the non viscous case and the rightmost term of equation (4.47) as the pressure increment due to the purely viscous case. Compared to a classical fractional step, the standard Laplacian has been replaced by the compatible Laplacian and the pressure has been updated to take into account the behaviour of the PSC in case of low mesh Reynolds number. It mainly consists of one iteration of the PCGU scheme, as noted in [64], except that equation (4.46) has been added to perform the projection step, which gives an exact discretely divergence free velocity due to the compatible Laplacian. However, to go just one step further, the PCG algorithm is recalled on Figure 4.1. The implementation is the one proposed in [55]. As usual, x_0 is the initial guess, r_j the residual, z_j the residual of the preconditioned system and p_j the search direction at iteration j. Applying the PCG on the PSC equation (4.30) and rearranging the terms with the previously mentioned preconditioner, one sees that the algebraic splitting corresponds exactly to the initialisation process equations (4.48a)-(4.48c) only. The final pressure of the modified algebraic splitting corresponds to the initial preconditioned residual, or equivalently to the first search direction. Furthermore, in the PCG algorithm, the first update of the unknown happens only at equation (4.48e). So that, in order for the first approximation to coincide with the final pressure value of the modified algebraic splitting, one must have $\alpha_0 = 1$. As α_j represents the value which maximizes the value of the quadratic functional associated with $\mathbf{A}$ in the p_j direction, or what is the same, brings the residuals orthogonals between each other, the pressure value obtained by the modified algebraic splitting is not in general the optimal value obtained for one iteration, except if $M^{-1} = A$ trivially. A different algebraic splitting could then be constructed by computing the value of α to update the pressure. It requires however one more matrix multiplication, or in the PSC context, one more inversion of the velocity system, which can not be used if one wants to verify the mass conservation. This would be very near to the Yosida method proposed in [52] with the complete preconditioner, as the final velocity would be the same and a pressure update would be performed.

More important are the controversial boundary conditions on pressure. As the fractional step does not reach the actualization step of the PCGU, the final pressure-like result of the fractional step *is not an approximation of the pressure but the residual of the preconditioned system*, so that boundary conditions for the pressure are the one inherited from the preconditioners. Whereas boundary conditions can be rather freely chosen for the preconditioner, they produce the

Given x_0:

$$r_0 = b - Ax_0 \quad (4.48a)$$
$$z_0 = M^{-1}r_0 \quad (4.48b)$$
$$p_0 = z_0 \quad (4.48c)$$

while(not converged):

$$\alpha_j = (r_j, z_j)/(Ap_j, p_j) \quad (4.48d)$$
$$x_{j+1} = x_j + \alpha_j p_j \quad (4.48e)$$
$$r_{j+1} = r_j - \alpha_j Ap_j \quad (4.48f)$$
$$z_{j+1} = M^{-1}r_{j+1} \quad (4.48g)$$
$$\beta_j = (r_{j+1}, z_{j+1})/(r_j, z_j) \quad (4.48h)$$
$$p_{j+1} = z_{j+1} + \beta_j p_j \quad (4.48i)$$

Figure 4.1: The Preconditioned Conjugate Gradient algorithm.

serious well-known drawback in the classical fractional step. A remedy is to change the preconditioner used, as proposed in this section, and also considered in [63, 51], but the main reason is constituted by the fact that the update of the pressure in the PCG is not performed. As will be seen in the last section, in case of a classical fractional step, it is better to impose the pressure on the free surface to the values given by the right-hand side of (4.45).

A straightforward extension of the PCGU and the algebraic splitting is the scheme used in this work, where the PCGU is used as a basic solver. If the convergence rate is too slow, or the maximum number of iteration is reached, the velocity iterate goes to the projection step (4.46), and the pressure is ultimately updated by equation (4.47). A complementary possibility is to evaluate the Reynolds mesh number and then activate or desactivate the proposed preconditioners. Concerning the velocity solve, a Symmetric Gauss Seidel (SGS) preconditioner is used to solve the velocity system implemented with Eisenstat's trick [26].

4.3.3 The mini element in details

Adding a bubble usually enhances stability. This enhancement can however take various forms, as illustrated by the $P1+/P1$ element and the $P1++/P1$ element. For the $P1+/P1$ element, equivalence between the bubble element and stabilized formulation has been proved [48, 49, 58, 5, 6]. Actually, the bubble gives the local Schur complement $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ which, in case of the Stokes problem with the $P1+/P1$ element is equivalent to a Laplacian due to the local property of the bubble as will be seen thereafter. However, the bubble for the $P1++/P1$ element gives stability for the linear part of the pressure only, the element being already stable for $\mathbf{P}_0$ pressure. Furthermore, if the

static condensation of the bubble in the mini element gives rise to a symmetric positive definite (SPD) block in $\mathbf{A}$ (up to the hydrostatic pressure mode), it is not the same by condensing the bubble and the linear part of the pressure of the $P1++/P1$. The block in $\mathbf{A}$ is then indefinite, reflecting the coupling between pressure and velocity. This result was verified by computing the eigenvalues of the block in $\mathbf{A}$ after condensation.

From now on, concentrating on the $P1+/P1$ element to highlight its properties, a monolithic discretization with the bubble element will give:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{A}_{ib} \\ \mathbf{B} & \mathbf{0} & \mathbf{B}_b \\ \mathbf{A}_{bi} & \mathbf{B}_b^T & \mathbf{A}_b \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{U}_b \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \\ \mathbf{F}_b \end{pmatrix} \tag{4.49}$$

Here, coupling between velocity components appears though the mass matrix, contrarily to the classical Stokes problem so that, after condensation of the bubble degrees, the system reads:

$$\begin{pmatrix} \mathbf{A} - \mathbf{A}_{ib}\mathbf{A}_b^{-1}\mathbf{A}_{bi} & \mathbf{B}^T - \mathbf{A}_{ib}\mathbf{A}_b^{-1}\mathbf{B}_b^T \\ \mathbf{B} - \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{A}_{bi} & -\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F} - \mathbf{A}_{ib}\mathbf{A}_b^{-1}\mathbf{F}_b \\ -\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b \end{pmatrix} \tag{4.50}$$

By assuming a "kind of" L^2-orthogonality [7] between the bubble and the linear part of the velocity, or by lumping the mass matrix, or by simply neglecting the numerical values due to the coupling [58], one obtains:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ -\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b \end{pmatrix} \tag{4.51}$$

so that, compared to the equivalent $P1/P1$ discretization, matrix $-\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T$ has been added in the mass conservation equation to stabilize the divergence operator, unstable in this case. Furthermore, the bubble right hand side has also been added to the pressure right hand side. The analogue of the PSC for the generalized Stokes problem of equation (4.30) reads:

$$(\tilde{\mathbf{B}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}}^T + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\mathbf{P} = \tilde{\mathbf{B}}\tilde{\mathbf{A}}^{-1}\mathbf{F} + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b \tag{4.52}$$

where

$$\tilde{\mathbf{B}} = \mathbf{B} - \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{A}_{bi} \tag{4.53}$$

$$\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{A}_{ib}\mathbf{A}_b^{-1}\mathbf{A}_{bi} \tag{4.54}$$

$$\tilde{\mathbf{B}}^T = \mathbf{B}^T - \mathbf{A}_{ib}\mathbf{A}_b^{-1}\mathbf{B}_b^T \tag{4.55}$$

Once more, supposing a L^2-orthogonality between the bubble and the linear part of the velocity, equation (4.52) reads:

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\mathbf{F} + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b \tag{4.56}$$

Writing the complete PCGU algorithm for the condensate bubble, one obtains the algorithm detailed on Figure 4.2. The three first equations correspond

to a classical fractional step scheme. Equation (4.61b) computes a predictive velocity and equation (4.61c) computes the initial residual of the pressure, the weak divergence of the predictive velocity. Then comes equation (4.61d) which constitutes the pressure Poisson or pressure correction equation, but stands here as the preconditioned system. Equation (4.61e) is the initialization of the search direction and does not appear in a classical fractional step. At this time, whether the projection step can be performed and the procedure exited, or an additional pressure update can be added and the procedure exited if the monolithic solver is not chosen. If it is, the procedure does not perform neither the projection step nor the pressure update and the PCGU follows until convergence. An additional possibility is to fix the maximum number of PCGU iterations and if reached, to go to the projection step with or without the pressure update, as already mentioned.

It is also interesting to see the stabilization effect added by the bubble condensation [48, 6], at the global level on the Pressure Schur Complement, and at the local level on the pressure Poisson equation. If a first order fractional step scheme is chosen with a standard Laplacian equation, the only difference appears due to the bubble right hand side in the velocity residual on equation (4.61c) at the global level. It happens to be natural as the fractional step possesses a rather good stability for a first order scheme [19]. For a second order scheme, the operator $\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T$ enhances the stability of the pressure Schur Complement but only at the right hand side of equation (4.61c), so that it is really the divergence operator of the $P1/P1$ discretization which is stabilized, and no additional term appears in the standard Laplacian operator. Finally, the additional term $\mathbf{B}_b\mathbf{M}_b^{-1}\mathbf{B}_b^T$ brings the compatible Laplacian operator stable in equation (4.61d).

As both operators $\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T$ and $\mathbf{B}_b\mathbf{M}_b^{-1}\mathbf{B}_b^T$ enhance the stability of the scheme, the next step is to explicitly write the behaviour of these operators. Using the hat-function bubble, which is piecewise linear on each element and null on the boundary of the element, the numerical integration is performed exactly as the shape function gradient is constant on the element, so that, denoting by V_k the volume of element K and d the dimension:

$$\begin{aligned}
\mathbf{B}_{bij} &= -\int_K \psi_i \nabla \cdot \mathbf{N}_{bj}\, dV && (4.57a)\\
&= \int_K \nabla\psi_i \cdot \mathbf{N}_{bj}\, dV && (4.57b)\\
&= \nabla\psi_i \times \frac{V_K}{d+1} && (4.57c)
\end{aligned}$$

where an integration by parts has been performed. Supposing that no velocity coupling appears due to boundary conditions, and after lumping the bubble mass matrix, the velocity operator for the bubble reads:

$$\mathbf{A}_b = \Big(\frac{\rho}{(d+1)\times\delta t} + \theta\mu(\nabla N_b)^2\Big) \times V_K \qquad (4.58)$$

Finally, denoting by $\mathbf{L}$ the elementary Laplacian matrix, the bubble stabilization operator for the Pressure Schur Complement reads:

$$\begin{aligned}
\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T &= \frac{V_K}{(d+1)^2} \times \frac{1}{\frac{\rho}{(d+1)\times\delta t} + \theta\mu(\nabla N_b)^2}\nabla\psi_i\nabla\psi_j && \text{(4.59a)}\\
&= \frac{1}{(d+1)^2} \times \frac{1}{\frac{\rho}{(d+1)\times\delta t} + \theta\mu(\nabla N_b)^2}\,\mathbf{L} && \text{(4.59b)}\\
&= \tau_K\,\mathbf{L} && \text{(4.59c)}
\end{aligned}$$

where τ_K is the stabilization parameter given by the bubble. It can be seen that the bubble parameter takes into account the inertial and the viscous operators. Should convection have been considered, another term would have appeared in the stabilization parameter. This kind of stabilization parameter is very similar to other methods [43, 20], the only difference appears in the fact that the stabilization coefficient is given by the bubble shape function. In the same way, the bubble stabilization for the pressure Poisson equation reads:

$$\mathbf{B}_b\mathbf{M}_b^{-1}\mathbf{B}_b^T = \frac{\delta t}{(d+1)\times\rho}\mathbf{L} \tag{4.60}$$

It must be noted however that *only the inertial operator* appears in the stabilization parameter. As seen before, the pressure Poisson equation is a preconditioner for the Uzawa operator efficient only at high Reynolds mesh number. It is then natural that only the inertial term appears in the stabilization parameter. It must be noted also that, due to the inertial term of null order, the stabilization introduced is only second order accurate in space, which differs from other methods which use a fourth order damping whether directly [39], or by another method [57].

Finally, the PCGU stabilized with the bubble element is rather close to predictor-corrector schemes, like the one proposed in [8, 21]. However, apart from considering only the Laplacian preconditioner, there is no guarantee of convergence of the iterative scheme. Furthermore, this scheme does not possess the optimal properties of the conjugate gradient algorithm. Nevertheless, this remark is only valid for a symmetric velocity operator. If not, matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is not symmetric and the PCGU can not be applied whereas the predictor-corrector remains valid. Extensions to the non symmetric case of the PSC are proposed in [33].

4.3.4 A preconditioner for the preconditioner

As noted in [38], the costly part of a classical fractional step at high mesh Reynolds number is the Laplacian solve, mainly because if a small time step is used, the velocity system is very well conditioned whereas the time step has no influence on the pressure Laplacian. The same happens for the discrete Laplacian.

As already mentioned, the use of the compatible Laplacian $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T\mathbf{P} = \mathbf{F}_p$ reflects the discretization of the Laplacian by a mixed formulation. As noted

Given $\mathbf{P}_0$:

$$\begin{aligned}
\mathbf{R}_0 &= (\mathbf{B}\mathbf{A}^{-1}\mathbf{F} + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b) && (4.61a)\\
&- (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\,\mathbf{P}_0 \\
&\text{Computed as:} \\
\mathbf{A}\mathbf{U}_0 &= \mathbf{F} - \mathbf{B}^T\mathbf{P}_0 && (4.61b)\\
\mathbf{R}_0 &= \mathbf{B}\mathbf{U}_0 + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{F}_b - (\mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\,\mathbf{P}_0 && (4.61c)
\end{aligned}$$

$$\begin{aligned}
\mathbf{G}_0 &= (\frac{\rho}{\delta t}(\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{M}_b^{-1}\mathbf{B}_b^T)^{-1} + \theta\mu\mathbf{M}_p^{-1})\,\mathbf{R}_0 && (4.61d)\\
\mathbf{W}_0 &= \mathbf{G}_0 && (4.61e)
\end{aligned}$$

(1) Projection step: $\mathbf{U}_{n+1} = \mathbf{U}_0 - \frac{\delta t}{\rho}\mathbf{M}_V^{-1}\mathbf{B}^T\mathbf{G}_0$. End of FS.
Update the pressure: $\mathbf{P}_{n+1} = \mathbf{P}_0 + \mathbf{G}_0 + \theta\mu\mathbf{M}_p^{-1}\mathbf{R}_0$. End of MFS.

Main loop:

$$\begin{aligned}
\alpha_j &= (\mathbf{R}_j, \mathbf{G}_j)/((\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\,\mathbf{W}_j, \mathbf{W}_j) && (4.61f)\\
&\text{Computed as:} \\
\mathbf{A}\mathbf{Z}_j &= \mathbf{B}^T\mathbf{W}_j \\
\mathbf{V}_j &= \mathbf{B}\mathbf{Z}_j + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T\,\mathbf{W}_j \\
\alpha_j &= (\mathbf{R}_j, \mathbf{G}_j)/(\mathbf{V}_j, \mathbf{W}_j)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}_{j+1} &= \mathbf{P}_j + \alpha_j\mathbf{W}_j && (4.61g)\\
\mathbf{R}_{j+1} &= \mathbf{R}_j - \alpha_j(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{A}_b^{-1}\mathbf{B}_b^T)\,\mathbf{W}_j && (4.61h)\\
&\text{Computed as:} \\
\mathbf{R}_{j+1} &= \mathbf{R}_j - \alpha_j\mathbf{V}_j
\end{aligned}$$

$$\begin{aligned}
\mathbf{U}_{j+1} &= \mathbf{U}_j - \alpha_j\mathbf{Z}_j && (4.61i)\\
\mathbf{G}_{j+1} &= (\frac{\rho}{\delta t}(\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T + \mathbf{B}_b\mathbf{M}_b^{-1}\mathbf{B}_b^T)^{-1} + \theta\mu\mathbf{M}_p^{-1})\mathbf{R}_{j+1} && (4.61j)\\
\beta_j &= (\mathbf{R}_{j+1}, \mathbf{G}_{j+1})/(\mathbf{R}_j, \mathbf{G}_j) && (4.61k)\\
\mathbf{W}_{j+1} &= \mathbf{G}_{j+1} + \beta_j\mathbf{W}_j && (4.61l)
\end{aligned}$$

if (NITER>NITERMAX) GOTO (1)
Until convergence

Figure 4.2: The PCGU algorithm with condensation of the bubble.

in [40], the function spaces of the generalized Stokes problem and the mixed Laplacian are different, so that a stable Stokes element could be unstable for a mixed Laplacian problem. The invertibility and the condition number of system (4.43) relies on the weak inf-sup condition:

$$\delta \leq \min_{q \neq 1} \max_{v \neq 0} \frac{|(q, \nabla \cdot v)|}{\|v\| \|\nabla q\|} \tag{4.62}$$

and not on the "classical" inf-sup condition. This condition has been used in [65] to prove the classical inf-sup under adequate assumptions for the Taylor-Hood element. The matrix form of this condition reads:

$$\delta^2 = \min_{q \neq 1} \frac{(\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T q, q)}{(\mathbf{K}_p q, q)} \tag{4.63}$$

In [41], it is shown that, in particular, this condition is met for the bubble element and for the $P2/P0$ by generalizing the norm of the pressure for discontinuous approximations. The same arguments can be used to prove (4.62) for the $P1++/P1$. Equation (4.62) gives the invertibility of matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ and, by using the boundedness of operator B with these "wrong" norms, proves that matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ is spectrally equivalent to a Laplacian so that, in both cases of standard or compatible discretization, the condition number behaves like $O(h^{-2})$ [4]. It is SPD up to a hydrostatic pressure mode, so that it is classically solved by a preconditioned conjugate gradient (PCG).

If the matrix must be assembled, it could be created globally as a sparse matrix product, or advantage could be taken of the grid so that the matrix is constructed locally by moving from pressure nodes to velocity nodes to pressure nodes in case of continuous pressure, or from element to velocity node to element in case of discontinuous pressure. In [17], the preconditioner is solved using a direct solver, which appears to be prohibitive for large meshes. In [13], no comment is made on how to solve the preconditioner. In [52, 36], it seems to be assembled at once, and solved by the BiCGStab. Finally in [56], a **QR** decomposition of $\mathbf{B}\mathbf{M}_V^{-1/2}$ is advocated, as the compatible Laplacian pressure is solved twice. The **QR** decomposition allows to performe two triangular solves per equation. Except in [17] for 2D examples, the inverse of the velocity mass matrix has been previously lumped in all other papers. A discussion on the error commited by the mass lumping in this case can be found in [36].

In this work, a different approach is adopted. Three main requirements could be formulated:

- only the action of matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ must be needed to avoid to assembly and store this matrix, particularly in case of a Lagrangian description where all matrices must be assembled every non-linear iteration so that all ILU(k)-type preconditioners are discarded.

- the preconditioner must be robust and not fear a null pivot during factorization.

- evidently the preconditioner must be efficient in reducing the condition number of the system to be solved.

Mainly, solving (4.45) amounts to solve a least square problem. In this work, the preconditioner chosen is the one presented in [10] and used for least-square problems in [11]. The preconditioner is based on an incomplete $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$-orthogonalization by a Modified Gram-Schmidt (MGS) [55] procedure, and only requires the action of the matrix and not its factorization. Furthermore, due to the special pivot used during the $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$-orthogonalization, it can not become null. It is proved in [10] that the Cholesky factors can be recovered from the orthogonalization process. In order to conserve a sparse preconditioner, a dropping rule is applied on the $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$-orthogonalized vectors. A second threshold is applied on the value of the $\mathbf{L}\mathbf{L}^T$ factor in order to have a sparser triangular solve. As it will be seen, this second threshold is particularly efficient to conserve the relevant factors, keeping the preconditioner very sparse, resulting in a low storage and very efficient preconditioner. In the extreme case of no fill-in allowed, the SAINV preconditioner, the predecesor of the RIF preconditioner, is equivalent to a Jacobi preconditioner, and the RIF preconditioner to a SGS one. The version used in this work is the right-looking variant or rank-one update. It must be noticed that, in order for the preconditioner to be efficient, matrix-vector products must be done in *sparse-sparse* mode [55]. Storing matrix $\mathbf{B}$ gives automatically the possibility to perform these products with matrix $\mathbf{B}^T$. Due to the bubble condensation, the version of the RIF preconditioner implemented in this work stands between the RIF preconditioner for SPD matrix, and the one for least square problems. The whole algorithm is presented in Figure 4.3. Numerical results to illustrate the efficiency of the preconditioner are provided in the last section. The main reason of its success is due to the fact that the dropping is applied on the inverse of the incomplete factors instead of the factors themselves, so that $\|\mathbf{I} - \mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}\|$ is minimized instead of $\|\mathbf{A} - \mathbf{L}\mathbf{L}^T\|$. The triangular $\mathbf{L}\mathbf{L}^T$ solve is then much more stable (see e.g. [14] and references therein). All three requirements are then fully met. In order to have a least-square problem, equation (4.45) is replaced by:

$$\frac{\delta t}{\rho}\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T\Delta\mathbf{P} = \mathbf{B}\tilde{\mathbf{V}} \tag{4.64}$$

where:

$$\tilde{\mathbf{B}} = \mathbf{B}\mathbf{M}_L^{-1/2} \tag{4.65}$$

with $\mathbf{M}_L$ the lumped diagonal velocity mass matrix, as $\mathbf{M}_L > 0$ for the $P1+/P1$ and the $P1++/P1$ elements. Furthermore, a scaling is applied on the matrix to have an independent threshold value so that the system to be solved is:

$$\frac{\delta t}{\rho}\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T\Delta\tilde{\mathbf{P}} = \mathbf{D}^{-1/2}\mathbf{B}\tilde{\mathbf{V}} \tag{4.66}$$

where:

$$\tilde{\mathbf{B}} = \mathbf{D}^{-1/2}\mathbf{B}\mathbf{M}_L^{-1/2} \tag{4.67}$$

with $\mathbf{D}$ the diagonal of $\mathbf{B}\mathbf{M}_L^{-1}\mathbf{B}^T$ and $\tilde{\mathbf{P}} = \mathbf{D}^{1/2}\mathbf{P}$.

As a final remark, care must be taken when some connex components of the fluid only have Dirichlet boundary conditions for the velocity, as pressure is defined only up to a constant in each of these connex components. Matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ is singular and has as much null eigenvalues as connex components with only Dirichlet boundary conditions for the velocity. The algorithm then finds all the connex components, where connexity is different for continuous and discontinuous pressures, imposes one Z_i vector in the preconditioner, and the same pressure degree of freedom in matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$. After the resolution, the solution is corrected to maintain a null mean-value pressure. In [27], it is noted that singular systems are not a problem for iterative solvers if a compatibility condition is verified for the discrete system. However in practice, convergence was much faster by imposing some degrees of freedom for the standard Laplacian discretization, and even divergence was observed for matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$ without imposing anything for the 3D cavity example.

4.3.5 Mass lumping

In this section, a special comment on mass lumping is emphasized. The first step of the fractional step, or of the algebraic splitting, consists in solving a discrete elliptic or parabolic equation with a mass and a stiffness matrix. It is well-known that such discretization could not verify a discrete maximum principle under the appropriate assumptions [18]. Such instabilities have not been reported with the fractional step or the algebraic splitting in the literature, perhaps because of the fact that a high mesh Reynolds number could be avoided in the Eulerian formulation by a larger time step.

However, solving a classical dam break problem with water with a density of $10^3 kg.m^{-3}$, a characteristic mesh length of $10^{-1}m$, a time step of $10^{-3}s$ and a dynamic viscosity of $10^{-3}kg.m^{-1}.s^{-1}$ implies a mesh Reynolds number of 10^7 so that the velocity Laplacian contribution to matrix A in equation (4.29) is almost inexistent. The velocity system amounts then to solve a mass matrix scaled by the density and the inverse of the time step. As noted in [62, 18], the finite element mass matrix for linear elements does not possess a maximum principle property as it is not a M-matrix. At the opposite, the sum of the lumped mass matrix with the velocity Laplacian, scaled by the viscosity, possesses this property assuming that no angle in the triangulation is bigger than $\pi/2$. For such high values of the mesh Reynolds number, instabilities were observed and the replacement by the lumped mass matrix amounts to be an efficient and accurate solution, as well as allowing the iterative solver to converge faster. Here, the discretization under consideration is not exactly performed by linear elements as the bubble shape function is added, which can produce non negative extra diagonal terms. Indeed, it is equivalent to solve the same problem with linear elements on a grid where each element has been decomposed in three triangles in two dimensions and four tetrahedras in 3D, so that at the bubble nodes, the angles are greater than $\pi/2$. The mass matrix is still not a M-matrix and the stiffness matrix could have lost this property.

