UNIVERSITAT POLITÈCNICA
DE CATALUNYA

CIMNE

Master of Science in Computational Mechanics

Master Thesis:

DG-XFEM FORMULATION FOR THE UNSTEADY INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

Carles Estruch i Tena

Advisors:
Esther Sala-Lardies
Adeline de Villardi de Montlaur

Barcelona, June 2011

ACKNOWLEDGEMENTS

It is a pleasure to thank those who made this thesis possible.

First of all, I owe my deepest gratitude to my advisors Esther and Adeline. This thesis would not have been possible without their help, guidance and widespread availability to solve any aspect of this work.

Secondly, but not least, I would like to thank Pedro and Lelia for all their support and help given in the resolution of all the problems that have arisen in these two years of master.

I should also add that it has been an honour for me to do my internship in CIMNE Castelldefels under the supervision of Pere-Andreu, to whom I also want to thank for all what I have learned and for the space that has given to me to do my thesis.

Lastly, I would like to thank my colleagues, who are in the same situation, for all the great moments spent together, especially those who have nothing to do with academic issues and helped us to clear our minds and unwind from work.

ABSTRACT

This Master Thesis proposes a combined formulation of the Discontinuous Galerkin Method (DG) with solenoidal basis functions and the eXtended Finite Element Method (XFEM), in order to solve the incompressible Navier-Stokes equations for unsteady flows around a solid object, providing high orders of accuracy in space and time. This DG-XFEM formulation simplifies the meshing process using structured meshes that also do not need to be updated at every time step if the object moves, reducing the computational cost.

In the DG-XFEM formulation a fixed structured mesh is used and its elements are classified in three groups, which receive a different treatment. First, the elements inside the solid object are excluded in the calculations since it is treated as a void. Second, the elements belonging to the fluid are calculated as in the DG solenoidal formulation. Third, for the elements cut by the interface integration is modified using XFEM in order to take into account only the fluid region, considering curved integration cells to accurately compute integrals in high-order elements; straight afterwards it is solved again with the DG solenoidal formulation. In the DG solenoidal formulation incompressible flows are first solved for velocity and only part of the pressure's degrees of freedom (hybrid pressure), reducing the overall size of the system to be solved, while the rest of pressure degrees of freedom (interior pressure) is computed as a post-processing.

A numerical validation of the method is given with the simulation of the classical benchmark test of the flow past a cylinder, showing its good performance in several cases tested.

Contents

List of Figures **ii**

1 Introduction **1**
1.1 Objectives . . . 1
1.2 Discretization methods: state of the art . . . 1
1.2.1 Discontinuous Galerkin method . . . 2
1.2.2 Solenoidal basis functions . . . 2
1.2.3 eXtended Finite Element Method . . . 2
1.2.4 Rosenbrock time integration . . . 3
1.3 Overview of the thesis . . . 4

2 Problem Statement **5**
2.1 Previous definitions . . . 5
2.2 Governing equations . . . 6
2.3 DG formulation . . . 6
2.4 Solenoidal basis functions . . . 7
2.5 DG-XFEM formulation . . . 9
2.5.1 Interface tracking: level-set method . . . 9
2.5.2 Treatment of voids . . . 10
2.5.3 Integration for high-order elements . . . 11
2.6 Time discretization . . . 12

3 Numerical Validation **14**
3.1 Numerical code . . . 14
3.2 Benchmark test: flow past a cylinder . . . 15

4 Conclusions **21**
4.1 Summary and contributions . . . 21
4.2 Future research . . . 22
4.3 Applications . . . 23

Bibliography **24**

List of Figures

2.1 Discretized fluid domain with a solid object inside 10
2.2 Integration error committed with the standard linear approximation of the interface 11
2.3 Curved interface inside a fourth-order element and different ways to approximate it 12

3.1 Domain discretization with fourth-order elements. Comparison between DG and DG-XFEM formulations 15
3.2 Vortices comparison at low Reynolds numbers. Velocity vectors field with a fourth-order velocity approximation 16
3.3 Mean velocity field comparison at low Reynolds numbers with a fourth-order velocity approximation 17
3.4 Flow evolution in time. Mean velocity field at different time steps for $Re = 100$ and fourth-order velocity approximation . 18
3.5 Symmetric vortices. Velocity vectors field at $t = 3$ for $Re = 100$ and fourth-order velocity approximation 19
3.6 Evolution of the lift coefficient in time for $Re = 100$ with second-order velocity approximation and first-order for pressure 19
3.7 Flow evolution between periodic time steps. Velocity vectors field at different time steps for $Re = 100$ and fourth-order velocity approximation 20

Chapter 1

Introduction

1.1 Objectives

In order to determine the flow around an undeformable solid, a Discontinuous Galerkin (DG) formulation with solenoidal basis functions is proposed in [24, 26], which provides high orders of accuracy in space and time for unsteady incompressible flows. In this case, this is done using a computational mesh that covers the fluid domain, matching the material interfaces. However, following this approach, if the solid moves remeshing is needed.

To avoid this, we propose to consider a fixed mesh covering the whole domain (fluid and solid) and use the eXtended Finite Element Method (XFEM) to solve the problem. The introduction of XFEM [7] with a level-set method suppresses the need to mesh the discontinuity surfaces, adding more freedom in the meshing process. As there is no need to adapt the mesh around the object, structured meshes can be used. Moreover, in the case that the object moves, there will be no need to remesh at every time step.

The aim of this thesis is to derive a DG formulation with solenoidal approximations combined with XFEM in order to solve the incompressible Navier-Stokes equations for unsteady flows. The basic idea of this DG-XFEM formulation is to combine the good features of both methods to obtain better results than DG in terms of simplicity of mesh implementation and computational cost.

1.2 Discretization methods: state of the art

A brief explanation about the methods mentioned in section 1.1 is shown below. A short reference to the time integration is also given.

1.2.1 Discontinuous Galerkin method

The Discontinuous Galerkin method was first developed in 1973 by Reed and Hill [32], but it was not used for CFD simulations until the 90's by Cockburn and Shu [18]. The solution of the Navier-Stokes equations with the DG method was first accomplished by Bassi and Rebay [4] in 1997. As the method gained more attention in the CFD research community, further advances have come fairly rapidly (see for instance [35, 15, 17, 31]). Researchers now use the DG method to perform simulations of a wide variety of flow regimes. The method has been adapted to use it with compressible and incompressible, steady and unsteady, as well as laminar and turbulent conditions.

The DG method combines features of both Finite Element (FEM) [9] and Finite Volume Methods (FVM) [3]. The solution is represented within each element as a polynomial approximation (as in FEM), while the inter-element convection terms are resolved with upwinded numerical flux formulae (as in FVM). Theoretically, solutions may be obtained up to an arbitrarily high order of accuracy [14]. In addition to that, DG also permits the formulation of very compact numerical schemes. This is due to the fact that the solution representations in each element are purposefully kept independent of the solutions in other cells, with inter-element communication occurring only between adjacent cells (elements sharing a common face). This characteristic, along with other favourable numerical properties, makes this method extremely flexible (easily handling a wide variety of element types and mesh topologies) and also allows a number of adaptive techniques (both h- and p-refinement) and solver acceleration strategies to be implemented in a rather straightforward manner. The drawback of a DG formulation is in general its cost because of the duplication of degrees of freedom at the elements' boundaries. This overhead of degrees of freedom is becoming less significant for high-order approximations.

1.2.2 Solenoidal basis functions

Solenoidal basis functions are useful in the Stokes and Navier-Stokes problems with incompressibility conditions (see for instance [2, 25, 28]). Using solenoidal approximations in incompressible problems allows to split the velocity space into the direct sum of a solenoidal part and an irrotational part. Then the problem consists of an uncoupled problem, allowing to decrease the total number of degrees of freedom and, consequently, the computational cost.

1.2.3 eXtended Finite Element Method

XFEM was developed in 1999 by Belytschko and collaborators [6, 5] to help alleviate shortcomings of the Finite Element Method and is used to model

the propagation of discontinuities such as cracks or material interfaces. The idea behind the method is to retain most advantages of meshfree methods while minimizing their negative sides [22].

This method was developed to ease difficulties in solving problems with localized features that are not efficiently resolved by mesh refinement. One of the initial applications was the modelling of fractures in a material [6]. In this original implementation, bases that include crack opening displacements are obtained by means of adding —for the nodes belonging to elements cut by the interface— discontinuous basis functions to the standard polynomial shape functions. A key advantage of XFEM is that in such problems the finite element mesh does not need to be updated to track the crack path. Subsequent research, for example [13, 21, 36], has illustrated the more general use of the method for problems involving singularities, material interfaces, regular meshing of micro-structural features such as voids, and other problems where a localized feature can be described by an appropriate set of basis functions.