Goal: Build a Modified Gram-Schmidt (MGS) right-looking $\mathbf{A}$-orthogonalization where $\mathbf{A} = (\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T + L)$ of a basis of sparse vectors $\{\mathbf{Z}_i\}$. TOL and TOL_p are two given tolerances.
Initialization:

- For each pressure degree of freedom, build the linked list of the point neighbours of the neighbours $NEIGH[i]$ for continuous pressure, or the neighbour elements for discontinuous pressure.
- Build the linked list of $\mathbf{L}^T$ for sparse products
- Create a list $ROW[i]$ which points on the rows of the $\mathbf{Z}_i$.
- Initialize $\mathbf{Z}_i[i].push(1)$, $ROW[i].enqueue(i)$.

Main loop:

$DO\ i = 1, 2, .., n$

//Compute products in sparse-sparse mode

$\mathbf{w}_V = \mathbf{M}_V^{-1/2}\mathbf{B}^T\mathbf{Z}_i$

$\mathbf{w}_p = \mathbf{L}\mathbf{Z}_i$

$p_i = (\mathbf{w}_V, \mathbf{w}_V) + (\mathbf{w}_p, \mathbf{w}_p)$

//Store pivot in RIF preconditioner

$RIF[i][i] = p_i$

Create a stack $TEST[i]$ from $ROW[i]$ and $NEIGH[i]$ such that $(\mathbf{Z}_i, \mathbf{Z}_j)_{\mathbf{A}} \neq 0$

$DO\ k = 1, 2, ..\mathbf{TEST}.size$

$j = TEST[k]$

$p_j = (\mathbf{w}_V, \mathbf{M}_V^{-1/2}\mathbf{B}^T\mathbf{Z}_j) + (\mathbf{w}_p, \mathbf{Z}_j)$

$l_{ji} = p_i/p_j$

$if\ (l_{ji} < TOL_p)\ RIF[i][j] = l_{ji}$

//Apply a dropping rule to $\mathbf{Z}_j$:

$\mathbf{Z}_j = \mathbf{Z}_j - l_{ji}\mathbf{Z}_i$

$if(Z_j[l] < TOL)$ drop $Z_j[l]$

$else\ ROW[l].push(j)$

$ENDDO$

$ENDDO$

Figure 4.3: The RIF algorithm.

Alternatively, the introduction of the bubble in a reaction-diffusion problem has been interpreted in [32] as a stabilization applied to the reactive term, albeit not sufficient to eliminate the wiggles. From a different point of view, the stabilization comes from the fact that, as the mesh is finer, the reactive term is less strong than with the original mesh. Numerical experiments agree with the results of [32] so that lumping the mass matrix at high mesh Reynolds number is then still a necessity. At low mesh Reynolds number, numerical experiments seem to indicate that the maximum principle is still verified with the stiffness matrix without lumping the mass matrix.

This phenomenon was not observed for the $P1++/P1$ mass matrix, which contains negative extra diagonal terms. In the same way, replacing the "hat function" by the cubic bubble implies to use a 13 Gauss points formula to integrate exactly the 6^{th} degree mass matrix. Using reduced integration of 6 points reduces the oscillations, and using a three points formula eliminates the wiggles. Lumping the mass matrix is known to deteriorate the solution in some cases [60], but as mentioned in [37] the matching of the second order Crank-Nicolson time integration and the mass lumping constitutes an appropriate choice. Therefore, the classical drawback of the Uzawa scheme which needs an accurate inversion of matrix A is somehow limited to low mesh Reynolds number, as much as the velocity mass matrix is lumped, which accelerates further the convergence of the PCG for the velocity system. .

4.4 Mass conservation

This section is the *raison d'être* of the solver part, and constitutes the main improvement with respect to previous work [3]. Mass conservation is a difficult problem, especially during a large numerical simulation. The non respect of it happens equally in an Eulerian or a Lagrangian framework. However, as the domain changes in a Lagrangian formulation, it is more easily perceptible.

Discontinuous pressure elements have the attractive property to verify locally the conservation equation. It means that the weak divergence will be equal to zero to the machine precision on each element. However, as already noted in Chapter 2 , writing the equation of mass conservation in a traditional Eulerian way implies that a time derivation has already been performed, so that only an accurate time integration will imply that a small value of the weak divergence will produce a Jacobian value of 1 for each element in case of discontinuous pressure or on the patch of each node in case of continuous pressure. A reliable measure of the compliance of the mass conservation has been proposed in [45] by computing:

$$\max_K |\int_K \nabla \cdot \mathbf{v}_K dK| \tag{4.68}$$

where $\mathbf{V}_K$ is the velocity vector on element K. This measure is obtained at no additional cost for discontinuous pressure elements by using the PCGU, as convergence is measured by the norm of the residual $\mathbf{BV}$ which contains the precedent measure for the constant part of the pressure. For continuous pressure

elements, apart from the norm of the residual, we propose the measure:

$$\sum_i \int_{Patch} \psi_i \nabla \cdot \mathbf{v} d\Omega_i = \int_\Omega \nabla \cdot \mathbf{v} d\Omega \tag{4.69}$$

where the integration domain is the patch of node i, indeed all the elements which contain i, and $\mathbf{v}_j$ is the velocity vector of all the neighbouring nodes plus i of node i. It is not a norm, but represents the global mass conservation, which is the best that can be achieved by continuous pressure elements. Results with both the Euclidian norm of the residual and the last measure are given in the last section.

By using the classical fractional step, the introduction of the Laplacian pressure matrix produces the controversial Neuman boundary condition for the normal pressure derivative, but above all in case of free surface problems, implies to impose the pressure at the free surface as a Dirichlet condition. Taking the weak divergence of the end-of-step velocity, the velocity degrees of freedom belonging to the free surface can not verify the null divergence as the pressure nodes associated with the mass conservation equation have been removed from the Laplacian equation. Furthermore, during a dam break, elements or groups of elements can fly away and mix thereafter. Imposing a null pressure on all the nodes of one element produces a null pressure in all the element and the impossibility to satisfy the mass conservation, as will be seen below.

The main problem by using discontinous pressure elements appears to be that, as in 3D the number of elements is roughly 6 times the number of points so that the pressure dofs grow very quickly. A first idea is to define a constant pressure nodally on a cell around a vertex defined by dividing each element by its medians. The area of each subcell is one third of the area of the element, and for a constant pressure, the elemental contribution of the coupled velocity-pressure term reads:

$$\begin{aligned} \mathbf{B}_{bij} &= -\int_{K/3} \psi_i \, \nabla \cdot \mathbf{N}_j \, dV && \text{(4.70a)} \\ &= -\int_{K/3} \nabla \cdot \mathbf{N}_j \, dV && \text{(4.70b)} \\ &= -\nabla \cdot \mathbf{N}_j \times \frac{V_K}{d+1} && \text{(4.70c)} \end{aligned}$$

which is exactly the elemental contribution of the P^1/P^1 known to be unstable. It could maybe be however a good departure to construct a stabilization method.

4.5 Numerical examples

In this section, the theoretical predictions of the spatial and time discretizations described in the previous sections for the generalized Stokes problem are illustrated on various numerical examples. The free-surface problem will be dealt

with in the next chapter. Nevertheless, various tests in this section involve free surface problems, as they are tackled from the solver point of view. Finally, the cylinder benchmark is used to validate the Lagrangian formulation in a typical Eulerian context.

4.5.1 Preconditioner results

The efficiency of the algebraic splitting relies mainly on the efficiency of the RIF preconditioner. It must be noted that, for the PCGU, the preconditioner is assembled only once during the Uzawa iteration process, so that its cost is amortized over the iterations.

Its performances are illustrated in Table 4.1 on a 3D dam break example presented below with a $P1+/P1$ element. Each result consists in the storage requirement of the RIF normalized with the one of the $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$ matrix in the upper left corner, the computational time for the RIF construction in seconds in the lower left corner, the iteration number of the PCG applied to matrix $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$ at the upper right corner, and the computational time in seconds of the PCG in the lower right corner. These results are reported with various thresholds applied on the inverse factors and then on the factors themselves as a post filtration for a tolerance of $\|b\|/\|r^n\|$ of 10^{-13} for the PCG.

During the factorization, for each $\mathbf{Z}_i$ vector forming the $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$-orthogonal basis, scalar products must be done with others $\mathbf{Z}_j$ vectors. A computation for each vector against the others would give a complexity of the process of $O(n^2)$, making the whole process very inefficient. As the $\tilde{\mathbf{B}}^T\mathbf{Z}_i$ vectors have only few components, advantage can be taken from the sparsity in order to compute for each $\mathbf{Z}_i$ only the $\mathbf{Z}_j$'s that will give a non vanishing scalar product. This task is achieved by computing in a first pass the neighbours of the neighbours for each point in the continuous pressure case, and the neighbours of each element in the discontinuous pressure case .

As already mentioned, the $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$ matrix is spectrally equivalent to a Laplacian, which means that it is bad conditioned but not ill-conditioned. For example, the PCG converges without preconditioning. The results reported in Table 4.1 illustrate the nice behaviour of the RIF preconditioner. The preconditioner behaves exactly as predicted; the bigger the tolerance, pre or post, the better the iteration number. For the same example with the $P1++/P1$, the RIF preconditioner reaches a gain in time of 30% with optimal parameter values with respect to the diagonal preconditioner. Finally, the low storage requirements due to the post-filtration makes RIF a very efficient preconditioner.

4.5.2 Behaviour of the Uzawa operator

The main goal of this section is to compare the results obtained by the classical fractional step, the modified algebraic splitting and a monolithic solution obtained with the fully converged PCGU on the first step of a dam break problem and on the no-flow problem.

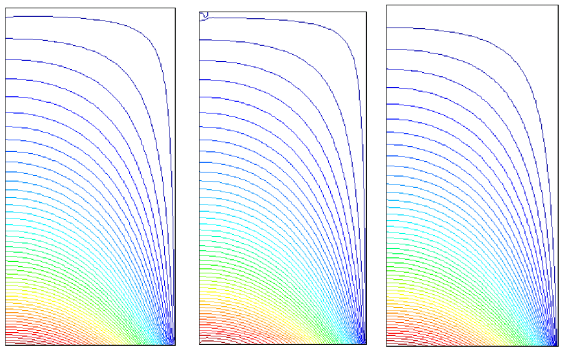
SAINV	Post Filtration Threshold									
drop tol.	0		1e-3		1e-2		1e-1		1	
0.001	3.2	22	0.91	23	0.41	29	0.04	73	0.02	101
	207	3	207	2	206	2	207	4	206	5
0.01	1.2	25	0.8	25	0.4	31	0.04	72	0.02	101
	27	3	27	2	27	3	26	5	27	7
0.1	0.51	39	0.5	39	0.36	42	0.04	73	0.02	102
	1	3	1	3	1	4	1	5	1	7

Table 4.1: RIF preconditioner with various pre and post dropping values. The columns represent various SAINV drop tolerances and lines various post filtration thresholds. For each case, the storage requirement is reported in the upper left corner, the computational time in seconds for the RIF construction in the lower left, the iteration number of the PCG in the upper right, and the computational time in seconds for the PCG only in the lower right.

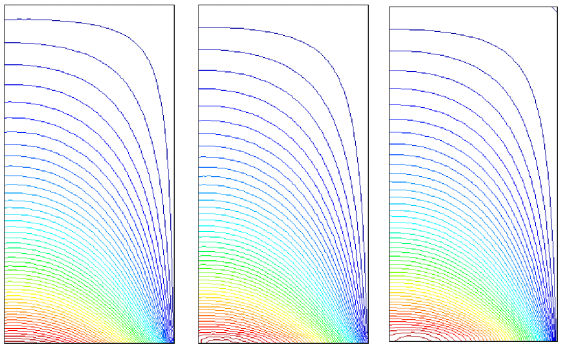
Onset of a 2D dam break

The example studied is the first step at $t = 0$ of a 2D dam break, which is a typical free surface problem. Initially, the fluid occupies a rectangle of dimensions 0.3×0.6, the tank is 1×1 and the mesh is unstructured of size 0.01. The spatial discretization is performed with a $P1 + /P1$ element. As explained in Section (4.3.1), the Laplacian preconditioner is expected to behave well with high mesh Reynolds number, the pressure mass matrix on low mesh Reynolds number, and the optimal preconditioner on the whole range of mesh Reynolds number. Therefore, the comparison is performed at three different mesh Reynolds number regimes, a high, middle and low one. Results are displayed on Figure 4.4. The first test is performed with $Re_m = 10^5$, the second with $Re_m = 1$ and the third with $Re_m = 0.01$. For each row, the pictures represent the results of the PCGU, the Modified Algebraic splitting (MAS) and the Fractional Step (FS), where it is reminded that a standard Laplacian is used for the pressure Poisson equation and no final update is performed for the pressure. As expected, at $Re_m = 10^5$ the pressure isolines are in very good agreement for the MAS and the FS respect to the monolithic scheme, which explains the success of the fractional step. At $Re_m = 1$, slight differences begin to appear at the bottom of the wave between the MAS and the FS, the MAS being closer to the PCGU than the FS. At $Re_m = 0.01$, the results obtained by the Pressure Laplacian are still driven by the gravity force, so that they are very far from the PCGU results, making the solution inaccurate. The MAS stays rather close to the PCGU, providing a cheap approximation of the monolithic scheme.

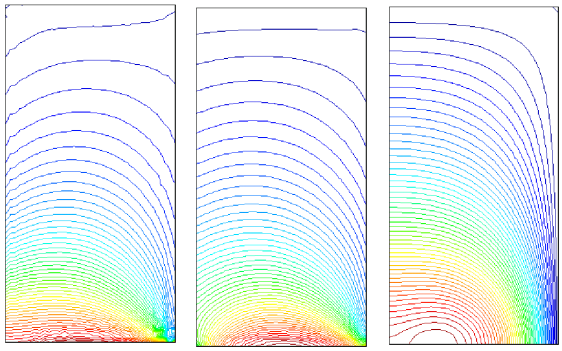
Remark 4.5.1 Here, emphasis is obviously put on free surface flows, due to the Lagrangian formulation. As mentioned in Section 4.3.2, the standard Laplacian preconditioning imposes the pressure as a Dirichlet boundary condition, and prohibits or delates the convergence of the PCGU. For this simple 2D example



(a) PCGU, MAS and FS at $Re_m = 10^5$



(b) PCGU, MAS and FS at $Re_m = 1$



(c) PCGU, MAS and FS at $Re_m = 10^{-2}$

Figure 4.4: Pressure isolines at the beginning of a 2D dam break result for the PCGU, the Modified Algebraic Splitting and the Fractional Step at mesh Reynolds number $\{10^5, 1, 10^{-2}\}$ with a $P1 + /P1$ element. The tank sides are at the left and at the bottom.

at $Re_m = 1$, the PCGU converges with a residual norm of 10^{-13} with 17 iterations with the compatible Laplacian and the pressure mass matrix, whereas 91 iterations were necessary for the standard Laplacian and the pressure mass matrix. A better solution is to impose the standard Laplacian to the residual of the PCGU. The PCGU then converges with 29 iterations. However, advantage can not be taken for the fractional step because it equally fixes the pressure degrees of freedom at the free surface. It is possible that this procedure reduces the mass loss for standard fractional step. For the 3D example presented in the next section, the results are still more significant; at high Re_m, the PCGU with the compatible Laplacian needs only three iterations to converge with the pressure mass matrix, but 482 iterations of standard Laplacian are necessary by imposing the preconditioned system to zero at the free surface, and 390 iterations are obtained by imposing the preconditioned system to the residual value. Without Laplacian preconditioner, 95 iterations are necessary to reach convergence. With only the standard Laplacian, the norm of the residual does not decrease after 16 iterations, and the PCGU does not converge. These results show the difficulty of the standard Laplacian to maintain a null divergence on the free surface so that all the schemes with a standard Laplacian will produce an important mass loss around the free surface.

The no-flow test

Another interesting numerical example is the no-flow test [35], in order to verify the pressure boundary layer due to the fractional step, and to compare the end-of-step velocity of the fractional step and the algebraic splitting to the monolithic solution, where the exact solution is the hydrostatic pressure and a uniformly null velocity. Here, the fluid is water so that the mesh Reynolds number is high for a reasonable time step, namely $Re_m = 10^5$ here. The geometry is such that one side is oblic, in order for the condition imposed on the normal derivative of the pressure to not coincide with the exact normal derivative on the wall side. Results are performed with the PCGU, the algebraic splitting of first and second order, and the classical first and second order fractional step. Second order means that the pressure of the first order has been reinjected in the momentum equation. The tank dimension is 2×2 and the oblic part corresponds to the diagonal of the 1×1 lower right square. The spatial discretization has been performed with a $P1 + /P1$ element, and the mesh is unstructured with size 0.05. Figure 4.5(a) shows the whole geometry whereas all other results are centred on the oblic side. In Figure 4.5, a consistent velocity mass matrix is used in (4.29) whereas a lumped mass matrix is used in Figure 4.6. The maximum velocities for the five examples with consistent and lumped mass matrix are depicted in Table 4.2. As the solution belongs to the discrete spaces, the monolithic solution displays the exact solution without pressure boundary layer as expected. However, it is rather surprising to see the pressure isolines of the first order algebraic splitting bending around the oblic boundary with the consistent velocity mass matrix, nonetheless with a curvature which does not reflect the pressure boundary layer. Examining the maximum velocity norm for

	PCGU	1^{st} split	2^{nd} split	1^{st} frac.	2^{nd} frac.
Consistent	1e-14	5e-3	2e-3	6e-3	1e-3
Lumped	1e-13	1e-7	1e-12	8e-3	3e-3

Table 4.2: Maximum velocity norm for the no-flow test with consistent and lumped mass matrix with $Re_m = 10^5$ for the PCGU, the algebraic splitting and the fractional step.

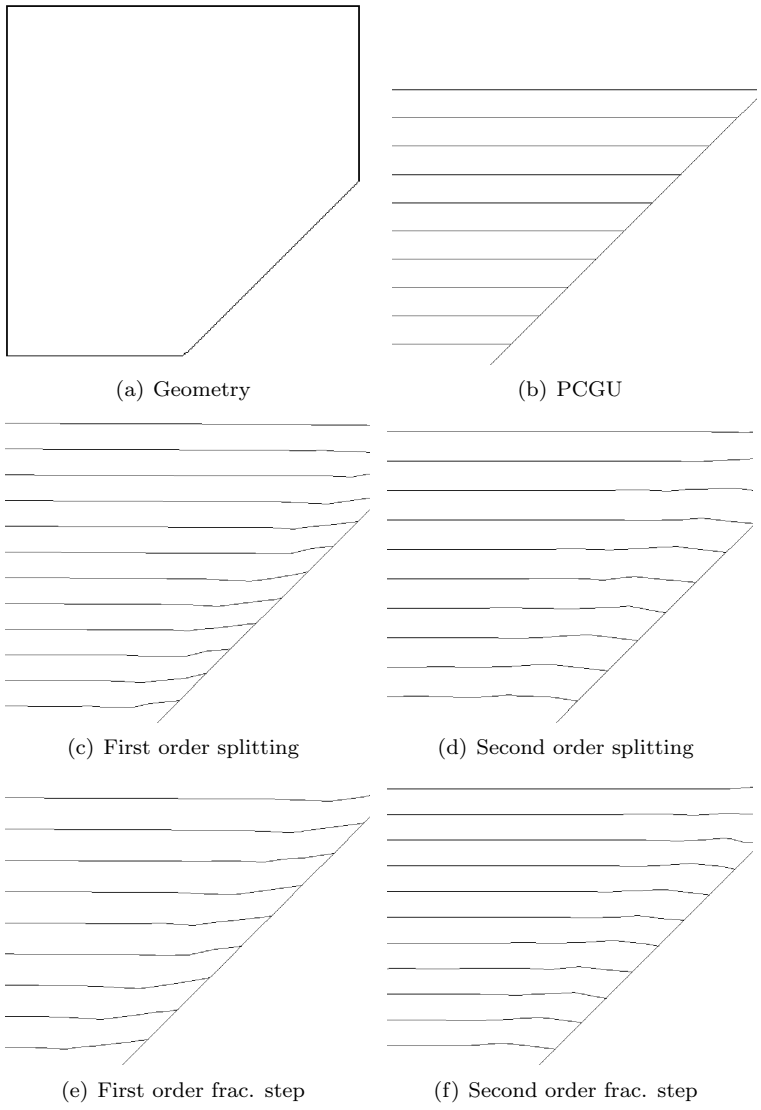
these examples, it is seen that instabilities are created by the consistent mass matrix which generate the bending of the pressure isolines for the algebraic splitting. In fact, the first step of the classical fractional step is a discretization of a parabolic equation, well-known for the instabilities as stated in Chapter 6. On the other hand, the lumped mass matrix gives a velocity guess without instabilities, as seen in Table 4.2, and no pressure boundary layer due to a possible $\frac{\partial p}{\partial n} = 0$ condition appear for the algebraic splitting whereas it is clearly present for the classical fractional step. Using a lower mesh Reynolds number stresses the bad preconditioning of the fractional step or the algebraic splitting to the Uzawa operator, as explained for the last example. Whereas the solution is still the same, the maximum velocity norm is around 0.1 and does not produce the correct velocity field. The pressure is far better with the algebraic splitting and the scaled pressure mass matrix, but iterations to the monolithic scheme seem to be unavoidable.

Remark 4.5.2 If the original velocity mass matrix is lumped, convergence of the PCGU preconditioned by the compatible Laplacian was observed in two iterations for a mesh Reynolds number of 10^7. It is in perfect agreement with the theoretical behaviour as, for such a high value of mesh Reynolds number, the Uzawa operator contains almost only the inertial part. Inverting matrix $BM_L^{-1}B^T$ is then almost equivalent to inverting the Uzawa operator. If one accepts to lump the velocity mass matrix, the solution of the monolithic scheme at high mesh Reynolds number is obtained very efficiently and constitutes an appealing scheme.

4.5.3 Mass conservation

The studied problem is the same as above, namely a dam break test, except that it is a 3D dam break and it is performed over 5 seconds of computational time. It involves breaking waves with water, strong contact and mixing, as senn in the next chapter, so that it is a good test to compare mass loss. The tank geometry is $10 \times 10 \times 10$ and initially the fluid volume is $3 \times 3 \times 6$ at the corner. The mesh is unstructured with size 0.3. The initial Reynolds mesh number is 10^5, which varies due to the variable time step chosen so that the Courant number be equal to 1. The results have been performed with three different elements:

- a $P1/P1$ backward Euler stabilized by a first order FS.

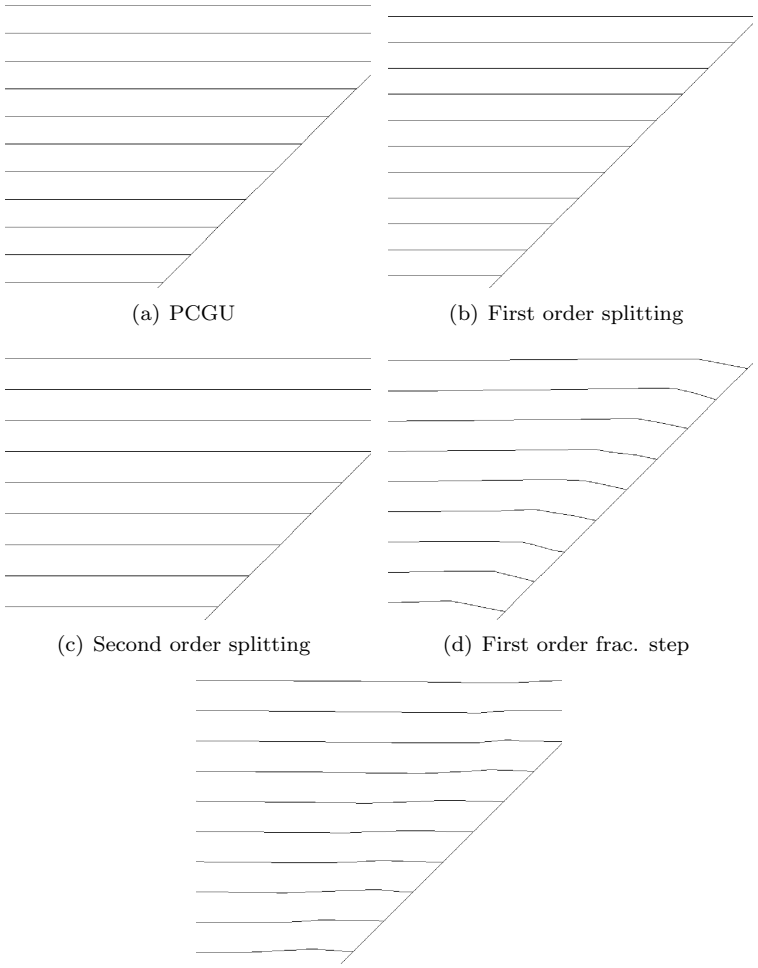


(a) Geometry (b) PCGU

(c) First order splitting (d) Second order splitting

(e) First order frac. step (f) Second order frac. step

Figure 4.5: The no-flow test with consistent velocity mass matrix for the PCGU, first and second order splitting, and first and second order fractional step schemes. The pictures represent the geometry and detail on the oblic side.



(a) PCGU (b) First order splitting

(c) Second order splitting (d) First order frac. step

(e) Second order frac. step

Figure 4.6: The no-flow test with lumped velocity mass matrix for the PCGU, first and second order splitting, and first and second order fractional step schemes. The pictures represent a detail on the oblic side.

Element	Velocity nodes	Pressure nodes
P1++/P1	163735	195940
P1+/P1	63345	14360
P1/P1	14360	14360

Table 4.3: Number of velocity and pressure nodes for the P1++/P1, P1+/P1 and P1/P1 elements for a 3D dam break.

- a $P1++/P1$ with discontinous linear pressure MAS.
- a $P1+/P1$ with MAS, FS and PCGU.

The dam break results have not been displayed as other 3D dam break results are presented in the next chapter. In Table 4.3, the number of velocity and pressure degrees of freedom are reported. It must be noted that the bubble is condensed for the velocity system for the second element, and for the whole system for the third element. Values of the averaged minimal, maximum Jacobian for each element and the averaged Jacobian for all the elements are reported in Figures 4.7, 4.8 and 4.9. The curves were smoothed using Bezier curves in sake of clarity. These values correspond to the values obtained for the convergence of the non linear process. As the nodes are convected with their displacement values, an isoparametric element is used in the non-linear iteration. The time step is variable, but of the same order for all the examples, around 10^{-3} s. For the $P1++/P1$ element, the numerical integration is performed with 24 Gauss points, so that the terms involving the inf-sup condition are integrated with sufficient accuracy. It was observed that the velocity system could not be solved with less Gauss points per elements. An isoparametric formulation is used for the $P1++/P1$ so that the Jacobian takes into account all the shape functions, as the geometry of the bubble must also be interpolated. A negative Jacobian does not necessarily imply that the element screw itself but that the classical condition on the edges :

$$\|a_{i,K} - \tilde{a}_{i,\tilde{K}}\| \leq C\, h_K \tag{4.71}$$

where $a_{i,K}$ is the vertex of the element and $\tilde{a}_{i,\tilde{K}}$ is the mid edge of the corresponding straightsided element, is not fulfilled. It is remarkable that no problem was met during the computation, mainly explained by the small time steps used.

The computation for the linear elements and the bubble involved around two days of real time for a whole result of 20 seconds of computational time, whereas the P1++/P1 was stoped after two weeks for less than 5 seconds of computational time. Due to the discontinuous pressures and the second order time integration, the Jacobian values are very near from 1 for the $P1++/P1$ element in the three cases, as the discontinuous pressure implies a local mass conservation. However, as noted before, the elemental Jacobian value is not even weakly constrained to be 1 but rather the divergence of the velocity is weakly constrained to be null on each element. Here, time integration appears to be very important and the second order time integration of the $P1++/P1$

	$\|BV\|$	$\sum_i B_{ij}V_j$
P1/P1 1^{st} order FS	10^6	10^6
P1++/P1 MAS	4×10^{-10}	2×10^{-9}
P1+/P1 PCGU	1×10^{-12}	5×10^{-12}
P1+/P1 2^{nd} order FS	10^6	10^6
P1+/P1 MAS	4×10^{-14}	2×10^{-12}

Table 4.4: Maximum value of $\|BV\|$ and the value $\sum_i B_{ij}V_j$ for various elements and solvers.

seems to provide accurate Jacobian values very near from 1 up to 10^{-5} for the averaged Jacobian value.

For the first order in time $P1/P1$ element, results are completely different. Negative values of the Jacobian are even observed for the $P1/P1$ element with the classical fractional step at longer time, which does not preclude the non linear convergence. This is not surprising, as the measured Jacobian values are always the one of the previous iteration, so that it has been checked that all the previous Jacobian where positive and that the non linear convergence has been reached, but not that the Jacobian of the last iteration were also all positive. The imposition of the pressure on the free surface coupled with a first order integration in time gives by far the worst overall result. It must be noted that over the 7207 iterations performed in time, 348 backtrace iterations when the Jacobian of an element becomes negative were necessary to go back at the beginning of the time step whereas no backtrace iterations were performed for the PCGU and the MAS whatever the element.

The $P1+/P1$ element gives good values of Jacobian also up to 10^{-5} for the averaged Jacobian value, even if mass conservation is only applied on the patch of each node with the MAS and the monolithic. It can be seen that the results are slightly better with the PCGU, which was expected. However, the second order fractional step gives the worst results, better than the linear element due to the higher order time integration, but very far from the MAS and the PCGU.

The maximum value of the norm of the divergence and the proposed value to measure the mass loss is reported in Table 4.4 for the previous cases. The very high values of the fractional step are due to the cases where a group of elements have only free surfaces nodes, in which case the pressure is null on the whole domain, and no control could be achieved on the norm of the divergence on the whole domain. In the average, the value is around 1. For the PCGU case, $\|BV\|$ is the residual of the PCGU so that it is verified up to the tolerance, which was fixed to 10^{-12}. The global value is slightly less as it is not imposed by the algorithm, but really corresponds to the mass loss on the whole domain. For the MAS case, the values of $\|BV\|$ are better as they correspond to the tolerance of the system involved with matrix $\mathbf{B}\mathbf{M}_V^{-1}\mathbf{B}^T$, which was fixed to 10^{-13}

If until then, all the results coincide with the theoretical considerations, Figure 4.10 gives some unexpected results. The volume loss has been reported as the difference of the volume of the straight elements before and after moving the

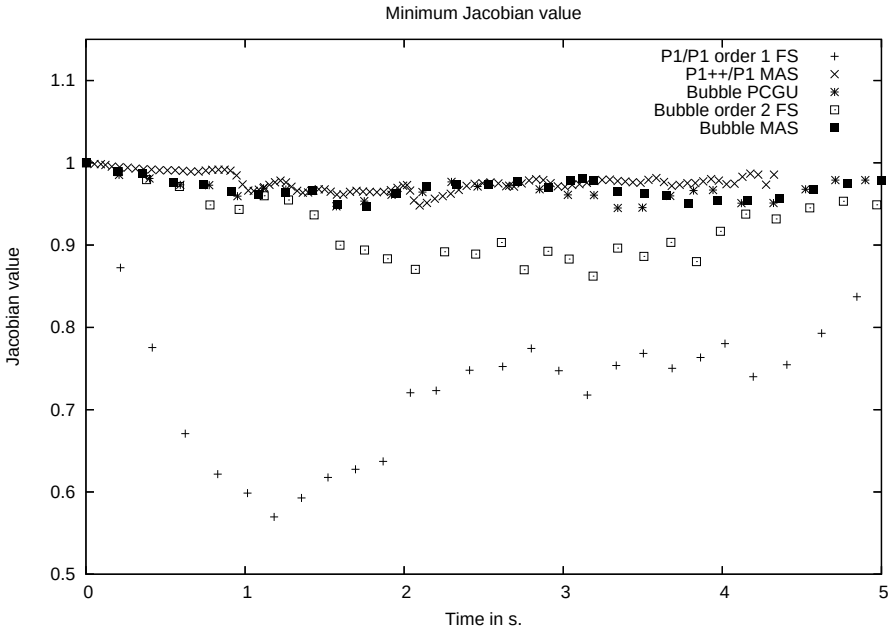


Figure 4.7: Minimal Jacobian value for various elements and solvers.

mesh divided by the initial volume. It must be noticed that, as the alpha shape method [25] is used to find the fluid boundaries, it could be also responsible for a mass loss. However, the volume loss reported on Figure 4.10 is *exclusively* measured as the difference between the volume of the straight elements between two time steps before remeshing. It is seen that, even if the $P1++/P1$ element provides a nice divergence norm, the computed volume loss is not as good as expected. It gives the best values but there is not an order of improvement, whereas it needs two orders more in time. It could be due to the fact that it does not take into account the curved surfaces of the element at the end-of step. The geometry becomes incresingly complicated if curved faces must be taken into account by the mesh generator at the next step so that only the volume loss of straight elements really matters. Furthermore, it is also seen that all the second order schemes do not give better volume conservation that the first order whereas the divergence and Jacobian measures were far better. As a θ-scheme is used, the measured Jacobian corresponds to a domain considered between n and $n+1$ whereas the volume is moved with the final computed displacement. This fact also explains the bad Jacobian values obtained by the first order scheme, as in this case, the Jacobian is computed with the whole displacement and not at an intermediate state where the displacement is less important. A second order backward differencing scheme could be a possible remedy but it deserves further investigation.

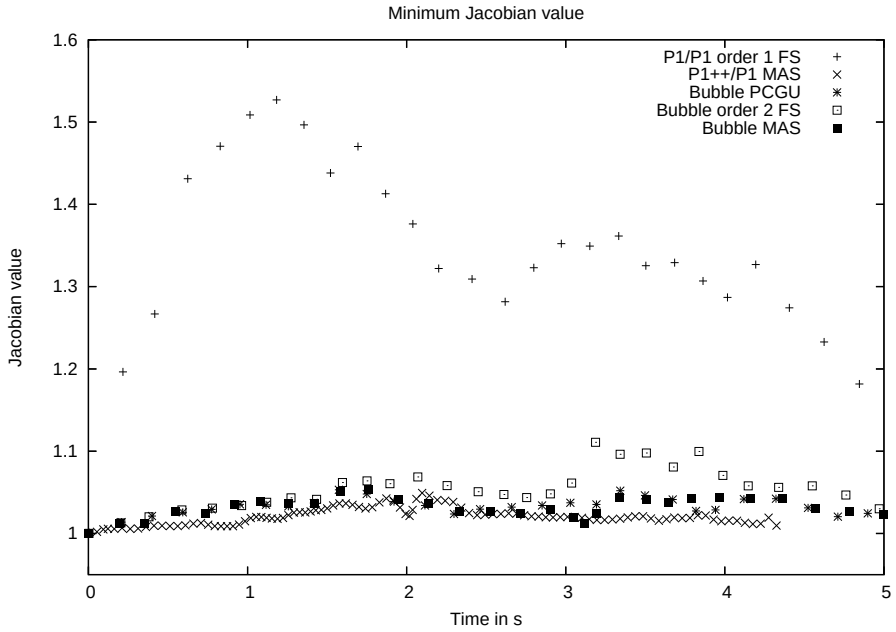


Figure 4.8: Maximum Jacobian value for various elements and solvers.

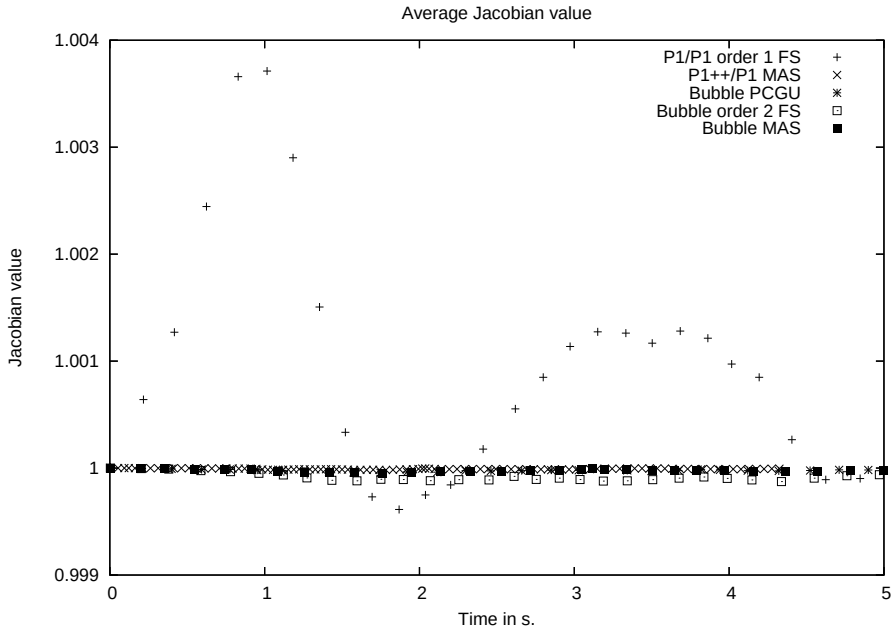


Figure 4.9: Average Jacobian value for various elements and solvers.

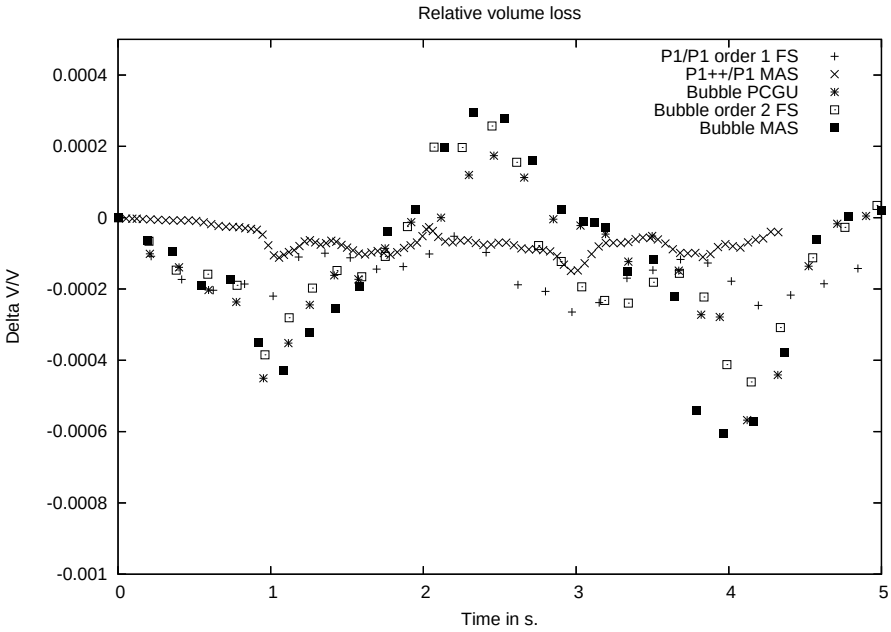


Figure 4.10: Relative volume loss for various elements and solvers.

Finally, a comparison between the monolithic solution with the PCGU, the Fractional Step and the Modified Algebraic Splitting is reported on Table 4.5 for various criterias for this 3D dam break example. For the FS and the MAS, the second order in time solvers are used so that the linear iteration is solved twice due to the reintroduction of the pressure in the momentum to avoid pressure interpolation errors. The assembly, meshing and solving times are reported in percent respect to the total real time reported in seconds. Then the time iteration number performed for the whole computation is reported, with the number of backtrace iterations due to possible negative Jacobians. Finally the average computational time step is reported in seconds. It is interesting to see that the computational cost of the compatible Laplacian is almost seven times the one of the standard Laplacian as the meshing time and the assembly time are almost the same, for a global time which is twice the one of the standard Laplacian. The condition number of both systems are of the same order. The standard Laplacian is solved by a PCG with the Eisenstat trick, which means that the preconditioner used is a SSOR preconditioner but the LU solve is saved so that there is only one matrix-vector product at each iteration. For the compatible Laplacian, each iteration involves two products of matrix $\tilde{\mathbf{B}}$ which is exactly three times larger than the standard Laplacian in 3D, one multiplication with a standard Laplacian to stabilize the $P1/P1$ operator, and the LU solve of the preconditioner, which means seven times the standard Laplacian plus the LU solve. The computational time measured experimentaly is thus in the

	PCGU	FS	MAS
Assembly	9	24	11
Solver	67	15	56
Meshing	18	49	25
Real time	217552	49816	101978
It. number	7086	7207	7154
Backtrace	0	348	0
Time step	0.00282	0.00277	0.00279

Table 4.5: Computational time for the PCGU, the FS and the MAS for various criterias for 14360 nodes.

expected order. The meshing time percentage varies greatly in the three cases as the solving time varies greatly. It must be noticed that the high Reynold mesh number is favorable for the monolithic to converge quickly with the compatible Laplacian compared to very viscous cases. Nevertheless, the solving time for the PCGU is 18 times the one of the FS for a global time multiplied by 4, whereas it is 3 times the one of the MAS for a global time multiplied by 2. As expected, the backtrace iterations are numerous for the fractional step due to the pressure imposition on the free surface and the impossibility to respect the mass conservation. It appears clearly why the fractional step is so popular.

4.5.4 Vortex shedding behind a cylinder

The cylinder example is a classical Eulerian benchmark [28]. The flow domain is a rectangle of dimensions $[0, 0] \times [16, 8]$, and a cylinder represented by a circle of diameter 1 is placed at point [4,4]. The Reynolds number is given by the diameter, the imposed velocity at the inflow and the kinematic viscosity. Two different results are presented, namely the standard value at $Re = 100$ for which the flow is laminar, and the case $Re = 10^6$, for which the flow is turbulent. No-slip conditions are applied on the cylinder whereas a velocity of [1,0] is imposed on the other parts except at the outflow for both cases. The results are obtained with the MAS. The mesh used is rather fine, containing around 6.10^4 nodes and $1,2.10^5$ elements. The mesh size is such that it is constant in a box of dimensions $[3, 3] \times [16, 8]$ centered at [8,4] with a value 0.02 and varies linearly in each coordinate until reaching a value of 0.3 on the sides, each straight line being delimited by one of the four arcs of parabole joining the point [8,4] with the four corners of both boxes. The second order Crank-Nicolson scheme is used in time.

Results of the isolines of the norm of the velocity and the pressure for various instants are reported on Figure 4.11 for $Re = 100$. The period of the oscillations of the von Karman vortex street for such $Re = 1$ number is around 6.0 which is in very good agreement with the reported ones. Some pressure oscillations seem to appear where the mesh is coarse. It seems to happen in nodes where the connectivity is lower or higher than 6, so that the meshing process should

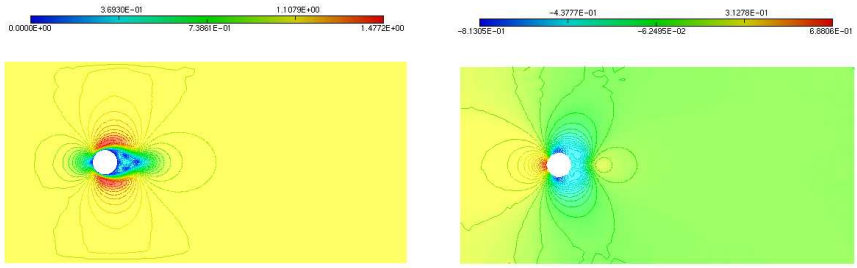
certainly take into account this parameter.

Results of the isolines of the norm of the velocity and the pressure for various instants are reported on Figure 4.12 for $Re = 10^6$. It must be noted that no turbulence model has been taken into account, as it should be firstly defined what means turbulence in a Lagrangian setting. However, the aim of this example is to illustrate that no numerical difficulties are found for such a high value of Reynolds number. The results are very different from Figure 4.11. The oscillations begin sooner and the period of oscillation is much faster, around $4.2s$.

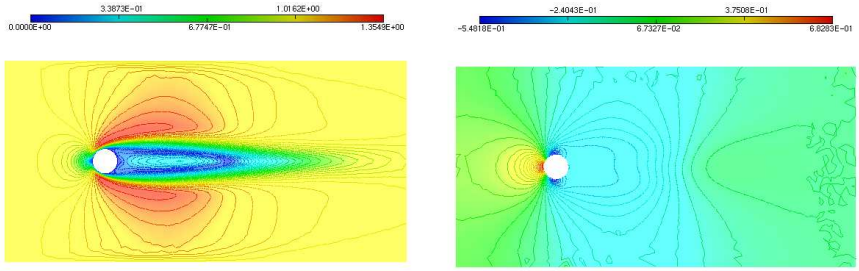
4.6 Conclusion

Various numerical aspects of a fully non linear incompressible Lagrangian formulation have been presented. First of all, the behaviour of the classical fractional step, the algebraic splitting and a variant with the introduction of the pressure mass matrix have been compared to conclude that the fractional step-like procedures are usefull numerical tools if they are used in the appropriate mesh Reynolds number, as are the equivalent preconditioners for the Uzawa operator. The first or second order approximation is obviously asymptotic and care must be taken to the underlying constants, as they also depend on the mesh discretization, the dynamic viscosity and the density. However, the monolithic solution remains still a general and unavoidable issue. A new scheme has been presented to achieve the monolithic solution in case of high mesh Reynolds number with very few iterations if one accepts to lump the velocity mass matrix.

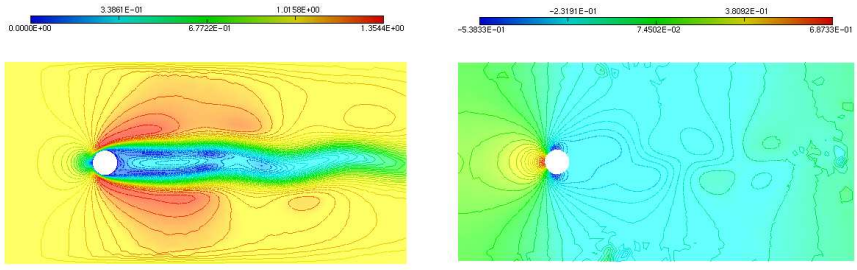
In order to achieve a relative mass conservation at each time step below 10^{-4} in a free surface context with Lagrangian description, the use of various spatial discretizations with continuous and discontinuous pressure has been implemented. However, it was observed that the introduction of discontinuous elements did not lead to much more accurate results, mainly because of the unavoidable use of isoparametric elements which generate volumes that can not be recovered by straight simplex elements at such a low tolerance, whereas the global volume of the latters measures what really matters. Furthermore, it implies an increase of two orders in time which, even if it provides better results for the same mesh, is easily outperformed by slightly increasing the degrees of freedom of C^0 pressure elements. In view of these results, the bubble element still happens to constitute a correct solution, providing a very good compromise between accuracy and efficiency. The key point in mass conservation with free surface problems, at least from a divergence point of view, is that algebraic splitting methods leave boundary pressure conditions as a Neumann boundary condition. This result applies equally to an Eulerian description, but do not appear so clearly, except by computing the volume lost by moving the mesh with the end-of-step velocity. The "ultimate" mass conservation scheme must impose in its formulation that the Jacobian be equal to one, independently of the computational cost and the description, the time integration being as important as the norm of the divergence as the precision increases.