The principle of this method [7] is the enrichment of the polynomial approximation space of the classical Finite Element Method, so that it is able to naturally reproduce the discontinuities. The construction of the enrichment is frequently done with the level-set method, developed in 1988 by Osher and Sethian [30], since it simplifies the process of tracking interfaces' evolution. By means of this method, it is not only possible to determine where the enrichment is needed but it also facilitates its construction. The idea is to define interfaces implicitly by means of the zero-level of a scalar function within the domain.

Another key point in XFEM is the modification of the integration in the elements affected by the discontinuities. For the elements cut by the interface, integration rules are modified to be able to compute accurate integrals at its both sides. Usually, the interface is approximated by a linear function in each element. However, for high-order elements with curved interfaces there is the necessity of using also high-order integration methods that can be adapted to these shapes depending on the degree of interpolation, as it is presented in [11, 19].

1.2.4 Rosenbrock time integration

In the past, numerical techniques for flow simulations has focused mainly on steady state calculations due to computational costs. However, many physical phenomena are unsteady, creating the need for efficient numerical formulations for this kind of problems. A very well-known example of unsteady problem is the generation of the so-called von Kármán vortices in a flow past a cylinder.

In this thesis, a 4-stage ROSI2Pw method has been used, which belongs to the family of the Rosenbrock methods. As explained in [26], these

methods are derived from Singly Diagonally Implicit Runge-Kutta (SDIRK) methods, which are a special case of IRK methods where all the diagonal coefficients of the Butcher array are identical. Rosenbrock methods avoid the solution of nonlinear systems, reducing the computational cost. At each time step, velocity and pressure are updated using a similar formulation as in a standard Runge-Kutta (RK) method [27].

1.3 Overview of the thesis

The thesis is structured as follows:

Chapter 2 contains the problem statement, with a few previous definitions (2.1) followed by the presentation of the governing equations (2.2). Then the DG formulation (2.3) with solenoidal basis functions (2.4) is stated and combined afterwards with XFEM (2.5). Finally, the time discretization (2.6) is explained.

Chapter 3 gives a short explanation about the MATLAB code used in this thesis (3.1), followed by the numerical validation of the code by means of the classical benchmark test of the flow past a cylinder (3.2).

Chapter 4 presents the conclusions of the thesis with a summary of all the work done, highlighting its contributions to the existing formulation (4.1). Future research in this topic (4.2) and possible applications of the code (4.3) are also commented.

Chapter 2

Problem Statement

2.1 Previous definitions

Following [26] and [25], let's consider $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$ an open bounded domain in a space with $\mathtt{n_{sd}}$ spatial dimensions and with boundary $\partial\Omega$. Now divide this domain into $\mathtt{n_{el}}$ disjoint subdomains Ω_i,

$$\overline{\Omega} = \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \text{for } i \neq j$$

with piecewise linear boundaries $\partial\Omega_i$, which define an internal interface Γ

$$\Gamma := \Big[\bigcup_{i=1}^{\mathtt{n_{el}}} \partial\Omega_i\Big] \backslash \partial\Omega$$

The *jump* $[\![\cdot]\!]$ and *mean* $\{\cdot\}$ operators are defined along Γ using values from the elements of both sides of the interface, Ω_i and Ω_j (with exterior unit normals $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$), and also extended along the exterior boundary $\partial\Omega$:

$$[\![p\,\boldsymbol{n}]\!] = \begin{cases} p_i\,\boldsymbol{n}_i + p_j\,\boldsymbol{n}_j = \boldsymbol{n}_i(p_i - p_j) & \text{on } \Gamma \\ p\,\boldsymbol{n} & \text{on } \partial\Omega \end{cases} \tag{2.1}$$

$$\{p\} = \begin{cases} 1/2p_i + 1/2p_j & \text{on } \Gamma \\ p & \text{on } \partial\Omega \end{cases} \tag{2.2}$$

for p a scalar.

In some equations, $(\cdot,\cdot)$ will denote the $\mathcal{L}_2$ scalar product in Ω, that is for scalars

$$(p,q) = \int_\Omega p\,q\,d\Omega \tag{2.3}$$

Analogously, $(\cdot,\cdot)_\Upsilon$ denotes the $\mathcal{L}_2$ scalar product in any domain $\Upsilon \subset \Gamma \cup \partial\Omega$. For instance,

$$(p,q)_\Upsilon = \int_\Upsilon p\,q\,d\Gamma \tag{2.4}$$

In equations (2.1), (2.2), (2.3) and (2.4), see [25] for the case with vectors and tensors.

2.2 Governing equations

The equations of the model to be solved are the Navier-Stokes equations for unsteady incompressible flows, which in the strong form read as follows[1]

$$\frac{\partial \boldsymbol{u}}{\partial t} - 2\boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{u}) + \boldsymbol{\nabla}p + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = \boldsymbol{f} \qquad \text{in } \Omega\times]0,T[\tag{2.5a}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega\times]0,T[\tag{2.5b}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D\times]0,T[\tag{2.5c}$$

$$-p\boldsymbol{n} + 2\nu(\boldsymbol{n}\cdot\boldsymbol{\nabla}^{\mathtt{S}})\boldsymbol{u} = \boldsymbol{t} \qquad \text{on } \Gamma_N\times]0,T[\tag{2.5d}$$

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}) \qquad \text{in } \Omega \tag{2.5e}$$

where $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\boldsymbol{f} \in \mathcal{L}_2(\Omega)$ is the source term, $\boldsymbol{t}$ the boundary tractions, $\boldsymbol{u}$ the flux velocity, p its pressure, ν the kinematic viscosity and $\boldsymbol{\nabla}^{\mathtt{S}} = \frac{1}{2}(\boldsymbol{\nabla} + \boldsymbol{\nabla}^T)$.

2.3 DG formulation

The problem (2.5) can be discretized following a Discontinuous Galerkin Interior Penalty Method (DG-IPM) as it is done in [28] for the steady Navier-Stokes equations and extended afterwards for an unsteady formulation in [26].

Let's introduce the discrete finite element spaces

$$\boldsymbol{\mathcal{V}}^h = \{\boldsymbol{v} \in [\mathcal{L}_2(\Omega)]^{\mathtt{n_{sd}}} \;;\; \boldsymbol{v}|_{\Omega_i} \in [\mathcal{P}^k(\Omega_i)]^{\mathtt{n_{sd}}} \quad \forall\Omega_i\}$$
$$\mathcal{Q}^h = \{q \in [\mathcal{L}_2(\Omega)] \;;\; q|_{\Omega_i} \in [\mathcal{P}^{k-1}(\Omega_i)] \quad \forall\Omega_i\}$$

where $\mathcal{P}^k(\Omega_i)$ is the space of polynomial functions of degree at most $k \geq 1$ in Ω_i.

Then the problem to solve is to find $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}^h\times]0,T[$ and $p_h \in \mathcal{Q}^h\times]0,T[$ such that $\forall\, \boldsymbol{v} \in \boldsymbol{\mathcal{V}}^h$, $\forall\, q \in \mathcal{Q}^h$ and $\forall\, t \in]0,T[$

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) + a\big(\boldsymbol{u}_h, \boldsymbol{v}\big) + c\big(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}\big) \\ \qquad\qquad + b\big(\boldsymbol{v}, p_h\big) + \big(\{p_h\}, [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!]\big)_{\Gamma\cup\Gamma_D} = l\big(\boldsymbol{v}\big) \\ b\big(\boldsymbol{u}_h, q\big) + \big(\{q\}, [\![\boldsymbol{n}\cdot\boldsymbol{u}_h]\!]\big)_{\Gamma\cup\Gamma_D} = (q, \boldsymbol{n}\cdot\boldsymbol{u}_D)_{\Gamma_D} \end{cases} \tag{2.6}$$

[1] Notice that, in (2.5a), the constant density has been absorbed into the pressure. In (2.5e), the initial velocity field $\boldsymbol{u}_0$ is assumed to be solenoidal, as will be seen in section 2.4.