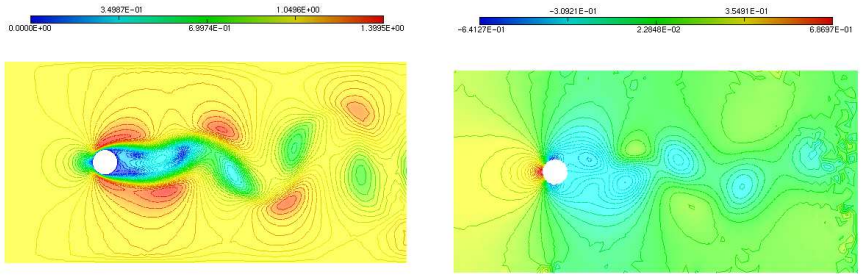
(a) t=2.1 s



(b) t=16.8 s

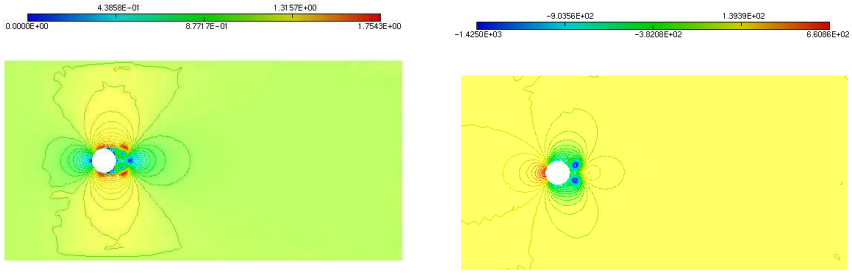


(c) t=31.2 s

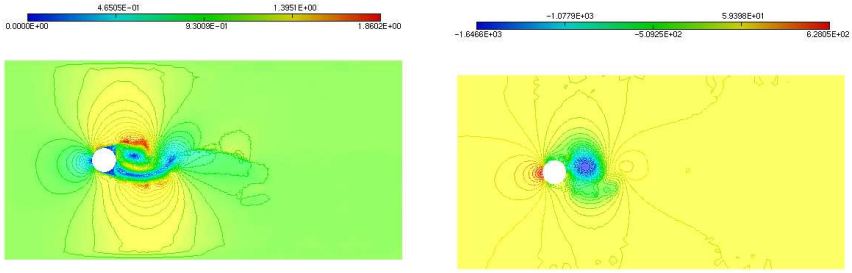


(d) t=48.6 s

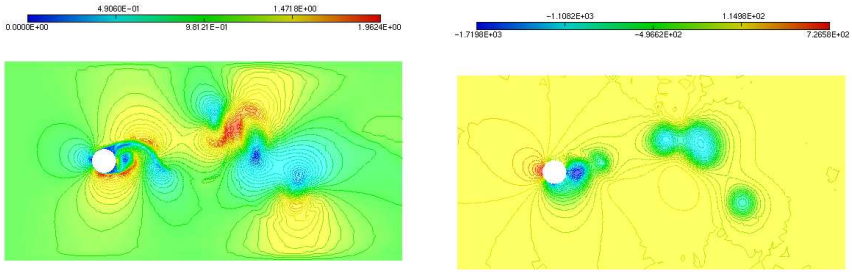
Figure 4.11: Vortex shedding behind a cylinder at $Re = 100$. The velocity norm is displayed at the left and the pressure at the right.



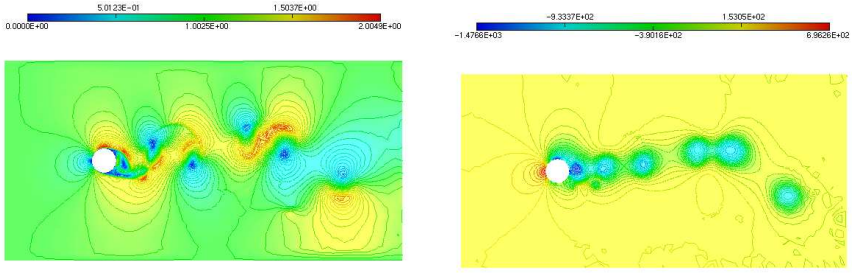
(a) t=2.1 s



(b) t=9.6 s



(c) t=16.5 s



(d) t=30.9 s

Figure 4.12: Vortex shedding behind a cylinder at $Re = 10^6$. The velocity norm is displayed at the left and the pressure at the right.

In view of the problems met in general, it is rather comfortable that no convective term appears in the discretization, which still more strenghtens the numerical difficulties as illustrated in [42]. Finally various numerical examples have illustrated the robustness of the proposed approach. The full understanding of the rather poor volume conservation of the second order θ-schemes tested in this work deserves further investigation.

Bibliography

[1] D.N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344, 1984.

[2] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming*. Stanford University Press, 1958.

[3] R. Aubry, S.R. Idelsohn, and E. Oñate. Particle Finite Element Method in fluid mechanics including thermal convection-diffusion. *Computer & Structures*, 83:1459–1475, 2005.

[4] O. Axelsson and V.A. Barker. *Finite Element Solution of Boundary Value Problems*. Classics in Applied Mathematics. SIAM, Philadelphia, 2001.

[5] C. Baiocchi, F. Brezzi, and L.P. Franca. Virtual bubbles and the Galerkin-Least-Squares methods. *Comp. Meth. Appl. Mech. Eng.*, 105:125–141, 1993.

[6] R.E. Bank and B.D. Welfert. A comparison between the mini element and the Petrov-Galerkin formulations for the generalized Stokes problem. *Comp. Meth. Appl. Mech. Eng.*, 83:61–68, 1990.

[7] O. Basset, H. Digonnet, H. Guillard, and T. Coupez. Multi-phase flow calculation with interface capture coupled solution. In M. Papadrakis, E. Oñate, and B. Schrefler, editors, *Computational Methods for Coupled Problems in Science and Engineering*. CIMNE, Barcelona, 2005.

[8] J.B. Bell and P.A. Colella. Second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 85:257–283, 1989.

[9] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[10] M. Benzi and M. Tuma. A robust incomplete factorization preconditioner for positive definite matrices. *Num. Lin. Alg. Appl.*, 10:385–400, 2003.

[11] M. Benzi and M. Tuma. A robust preconditioner with low memory requirements for large sparse least square problems. *Siam J. Sci. Comput.*, 25(2):499–512, 2003.

[12] F. Bertrand, M. Gadbois, and P.A. Tanguy. Tetrahedral elements for fluid flow problems. *Int. J. Num. Meth. Eng.*, 33:1251–1267, 1992.

[13] F. Bertrand and P.A. Tanguy. Krylov-based Uzawa algorithms for the solution of the Stokes equations using discontinuous-pressure tetrahedral finite elements. *J. Comput. Phys.*, 181(2):617–638, 2002.

[14] M. Bollhöfer. A robust ILU with pivoting based on monitoring the growth of the inverse factors. *Lin. Alg. Appl.*, 338:201–218, 2001.

[15] J.H. Bramble and J.E. Pasciak. Iterative techniques for time dependent Stokes problems. *Comp. Math. Applic.*, 33:13–30, 1997.

[16] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer, New York, 1991.

[17] J. Cahouet and J.P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. Num. Meth. Fluids*, 8:869–895, 1988.

[18] P.G. Ciarlet and P.A. Raviart. Maximum principle and uniform convergence for the finite element method. *Comp. Meth. Appl. Mech. Eng.*, 2:17–31, 1973.

[19] R. Codina. Pressure stability in fractional step finite element methods for incompressible flows. *J. Comput. Phys.*, 190:1579–1599, 2000.

[20] R. Codina and J. Blasco. A finite element formulation for the Stokes problem allowing equal velocity-pressure interpolation. *Comp. Meth. Appl. Mech. Eng.*, 143:373–391, 1997.

[21] R. Codina and O. Soto. Approximation of the incompressible Navier-Stokes equations using orthogonal-subscale stabilization and pressure segregation on anisotropic finite element meshes. *Comp. Meth. App. Mech. Eng.*, 193:1403–1419, 2004.

[22] M. Crouzeix and P.A. Raviart. Conforming and nonconforming finite element methods for solving the stationary Stokes equations. *RAIRO. Analyse Numérique*, pages 33–76, 1973.

[23] G. de Rham. *Variétés différentiables.* Hermann, Paris, 1960.

[24] E.J. Dean and R. Glowinski. On some finite element methods for the numerical simulation of incompressible viscous flow. In M.D. Gunzburger and R.A. Nicolaides, editors, *Incompressible computational fluid dynamics*, pages 17–65. Cambridge University Press, Cambridge, 1993.

[25] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, 1994.

[26] S.C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J. Sci. Stat. Comput.*, 2:1–4, 1981.

[27] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers.* Oxford University Press, 2005.

[28] M. Engelman and M. Jamnia. Transient flow past a circular cylinder: A benchmark solution. *Int. J. Num. Meth. Fluids*, 11:985–1000, 1990.

[29] M. Fortin. An analysis of the convergence of mixed finite element methods. *R.A.I.R.O. Anal. Numer.*, 11:341–354, 1977.

[30] M. Fortin and A. Fortin. Experiments with several elements for viscous incompressible flows. *Int. J. Num. Meth. Fluids*, 5:911–928, 1985.

[31] M. Fortin and A. Fortin. Newer and newer elements for incompressible flow. In R.H. Gallagher, G.F. Carey, J.T. Oden, and O.C. Zienkiewicz, editors, *Finite Elements in Fluids*, volume 6, pages 171–187. John Wiley and Sons Ltd., 1985.

[32] L.P. Franca and C. Farhat. Bubble functions prompt unusual stabilized finite element methods. *Comp. Meth. Appl. Mech. Eng.*, 123:299–308, 1995.

[33] F. Gauthier, F. Saleri, and A. Veneziani. A fast preconditioner for the incompressible Navier-Stokes equations. *Comput. Vis. Sci.*, 6:105–112, 2004.

[34] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equation*. Springer, 1986.

[35] P.M. Gresho, R.L. Lee, and R.L. Sani. Further studies on equal-order interpolation for Navier-Stokes. In *Proc. Fifth International Symp. on Finite Elements in Flow Problems*. Austin, Texas, 1984.

[36] M. Henriksen and J. Holmen. Algebraic splitting for Navier-Stokes equations. *J. Comput. Phys.*, 175:438–453, 2002.

[37] T.J.R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Dover Publishers,New York, 2000.

[38] R. Löhner. Projective prediction of pressure increments. *Comm. Num. Meth. Eng.*, 21:201–207, 2004.

[39] R. Löhner, C. Yang, E. Oñate, and S. Idelsohn. An unstructured grid-based, parallel free surface solver. *Appl. Numer. Math.*, 31:271–293, 1999.

[40] K.A. Mardal, X.C. Tai, and R. Winther. A robust finite element method for Darcy-Stokes flow. *SIAM J. Numerical Analysis*, 40:1605–1631, 2002.

[41] K.A. Mardal and R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numerische Mathematik*, 98(2):305–327, 2004.

[42] N. Nachtigal, S. Reddy, and L. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. Appl.*, 13:778–795, 1992.

[43] E. Oñate. A stabilized finite element method for incompressible viscous flow using a finite increment calculus formulation. *Comp. Meth. Appl. Mech. Eng.*, 182:355–370, 2000.

[44] C.C. Page and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *Siam J. Numer. Anal.*, 12(4):617–629, 1975.

[45] D. Pelletier, A. Fortin, and R. Camarero. Are FEM solutions of incompressible flows really incompressible ? (or how simple flows can cause headaches!). *Int. J. Num. Meth. Fluids*, 9:99–112, 1989.

[46] E. Pichelin and T. Coupez. A Taylor discontinuous Galerkin method for the thermal solution in 3d mold filling. *Comput. Meth. Appl. Mech. Engrg.*, 178:153–169, 1999.

[47] R. Pierre. Regularization procedures of mixed finite element approximations of the Stokes problem. *Num. Meth. Part. Diff. Eq.*, 5:241–258, 1989.

[48] R. Pierre. Optimal selection of the bubble function in the stabilization of the P1-P1 element for the Stokes problem. *Num. Meth. Part. Diff. Eq.*, 32(4):1210–1224, 1995.

[49] R. Pierre. Simple C^0-approximations for the computations of incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, 68:205–227, 1995.

[50] O. Pironneau. *Méthodes des Eléments finis pour les fluides*. Masson, Paris, 1988.

[51] A. Prohl. *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*. Teubner, Stuttgart, 1997.

[52] A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 188:505–526, 2000.

[53] P.A. Raviart and J.M. Thomas. *Introduction à l'Analyse Numérique des Equations aux dérivées Partielles*. Masson, 1988.

[54] J. Roberts and J. Thomas. Mixed and hybrid methods. In P. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume 2, pages 523–640. North-Holland, Amsterdam, 1991.

[55] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003. Second edition.

[56] F. Saleri and A. Veneziani. Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations. *SIAM J. Num. Anal.*, 43:174–194, 2005.

[57] O. Soto, J. Löhner, R. Cebral, and F. Camelli. A stabilized edge-based implicit incompressible flow formulation. *Comp. Meth. App. Mech. Eng.*, 193:2139–2154, 2004.

[58] A. Soulaimani, M. Fortin, G. Dhatt, Y. Ouellet, and F. Bertrand. Simple continuous pressure elements for two- and three-dimensional incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, 1:47–69, 1987.

[59] R. Stenberg. Analysis of mixed finite element methods for the stokes problem: A unified approach. *math. Comp.*, 42:9–23, 1984.

[60] G. Strang and G. Fix. *An Analysis of the Finite Element Method.* Wellesley-Cambridge Press, 1973.

[61] R. Témam. *Navier-Stokes Equations.* North-Holland, 1984.

[62] V. Thomée. *Galerkin finite element methods for parabolic problems.* Springer Verlag, Berlin/New York, 1984.

[63] L.J.P. Timmermans, P.D. Minev, and F.N. Van De Vosse. An approximate projection scheme for incompressible flow using spectral elements. *Int. J. Num. Meth. Fluids*, 22:673–688, 1996.

[64] S. Turek. *Efficient solvers for incompressible flow problems - an algorithmic and computational approach.* Springer, Berlin, Heidelberg, New York, 1999.

[65] R. Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO Anal. Numer.*, 18:175–182, 1984.

Chapter 5

Free surface flows

In this chapter, the free surface problem is considered. The main difficulty is due to the fact that the boundary of the body under study will move and deform, due to mechanical effects given by the computed solution, so that the location of the free surface is somehow an additional unknown of the problem. This problem is also often called the moving boundary problem. Apart from the convection of the body, which would be straightforward without interaction with solid walls in a Lagrangian formulation, the topology of the body will change with the boundary evolution due to intersection and separation between fluid and wall, and fluid and fluid. This is by far the most difficult aspect. In a solid mechanics context, this kind of problem was tackled from the contact point of view, mainly because, even if there is contact, the boundaries of the body remain with the same topology. However, fixed grid methods offer an appealing and much cheaper way to take contact into account at the price of the resolution up to the mesh size.

Various methods were tested in this work, mainly depending on the context that one assumes to be known to determine the boundary. A first approach would be to consider the problem as a classical meshing problem by following the boundaries of the body and generate classically a body conforming mesh [7]. A second approach consists in taking a cloud of points as sole knowledge, and to try to prune a shape from this cloud of points [12]. A last possibility is to enrich the knowledge of the context by not simply considering a given cloud of points, as the problem under study is a fluid problem with given initial boundaries and materials [33]. Other methods exist and have been commented in Chapter 1 so that the aim of this section is to highlight specific aspects that were used in this work. These above mentioned three aspects are detailed below. Finally, numerical results are presented, comparing the different approaches, and validated against a 2D dam break, and the cylinder with slot example.

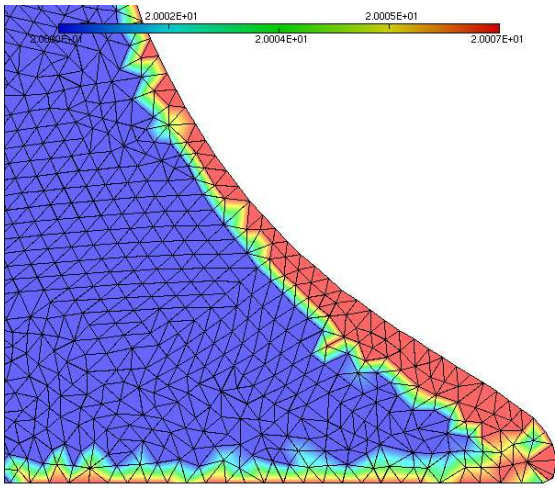


Figure 5.1: Intersection and folding between the fluid and wall boundary. The temperature field has been displayed with an external heat flux.

5.1 The classical approach

As a classical mesh generation scheme, a solution to explicitly know the boundaries of the body under study is to follow this boundary during its motion. Should the body intersect other bodies, a classical contact methodology would find the contact area, the penetrated volume, and would repulse the other body [3]. Apart from the high computational costs incured by a contact formulation, for fluid and for example, damage and rupture mechanics, the topology of the body changes during the analysis, and various connex components may appear or merge due to the large deformations undergone by the body. The mesh is then highly distorded, and a remeshing or rezoning procedure becomes mandatory. It is then very difficult to find and numerically build the boundaries of the body due to the possible intersections and separations. Some heuristics must then be introduced [7]. Figure 5.1 shows a detail of a 2D dam break performed by following the boundary edges during the motion. Furthermore, a smoothing procedure [39] was applied on the wall and on the fluid to regularize the boundary contour. Finally, the velocities of the points of the wall that limit the fluid and gas phases were extrapolated to further smooth the boundary discretization. However, it clearly appears that some of the points of the boundary become closer and closer, and much worse, the boundary edges cross the wall edges, so that the computation must stop. Temperatures are displayed for convenience with an external heat flux.

5.2 The cloud of points approach

For very different applications, objects are only known by points belonging on the surface or to the inner volume, but no information is explicitly known about the definition of the surface of these objects. This occurs in medical imagery, where the datas are scanned and millions of points are collected by the sampling machine, in computer vision, and graphics. The target is to reconstruct a surface of a volume passing through these points to calculate geometrical properties, such as areas, volumes, axes of inertia, surface normals, curvature and elementary shapes [4]. The difficulty is due to the fact that there exists an infinity of such surfaces so that the problem is ill-posed. Different criterias have then been proposed such as the maximization of the volume, the minimization of the surface, of the length of the edges, or angles, and so on. Various approaches are:

- The first idea consists in considering the surface as an elastical membrane. An initial large membrane encloses the data at the beginning. Then a deformation process is performed to minimize the elastic energy [33, 43]. A good initial guess must be provided and only local minimals can be reached.

- The combinatorial approach constructs a geometric data structure such as the Delaunay triangulation of the point set and try to extract from this structure a set of facets that approximates the surface. The first results in this directions are the alpha-shape concept of Edelsbrunner et al. [15] and the sculpting method of Boissonat [4]. Amenta *et al.* [1, 2] prove in 3D that the output of the algorithm is topologycally valid and convergent both pointwise and in surface normals, given assumptions of the density of the sampling point, mainly relying on an approximation of the medial axis, which is the generalization of the Voronoi diagram of points to surfaces.

- Another idea is to see the surface as a level surface of an implicit function and to try to find the zero of this function. Then a voxel or marching cube algorithm is performed, the intersection with the implicit surface computed, and a first rough surface triangulation created. [22]. Another strategy is proposed by Boissonnat *et al.* [5] by interpolating the implicit surface through the Natural Neighbours shape function [37, 38].

- A last idea is to directly generate the surface by a traditional surface mesher [28] by applying a strategy to generate locally the best triangle possible relying on a subset of the given cloud of points.

In this work, the alpha shape concept [15] has been extensively used to find the surface shape inside the Delaunay triangulation. The advantage is obvious as we already need the mesh for the computation. As the system is constantly moving, the location of the boundaries must also be redefined accordingly. The alpha shape method could be formulated as:

Definition 5.2.1 *Given a particle distribution depending on h(x), where h(x) is the minimum distance between two particles, all particles on an empty sphere with a radius r(x) larger than $\alpha h(x)$ are considered as boundary particles.*

In this criterion, α is a parameter close to, but greater than one. This definition is extracted from [23] and corresponds to a possible utilisation of the alpha shape in a finite element context. The exact definition of the alpha shape, formulated by their authors is:

Definition 5.2.2 *Given a real number α, with $0 < \alpha < \infty$, let an $\alpha - ball$ be an open ball with radius α, $\mathcal{S}$ a finite set of points in $\mathcal{R}^3$ and T any subset of $\mathcal{S}$. A simplex is said to be α-exposed if there is an empty α-ball b with $T = \partial b \cap \mathcal{S}$, where ∂b is the plane or sphere bounding b. A fixed α thus defines sets $F_{k,\alpha}$ of α-exposed k-simplices for $0 \geq k \geq 2$. The α-shape of $\mathcal{S}$ is the polytope whose boundary consists of triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$, and the vertices in $F_{0,\alpha}$.*

Roughly speaking, the alpha shape compares the value of the radii of the simplices of the triangulation and the α parameter provides a cut-off value to consider the elements with radius bigger than α as 'outside'. The first definition of the alpha-shape is equivalent to the second one thanks to the Delaunay triangulation property of being the dual of the Voronoi Diagram. If $\alpha = \infty$ the convex hull of the cloud of points is obtained, if $\alpha = 0$, the shape consists in only the points of $\mathcal{S}$. A straightforward extension to a cloud of points distributed respect to a size distribution map has been done in this work, as seems also to have been done in [14]. The extension to the anisotropic case is not so obvious as the geodesics are not straight lines anymore, and the parameter still more sensible. However, Teichmann and al. [40] propose some extensions to the method.

At the implementation level, the alpha shape will determine a surface for a cloud of points representing a volume, without the assumption that this cloud of points are only points of the initial surface, which is an important difference with the other methods. The points of the last time step, or the initial given cloud of points at time zero, are filtered and inserted in the triangulation. The alpha shape is then computed, and all the inner edges of fluid are examined to compute their length and generate new waves of points. Figure 5.2 describes the process for the simple shape of the letter 'A'. The points of the letter are inserted in a bounding box and the Delaunay triangulation is computed. Given a parameter, the resulting shape is found, and the maximal circles of radius alpha empty from other points are also shown.

The drawbacks of the alpha shape are various. As the surface is extracted from the faces of the Delaunay triangulation, the surface appears to be very rough. Some practical criterias on the surface roughness are given in [17, 18]. Furthermore, as already mentioned in Chapter 3, the mesh must be as close as a Delaunay mesh in order for the alpha shape to produce adequate results. As seen in the same chapter, this requirement is not the main aim at producing

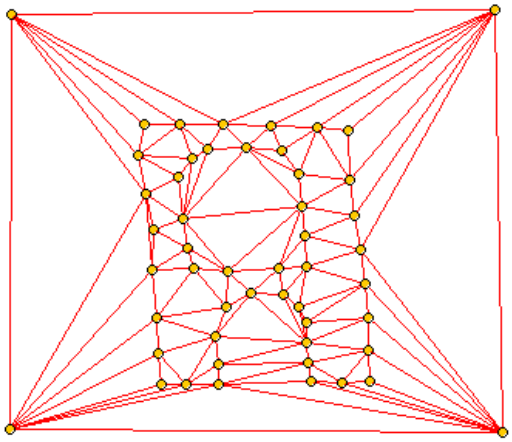
meshes for finite elements computations, and both requirements could sometimes be antinomics. Also, the necessary introduction of Steiner points in a 3D context perturbs the alpha-shape context and shows the limitation of the method. In the same way, the boundaries can not be generally remeshed, as it could produce new elements that will be considered as fluid, generating endless fluid mass. It must be also be noticed that the alpha shape needs a complete reconstruction of the mesh at each time step in a 3D context. However, only some elements are really distorded due to the large displacements, so that a local mesh optimization method would be much more efficient and easy to parallelize than the whole Delaunay mesh regeneration. Finally, some holes may appear in the fluid volume, greatly disturbing the pressure distribution as illustrated in the numerical section.

5.3 The Level Set approach

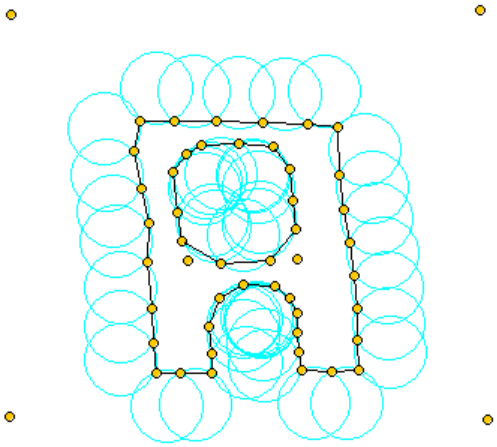
In the Level Set approach, the surface is considered implicitely as the zero isocontour of some function in a usually fixed volumic domain [33]. The evolution of the surface is then computed and updated, as it is not known implicitely.