where

$$a\big(\boldsymbol{u},\boldsymbol{v}\big) := \big(2\nu\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{u},\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{v}\big) + C_{11}\left(\llbracket\boldsymbol{n}\otimes\boldsymbol{u}\rrbracket,\llbracket\boldsymbol{n}\otimes\boldsymbol{v}\rrbracket\right)_{\Gamma\cup\Gamma_D} - \big(2\nu\{\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{u}\},\llbracket\boldsymbol{n}\otimes\boldsymbol{v}\rrbracket\big)_{\Gamma\cup\Gamma_D} - \big(\llbracket\boldsymbol{n}\otimes\boldsymbol{u}\rrbracket,2\nu\{\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{v}\}\big)_{\Gamma\cup\Gamma_D}, \quad (2.7a)$$

$$c\big(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}\big) := \frac{1}{2}\Big[-((\boldsymbol{w}\cdot\boldsymbol{\nabla})\boldsymbol{v},\boldsymbol{u}) + ((\boldsymbol{w}\cdot\boldsymbol{\nabla})\boldsymbol{u},\boldsymbol{v}) + \sum_{i=1}^{\mathtt{n_{el}}}\int_{\partial\Omega_i\setminus\Gamma_N}\frac{1}{2}\left[(\boldsymbol{w}\cdot\boldsymbol{n}_i)(\boldsymbol{u}^{ext}+\boldsymbol{u}) - |\boldsymbol{w}\cdot\boldsymbol{n}_i|\left(\boldsymbol{u}^{ext}-\boldsymbol{u}\right)\right]\cdot\boldsymbol{v}d\Gamma + \int_{\Gamma_N}(\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{u}\cdot\boldsymbol{v}d\Gamma\Big], \quad (2.7b)$$

$$b\big(\boldsymbol{v},p\big) := -\int_{\Omega} q\,\boldsymbol{\nabla}\cdot\boldsymbol{v}\,d\Omega \quad (2.7c)$$

and

$$l\big(\boldsymbol{v}\big) := (\boldsymbol{f},\boldsymbol{v}) + (\boldsymbol{t},\boldsymbol{v})_{\Gamma_N} + C_{11}\left(\boldsymbol{u}_D,\boldsymbol{v}\right)_{\Gamma_D} - \big(\boldsymbol{n}\otimes\boldsymbol{u}_D,2\nu\boldsymbol{\nabla}^{\mathtt{S}}\boldsymbol{v}\big)_{\Gamma_D}. \quad (2.7d)$$

C_{11} is the penalty parameter and $\boldsymbol{u}^{ext}$ denotes the exterior trace of $\boldsymbol{u}$ taken over the side/face under consideration (for further details see [25]).

2.4 Solenoidal basis functions

Taking as a reference [25, 16, 10], solenoidal approximations can be introduced in the problem by splitting the velocity space $\boldsymbol{\mathcal{V}}^h$ into direct sum of a solenoidal part and an irrotational part $\boldsymbol{\mathcal{V}}^h = \boldsymbol{\mathcal{S}}^h \oplus \boldsymbol{\mathcal{I}}^h$, where

$$\boldsymbol{\mathcal{S}}^h = \big\{\boldsymbol{v}\in[\mathcal{H}^1(\Omega)]^{\mathtt{n_{sd}}} \mid \boldsymbol{v}|_{\Omega_i}\in[\mathcal{P}^k(\Omega_i)]^{\mathtt{n_{sd}}},\ \boldsymbol{\nabla}\cdot\boldsymbol{v}|_{\Omega_i}=0 \text{ for } i=1,\ldots,\mathtt{n_{el}}\big\}$$
$$\boldsymbol{\mathcal{I}}^h \subset \big\{\boldsymbol{v}\in[\mathcal{H}^1(\Omega)]^{\mathtt{n_{sd}}} \mid \boldsymbol{v}|_{\Omega_i}\in[\mathcal{P}^k(\Omega_i)]^{\mathtt{n_{sd}}},\ \boldsymbol{\nabla}\times\boldsymbol{v}|_{\Omega_i}=\boldsymbol{0} \text{ for } i=1,\ldots,\mathtt{n_{el}}\big\}$$

The solenoidal basis in an element Ω_i can be expressed as

$$\boldsymbol{\mathcal{S}}^h = \langle\ \phi_k^i\ \rangle_{k=1}^{\mathtt{n_{bfu}}} \quad (2.8)$$

where ϕ_k^i are the solenoidal vector bases, with $\mathtt{n_{bfu}}$ denoting the number of basis functions for the interpolation in that element.

As an example, in the case of a 2D triangle with a second-order approximation, $\mathtt{n_{bfu}} = 9$ and the solenoidal basis is

$$\boldsymbol{\mathcal{S}}^h = \left\langle \begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\x\end{pmatrix},\begin{pmatrix}x\\-y\end{pmatrix},\begin{pmatrix}y\\0\end{pmatrix},\begin{pmatrix}0\\x^2\end{pmatrix},\begin{pmatrix}2xy\\-y^2\end{pmatrix},\begin{pmatrix}x^2\\-2xy\end{pmatrix},\begin{pmatrix}y^2\\0\end{pmatrix}\right\rangle$$

with the irrotational complementary part being

$$\boldsymbol{\mathcal{I}}^h = \left\langle \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} x^2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ y^2 \end{pmatrix} \right\rangle$$

See for instance [2] for the construction of these spaces.

Under these circumstances, the problem (2.6) can be split in *two uncoupled problems*:

- The first one solves for *divergence-free velocities* and *hybrid pressures*:
  Find $\boldsymbol{u}_h \in \boldsymbol{\mathcal{S}}^h \times]0,T[$ and $\tilde{p}_h \in \boldsymbol{P}^h \times]0,T[$ solution of[2]

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) + a_{\mathtt{IP}}(\boldsymbol{u}_h, \boldsymbol{v}) + c(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}) \\ \qquad\qquad + (\tilde{p}_h, [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} = l_{\mathtt{IP}}(\boldsymbol{v}) \\ (\tilde{q}, [\![\boldsymbol{n}\cdot\boldsymbol{u}_h]\!])_{\Gamma\cup\Gamma_D} = (\tilde{q}, \boldsymbol{n}\cdot\boldsymbol{u}_D)_{\Gamma_D} \end{cases} \tag{2.9}$$

  $\forall \boldsymbol{v} \in \boldsymbol{\mathcal{S}}^h$, $\forall \tilde{q} \in \boldsymbol{P}^h$, $\forall\ t \in]0,T[$, with the forms defined in (2.7). The space of hybrid pressures (pressures along the sides) is:

$$\boldsymbol{P}^h := \left\{ \tilde{p} \mid \tilde{p} : \Gamma \cup \Gamma_D \longrightarrow \mathbb{R} \ \text{ and } \ \tilde{p} = [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!] \ \text{ for some } \ \boldsymbol{v} \in \boldsymbol{\mathcal{S}}^h \right\}$$

- The second problem, which requires the solution of the previous one, evaluates *interior pressures*:
  Find $p_h \in \mathcal{Q}^h \times]0,T[$ such that[3] $\forall \boldsymbol{v} \in \boldsymbol{\mathcal{I}}^h$ and $\forall t \in]0,T[$

$$\begin{aligned} b(\boldsymbol{v}, p_h) = l_{\mathtt{IP}}(\boldsymbol{v}) - \left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\right) - a_{\mathtt{IP}}(\boldsymbol{u}_h, \boldsymbol{v}) \\ - (\tilde{p}_h, [\![\boldsymbol{n}\cdot\boldsymbol{v}]\!])_{\Gamma\cup\Gamma_D} - c(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}) \end{aligned} \tag{2.10}$$

Thus, the discretized solution of the velocity and hybrid pressure obtained with the DG solenoidal formulation reads as follows:

$$\boldsymbol{u}_h(\boldsymbol{x}, t) = \sum_{k=1}^{\mathtt{n_{bfu}}} \phi_k^i(\boldsymbol{x}) \boldsymbol{u}_k^i(t) \qquad \text{for each element } \Omega_i \tag{2.11a}$$

$$\tilde{p}_h(\boldsymbol{x}, t) = \sum_{j=1}^{\mathtt{n_{bfp}}} \tilde{N}_j^e(\boldsymbol{x}) \tilde{p}_j^e(t) \qquad \text{on each side } \Gamma_e \tag{2.11b}$$

where $\tilde{N}_j^e$ are the shape functions for the hybrid pressure and $\mathtt{n_{bfp}}$ are the number of shape functions for the interpolation of the hybrid pressure.

[2] Observe that this problem, which has to be solved at each time step, shows an important reduction in the number of degrees of freedom with respect to the problem (2.6), as explained in [28].

[3] It is important to note that equation (2.10) can be solved element by element and pressure is its only unknown.

2.5 DG-XFEM formulation

In this section, the XFEM formulation is combined with the DG solenoidal approximation presented in sections 2.3 and 2.4 in order to obtain the flow around an undeformable solid with a simpler mesh that also does not need to be updated in time. First, the level-set method —usually related with XFEM to define the interface— is explained in section 2.5.1. Second, the treatment of the solid as a void is stated in section 2.5.2. Finally, in section 2.5.3, the modification of the integration in the elements affected by the interface is presented.

2.5.1 Interface tracking: level-set method

Considering that our domain of study Ω is a fluid with a solid material inside, let's name these two regions as Ω_f and Ω_s respectively, such that $\Omega = \Omega_f \cup \Omega_s$ and $\bar{\Omega}_f \cap \bar{\Omega}_s = \Gamma_s$. A level-set function [7] is any continuous function $\varphi(\boldsymbol{x}), \boldsymbol{x} \in \Omega$, that is negative in one subdomain and positive in the other, while the zero-level is the position of the closed interface Γ_s. The *signed distance function* [29] is a particularly useful level-set function, defined as

$$\varphi(\boldsymbol{x}) = \pm \min_{\boldsymbol{x}^* \in \Gamma_s} \|\boldsymbol{x} - \boldsymbol{x}^*\|, \quad \forall \boldsymbol{x} \in \Omega \tag{2.12}$$

where $\|\cdot\|$ denotes the Euclidean norm. The sign is different on the two sides of the interface; in our case:

$$\varphi(\boldsymbol{x}) \begin{cases} > 0 & if\ \boldsymbol{x} \in \Omega_f \\ = 0 & if\ \boldsymbol{x} \in \Gamma_s \\ < 0 & if\ \boldsymbol{x} \in \Omega_s \end{cases}$$

For a discretized domain, the values of the level-set function are stored at the nodes $\varphi_i = \varphi(\boldsymbol{x}_i)$, and the function is interpolated as

$$\varphi^h(\boldsymbol{x}) = \sum_{i \in I} N_i(\boldsymbol{x}) \varphi_i \tag{2.13}$$

using standard FE shape functions $N_i(\boldsymbol{x})$ as interpolation functions, where I is the set of all nodes in Ω. The representation of the discontinuity as the zero-level of $\varphi^h(\boldsymbol{x})$ is only an approximation of the real position, which improves with mesh refinement.

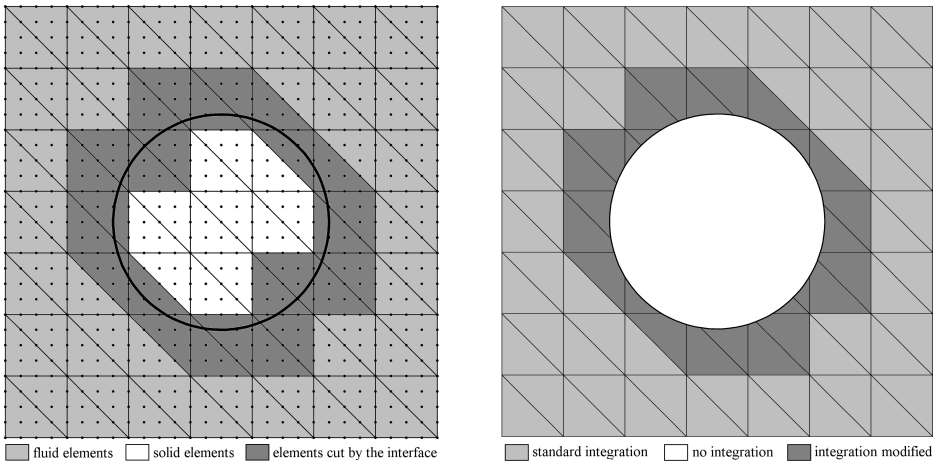
If the interface Γ_s moves during the simulation, as it is commented in section 4.2, $\varphi(\boldsymbol{x})$ is also a function of time $\varphi(\boldsymbol{x}, t)$ and the level-set function needs to be updated at each time-step. If the solid moves due to the inflow velocity, a simple transport equation is solved to update the level-set.

2.5.2 Treatment of voids

If we consider now our domain $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$ discretized with a structured mesh of $\mathtt{n_{el}}$ elements and the solid object placed in its interior, it is clear that there will be some elements cut by its interface Γ_s. In our case, the treatment of this problem differs a little from the general XFEM formulation.

In a general XFEM formulation, the interpolation would need to be enriched for velocity and hybrid pressure in the nodes of the elements affected by the interface, being expressed as the summation of the DG solenoidal part from equation (2.11) and another part containing the *enrichment function* $\psi(\boldsymbol{x})$. This function incorporates information about discontinuities and other singularities into the approximation space. In this case, this function separates the fluid (where all the calculations are done) and the solid (in which no calculations are considered), and its expression is given by Sukumar et al. in [34]:

$$\psi(\boldsymbol{x}) = \begin{cases} 1 & if\ \boldsymbol{x} \in \Omega_f \\ 0 & if\ \boldsymbol{x} \in \Omega_s \end{cases} \tag{2.14}$$



(a) Elements classification (b) Integration cells

Figure 2.1: Discretized fluid domain with a solid object inside

In this work, since the solid part is not taken into account in the calculations, it is treated as a void. Thus, we will not enrich the interpolation but modify the integration to consider only the fluid region in the elements cut by the interface. According to this, the elements can be classified into different categories, as shown in figure 2.1a:

(i) *Fluid elements*: standard integration is done following the DG solenoidal formulation presented in sections 2.3 and 2.4.

(ii) *Solid elements*: the degrees of freedom associated with their nodes are removed from the system of equations.

(iii) *Elements cut by the interface*: integration is done only in the region belonging to the fluid part, as explained next in section 2.5.3.

2.5.3 Integration for high-order elements

As mentioned in section 2.5.2, a modification of the integration is required in the elements cut by the interface (figure 2.1b shows the different integration cells inside the elements). A standard approach is to use a linear approximation of the interface inside each element and integrate only in the area containing the fluid (figure 2.3b). For high-order elements, as it is our case, this is not enough. Figure 2.2 shows the integration error committed with this approach.

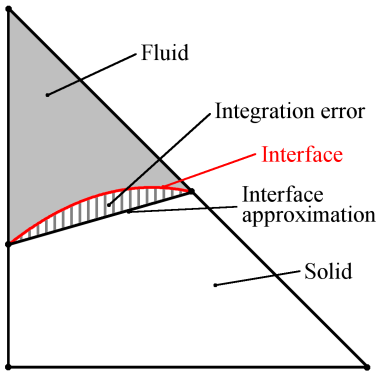
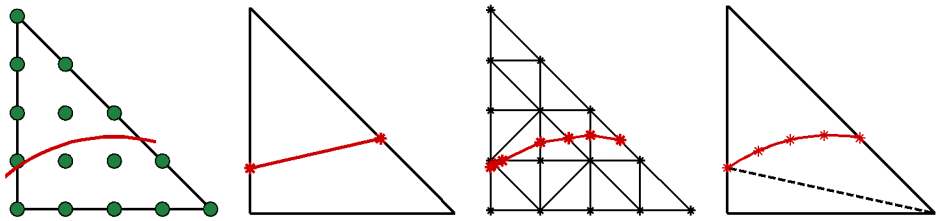


Figure 2.2: Integration error committed with the standard linear approximation of the interface

To improve the approximation of the interface, there are several options, as shown in figure 2.3:

- Möes et al. [19] consider a *piecewise linear approximation* inside each element, which consists of the linear approximation of the interface commented before, but with recursive refinements within each element (figure 2.3c). The problem is that the level of refinement depends on the size of the element. In fine meshes, the approximation of the interface also needs to be refined in each element.

- Cheng and Fries [11] consider the possible curvature of the interface by using curved integration cells with a *p-th degree approximation* (2.3d), where p is the order of the polynomials used to approximate the interface, which is the same as the degree of the elements.



(a) Real interface (b) Standard linear (c) Piecewise linear (d) p-th degree

Figure 2.3: Curved interface inside a fourth-order element and different ways to approximate it

In this thesis, the strategy used is similar to the one proposed in [11], but taking into account that the interface inside high-order elements can be quite complex. This consideration means that, if necessary, the elements can be subdivided in order to better approximate the interface. After that, integration is done in the cells belonging to the fluid.

The points defining the p-th degree polynomial inside an element are not only used to define the integration cells, but they are also used as the nodes of a one-dimensional mesh needed to compute contributions along the interface. Note that this good approximation of the interface not only improves integration results inside the element, but also affects the imposition of the boundary conditions.