One of the main advantages of the Level Set formulation is due to the fact that pinching and merging are resolved naturally up to the grid discretization level. This attractive property was already noted in [36] and [9], where a first algorithm was devised which computed the exact intersection, giving rise to a very rough surface discretization, whereas the Level Set approach seemed to be much more taylored to smooth the contour. In practice, the studied domain is fully meshed and the implicit function convected with the computed velocity. The air domain may or may not be taken into account for efficiency reasons, although [8] shows that at least the gas effects must be considered. If the velocity of the air is not considered, some extrapolation of the velocity and the pressure in the air domain around the free surface must be done to correctly advect the implicit function [30]. Usually, slip conditions are applied to the velocity system so that the Level Set could evolve on the walls, but it is not necessary if the velocity and the Level Set are interpolated by different shape functions [34].

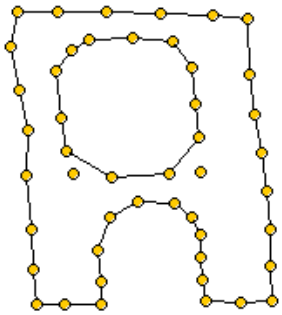
A partial drawback appears because, as the free surface usually do not coincide with faces of the mesh, the discontinuity in the pressure gradient between air and water can not be properly described. A remedy proposed in [11] consists in modifying locally the pressure shape function to recover this discontinuity. Another possibility would be to adaptively remesh around the free surface. The typical example where instability can occur with a two phase flow arises in the simplest hydrostatic pressure column. Another way to see it could be that, if spatial integration is performed with one Gauss point, the 'effective' free surface will present sharp irregularities as some elements will be selected as fluid, and some other not. This sharp front produces instability as the pressure isolines can not be horizontal. As noted in [11], increasing the Gauss point number does not improve the solution, as the pressure gradient still stays continuous in the element.



(a) The Delaunay triangulation.



(b) The maximal empty spheres.



(c) The resulting shape.

Figure 5.2: The α-shape concept, from *http://cgm.cs.mcgill.ca/ godfried/teaching/projects97/belair/alpha.html*.

Furthermore, a constant reinitialization procedure must be performed if a smooth distance Level Set function is chosen. If another type of function, such as discontinuous function is used, care must be taken on the monotonicity of the convective operator [30].

5.4 The projection method

As already said, the main difficulty with free-surface problems is created by the fact that the domain evolves in time, so that, as remeshing is mandatory in a Lagrangian formulation, the domain must be redefined to allow the introduction and suppression of nodes, which will be moved by physical reasons and not by mesh optimization purposes. This problem is then at first view a simple mesh spatial interpolation problem as the mesh of the old time step will be moved, so that its topology, boundary and all geometrical properties are known. However, there are at least three difficulties appearing due to the physical problem at hand:

- The resolution of the no-slip velocity at the walls, which if no special treatment is done, will provoke the endless stretching of an element which has a face on the wall, and the endless increse of negative pressure to maintain the mass conservation, in this element or on the whole domain.
- The resolution of the free surface merging, due to breaking waves, mould filling and so on.
- The resolution of separation of the fluid if the velocity in an element is such that two nodes of the element move away so that the fluid pinch apart.

If the first problem has a strong physical nature, the two others are much more geometrical as it concerns the topology of the free surface. Two important physical phenomenas in free surface problems are surface tension and cavitation. The first plays an important role in the formation of liquid drops [13], and the second in the creation of internal free surfaces. So that both phenomena contribute to the topological changes of the free surface, and the limit between the wetted and the still dry wall part. However, they are not taken into account in this work and a more 'macroscopical' approach must be used. There is a 'kinematic paradox' [25] at the moving contact line as the usual boundary condition for viscous problem is the no-slip condition, which obviously contrasts with the fact that the contact line will evolve in space. The usual boundary condition used in an Eulerian implementation is the slip condition [10, 19, 20, 30], which partially resolves the problem at the expense of loosing any wall boundary layer. Another interesting alternative is to consider a Signorini contact condition at the wall coupled with a boundary condition [31].

The fact that implicit functions resolve these pinching and merging phenomenas is mainly due to the numerical diffusion introduced by the grid resolution of the presence function. Furthermore, if a Lagrangian method is used, it is

pleasant that the boundaries of the domain coincide with the boundary simpleces, as does the alpha shape method. This is typically reminiscent of a P^0 interpolation of the presence function. As some diffusion is needed, the idea is to simply perform a P^0 projection from one time level to the other to determine which element are of fluid type, and which are empty. As noted in Chapter 3, the interpolation process is very fast and robust so that the global algorithm is negligeably penalized compared to the alpha-shape method. In order to resolve the kinematic problem, the last line of fluid elements is projected on the wall, the discrete surface remeshed [27, 41, 16] and the projection performed, so that the no-slip condition is conserved from the alpha-shape method, and the contact line evolves, corresponding to a zero order extrapolation [21]. Compared to the alpha-shape, the main advantages are that no spurious holes appear in the mesh so that the pressure distribution does not have to satisfy rough boundary conditions and the numerical results are much smoother. Furthermore, the mesh procedure becomes independent of the method as it does not need to be strictly Delaunay. Steiner points are neither a problem and they will be interpolated like other points if needed. Finally, the boundary discretization does not perturb the method so that it is then possible to remesh it if necessary. The proposed method, whereas designed for other reasons, could be related to the 'transport+projection' schemes usually used in Eulerian formulation [6, 35, 24], and reminiscent of the Godunov scheme [21]. They provide unconditionaly stable schemes due to the absence of the convective term but they also somehow transfer the difficulty to the geometry by finding forwards or backwards on the characteristic lines the old values for each considered node. However, the difference relies once again on the fact that the Euler formulation considers the deformed configuration instead of the initial configuration as the reference configuration.

5.5 Numerical examples

The main aim of this section is to validate the aforementioned numerical method on classical benchmarks, and to compare it with the other methods used. To illustrate convection, the example of the cylinder with a slot is used. Then the 2D dam break is performed and compared with analytical and numerical results. Finally, a 3D dam break has been runned to show the robustness of the method.

5.5.1 Rotation of a cylinder with a slot

This example is a classical benchmark for the convection equation, see [42, 29, 26, 24]. A cylinder of radius 0.3 has a slot of width 0.1, which leaves a bridge of 0.1 at the upper part. A rigid body rotation in a nonuniform velocity field is given by $v(x, y) = (-y, x)$. If this example would be considered without the P^0 interpolation by conserving the boundaries, the exact solution would be trivially obtained up to time integration errors with the Lagrangian method. It is however interesting to see the effects of the projection with this example.

Computation was performed up to a quarter revolution in [29], and up to a whole revolution in [26, 24]. It must be noted that the Flux Corrected Transport (FCT) paradigm [42, 29, 26] and the Discontinuous Galerkin method (DG) [24] are among the best Eulerian methods available for convection dominant cases with discontinuities. Figure 5.3 shows the results up to 15 revolutions. 33333 iterations were performed with a time step of 10^{-3}. Even if the shape of the cylindre begins to change due to the time integration, the contours of the discontinuity are very well conserved. The Lagrangian formulation shows its great accuracy to move here a given discontinous function, and more generally a free surface contour. The size mesh is 0.02 as in [24], but only the inner cylinder has been meshed. However, only the boundary points matter, and the inner mesh could have been made only with boundary points. It must be noted that an important and implicit issue here relates the mesh size and the cylinder boundaries. If the boundaries are Delaunay conforming, they naturally appear in the new mesh where the projection takes place, and the shape of the cylinder is then perfectly reproduced up to the time integration. It is also interesting to see how the spikes of the slot deform, particularly after 5 rotations, giving rise to almost degenerated triangles. and how the projection improves the shape by rejecting the too much deformed triangles

5.5.2 2D dam break

This is a classical benchmark for free surface flows, and has been used by various authors for experimental [32] and numerical results, see [30, 19] and references therein. The first result is given by Figure 5.4, which compares the evolution of the horizontal location of the free surface $\delta = x/a$ along the bottom wall respect to a normalized time $\tau = t\sqrt{2g/a}$ for the experimental result and the projection method. Even if the global shape was similar between the alpha shape method and the projection method, the alpha shape method has not been displayed due to the irregularity of the contour at the rightmost point. Here, a is the base length of the wave, the height is $2a$, and the length of the reservoir is $4a$. It must be noted that the boundary condition considered in this work is the no-slip boundary condition as opposed to the previously cited references. The results were obtained with the monolithic iterative solver and a mesh size of 0.02 as the one used in [19]. The overall agreement is good, even if some slight discrepancies appear at the beginning of the experiment, as appear in the numerical results of previously cited works. In the paper of Martin and Moyce, they report some difficulties at the begining of the experiment which could explain the different results. As the viscosity is rather low, the influence of the boundary layer is very limited, which explains that results are almost the same with and without the no-slip condition.

Figures 5.5 and 5.6 compare a 2D dam break with the alpha shape method and with the projection method for the pressure field solution with a monolithic iterative solver. The creation of holes in the liquid domain due to the alpha shape method forces the pressure to weakly verify the null stresses on the boundary of the holes and destabilizes the pressure distribution, as seen in

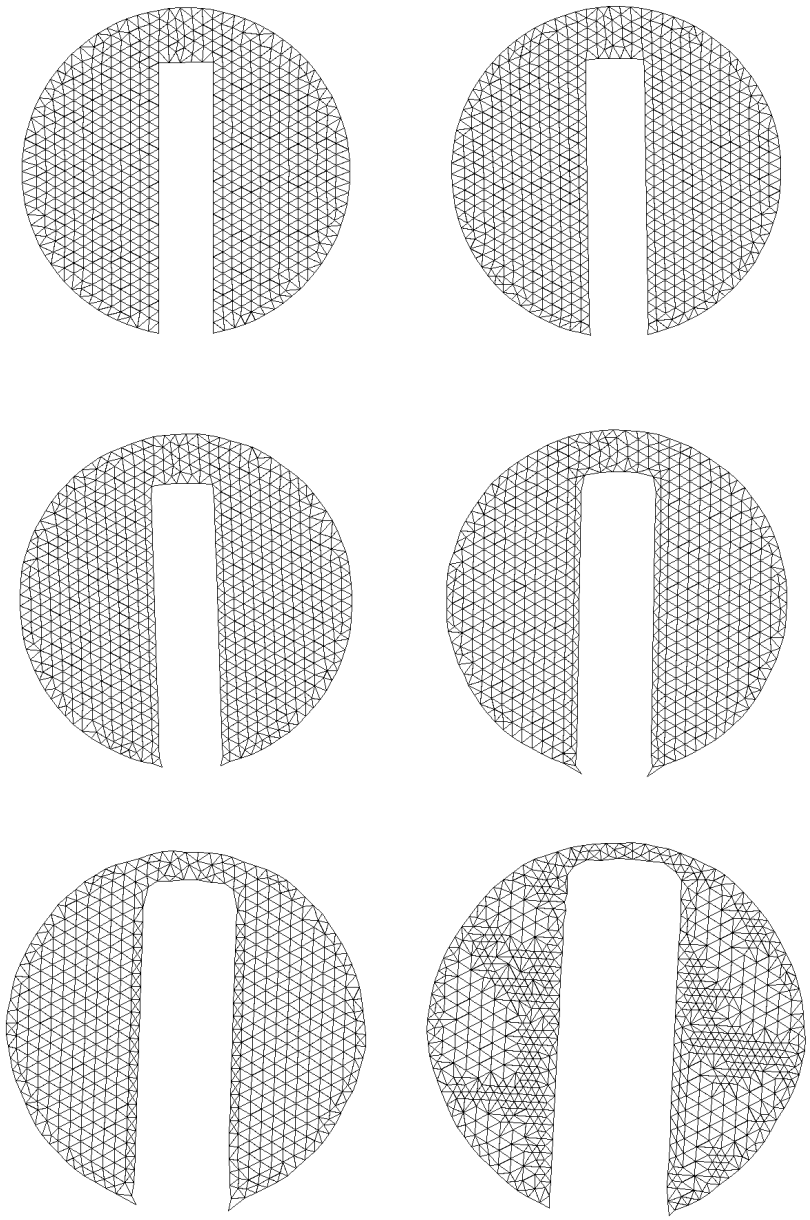


Figure 5.3: Cylinder with a slot for {0,1,2,5,10,15} revolutions.

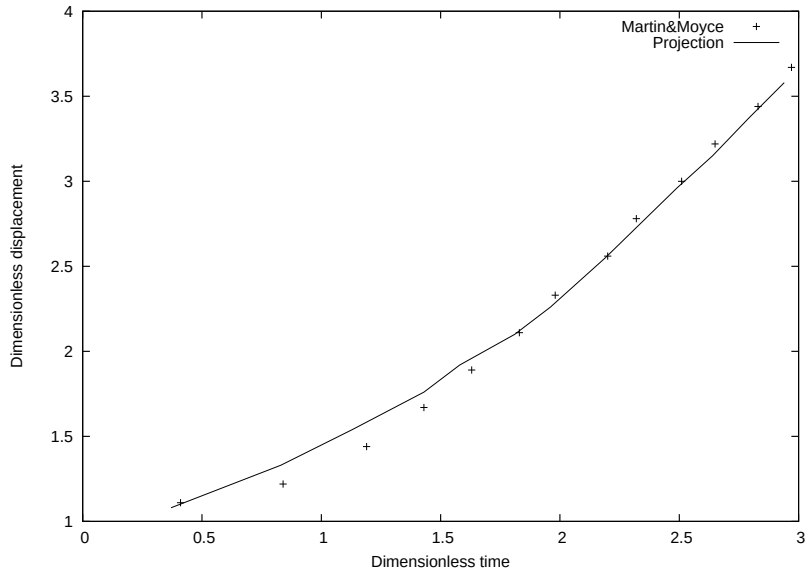


Figure 5.4: 2D dam break: comparison of the horizontal displacement with experiment.

the second subfigure. However, it must be noted that the position of the free surface is somehow similar in both cases, at least for the not too long time, which illustrates the well known fact that even if pressure results are not so good, velocity results provide an accurate solution. The surface for the alpha shape method has not been smoothed, whereas it has been slighlty smoothed with the projection method by one smoothing iteration per time step.

5.5.3 3D dam break

This example is the classical 3D dam break with a solid cube at the middle of the domain, and is illustrated in Figure 5.7 at various instants. The mesh is composed by 2×10^5 nodes and around 10^6 elements with a size map concentrated around the solid cube. The main aim of this example, as there is no analytic solution provided, is to illustrate the robustness of the method, as 400 seconds of computed time were performed with a time step of 10^{-3}, so that around 10^5 complete meshings and interpolations were performed for the whole computation. This example was runned with the alpha shape method, as can be seen with the rather rough free surface shape.

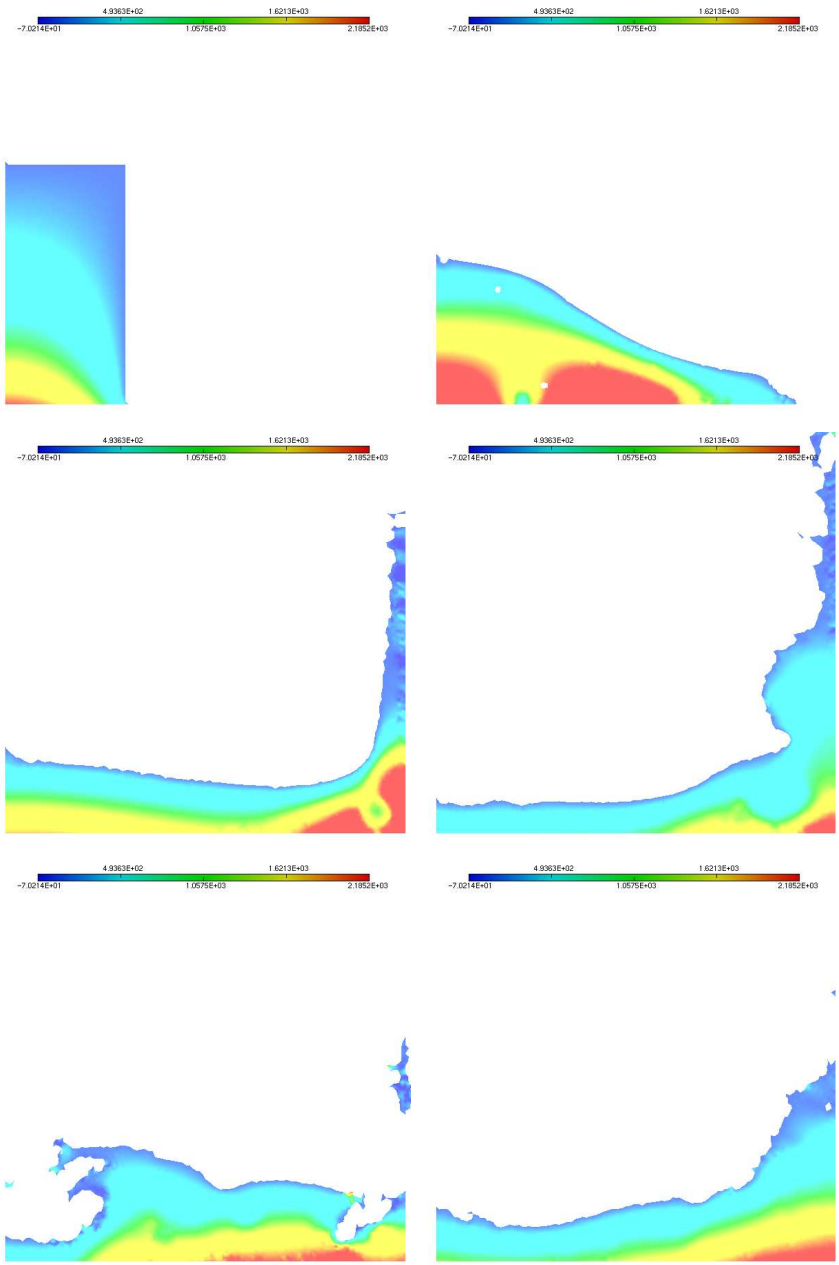


Figure 5.5: Pressure distribution with alpha shape at t={0.001,0.22,0.52,0.9,1.25,2.3}.

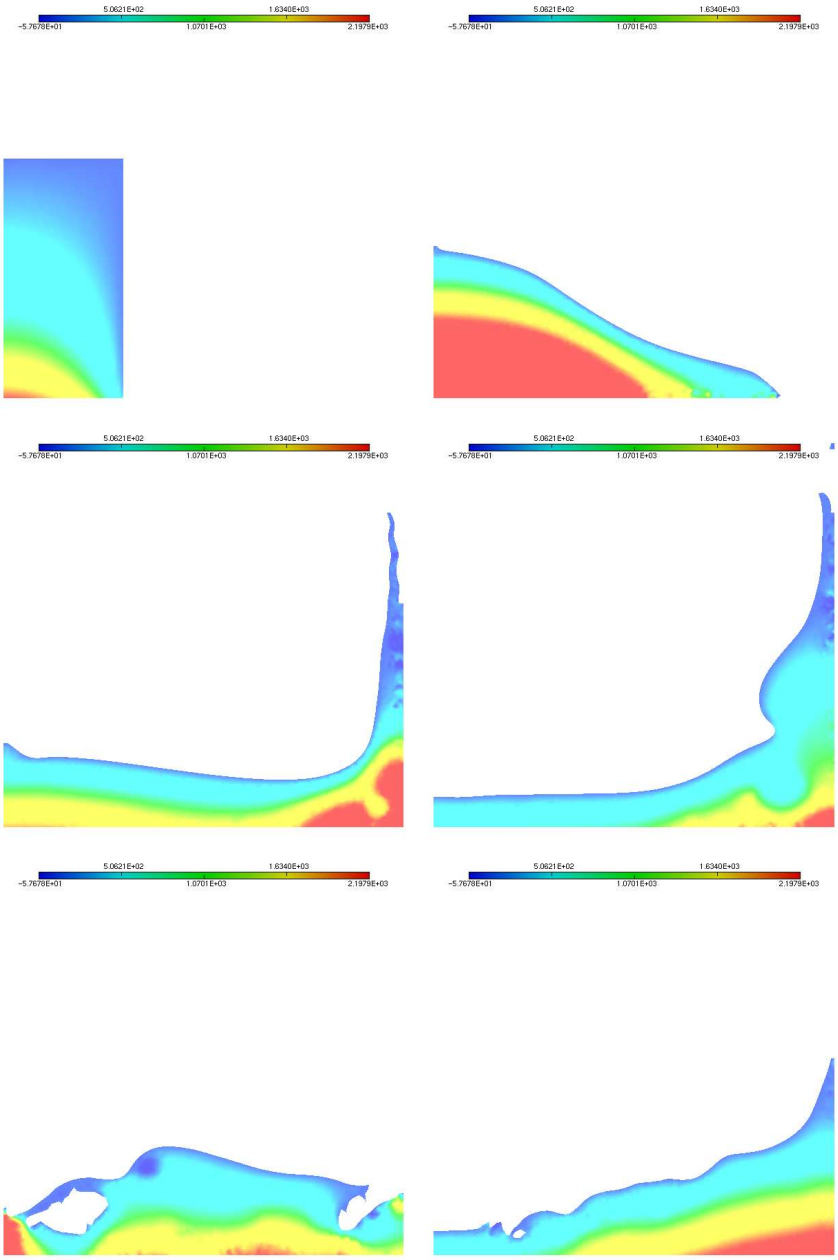


Figure 5.6: Pressure distribution without alpha shape at t={0.001,0.22,0.52,0.9,1.25,2.3}.

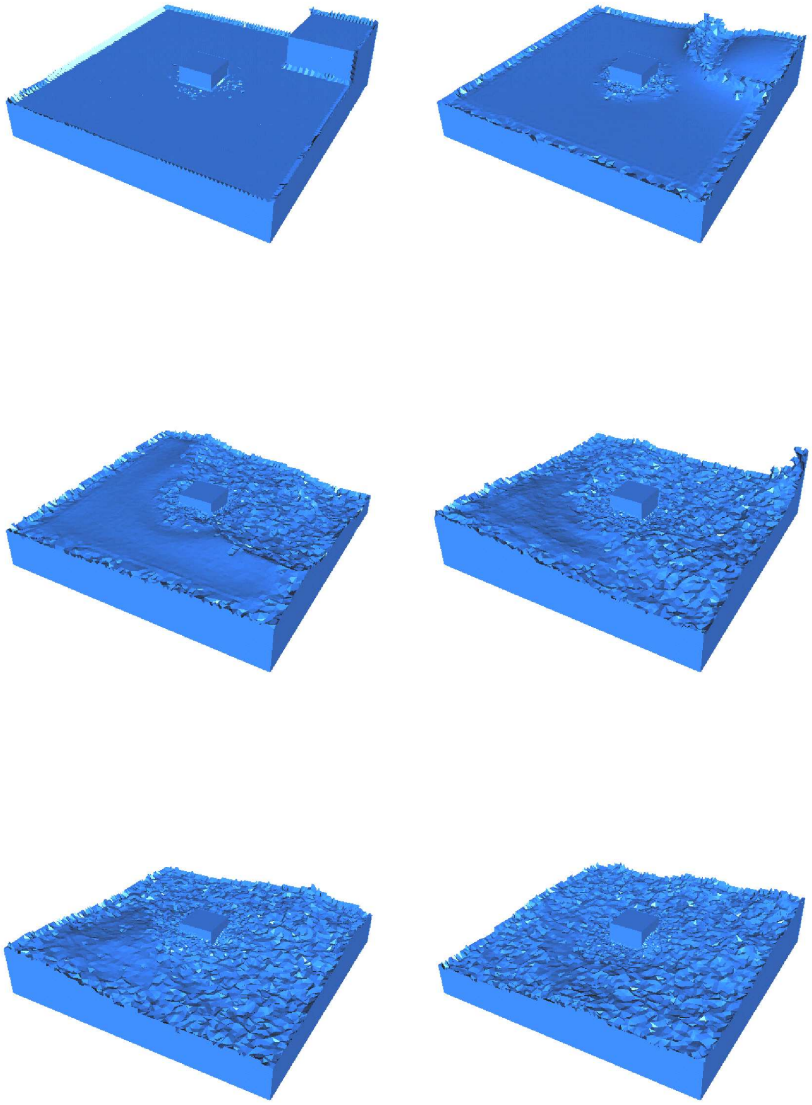


Figure 5.7: 3D dam break at t={0.1,0.9,2.3,4,5,6}.