2.6 Time discretization

The spatial discretization using the solenoidal DG-XFEM formulation of the unsteady incompressible Navier-Stokes problem (2.5) can be written as

$$\begin{cases} \mathbf{M}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} + \mathbf{C}(\mathbf{u})\mathbf{u} + \mathbf{G}\mathbf{p} = \mathbf{f_1} \\ \mathbf{G}^T\mathbf{u} = \mathbf{f_2} \end{cases} \tag{2.15}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ the diffusion matrix, $\mathbf{C}$ the convection matrix, $\mathbf{G}$ the discrete gradient/divergence matrix, $\mathbf{u}$ and $\mathbf{p}$ the vectors of nodal values, $\dot{\mathbf{u}}$ denotes the time derivative, and $\mathbf{f_1}$ and $\mathbf{f_2}$ vectors taking into account force term and boundary conditions. This system of $\mathtt{n_{dof}}$ degrees of freedom can also be written as

$$\begin{cases} \mathbf{M}\dot{\mathbf{u}} = \mathcal{F}(t, \mathbf{u}, \mathbf{p}) \\ 0 = \mathcal{G}(t, \mathbf{u}) \end{cases} \tag{2.16}$$

with $t \in]0, T[$ and where

$$\begin{aligned} \mathcal{F}(t, \mathbf{u}, \mathbf{p}) &= \mathbf{f_1} - \mathbf{K}\mathbf{u} - \mathbf{C}(\mathbf{u})\mathbf{u} - \mathbf{G}\mathbf{p}, \\ \mathcal{G}(t, \mathbf{u}) &= \mathbf{G}^T\mathbf{u} - \mathbf{f_2}. \end{aligned} \tag{2.17}$$

The system of equations (2.16) is stated as a system of Differential Algebraic Equations (DAEs), corresponding to the conservation of momentum equation plus the incompressibility constraint. Navier-Stokes equations are an index-2 DAE system. For this reason, following [26], the time discretization is done with a 4-stage ROSI2Pw method, which belongs to the family of the Rosenbrock methods. At each time step, $\mathbf{u}$ and $\mathbf{p}$ are updated as in a standard s-stage Runge-Kutta method, which for the system of equations (2.16) reads

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{l}_i \\ \mathbf{p}^{n+1} &= \mathbf{p}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{k}_i \end{aligned} \tag{2.18}$$

where $\mathbf{l}_i$ and $\mathbf{k}_i$ are the solution of a linearized system of equations. For further details of this time integration method, see [26].

Chapter 3

Numerical Validation

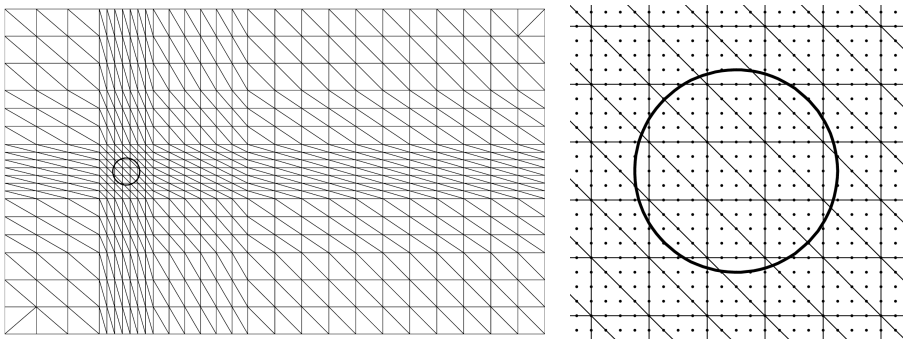
3.1 Numerical code

The results shown in this section have been obtained with a numerical code implemented in MATLAB. Originally, this code was developed by Montlaur in [24], with subsequent improvements in [26], to solve problems involving incompressible flows using a DG formulation with solenoidal approximations, as explained in sections 2.3, 2.4 and 2.6. In this thesis, the code has been adapted to solve the incompressible Navier-Stokes equations for unsteady flows with this formulation, adding the XFEM formulation for voids shown in section 2.5. Conceptually, the main modifications introduced in the code are:

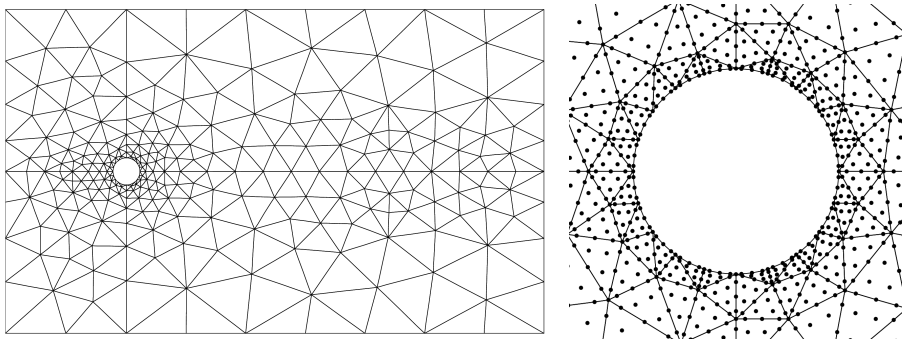
(i) Substitution of the existing non-structured mesh —constrained to be adapted around the solid object— by a structured one without this requirement. To better appreciate the differences, figure 3.1 shows the mesh used in this thesis compared with the one used in the case of having only a DG formulation. The interface is treated with the level-set method, seen in section 2.5.1.

(ii) Detection of all the cut points between the mesh and the object's interface and classification of the elements and edges depending if they are completely inside the object, outside, or cut by the interface (see figure 2.1a). Definition of the object's faces in each element cut by the interface.

(iii) Modification of the quadrature (2D or 1D) in the elements and edges cut by the interface (see figure 2.1b). The elements and edges inside the object are not taken into account in the calculations, while the ones belonging to the fluid are treated like before in the original code (for more details see section 2.5.3). The object's faces are treated as exterior faces of the domain, without modifying the quadrature.



(a) Whole domain of study (DG-XFEM) (b) Close-up (DG-XFEM)



(c) Whole domain of study (DG) (d) Close-up (DG)

Figure 3.1: Domain discretization with fourth-order elements. Comparison between DG and DG-XFEM formulations

In order to check this code, it has been applied to the solution of a well-known benchmark test. It is the classical problem of the flow past a cylinder, shown in section 3.2.

3.2 Benchmark test: flow past a cylinder

The simulation of the vortices behind a fixed circular cylinder is a popular benchmark problem to test the performance of numerical algorithms for solving the unsteady Navier-Stokes equations. The simplicity of this problem is that its characteristics are determined solely by the Reynolds number, which is defined by

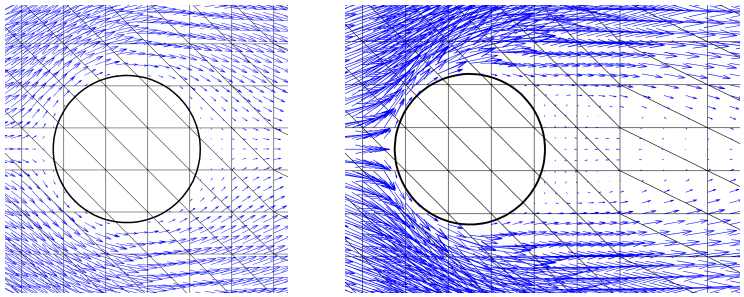
$$Re = \frac{u_\infty D}{\nu} \tag{3.1}$$

where u_∞ is the inflow velocity, D is the diameter of the cylinder and ν the kinematic viscosity. Herein, it is established a constant velocity $u_\infty = 1$.

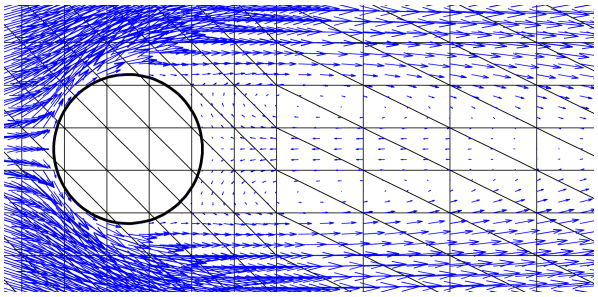
The domain of simulation is a rectangle of dimension 20x12 with the

cylinder of diameter $D = 1$ centered vertically and separated by a horizontal distance of 4.5 from the left side, which is the inflow. Dirichlet boundary conditions are imposed on the inflow, $\boldsymbol{u}_D = (u_\infty, 0)$, and circle, $\boldsymbol{u}_D = (0, 0)$; while null Neumann conditions are set on the outlet, top and bottom of the domain. Initial conditions are prescribed in all the domain with a velocity field $\boldsymbol{u}_0 = (u_\infty, 0)$, except on the cylinder, where $\boldsymbol{u}_0 = (0, 0)$.