5.6 Conclusion

In this chapter, the free surface problem has been considered, various methods have been tested and compared, in a Lagrangian context, against analytical and numericaly established results, and the differences with an Eulerian implementation commented. The Lagrangian formulation gives naturally an accurate description of the free surface until merging is considered, which happens very soon in a simple 2D dam break example. At this point, and due to the large deformations of the fluid, the correct characteristics must be deleted in order to maintain a good mesh quality, and to avoid a much more expensive contact formulation. An appealing advantage of the Lagrangian formulation is constituted by the fact that the non linear character of the Navier-Stokes equations is fully solved as no oscillations appear between elements filled of gas and water, as appears in an Eulerian formulation, so that an explicit position of the free surface is not required.

Bibliography

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational Geometry*, pages 39–48, 1998.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, 1998.

[3] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[4] J.D. Boissonat. Geometric structures for three-dimensional shape representation. *ACM transactions on graphics*, 3:266–286, 1984.

[5] J.D. Boissonat and F. Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry*, 22:185–203, 2002.

[6] J.P. Boris and D.L. Book. Flux-corrected transport I. SHASTA, a fluid transport algorithm that works. *J. Comput. Phys.*, -11:38–69, 1973.

[7] H. Borouchaki, A. Cherouat, P. Laug, and K. Saanouni. Adaptive remeshing for ductile fracture prediction in metal forming. *C.R. Mécanique*, 330:709–716, 2002.

[8] A. Caboussat. *Analysis and Numerical Simulation of Free Surface Flows*. PhD thesis, Ecole polytechnique fédérale de Lausanne, 2003.

[9] J.R. Cebral, R. Löhner, P.L. Choyke, and P.Y. Yim. Merging of intersecting triangulations for finite element modeling. *J. of Biomechanics*, 34:815–819, 2001.

[10] R. Codina and O. Soto. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. *Int. J. Num. Meth. Fluids*, 40:293–301, 2002.

[11] A.H. Coppola-Owen and R. Codina. Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *Int. J. Num. Meth. Fluids*, 2005. Accepted to publication.

[12] F. Del Pin. *The Meshless Finite Element Method Applied to a Lagrangian Particle Formulation of Fluid Flows.* PhD thesis, Santa Fe, 2003.

[13] P.G. Drazin. *Introduction to Hydrodynamic Stability.* Cambridge University Press, Cambridge, 2002.

[14] H. Edelsbrunner. Weighted alpha shapes. *Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana*, 1992.

[15] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13(1):43–72, 1994.

[16] P.J. Frey. About surface remeshing. In *9th International Meshing Roundtable*, pages 123–136. Sandia National Laboratories, 2000.

[17] P.J. Frey and H. Borouchaki. Geometric evaluation of finite element surface meshes. *Finite Elem. Anal. Des.*, 31(1):33–53, 1998.

[18] P.J. Frey and H. Borouchaki. Geometric surface mesh optimization. *Comput. Visual. Sci.*, 1:113–121, 1998.

[19] P. Hansbo. The characteristic streamline diffusion method for the time-dependent incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 99:171–186, 1992.

[20] P. Hansbo. A Crank-Nicolson type space-time finite element method for computing on moving meshes. *J. Comput. Phys.*, 159:274–289, 2000.

[21] C. Hirsch. *Numerical Computation of Internal and External Flows, Volume I & II.* Numerical Methods in Engineering, Wiley, New York, 1995.

[22] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, 1992.

[23] S.R. Idelsohn, E. Oñate, N. Calvo, and F. Del Pin. The Meshless Finite Element Method. *Int. J. Num. Meth. Eng.*, 58(6):893–912, 2003.

[24] C. Johnson. A new approach to algorithms for convection problems which are based on exact transport + projection. *Comp. Meth. Appl. Mech. Engrg.*, 100:45–62, 1992.

[25] S.F. Kistler. Hydrodynamics of wetting. In J.C. Berg, editor, *Wettability*, pages 311–430. Marcel Dekker Inc., New York, 1993, 1993.

[26] D Kuzmin and S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.*, 175(2):525–558, 2002.

[27] R. Löhner. Regridding surface triangulations. *J. Comput. Phys.*, 126:1–10, 1996.

[28] R. Löhner. Surface reconstruction from a cloud of points. *CIMNE Publications.*, 88, 1996.

[29] R. Löhner, K. Morgan, M. Vahdati, J.P. Boris, and D.L. Book. FEM-FCT: combining unstructured grids with high resolution. *Comm. Appl. Num. Meth.*, 4:717–729, 1997.

[30] R. Löhner, C Yang, and E. Oñate. Large-scale simulation of flows with violent free surface motion. In P. Bergan, J. García, E Oñate, and T. Kvamsdal, editors, *Computational Methods in Marine Engineering 2005*. CIMNE, Barcelona, 2005.

[31] V. Maronnier, M. Picasso, and J. Rappaz. Numerical simulation of free surface flows. *J. Comput. Phys.*, 155:439–455, 1999.

[32] J.C. Martin and W.J. Moyce. An experimental study of the collapse of a liquid column on a rigid horizontal plane. *Phil. Trans. Royal Soc. London*, A244:312–324, 1952.

[33] S. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002.

[34] E. Pichelin and T. Coupez. Finite element solution of the 3D mold filling problem for viscous incompressible fluid. *Comput. Meth. Appl. Mech. Engrg.*, 163:359–371, 1998.

[35] O. Pironneau. On the transport-diffusion algorithm and its application to the Navier-Stokes equations. *Numer. Math.*, 38:309–332, 1982.

[36] A. Shostko, R. Löhner, and W.C. Sandberg. Surface triangulation over intersecting geometries. *Int. J. Num. Meth. Eng.*, 44:1359–1376, 1999.

[37] R. Sibson. A vector identity for the Dirichlet tesselation. *Math. Proc. Camb. Phil. Soc.*, 87:151–155, 1980.

[38] R. Sibson. A brief description of natural neighbour interpolation. In V. Barnet, editor, *Interpreting Multivariate Data*, pages 21–36. John Wiley & Sons, Chichester, 1981.

[39] G Taubin. A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.

[40] M. Teichmann and M. Capps. Surface reconstruction with anisotropic density-scaled alpha shapes. In *IEEE Visualization'98*, North Carolina, USA, 1998.

[41] R Tilch and R. Löhner. Advances in discrete surface grid generation: Towards a reliable industrial tool for CFD. *AIAA-02-0862*, 2002.

[42] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31:335–362, 1979.

[43] H.K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, 2001.

Chapter 6

Coupled thermal flow with solidification

In this chapter, the thermal effects on the body under study is also considered. The main difficulty appears through the coupled influence of the displacement and the heat effects. This coupling can be partial if only motion plays a role on the heat equation through convective heat transfer. It can be made total by considering the Boussinesq model which adds a gravitational term in the momentum equation driven by the temperature distribution. It can still be stronger by considering the effect of solidification during the motion, where the viscosity will also be dependent of the temperature. The three next sections will present each of these coupling by order of gradation, and evidently also by difficulty. A last section provides numerical examples of the three previous sections, and validate the approach with the classical thermaly driven cavity, and the Rayleigh-Bénard instability. As in the precedent chapter, standard notations are used instead of the more awkward Lagrangian nomenclature of Chapter 2.

6.1 The thermal problem

In this section, the thermal problem is presented quickly, as it is now a well-known problem, the functional setting is introduced, and the discretization through the Galerkin method performed.

6.1.1 The heat equation

Neglecting mechanical effects and volume changing, the conservation of energy reads:

$$\rho \frac{\partial u}{\partial T} \frac{dT}{dt} = -\nabla \cdot \mathbf{q} + f_T \tag{6.1}$$

where u is the specific internal energy, $\mathbf{q}$ the heat flux, f_T a potential thermal source, ρ the density and T is the temperature. By using the Fourier law as constitutive law:

$$\mathbf{q} = -\kappa : \nabla T \tag{6.2}$$

where κ is the conductivity tensor, and defining the specific heat as:

$$C = \frac{\partial u}{\partial T} \tag{6.3}$$

the heat transfer equation reads:

$$\rho C \frac{dT}{dt} - \nabla \cdot (\kappa : \nabla T) = f_T \tag{6.4}$$

The heat equation is rich in various boundary conditions defined as:

$$\Gamma_D = \{\partial\Omega \,|\, T_{|\Gamma} = T_\Gamma\} \tag{6.5}$$

$$\Gamma_N = \{\partial\Omega \,|\, \mathbf{q} \cdot \mathbf{n} = -\overline{q},\ \mathbf{n} \text{ exterior normal to } \Gamma_N\} \tag{6.6}$$

where classically Γ_D is the Dirichlet part of the boundary and Γ_N is the Neumann part of the boundary, and $\overline{q}$ an imposed incoming flux.

$$\Gamma_C = \{\partial\Omega \,|\, \mathbf{q} \cdot \mathbf{n} = h_C(T - T_{env})\} \tag{6.7}$$

where Γ_C is the convection boundary condition, h the convection heat transfer coefficient, which may be temperature dependent, and T_{env} the environmental temperature.

$$\Gamma_R = \{\partial\Omega \,|\, \mathbf{q} \cdot \mathbf{n} = h_R(T - T_R)\} \tag{6.8}$$

where Γ_R is the radiation boundary condition, T_R the temperature of the external radiation source and :

$$h_R = \sigma_R \epsilon_R (T^2 + T_R^2)(T + T_R) \tag{6.9}$$

is a radiative coefficient, where $\sigma_R = 5.66961$ is the Stefan-Boltzmann constant, and ϵ_R the emissivity, $0 \leq \epsilon \leq 1$.

6.1.2 Weak form and functional setting

Multiplying equation (6.4) by a test function N, summing over the whole domain Ω, and integrating by parts some terms, the system reads:

$$\int_\Omega \rho C \frac{DT}{Dt} N \, dV + \int_\Omega \kappa : \nabla T \cdot \nabla N \, dV = \int_\Omega f_T N \tag{6.10}$$

At this point the classical functional setting can be introduced:

Problem 6.1.1 *Given $T_0 \in \mathbf{L}^2(\Omega)$ and $\mathbf{f_T} \in \mathbf{L}^2(\Omega)$, find T in $L^2(0,T;\mathbf{H}^1_\Gamma(\Omega))$ solution of equation (6.10) for all N belonging to $\mathbf{H}^1_0(\Omega)$, and T satisfying the initial condition.*

The weak form of the heat equation is the classical parabolic problem, and existence and uniqueness are classical results obtained in the same way as in Chapter 4.

6.1.3 Discretization

Using the Galerkin method for the spatial discretization and a θ-scheme in time, the algebraic system reads:

$$(\frac{\mathbf{M}_T}{\delta t} + \theta \mathbf{K}_T(\mathbf{U}))\mathbf{T}^{n+1} = \mathbf{F}_T + (\frac{\mathbf{M}_T}{\delta t} - (1-\theta)\mathbf{K}_T(\mathbf{U}))\mathbf{T}^n \tag{6.11}$$

where the previous matrices are:

$$\mathbf{M}_{Tij} = \int_\Omega \rho C N_i N_j \, dV \tag{6.12}$$

$$\mathbf{K}_{Tij} = \int_\Omega \kappa : \nabla N_i \cdot \nabla N_j \, dV \tag{6.13}$$

One common numerical problem in mould filling appears with the thermal shock between the cold mould and the boiling casting, so that wiggles constantly appear with a classical discretization. A short review of possible remedies is commented in [15]. A mixed discretization temperature/heat flux may avoid this problem. In order to compare with the last section, it must be noted that, contrarily to the generalized Stokes system (4.28), an inf-sup condition is not required due to the presence of the SPD temperature mass matrix [12, 18], so that an equal order discretization appears to be very attractive computationally. Furthermore, applying a block-Gaussian elimination as in the previous part, the same kind of system must be solved and the solver of all the previous section is readily available. For example, the mixed temperature equation to be solved reads:

$$(\frac{\mathbf{M}_T}{\delta t} + \theta \mathbf{B}\mathbf{M}_q^{-1}\mathbf{B}^T)\mathbf{T}^{n+1} = \mathbf{F}_T + (\frac{\mathbf{M}_T}{\delta t} - (1-\theta)\mathbf{B}\mathbf{M}_q^{-1}\mathbf{B}^T)\mathbf{T}^n \tag{6.14}$$

Matrix $\frac{\mathbf{M}_T}{\delta t} + \theta \mathbf{B}\mathbf{M}_q^{-1}\mathbf{B}^T$ is SPD so that it can be solved by PCG. However, compared to equation (4.45), the presence of the temperature mass matrix provides a much better conditioned operator so that, after having lumped the flux mass matrix, it could be interesting to solve it as in the previous section. Nonetheless, as mentioned in section 4.3.5, the temperature mass lumping does not only produce an additional diffusion balanced by the Crank-Nicolson time integrator but guarantees under reasonable mesh assumptions a monotonic scheme as the discretization is performed only with linear elements here. This valuable property compared with the increased storage of the mixed temperature/heat flux discretization makes the classical scheme much more attractive. Furthermore, at high mesh Reynolds number, the matrix becomes much more diagonally dominant and only a few iterations are necessary to converge with the PCG.

Coupling the thermal equation with the Navier-Stokes equations in this case is easily performed as the only non-linear problem is the mechanical part. At the implementation level, the fluid solver is firstly solved with its own non-linearity, and then the thermal problem is computed, as no interaction takes place between each other, which constitutes a whole time step.

6.2 The Boussinesq approximation

In this section, the Boussinesq approximation is introduced, the dimentionless problem presented and the discretization performed.

6.2.1 The modified equation

In order to take into account the variation with temperature of the material properties of the fluid, Boussinesq [1] introduced a variation of the density depending of the temperature in the gravitational force term only, supposing that the acceleration of the fluid is much smaller than the gravity. This provides the driving mechanism of the convective motions. The Boussinesq approximation for incompressible fluids [20] is then:

$$\rho = \rho_0(1 - \alpha(T - T_0)) \tag{6.15}$$

where α is the volume/thermal expansion coefficient of the fluid. Linearizing the mass, momentum and energy conservation equations, considering the thermal diffusivity as scalar, and writing them in dimensionless form by introducing a characteristic length L, time L^2/κ, velocity κ/L, pressure $\kappa^2\rho/L^2$ and temperature $\Delta T/Ra$ one obtains:

$$\nabla \cdot v = 0 \tag{6.16}$$

$$\frac{\partial v}{\partial t} = -\nabla p + Pr \times T\mathbf{e}_z + Pr \times \Delta v \tag{6.17}$$

$$\frac{\partial T}{\partial t} = \Delta T + Ra \times w \tag{6.18}$$

where w is the vertical velocity, $\mathbf{e}_z$ the unit vector in the vertical direction, Ra is the Rayleigh number, defined as:

$$Ra = \frac{\alpha g \Delta T L^3}{\nu\kappa} \tag{6.19}$$

with g is the gravity acceleration, and Pr is the Prandtl number, defined as:

$$Pr = \frac{\nu}{\kappa} \tag{6.20}$$

The Rayleigh number represents a nondimensional measure of the vertical temperature diffusion. It reflects the destabilizing effect of buoyancy to the stabilizing effect of molecular diffusion of momentum and buoyancy [10]. The Prandtl number characterizes the regime of convection of the flow by approximating the ratio of momentum diffusivity and thermal diffusivity [14].

6.2.2 Discretization

At the algebraic level, the fully coupled thermo-mechanical system then reads:

$$\begin{pmatrix} \mathbf{A_V}(\mathbf{U}) & \mathbf{B}(\mathbf{U})^T & \rho g \alpha \mathbf{M}_{VT} \\ \mathbf{B}(\mathbf{U}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_T}(\mathbf{U}) \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{P} \\ \mathbf{T} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{F}}_v \\ \mathbf{0} \\ \tilde{\mathbf{F}}_T \end{pmatrix} \tag{6.21}$$

There is no coupling between the velocity and the temperature due to the convective term but the coupling appears through the gravity force. Once more, all matrices are dependent of the displacement **U**, as emphasized. The whole matrix is not symmetric, and advantage can be taken by the fact that, freezing the velocity and displacement in the thermal problem gives rise to a linear problem, as in the last section. Here, as the Boussinesq term couples the velocity and the temperature, a global non linear loop must be performed. However, it was found that it was enough to include the coupling term in the non-linear displacement loop to achieve convergence. The system is then solved in a staggered way by considering the last known temperature for the velocity system and the last known displacement, for the computation of the temperature shape functions, until convergence.

6.3 Solidification

In this section, the solidification of the fluid is considered. The problem is then a phase change-problem where part of the fluid has solidified, part of the fluid is still fluid, and part of the fluid stays in between these two states, the 'mushy' zone. Here, the solidification problem is recalled, but it is fully presented in [3].

6.3.1 The phase-change problem

In phase-change problems, the specific internal energy is defined as:

$$u = \int_{T_{ref}}^{T} C dT + L f_{pc} \tag{6.22}$$

where T_{ref} is a reference temperature, C is the specific heat capacity, L is the latent heat released in a freezing problem, and f_{pc} is the phase change function defined by:

$$f_{pc}(T) = \begin{cases} 0, & T \leq T_m \\ 1, & T > T_m \end{cases} \tag{6.23}$$

for the isothermal case, and:

$$f_{pc}(T) = \begin{cases} 0, & T \leq T_s \\ 0 < g(T) \leq 1, & T_s < T \leq T_l \\ 1, & T > T_l \end{cases} \tag{6.24}$$

for the non-isothermal case. Here, T_m is the melting temperature, and T_s and T_l are the solid and liquid temperature respectively. Function $g(T)$ corresponds to the mushy zone and can be obtained by a microstructure model. In this work, a simple linear interpolation is performed between T_s and T_l so that:

$$g(T) = \frac{T - T_s}{T_l - T_s}, \quad T_s \leq T < T_l \tag{6.25}$$

This interpolation is also performed for the viscosity dependance on the temperature, where a large numerical value represents an infinite viscosity for the solid part. Inserting equation (6.22) into equation (6.1), the final solidification equation reads:

$$\rho\Big(C + L\frac{\partial f_{pc}(T)}{\partial T}\Big)\frac{dT}{dt} = \nabla\cdot(\kappa:\nabla T) + f_T \tag{6.26}$$

Another way to write equation (6.26) is to write the appearent heat capacity term due to phase change at the left hand side as an additional source term as:

$$\rho C\frac{dT}{dt} = \nabla\cdot(\kappa:\nabla T) + f_T - L\dot{f_{pc}} \tag{6.27}$$

This problem is non-linear because the position of the front is not known. At the front location, for an isothermal problem, the heat balance gives [9]:

$$(\mathbf{n}\cdot(\mathbf{k}:\nabla T))_s - (\mathbf{n}\cdot(\mathbf{k}:\nabla T))_l = \rho L v(\mathbf{x},t) \tag{6.28}$$

so that there is a jump in the heat flux at the front location.

6.3.2 Discretization

The main difficulty is the discontinuity arising in the isothermal case in the enthalpy at the phase-change front. Various solutions proposed in the literature were presented in Chapter 1. In this work, the solidification method of [4] is used, as it ingenuously avoids the difficulty through the introduction of the solid fraction. Performing a standard Galerkin method in space, a θ-scheme in time, and a Newton-Raphson linearisation, the discrete equation of the weak form in a non incremental form reads:

$$\begin{aligned}(\frac{\mathbf{M}_T + \mathbf{M}_{sol}}{\delta t} + \theta\mathbf{K}_T(\mathbf{U}))\mathbf{T}^{n+1} &= (\frac{\mathbf{M}_T}{\delta t} - (1-\theta)\mathbf{K}_T(\mathbf{U}))\mathbf{T}^n + \\ &\mathbf{F}_T - \frac{\mathbf{L}^{n+1} - \mathbf{L}^n}{\delta t} + \frac{\mathbf{M}_{sol}}{\delta t}\mathbf{T}_{ni}\end{aligned} \tag{6.29}$$

where the previous matrices and vectors are:

$$\mathbf{M}_{solij} = \int_\Omega \rho L\frac{\partial f_{pc}(T)}{\partial T}N_i N_j\, dV \tag{6.30}$$

$$\mathbf{K}_{Tij} = \int_\Omega \kappa:\ \nabla N_i\cdot\nabla N_j\, dV \tag{6.31}$$

$$\mathbf{L}_i^{n+1} = \int_\Omega \rho L N_i f_{pc}^{n+1}\, dV \tag{6.32}$$

Matrix $\mathbf{M}_T$ is the same as before and $\mathbf{T}_{ni}$ is the value of the temperature vector at the previous non linear iteration.

Another way of writing the phase change part of (6.30) for the isothermal phase change [16] can be derived by writing the solid fraction as:

$$f_{pc}(T) = H(T - T_m) \tag{6.33}$$

where $H(T - T_m)$ is the Heaviside function, so that (6.30) reads:

$$\tilde{\mathbf{M}}_{solij} = \int_\Omega \rho L \frac{\partial H(T)}{\partial T} N_i N_j \, dV \tag{6.34}$$

$$= \int_\Omega \rho L \delta(T - T_m) N_i N_j \, dV \tag{6.35}$$

By finding a bijection between each point of Ω, and the temperature and its normal section, we obtain:

$$\tilde{\mathbf{M}}_{solij} = \int_T \int_S \rho L \delta(T - T_m) N_i N_j J \, dT dS \tag{6.36}$$

where S is an hyperplane normal to the isothermal curve. It can be shown [16] that:

$$J = \frac{1}{|\nabla T|^2} \tag{6.37}$$

and that, as the hyperplane S has the same temperature for each value of the integrated temperature:

$$dS = |\nabla T| \, ds \tag{6.38}$$

were ds the curvilinear abscissa so that, finally:

$$\tilde{\mathbf{M}}_{solij} = \int_\Gamma \rho \frac{L}{|\nabla T|} N_i N_j J \, ds \tag{6.39}$$

where Gamma is the locus of the interface.

The derivative of the solid fraction respect to the temperature in (6.30) is then numericaly evaluated through its secant so that the solidification mass matrix reads:

$$\mathbf{M}_{solij} = \int_\Omega \rho \Big(C + L \frac{f_{pc}(T_i^{n+1}) - f_{pc}(T^n)}{T_i^{n+1} - T^n} \Big) N_i N_j \, dV \tag{6.40}$$

where T_i^{n+1} refers to the temperature array at time $n+1$ during the non-linear iteration number i.

Here, the problem is much more complex as both the mechanical and thermal problem are non linear and coupled through the Boussinesq term. A global non-linear loop is then necessary as is a local non-linear loop for each subproblem. The non-linear solver strategy is shown on Figure 6.1. All non-linearities are solved with a fixed point method. Furthermore, due to the high viscosity values, the stress-divergence form is used so that the velocity components are coupled to take into account the correct boundary condition in case of free surface. The whole process is then computationally very intensive.

(1) For each time step:

- Global staggered non linear loop: given T^0_{n+1}
- $Uglobal = 0$
- $DO\ i = 1, 2, .., n$
  - (2) Solve the Navier-Stokes equations:
    - $Ulocal = 0$
    - $DO\ j = 1, 2, .., n$
      - (2) Solve the linearized Navier-Stokes equations considering temperature T^i_{n+1} and obtain a new displacement U^{j+1}_{n+1}
      - Check convergence:

$$\frac{\|U^{j+1}_{n+1} - Ulocal\|_{L^2}}{\|U^{j+1}_{n+1}\|_{L^2}} \leq \epsilon_{Ulocal}$$

      - if (no converged): $Ulocal = U^{j+1}_{n+1}$ goto 2
      - else goto 3
  - (3) Solve the thermal problem:
    - $Tlocal = 0$
    - $DO\ k = 1, 2, .., n$
      - Solve the linearized thermal problem considering displacement U^{j+1}_{n+1} and obtain a new temperature T^{k+1}_{n+1}
      - Check convergence:

$$\frac{\|T^{k+1}_{n+1} - Tlocal\|_{L^2}}{\|T^{k+1}_{n+1}\|_{L^2}} \leq \epsilon_T$$

      - if (no converged): $Tlocal = T^{k+1}_{n+1}$ goto 3
      - else goto 4
  - (4) Check global convergence:

$$\frac{\|U^{i+1}_{n+1} - Uglobal\|_{L^2}}{\|U^{i+1}_{n+1}\|_{L^2}} \leq \epsilon_{Uglobal}$$

    - if (no converged): $Uglobal = U^{i+1}_{n+1}$ goto 2
    - else $t = t + dt$ goto 1

Figure 6.1: The staggered scheme for coupled thermo-mechanical analysis.

6.4 Numerical examples

In this section, various numerical experiments are presented to illustrate the thermal equations presented above. The first example reproduces the cylinder benchmark of Chapter 4. The second example has been chosen to validate the method by comparison with previous established numerical results obtained for the same problem using an Eulerian formulation. The third example describes a typical free surface problem with contact on a solid. The fourth and fifth examples involve a complete thermo-mechanical coupling with the classical Rayleigh-Bénard instability. However, the fifth example introduces a not so classical free-rigid boundary condition. Finally, a last example illustrates the solidification process with an isothermal phase-change.

6.4.1 Vortex shedding behind a cylinder

The same example as in Chapter 4 has been computed with the heat equation. Here, the coupling is only partial as, as already seen, the temperature has no influence on the displacement in this case. The initial temperature is fixed at 20^0C for the fluid and at 60^0C for the cylinder. The temperature is imposed on the cylinder surface and all other boundaries are adiabatic. Figures 6.2 and 6.3 illustrate the temperature distribution at the same instants as in Chapter 4. The thermal capacity is fixed at $2400J.(kg^0C)^{-1}$, the thermal conductivity at $0.25W.(m^0C)^{-1}$. At $Re = 100$, the results are in very good agreements with the experimental results presented in [13] p. 112 and p. 150. At $Re = 10^6$ the fluid is turbulent, and so is the heat transport. In both cases, the Lagrangian formulation provides a very accurate treatment of the convective term.

6.4.2 The cavity problem

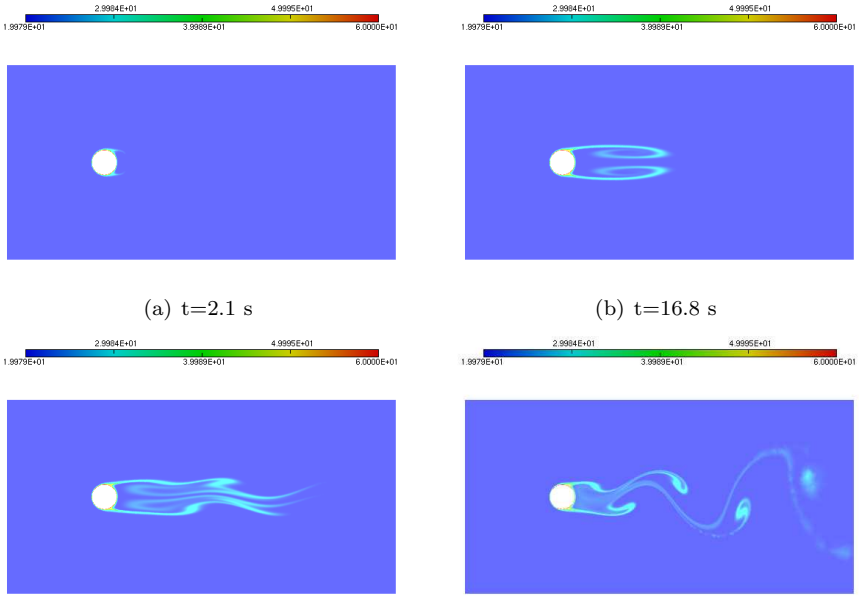
This example is a classical benchmark for Eulerian formulations. The fluid is inside a cavity, the left wall is heated isothermally to 20.5^oC, and the right wall is heated isothermally to 19.5^oC, the other sides being adiabatic. The fluid is initially at 20^oC, which is the reference temperature. The Rayleigh number, defined as:

$$Ra = \frac{\alpha g \Delta T L^3}{\nu \kappa} \tag{6.41}$$

has been chosen as in [17], and is equal to 10^6. The Prandtl number, defined as:

$$Pr = \frac{\nu}{\kappa} \tag{6.42}$$

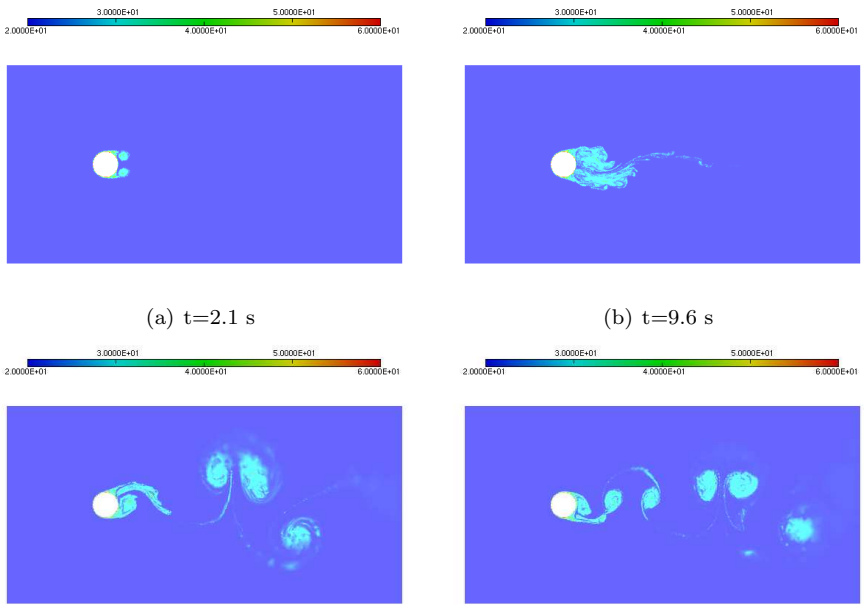
is equal to 1. The cavity is a square, as in [17]. The mesh is composed of 16400 nodes during the whole analysis, and 32000 elements at the beginning, 2/3 for the fluid part. As explained in the last section, if a node is too close to another one, it is removed and placed in another part, but the total number of nodes remains constant. The temperature and velocity distributions are presented in Figure 6.4 at different time values. The results are in perfect agreement with the



(a) t=2.1 s (b) t=16.8 s

(c) t=31.2 s (d) t=36.3 s

Figure 6.2: Thermal distribution for the vortex shedding behind a cylinder at $Re = 100$.



(a) t=2.1 s (b) t=9.6 s

(c) t=16.5 s (d) t=30.9 s

Figure 6.3: Thermal distribution for the vortex shedding behind a cylinder at $Re = 10^6$.

one obtained by Strada and Heinrich [17]. It should be noticed that these kind of examples are typical Eulerian examples, the configuration being particularly bad for the Lagrangian approach computationally. As a matter of fact, as the cavity is closed, there is much more contact with the wall than in usual free surface problems, and the space to fill for the element is much more reduced. However, the results obtained here demonstrate the validity of the method, even in this difficult case.

6.4.3 Mould filling

This example represents a mould filling by water at 60^oC. The mould initially has a temperature of 20^oC. The coupling between the mechanical and thermal parts is only partial due to the mesh movement, but the velocity is not influenced by the temperature. The contact is explicitly treated by the remeshing. The only potential problem could be to choose a time step too big so that the nodes go through the wall without having created an element of contact with it. However, the distortion of the mesh can not imply such non acceptable time steps. Numerical results are shown on Figure 6.5 at various instants during the filling and on Figure 6.6 to illustrate the thermal field through a cut of the mesh. It can be appreciated how the fluid fills the mould for a complicated geometry and produces a mixing by convection which diffuses quickly the temperature.

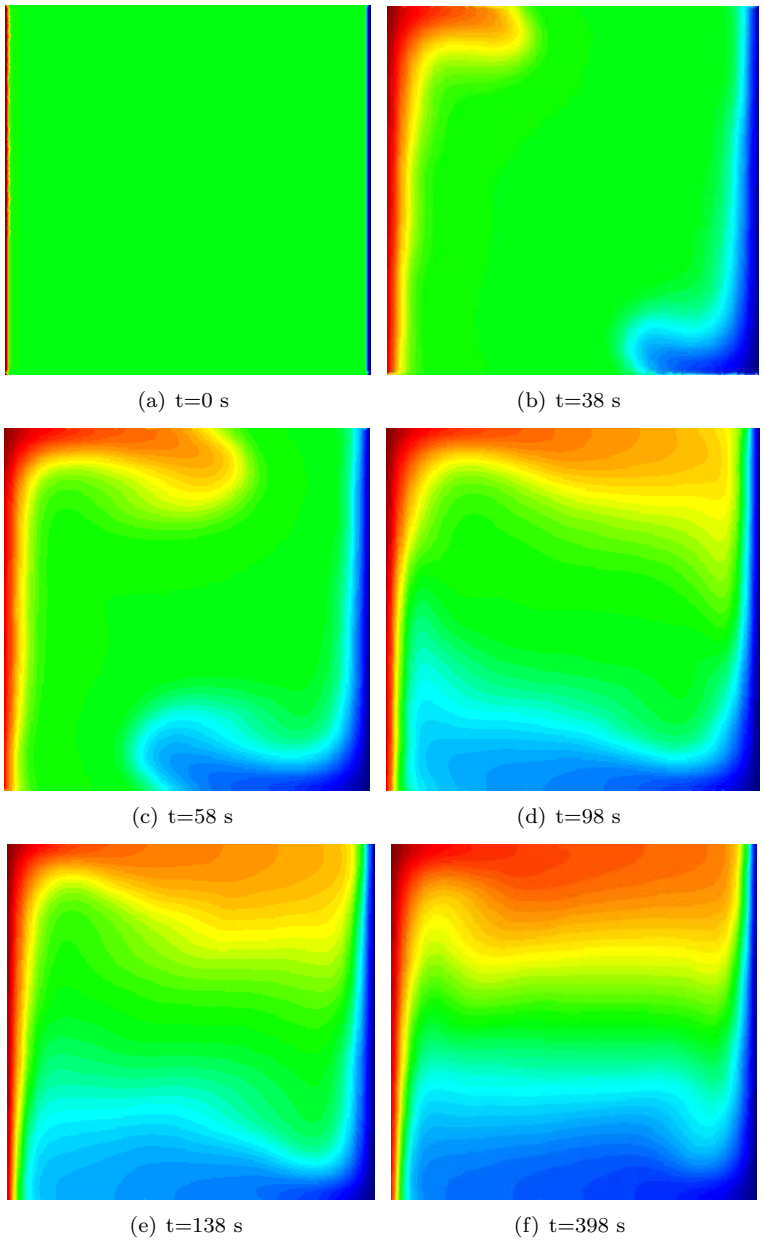
6.4.4 The Rayleigh-Bénard instability

This example is a classical problem of hydrodynamic instabilities, see [10] and [13] for a good description of it. The problem considered is a fluid initially at rest, which is heated from below. Assuming an infinitesimal perturbation of temperature, the Boussinesq approximation will induce a vertical movement towards the cooler region for the hotter lower part and viceversa, which will reinforce the initial perturbation. Given a critical Rayleigh number, the state of the flow will depend on the fact that its Rayleigh number is lesser or greater than the critical value. For $Ra < Ra_c$, no convection will occur and the flow is subcritical. If $Ra > Ra_c$, the instability will begin and produce the convection. The flow is then supercritical. For a slightly supercritical Rayleigh number, after passing through successive bifurcations, a steady state is reached as illustrated in [19], p 82. By increasing the Rayleigh number, other bifurcations occur after the primary bifurcations, according to the Prandtl Number, as depicted experimentally in [10], until reaching a turbulent flow.

By linearizing the mass conservation equation and the Navier-Stokes equations coupled with the heat equation in an Eulerian formulation, and by introducing a decomposition in normal modes, as performed in [11], a stability study leads to a solution of the form:

$$w = W(z)f(w,y)e^{st}, \qquad T = T(z)f(x,y)e^{st} \tag{6.43}$$

where s is a complex number representing the eigenvalue of the mode, w is the vertical component of the velocity, and f an unknown function. The solution

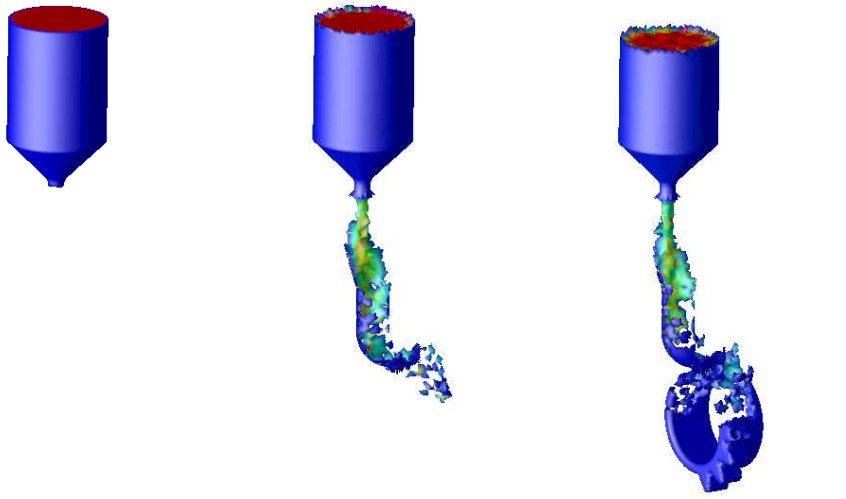


(a) t=0 s (b) t=38 s

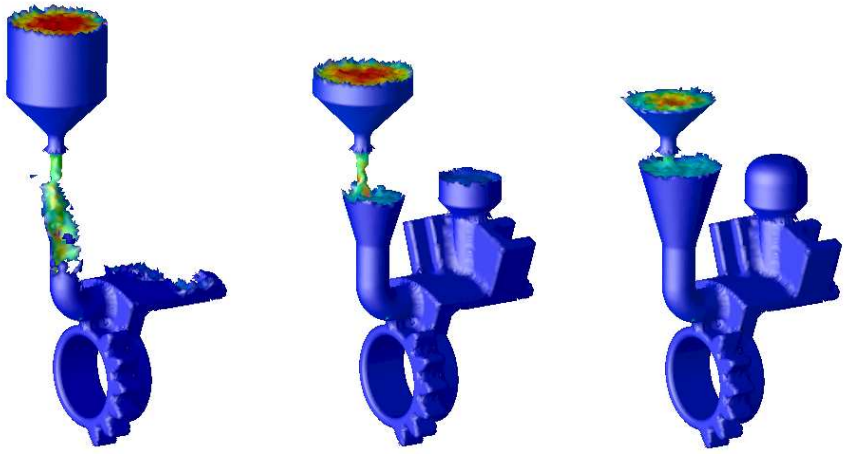
(c) t=58 s (d) t=98 s

(e) t=138 s (f) t=398 s

Figure 6.4: Temperature distribution in a closed cavity for $\theta \in \{19.5; 20.5\}$.



(a) t=0.1 s (b) t=5 s (c) t=15 s



(d) t=30 s (e) t=60 s (f) t=80 s

Figure 6.5: Temperature distribution at different instants during the mould filling.

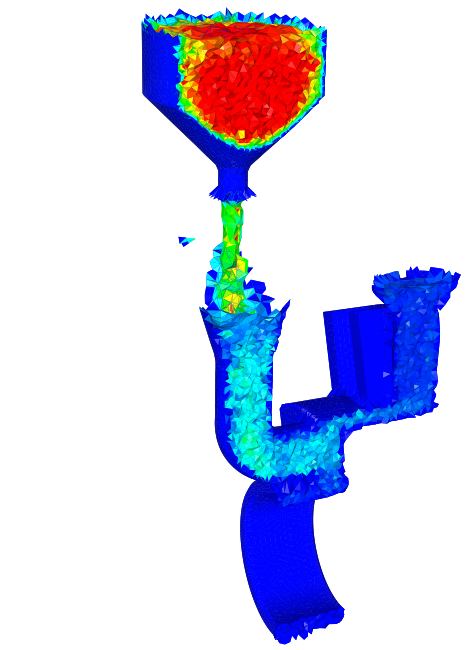


Figure 6.6: Temperature distribution through a mesh cut at t=50 s.

of the linearized equation with non physical free-free boundaries has the form:

$$W_n = A\, sin(n\pi z) \quad (n = 1, 2 \ldots) \tag{6.44}$$

The sign of the real part of s will decide of the stability of the flow. By solving the last equations with $s = 0$, and minimizing the Rayleigh number with respect to the wave number, it is possible to find the critical Rayleigh number, analytically for the free-free boundary condition, and numerically for the rigid-rigid and free-rigid boundary condition, as explained in [11] and [5]. For the rigid-rigid case, $Ra_c = 1708$.

In this numerical example, the bottom is heated isothermally at 21^oC, the top at 19^oC, and the reference and initial temperature of the fluid is 20^oC. The sides are adiabatic and $Ra = 10^5$ and $Pr = 10^{-1}$. The flow is then supercritical. The numerical results of Figure 6.8, which represent the norm of the velocity clearly depict the typical cell pattern observed experimentally when one horizontal side is much shorter than the other. These cells form rolls rotating in opposite direction for neighboring rolls, along the shortest side. However, what can not be well appreciated on the picture and is particulary remarkable, is that a quasi-steady state is reached, with a periodic oscillation of the temperature and the cells. This phenomenon is described in details in [14] by considering moderately nonlinear convection, and it is has to be noticed that the numerical results coincide with experimental ones for low Prandtl number,

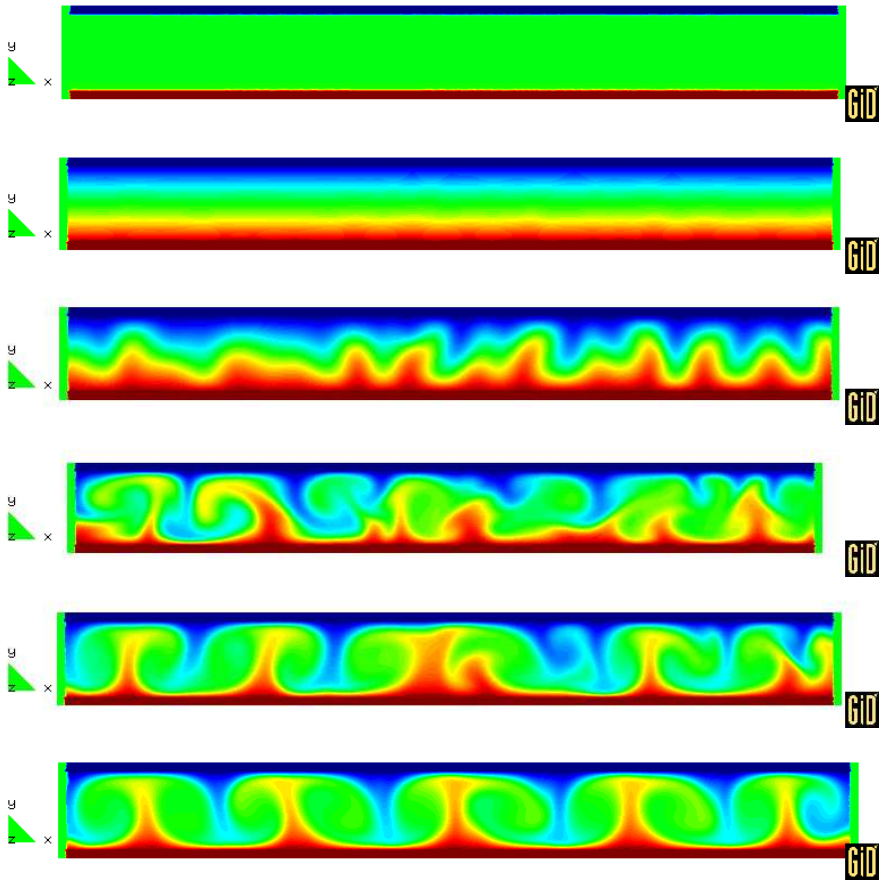


Figure 6.7: Temperature distribution for the rigid-rigid Rayleigh-Bénard instability for t={0.1, 38, 78, 108, 158, 400} and $\theta \in \{19; 21\}$.

which is exactly the situation here. The rolls oscillate as depicted in [6]. It is interesting to notice the interaction between the different cells before the quasi-steady state. The supercritical state could be observed through the approximate regularity of the cells at the last time step. The results are in agreement with [2] in the sense that the solution is time dependant for such a high Rayleigh number value. Figure 6.7 represents the temperature distribution at different moments and corresponds to the theoretical expected result of a supercritical flow.

6.4.5 The Rayleigh-Bénard instability with free surface

This example is rarely treated as it combines the difficulty of the thermal convection with the detection of free surfaces. The problem considered here does

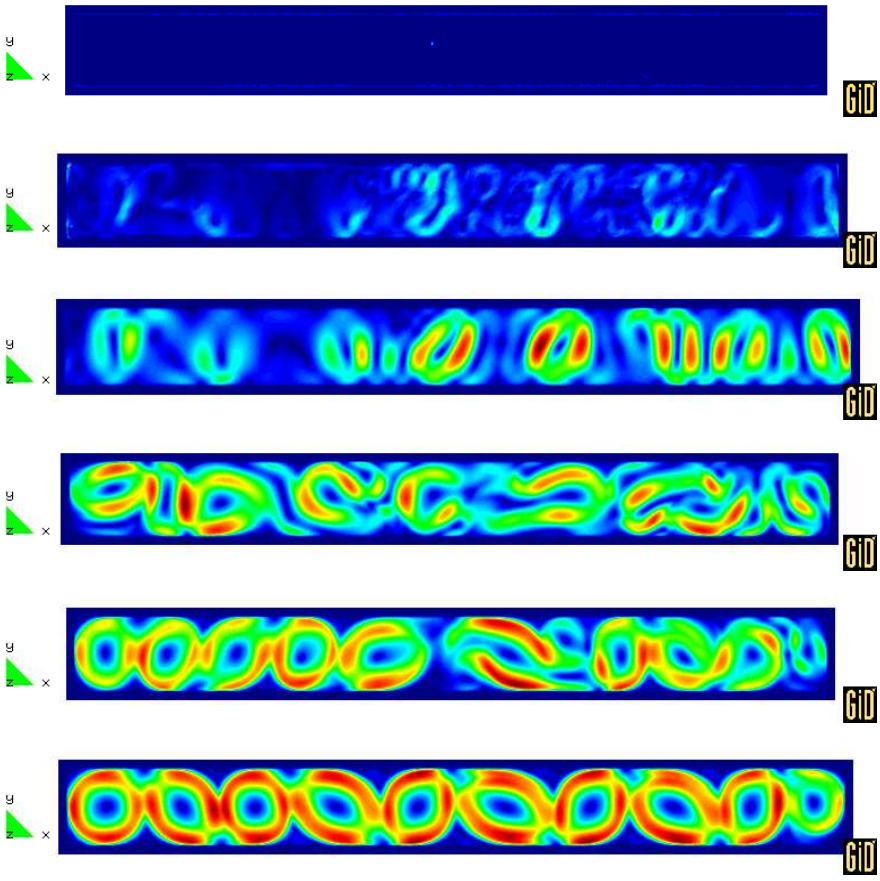


Figure 6.8: Velocity norm for the rigid-rigid Rayleigh-Bénard instability for t={0.1, 38, 78, 108, 158, 400}.

not take into account any surface tension, a phenomenon associated to the hydrodynamical instability of Bénard-Marangoni. Here, the coupling between temperature and displacement produces the instability, and circular cells are expected, compared to the appearance of hexagonal cells in the case of the Bénard-Marangoni instability [20].

The same conditions as in the previous example are chosen. A temperature of 19^oC is imposed on the free surface of the fluid. The same instants as above are reported to compare with the rigid-rigid boundary case. To follow the discussion on the critical Rayleigh number, the boundary conditions considered here imply $Ra_c = 1101$ theoretically so that the numerical experiment has a Rayleigh number far beyond Ra_c. As seen in Figure 6.10, the quasi-steady state has not been reached, due to the high Rayleigh number value. Some parts of the domain present a quasi static behaviour, but some cells are unstable. The rolls oscillate as in the preceeding example but some of the cells appear and disappear which is well observed at the last time step for the cells in the middle of the left part of the experiment. If the calculation of the critical Rayleigh number for the free-rigid case is well-known, it has not been possible to find in the literature theoretical results about the pattern of the cells for this kind of boundary conditions. Figure 6.9 represents the temperature distribution which contains the same unstable behaviour as the velocity. Figure 6.11 represents a detail of two cells with their temperature, norm of the velocity and velocity field.