The surface of the domain is discretized with a structured mesh of 1026 fourth-order elements[1], whose sizes are set by regions, having smaller elements near the cylinder to better capture the behaviour of the flow and also behind it, where the wakes will appear (see figures 3.1a and 3.1b). For velocity, fourth order solenoidal approximations are used; for pressure, standard approximations of third order.



(a) $Re = 1$ (b) $Re = 20$



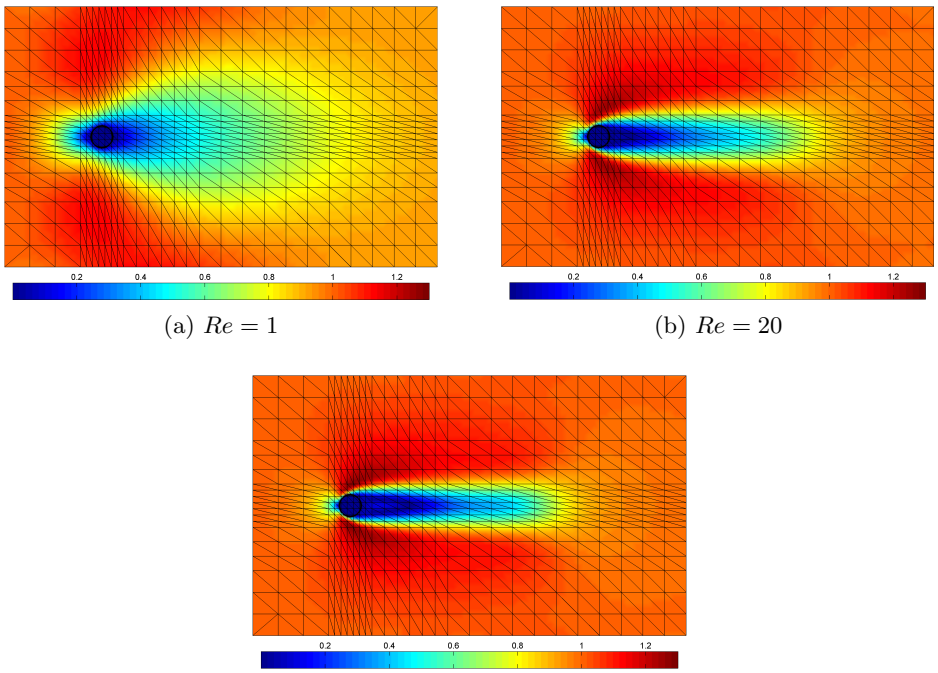
(c) $Re = 40$

Figure 3.2: Vortices comparison at low Reynolds numbers. Velocity vectors field with a fourth-order velocity approximation

Analyzing the flow patterns, it has been checked that for very low Reynolds numbers ($Re = 1 \div 10$) the flow is symmetric and steady, surrounding the cylinder without generating any eddies behind it (figure 3.2a). Increasing

[1]Recall that the addition of the XFEM features is done also with high-order interpolation methods, which allow to follow curved edges —like the cylinder boundary— and achieve also high-order integration in the elements affected by the interface, as explained in section 2.5.3.

the value until $Re = 20$, small vortices appear behind the cylinder (figure 3.2b). As exposed in [20], at Reynolds numbers around $Re = 40$, the steady solution for the flow starts to be unsteady and the vortices become wider, separating from the cylinder (figure 3.2c). Figure 3.3 also shows the mean velocity field for these cases. For higher Reynolds numbers an unsteady solution with periodic behaviour appears.

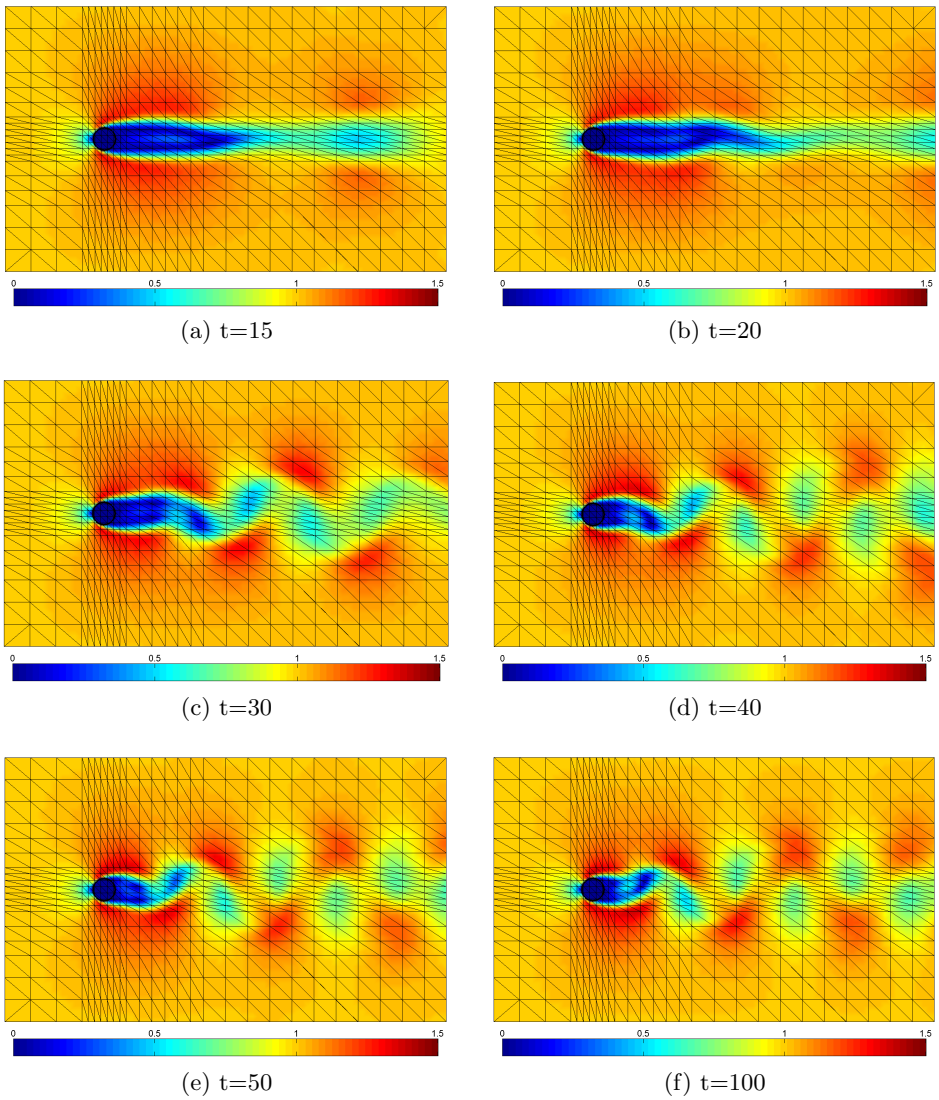


(a) $Re = 1$ (b) $Re = 20$

(c) $Re = 40$

Figure 3.3: Mean velocity field comparison at low Reynolds numbers with a fourth-order velocity approximation

In this thesis we have focused on the analysis of the flow for $Re = 100$ using a time step $\Delta t = 0.1$ in a time interval $[0, 100]$. Figure 3.4 shows the evolution of the flow during this time lapse. Starting from a non-physical situation determined by the initial conditions set before, the solution reaches a steady state condition as the time integration proceeds, with a symmetric pair of eddies appearing behind the cylinder in the first time steps, as shown in figure 3.5 (similar to the behaviour at low Reynolds numbers shown in figure 3.2). Between $t = 15$ and $t = 20$ we can see that the flow begins to be non-symmetric and at $t = 30$ the oscillation of the flow is remarkable. Around $t = 40, 50$ the instability of the eddies leads to a periodic vortex shedding from this point onwards, reaching the steady state of this periodic solution.



(a) t=15 (b) t=20

(c) t=30 (d) t=40

(e) t=50 (f) t=100

Figure 3.4: Flow evolution in time. Mean velocity field at different time steps for $Re = 100$ and fourth-order velocity approximation

The vortices, known as Von Kármán vortices, detach from the top and the bottom of the cylinder with associated oscillation in the lift forces, as shown in figure 3.6. The lift coefficient is calculated as,

$$C_L = \int_0^{2\pi} \boldsymbol{\tau}_y d\boldsymbol{\theta} \tag{3.2}$$

where $\boldsymbol{\tau}_y$ is the y-component of the normal component of the Cauchy stress tensor $\boldsymbol{\tau} = -p\boldsymbol{n} + 2\nu(\boldsymbol{n} \cdot \boldsymbol{\nabla}^{\mathbf{S}})\boldsymbol{u}$.

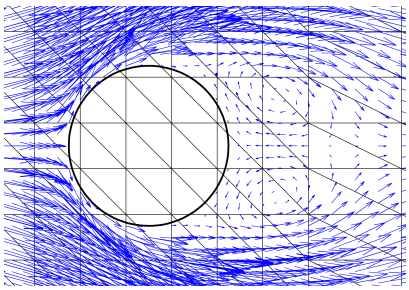


Figure 3.5: Symmetric vortices. Velocity vectors field at $t = 3$ for $Re = 100$ and fourth-order velocity approximation

From figure 3.6, the period of oscillation can be measured, obtaining an approximated value of $T = 5.9$. To observe the vortices and its periodicity, figure 3.7 shows the velocity vectors at different time steps, corresponding to consecutive instants with the vortices at the top and at the bottom of the cylinder. For example, figures 3.7a and 3.7c show the same vortex at the top of the cylinder at two different moments separated by a time lapse approximately equal to the period T (same happens with figures 3.7b and 3.7d).

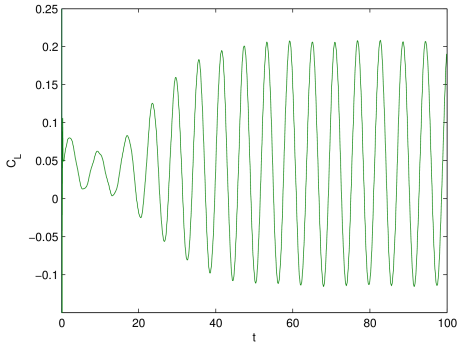
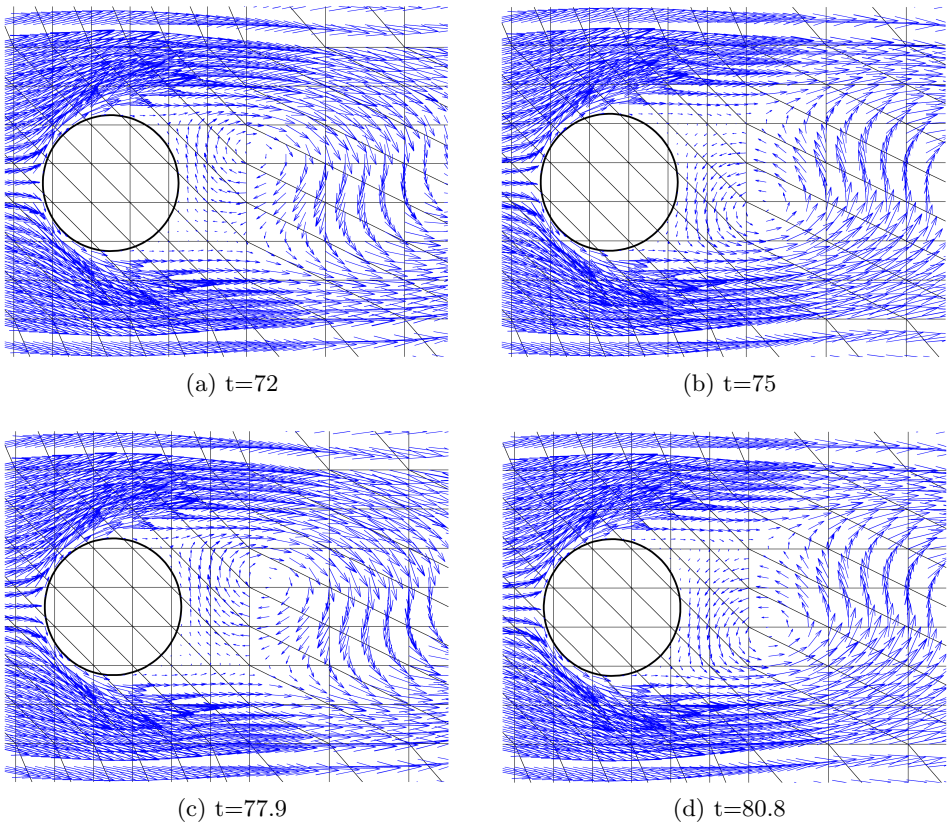


Figure 3.6: Evolution of the lift coefficient in time for $Re = 100$ with second-order velocity approximation and first-order for pressure

A usual way to check the results in a periodic problem like this one is considering the Strouhal number, which is a dimensionless number describing oscillating flow patterns. The definition is given by

$$S = \frac{fD}{u_\infty} \tag{3.3}$$

where f is the frequency of the vortex shedding, D is the characteristic length of the object (in this case the cylinder's diameter) and u_∞ is the velocity of the fluid.



(a) t=72 (b) t=75

(c) t=77.9 (d) t=80.8

Figure 3.7: Flow evolution between periodic time steps. Velocity vectors field at different time steps for $Re = 100$ and fourth-order velocity approximation

Applying equation (3.3) to the results obtained in the simulation we get a value of $S = 0.169$, which can be compared to the experimental results of Roshko in [33]. He found an empirical relation between the Reynolds and the Strouhal numbers that reads as follows:

$$S = \begin{cases} 0.212(1 - 21.2/Re) & \text{for } 50 < Re < 150 \\ 0.212(1 - 12.7/Re) & \text{for } 300 < Re < 2000 \end{cases} \tag{3.4}$$

In our case $Re = 100$, so the Strouhal number is $S = 0.167$, which is quite similar to the one we have obtained.

Having a look at the figures obtained in this section, it is clear that the behaviour seems to be the appropriate for this kind of flows. In any case, these have been compared with the ones presented by Montlaur in [24] and it has come to the conclusion that they seem to be similar.

Chapter 4

Conclusions

4.1 Summary and contributions

In this thesis, a DG-XFEM formulation with solenoidal basis functions has been proposed to solve the incompressible Navier-Stokes equations for unsteady flows around an object, allowing the use of high-order elements and integration schemes.

First, the problem is discretized following a Discontinuous Galerkin Interior Penaly Method (DG-IPM).

Second, the addition of the solenoidal basis functions allow to split the problem in two uncoupled problems, where the velocity and hybrid pressure can be obtained solving a reduced system of equations, while the interior pressure is computed in a post-process.

Third, the eXtended Finite Element Method (XFEM) is introduced making possible to solve the problem with a fixed structured mesh that covers the whole domain (both fluid and solid regions), which also does not need to be updated remeshing at every time step in the case that the object moves. The interface is defined using the level-set method and three kind of elements are set: fluid, solid and cut by the interface. Inside the fluid elements, the DG solenoidal formulation is employed. The solid is treated as a void, so the degrees of freedom associated with the nodes belonging to elements completely inside the solid are removed from the problem. Integration is modified in the elements affected by the interface in order to calculate only in their fluid region. Quadratures are modified in these elements to take into account that the interpolation is of high order, considering curved integration cells by means of approximating the interface with a polynomial of same degree as the elements and subdividing them, if necessary, depending on the complexity of the interface. The points defining the p-th degree polynomial are also used as a one-dimensional mesh to compute the contributions along the interface.

Finally, the system to be solved can be interpreted as a system of Dif-

ferential Algebraic Equations (DAE), which is discretized in time using a Rosenbrock method. This method works well for this high-order problem and the computational cost is lower than with other methods because it involves the solution of a linearized system of equations.

To solve this problem, a MATLAB code has been implemented and validated. To check it, the classical benchmark test of the flow past a cylinder has been applied. A structured mesh of fourth-order elements has been used to discretize the domain. Velocity and pressure have been calculated with fourth and third order approximations, respectively. Finally, the Strouhal number has been used as the reference parameter to compare the results obtained in this thesis with previous ones found in the literature, indicating the goodness of the code.

To sum up, the contribution of this thesis to the existing DG solenoidal formulation proposed in [26] is the addition of the XFEM features to be able to reproduce discontinuities inside elements without the requirement to adapt the mesh to the outline of the object. This DG-XFEM solenoidal formulation allows to obtain some significant improvements:

- Structured meshes can be used, simplifying the meshing process.
- There is no need to remesh at every time step in the case that the object moves or deforms, with the consequent reduction of computational cost.