6.4.6 Solidification with convection validation

The example chosen is the one proposed in [7] and references therein, as it offers numerical and experimental results. It considers the melting of a pure metal gallium cavity of aspect ratio of 2. Initially, the temperature is fixed to 28.3^oC and the left wall has an imposed temperature of 36^oC. The no-slip condition is imposed on all the wall sides. Physical values are reported in [7]. The viscosity of the solid part has been fixed to 10^6.

First of all, at the solver level, it is a difficult problem, as the viscosity will vary greatly in all the domain. As the phase-change is considered as isothermical, the viscosity has only two values, namely the one of the fluid and the one of the solid part. The first non linear iteration of the first time step is considered here. As the temperature is imposed at the left wall, all elements with one side will be considered as fluid elements, whereas the remaining part is solid. With the three preconditioners considered in Chapter 4, namely the standard Laplacian, the compatible Laplacian, and the pressure mass matrix, no one converge separately for this solid viscosity value, as does the conjugate gradient Uzawa without preconditioning. The iteration number of the association of the compatible Laplacian and the pressure mass matrix on one side, and of standard Laplacian with the pressure mass matrix on the other side is reported for various values of the solid viscosity in Table 6.1 and a tolerance of 10^{-12}. First of all, both converge always, which shows the importance of the association of both behaviours of the Uzawa operator, as commented in Chapter

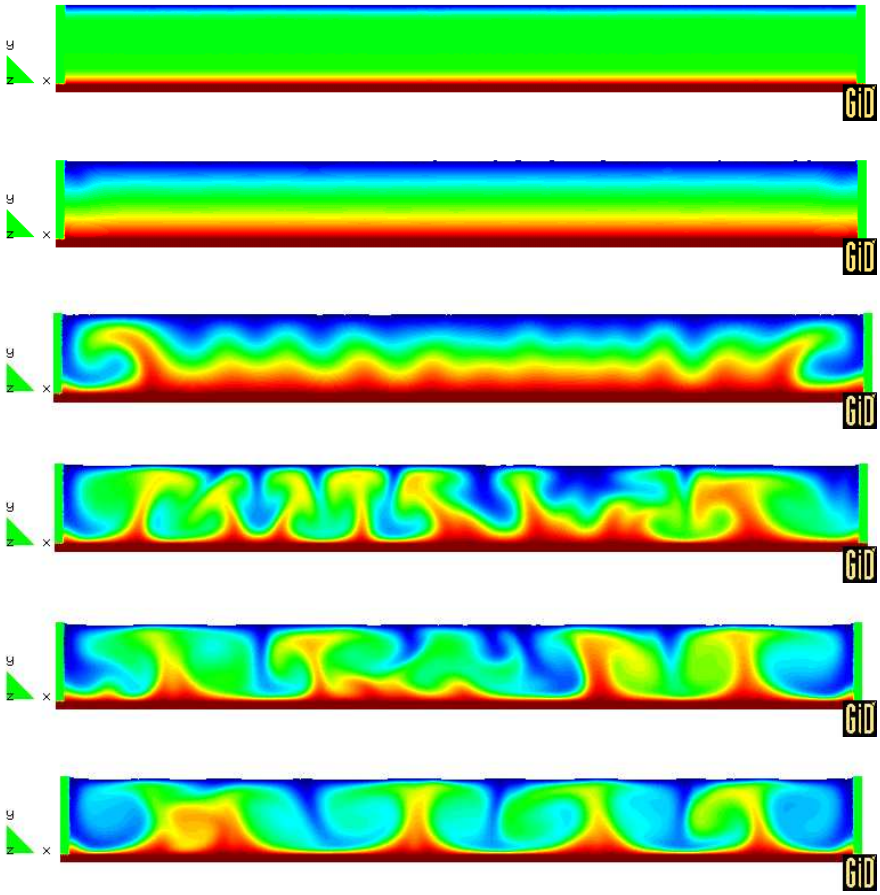


Figure 6.9: Temperature distribution for the free-rigid Rayleigh-Bénard instability for t={4, 38, 78, 108, 158, 400} and $\theta \in \{19; 21\}$.

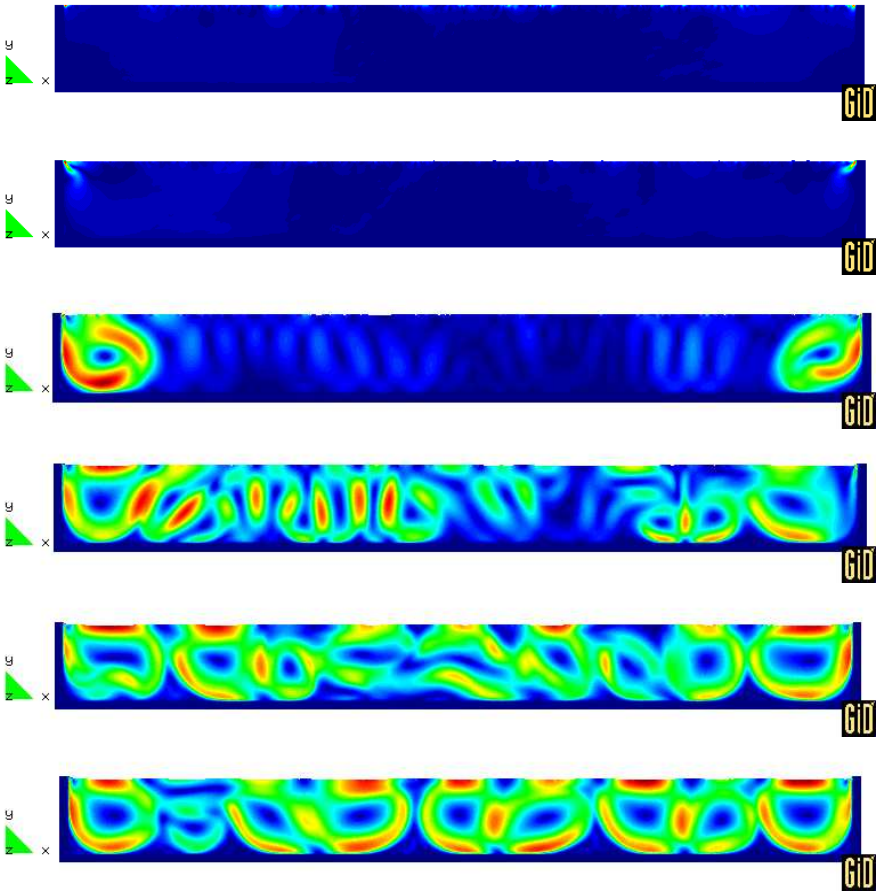
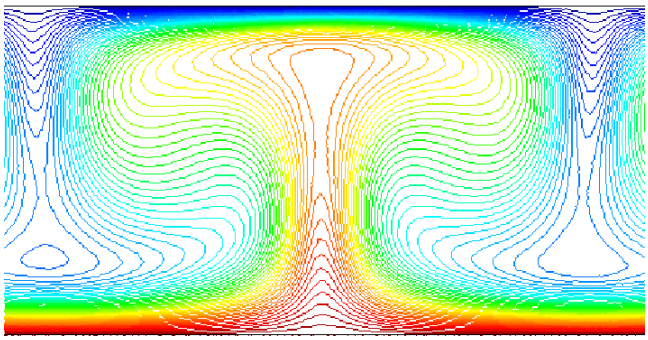
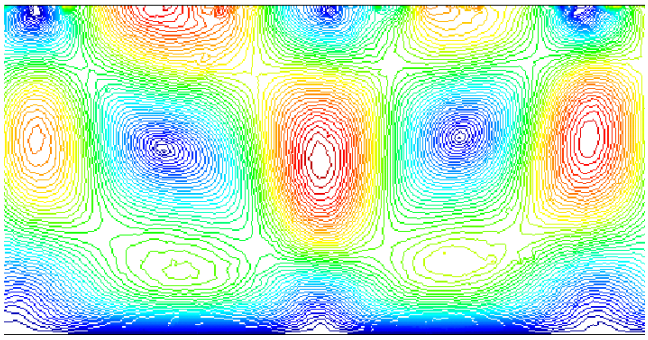


Figure 6.10: Velocity norm for the free-rigid Rayleigh-Bénard instability for t={4, 38, 78, 108, 158, 400}.



(a) Temperature isolines



(b) Velocity norm isolines

Figure 6.11: Detail of two cells for the free-rigid Rayleigh-Bénard instability.

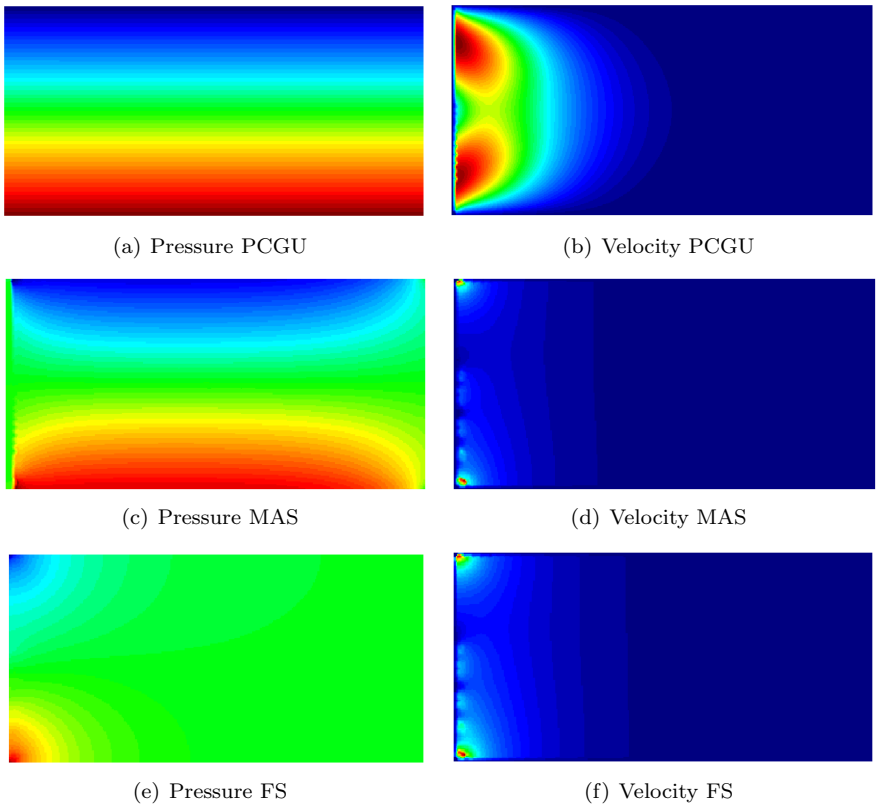
Solid viscosity	Compatible Lapl. Iter.	Standard Lapl. Iter.
0.00181	8	23
0.0181	20	36
0.181	36	48
1.81	42	48
18.1	46	45
181	47	44
1810	50	46
1e5	49	45
1e6	49	44

Table 6.1: Iteration number of the PCGU preconditioned by either the compatible Laplacian and the pressure mass matrix, or the standard Laplacian and the pressure mass matrix for various values of the solid viscosity, a tolerance of the PCGU of 10^{-12}.

4. The compatible Laplacian is much better than the standard Laplacian for low solid viscosities, which corresponds to the almost uniform viscosity case. However, as the solid viscosity grows, the iteration number of the compatible Laplacian grows faster than the standard Laplacian, for which each iteration is cheaper. For high values of the solid viscosity, the pressure mass matrix gives the behaviour of the solid part so that both Laplacian seem to reach a plateau. However the compatible Laplacian requires more iterations than the standard Laplacian. It is a surprising fact, as the compatible Laplacian is much nearer from the inertial part of the Uzawa operator than the standard Laplacian. The variable viscosity mixes both inertial and diffusive part in a complicated way so that further investigations must be done to fully understand this behaviour.

The pressure and the velocity distribution are reported on Figure 6.12 for the PCGU, the Modified Algebraic Splitting (MAS) and the Fractional Step (FS). Figure 6.12 illustrates the difficulty of the problem. As a matter of fact, around 50 iterations were necessary for the PCGU to converge, which means that, as the MAS is mostly equivalent to one iteration of the PCGU, the MAS results are far from the PCGU as seen in Figure 6.12. The results obtained by changing the compatible Laplacian for the standard Laplacian are very similar to the results presented for the MAS, as are the results obtained only with the pressure mass matrix in the MAS scheme. It evidently suggests that the Uzawa operator is much closer to the identity operator than to the pressure Laplacian. This is illustrated with the results of the FS, where only the pressure Laplacian is taken into account. It is a clear example that, in the general case, the monolithic resolution remains unavoidable.

Figure 6.13 displays the velocity norm at various instants. At the beginning of the simulation, a stream of melted fluid takes place in the whole height. As time goes on, the natural convection is more intense, and various cells appear. As the convection brings warmer fluid at the top of the cavity, the melting occurs

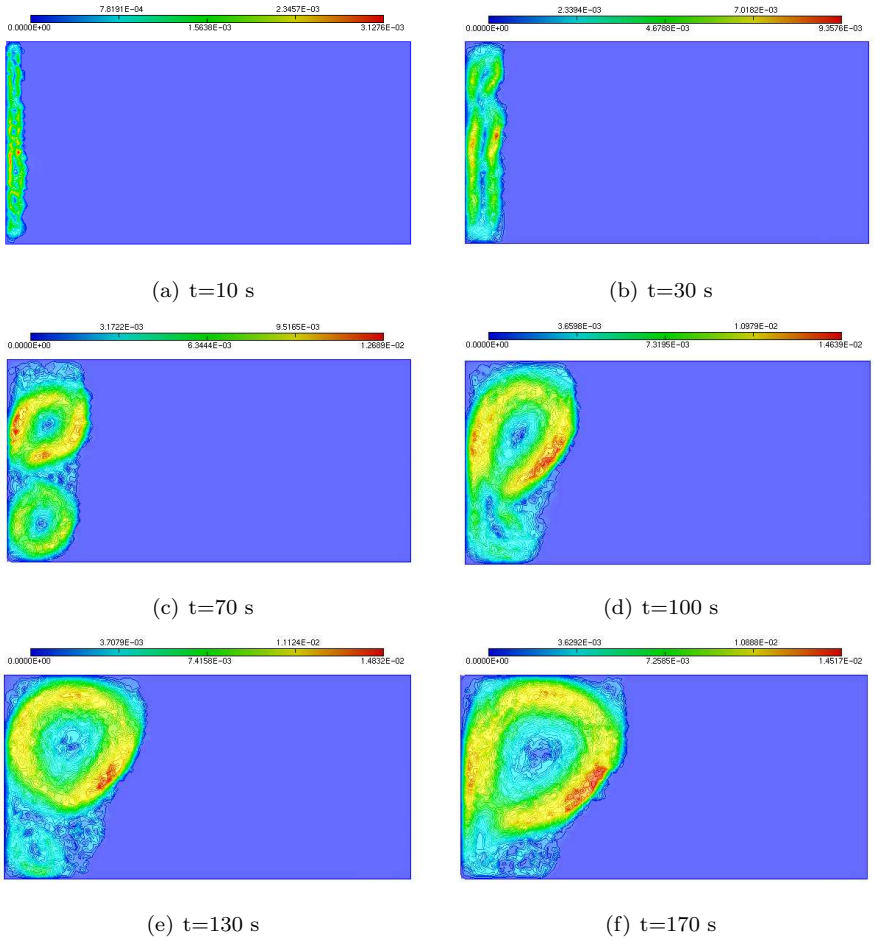


(a) Pressure PCGU (b) Velocity PCGU

(c) Pressure MAS (d) Velocity MAS

(e) Pressure FS (f) Velocity FS

Figure 6.12: Pressure and velocity distribution for the gallium solidification problem for the monolithic PCGU, the Modified Algebraic Splitting (MAS) and the Fractional Step (FS).



(a) t=10 s (b) t=30 s

(c) t=70 s (d) t=100 s

(e) t=130 s (f) t=170 s

Figure 6.13: Velocity distribution for gallium melting at various instants

faster in the upper part. If the qualitative results and the front evolution agree rather well with the references [8] and [7], the solidification time does not match, which means that not all the latent heat was absorbed during the solidification process. The convection is rather well resolved but the solidification process is responsible of the fact that the front moves too fast with a factor around 3. In this example, only the contribution of the tangent matrix was taken into account in a Picard iteration, neglecting the latent heat balance, as is done in the enthalpy method. The latent heat balance is not verified, and the latent heat not fully absorbed which explains the discrepancy with the time scale.

6.5 Conclusion

In this chapter, the thermal effect has been taken into account in the Lagrangian formulation of an incompressible Newtonian fluid. The semi-coupling due to the convective heat transfer in an Eulerian formulation is implicitely considered in the Lagrangian description, and once again no additional stabilisation has to be added in convection dominant flow cases. The fully coupled mechanical thermal flow is obtained by adding the Boussinesq approximation. Numerical results have shown that the non linearity is well resolved and a strong convergence has been obtained through the fixed point method. Finally, the solidification problem has been considered by modifying the temperature mass matrix, and the temperature dependance of the viscosity has been highlighted to simulate the mushy zone and an almost perfect solid zone. The temperature mass lumping and the second order Crank-Nicolson time integration provide an accurate and very efficient solver. Finally, the method has been validated with the cavity benchmark, and the classical Rayleigh-Bénard instability, where numerical results are in perfect agreement with the theoretical considerations. A 3D mould filling has also been presented to illustrate the robustness of the method.

Bibliography

[1] J. Boussinesq. *Théorie Analytique de la chaleur.* Gauthiers-Villars, Paris, 1903.

[2] F.H. Busse. Non linear properties of thermal convection. *Rep. Prog. Phys.*, 41:1929–1967, 1978.

[3] D. Celentano. *Un modelo thermomecánico para problemas de solidificación de metales.* PhD thesis, UPC, 1994.

[4] D Celentano, E. Oñate, and S. Oller. A temperature-based formulation for finite element analysis of generalized phase-change problems. *Int. J. Num. Meth. Eng.*, 37:3441–3465, 1994.

[5] S. Chandrasekar. *Hydrodynamic and Hydromagnetic Stability.* Clarendon Press (Oxford), 1961.

[6] R.M. Clever and F.H. Busse. Transition to time dependant convection. *J. Fluid Mech.*, 65:625–645, 1974.

[7] M. Cruchaga and D. Celentano. A finite element thermally coupled flow formulation for phase-change problems. *Int. J. Num. Meth. Eng.*, 34:279–305, 2000.

[8] J.A. Dantzig. Modelling liquid-solid phase changes with melt convection. *Int. J. Num. Meth. Eng.*, 28:1769–1785, 1989.

[9] S.H. Davis. *Theory of Solidification.* Cambridge University Press, 2001.

[10] P.G. Drazin. *Introduction to Hydrodynamic Stability*. Cambridge University Press, Cambridge, 2002.

[11] P.G. Drazin and W.H. Reid. *Hydrodynamic Stability*. Cambridge University Press, Cambridge, 1981.

[12] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equation*. Springer, 1986.

[13] E. Guyon, J.P. Hulin, and L. Petit. *Hydrodynamique physique*. EDP Sciences, 2001.

[14] E.L. Koschmieder. *Bénard Cells and Taylor Vortices*. Cambridge University Press, Cambridge, 1993.

[15] E. Pichelin and T. Coupez. A Taylor discontinuous Galerkin method for the thermal solution in 3d mold filling. *Comput. Meth. Appl. Mech. Engrg.*, 178:153–169, 1999.

[16] M. Storti, L.A. Crivelli, and S.R. Idelsohn. An efficient tangent scheme for solving phase-change problems. *Comp. Meth. App. Mech. Eng.*, 66:65–86, 1988.

[17] M. Strada and J.C. Heinrich. Heat transfer rates in natural convection at high Rayleigh numbers in rectangular enclosures: A numerical study. *Numerical Heat Transfer*, 5:81–93, 1982.

[18] V. Thomée. *Galerkin finite element methods for parabolic problems*. Springer Verlag, Berlin/New York, 1984.

[19] M. Van Dyke. *An Album of Fluid Motion*. The Parabolic Press, Stanford, CA, 1989.

[20] R.Kh. Zeytounian. Il y a cent ans...et aujourd'hui l'approximation de Boussinesq. *Matapli*, 71, april 2003.

Chapter 7

Conclusion and new perspectives

In this work, an incompressible Lagrangian fluid flow description applied to thermal convection was discussed. The Navier-Stokes equations in a Lagrangian formulation were derived. The non-linearities induced by the unknown final position were taken into account and solved by the Picard iteration method. The fluid solver applied to this problem was presented in details, and the mass conservation of various schemes and elements was deeply highlighted. The complete coupled thermal-fluid problem was also introduced. Numerical examples have illustrated the proposed methods in two and particularly three dimensions, which was the ambition of the present thesis.

The mesh generation part was fully described, as it has to support the convection of the nodes inherent to the method. In this work, the Delaunay based method was chosen, mainly due to its independance with a geometrical model, its ability to produce a few milion elements in a few minutes in a standard PC, and the possibility to reuse the points at each iteration to minimize the interpolation process. Details of particular features of the Lagrangian description were described in the implementation of the mesh generation context. Anisotropy in three dimensions was slightly considered to show the straightforward extensions of the method. Nevertheless, numerical problems for very large meshes may appear, and must be further investigated. Boundary meshing has not been tackled and constitutes also a natural extension of this work. Finally, the Delaunay based mesh generator would be easily extended to generate various hierarchical meshes for a multigrid application.

An iterative solver has been implemented based on the Preconditioned Conjugate Gradient Uzawa, and its tight relationships with the class of Fractional Step and Algebraic Splitting methods have been presented in details. The way to solve the monolithic system and to switch to the algebraic splitting is original, even if it resumes and unifies various results of the literature. The mass conservation has been greatly improved, and the role of the Fractional Step,

which is mainly responsible for the mass loss, has been discussed and illustrated through various examples.

At the numerical level, as the Lagrangian formulation does not include the convective term, no stabilisation method has been used. Furthermore, the absence of convective term brings the matrices symmetric, allowing a low storage requirement and iterative solvers with short term recurrences, minimization properties and uniform preconditioners. Finally, even if the Lagrangian description has been mainly used in free surface problems, it provides a very accurate convection and almost no diffusion in various other problems, as illustrated by the good agreement with various typical numerical Eulerian examples.

Free surface problems offer nevertheless a very appealing context for the Lagrangian description, as it provides an explicit description of the locus of the free surface, and no additional equation needs to be solved. Furthermore, the solution of the nonlinear mechanical problem is not perturbed by possible oscillations with elements that are fluid and then gaz. If the gaz effect is neglected, it allows to mesh only the domain filled by the fluid. Finally, boundary conditions are straightforward to impose on the free surface. Nevertheless, pinching and merging must be explicitely resolved.

The coupling of the mechanical solver with the thermal field was also presented. The Boussinesq approximation was considered to model natural flows, which produces a coupling with the velocity field. Solidification was introduced and partially validated with a reference example. The solidification process obviously deserves further investigation. Extension to the preconditioners used for the pressure Schur complement in this case must also be pursued, as the temperature dependant viscosity strenghtens the difficulty of approximating the Uzawa operator.

It must be noted that all this work was done in a strong numerical context, so that the physical models considered are the simplest one. The physics of jets and drops when waves break involves complicated phenomena as well as the dynamic of weting when the fluid slides on dry walls [2]. Surface tension was neither considered, nor turbulence effects which could be a very interesting theme in a Lagrangian formulation, as the Reynolds tensor comes mainly from the convective term, which is not present in a Lagrangian formulation. In the same way, the solidification process was simplified by interpolating linearly the mushy zone whereas very complicated anisotropic phenomena can occur during the solidification front advance [1]. The introduction of these physical phenomenas constitutes a possible future work.

Finally, the difficulties met during this work were of very different scales. The 3D meshing is mostly limited by the boundary recovery, which was one of the difficulties of the implementation part. The iterative monolithic solver with mass conservation was another difficulty. At the opposite, the introduction of the temperature, the Boussinesq approximation, and the convergence of the non linearity were a rather straightforward task. Obviously, the main difficulty is created by the convection of the mesh due to the large deformations.

Bibliography

[1] S.H. Davis. *Theory of Solidification*. Cambridge University Press, 2001.

[2] E. Guyon, J.P. Hulin, and L. Petit. *Hydrodynamique physique*. EDP Sciences, 2001.