4.2 Future research

The next stage to reach in the development of the MATLAB code is to make possible to move the cylinder. In [8] there is an excellent reference about the study of the flow past an oscillating cylinder (perpendicularly with respect to the flow direction). Actually, this work has already begun but it has not been finished. The key point in this modelling is how to transfer information of velocity and pressure between consecutive time steps. Since the interface moves during the simulation, the geometrical features and the degrees of freedom change at each step. The main difficulty in the code implementation is taking into account that a certain element initially belonging to the fluid domain can be cut by the interface at one step and in the next one become part of the solid and be excluded from the calculations, but after can be involved again when becoming a fluid element.

The current MATLAB code is able to consider fluxes around a solid object placed inside a fluid. Another possibility to develop the code is to make it possible to solve two-phase fluid flow, substituting the solid object with another fluid of different density. To do that, enrichment in the nodes affected by the interface is required. Velocity needs to be enriched with a ridge function, which allows to describe discontinuities in the derivatives, while for

pressure the same enrichment or a Heaviside function can be used (some references about this can be found in [13, 12, 21]). A typical benchmark test for this problem is the modelling of a rising bubble.

Another interesting point to consider would be the analysis of the computational cost and accuracy of the code compared with the previous one used in [26] or with other references. It is possible that for the steady cylinder the advantage of the DG-XFEM formulation is negligible, but it is for the oscillating cylinder when the improvements should appear in terms of computational cost. In this line, another work to consider may be also the optimization of the code.

4.3 Applications

With the current code, a first application that is possible to implement is to change the solid cylinder for any other object. A typical test can be for example the aerodynamic analysis of the flow around an airfoil or a study about the drag forces of objects with different shapes, among other simulations involving steady objects.

Once the development of the oscillating cylinder is ready, this would allow to introduce any moving object inside the fluid. In this case, a very interesting application would be to model the insect flight, studying the aerodynamics of a flapping wing placed inside the fluid. References of this kind of studies are the work of Ansari et. al in [1] and the research done by Kurtulus et. al in [23].

Bibliography

[1] Ansari, S. A., Zbikowski, R., and Knowles, K. Aerodynamic modelling of insect-like flapping flight for micro air vehicles. *Progress in Aerospace Sciences 42*, 2 (2006), 129–172.

[2] Baker, G. A., Jureidini, W. N., and Karakashian, O. A. Piecewise solenoidal vector fields and the Stokes problem. *SIAM Journal on Numerical Analysis 27*, 6 (1990), 1466–1485.

[3] Barth, T., and Ohlberger, M. Finite Volume Methods: foundation and analysis. In *Encyclopedia of Computational Mechanics*, vol. 1: Fundamentals. John Wiley & Sons, 2004, ch. 15, pp. 439–473.

[4] Bassi, F., and Rebay, S. A high-order accurate finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics 131*, 2 (1997), 267–279.

[5] Belytschko, T., and Black, T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering 45*, 5 (1999), 601–620.

[6] Belytschko, T., Dolbow, J., and Moës, N. A Finite Element Method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering 46*, 1 (1999), 131–150.

[7] Belytschko, T., and Fries, T. P. The eXtended/generalized Finite Element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering 84*, 3 (2010), 253–304.

[8] Blackburn, H. M., and Henderson, R. D. A study of two-dimensional flow past an oscillating cylinder. *Journal of Fluid Mechanics 385* (1999), 255–286.

[9] Brenner, S. C., and Carstensen, C. Finite Element Methods. In *Encyclopedia of Computational Mechanics*, vol. 1: Fundamentals. John Wiley & Sons, 2004, ch. 4, pp. 73–118.

[10] Carrero, J., Cockburn, B., and Schötzau, D. Hybridized globally divergence-free LDG methods. part I: the Stokes problem. *Mathematics of Computation 75*, 254 (2006), 533–563.

[11] Cheng, K. W., and Fries, T. P. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering 82*, 5 (2009), 564–590.

[12] Chessa, J., and Belytschko, T. An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension. *International Journal for Numerical Methods in Engineering 58*, 13 (2003), 2041–2064.

[13] Chessa, J., and Belytschko, T. An extended finite element method for two-phase fluids. *Journal of Applied Mechanics, Transactions of the ASME 70*, 1 (2003), 10–17.

[14] Ciarlet, P. G., and Raviart, P. A. Interpolation theory over curved elements, with applications to finite element methods. *Computer Methods in Applied Mechanics and Engineering 1*, 2 (1972), 217–249.

[15] Cockburn, B. Discontinuous Galerkin Methods for computational fluid dynamics. In *Encyclopedia of Computational Mechanics*, vol. 3: Fluids. John Wiley & Sons, 2004, ch. 4, pp. 91–127.

[16] Cockburn, B., and Gopalakrishnan, J. Incompressible finite elements via hybridization. part I: the Stokes system in two space dimensions. *SIAM Journal on Numerical Analysis 43*, 4 (2005), 1627–1650.

[17] Cockburn, B., Kanschat, G., and Schötzau, D. The local discontinuous Galerkin method for linearized incompressible fluid flow: a review. *Computers and Fluids 34*, 4-5 (2005), 491–506.

[18] Cockburn, B., and Shu, C.-W. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis 35*, 6 (1998), 2440–2463.

[19] Dréau, K., Chevaugeon, N., and Moës, N. Studied X-FEM enrichment to handle material interfaces with higher order finite element. *Computer Methods in Applied Mechanics and Engineering 199*, 29-32 (2010), 1922–1936.

[20] Fornberg, B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics 98*, 4 (1980), 819–855.

[21] Gross, S., and Reusken, A. An extended pressure finite element space for two-phase incompressible flows with surface tension. *Journal of Computational Physics 224*, 1 (2007), 40–58.

[22] Huerta, A., Belytschko, T., Fernandez-Mendez, S., and Rabczuk, T. Meshfree Methods. In *Encyclopedia of Computational Mechanics*, vol. 1: Fundamentals. John Wiley & Sons, 2004, ch. 10, pp. 279–309.

[23] Kurtulus, D. F., Farcy, A., and Alemdaroglu, N. Unsteady aerodynamics of flapping airfoil in hovering flight at low Reynolds numbers. *43rd AIAA Aerospace Sciences Meeting and Exhibit. Reno, Nevada (USA)* (10-13 January 2005), 15 pp.

[24] Montlaur, A. *High-order Discontinuous Galerkin methods for incompressible flows*. PhD thesis, Universitat Politècnica de Catalunya, September 2009. http://www.tdx.cat/TDX-0122110-183128.

[25] Montlaur, A., Fernandez-Mendez, S., and Huerta, A. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *International Journal of Numerical Methods in Fluids 57*, 9 (2008), 1071–1092.

[26] Montlaur, A., Fernandez-Mendez, S., and Huerta, A. High-order implicit time integration for unsteady incompressible flows. *International Journal for Numerical Methods in Fluids* (2011). Article submitted.

[27] Montlaur, A., Fernandez-Mendez, S., and Huerta, A. Métodos Runge-Kutta implícitos de alto orden para flujos incompresibles. *Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería 27*, 1 (2011), 77–91.

[28] Montlaur, A., Fernandez-Mendez, S., Peraire, J., and Huerta, A. Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations. *International Journal of Numerical Methods in Fluids 64*, 5 (2010), 549–564.

[29] Osher, S., and Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[30] Osher, S., and Sethian, J. A. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics 79*, 1 (1988), 12–49.

[31] Peraire, J., and Persson, P. The compact discontinuous galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing 30*, 4 (2008), 1806–1824.

[32] Reed, W., and Hill, T. Triangular mesh methods for the neutron transport equation. *Tech. Report LA-UR-73-479*, Los Alamos Scientific Laboratory (1973).

[33] Roshko, A. On the development of turbulent wakes from vortex streets. *National Advisory Comittee for Aeronautics* (1953). Technical note 2913.

[34] Sukumar, N., Chopp, D. L., Moës, N., and Belytschko, T. Modeling holes and inclusions be level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering 190*, 46-47 (2001), 6183–6200.

[35] Toselli, A. hp Discontinuous Galerkin approximations for the Stokes problem. *Mathematical Models and Methods in Applied Sciences 12*, 11 (2002), 1565–1597.

[36] Zlotnik, S., Díez, P., Fernàndez, M., and Vergés, J. Numerical modelling of tectonic plates subduction using x-fem. *Computer Methods in Applied Mechanics and Engineering 196*, 41-44 (2007), 4283–4293.